



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO



FACULTAD DE INGENIERIA

"SISTEMA DE CONTROL Y AUTOMATIZACION DE EVALUACIONES PARA EL CETIS 152"

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION

P R E S E N T A N :

CASTRO NICANOR MARIA ADRIANA

ESPINOSA PALMEROS ADRIANA

GONZALEZ OLVERA ALICIA

RIVERA DUEÑAS ALEJANDRA

DIRECTOR: ING. GLORIA MATA HERNANDEZ

MEXICO, D. F.

ENERO 2000

277386
S8E7E2



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Gracias a Dios:

Por las bendiciones que me ha dado.
Por la fuerza y fortaleza que ha entregado a mi espíritu
Por darme una luz de esperanza en los momentos de zozobro.

Gracias a mis Padres:

Por el apoyo que he recibido de ustedes a lo largo de toda mi vida.

Gracias a ti madrecita:

Que por tu cariño y ayuda he podido llegar a ser la mujer que soy el día de hoy
Por tu amor y comprensión que es muy grande. Por que si en algunos momentos has sentido que te he defraudado, esto no ha sido mi intención, yo solo he querido lo mejor para todos nosotros, pero solamente a lo largo de la vida y a través del tiempo uno va madurando y aprendiendo de sus errores. Te quiero mucho mamita y perdóname si alguna vez te he ofendido.

Gracias a ti Padre:

Por que yo sé que tu me quieres mucho, que siempre me has cuidado y protegido, pero que por cosas de la vida y de nuestro mutuo orgullo nos hemos enojado y distanciado. Solo te pido que recapacites y regreses al verdadero camino de tu vida, que hagas un esfuerzo y revalores a tus seres queridos. Ya olvida esas penas y sentimientos que te hacen sufrir, y da la cara a la vida, que esta solo se vive una vez, que pasado el tiempo este no regresa. Te quiero mucho papito.

Gracias a mis hermanos:

Alfonso gracias por ser mi hermano, y por darme tu apoyo en todo momento.
Amado gracias por todo ese cariño que me has brindado y que tal vez yo no he sabido ver, tal vez he sido un poco egoísta, pero yo te quiero muchísimo. Por que yo sé que siempre has estado a mi lado y me has apoyado en las buenas como en las malas.
Alejandro gracias por la enorme paciencia que siempre has tenido conmigo, por ayudarme en el cuidado de mis hijas, y por el apoyo que he recibido de ti.

Gracias a mis hijas:

Ana Karen y Diana qué son la luz de mi esperanza y el principal motivo de todos mis esfuerzos. Ustedes me han dado los momentos más bellos de mi vida, el haberles dado la vida es la experiencia más hermosa que ha tenido. Hijas mías, cuando ustedes puedan leer estas líneas, esperó que entiendan y comprendan todo el amor que les tengo.

Gracias a mi Tía María de la Luz:

Por este cariño familiar que nos une, por tus sabios consejos y el apoyo que he tenido de parte tuya cuando más lo he necesitado.

Gracias a mi comadre María Luisa:

Por ser mi mejor amiga y estar conmigo en todo momento.
Por tus sabios consejos y por mantener esta amistad que a través del tiempo sigue floreciendo. Gracias por velar por mí y mis hijas.

Gracias a Remedios:

Por que a pesar del poco tiempo que tenemos de conocernos, has sabido escucharme, apoyarme y aconsejarme dentro de mi mundo.
Por que eres una persona ejemplar y digna de llamar amiga.

Gracias a la UNAM:

Por abrirme la luz del conocimiento y darme la preparación académica que me ha hecho ser lo que el día de hoy y mañana seré.

Adriana Castro Nicanor

Dedico este trabajo:

A mis padres, por todo el apoyo que nos han brindado a mí y a mis hermanos, por la confianza depositada en cada uno de nosotros y principalmente por alentarnos a ser cada día mejores.

Gracias por haberme brindado su cariño y comprensión, por sus consejos, ya que sin todo esto nunca hubiese llegado al final de este camino por el cual me han guiado, nunca dejaré de agradecer a Dios por haberme dado unos padres tan maravillosos como ustedes.

A la persona que en estos últimos años me ha impulsado como pareja, persona, mujer, pero sobre todo como profesionalista a Israel, porque llegó a mi vida en el momento preciso, primero como compañero, posteriormente como un gran amigo y ahora es mi presente. Gracias por todo el apoyo brindado.

Durante todo este tiempo, me he encontrado con distintas personas que han formado parte importante en mi vida, debido al apoyo, a los consejos, confianza, pero sobre todo por estar junto a mí en los momentos más relevantes de mi vida y por los momentos de alegría y risas que hemos compartido por eso quiero agradecerles a cada uno de ellos por brindarme su amistad a Marcela, Alejandra, Manibel, Alicia, Paul y Armando.

Agradezco a mis compañeras de tesis, Alejandra, Alicia, Adriana C, por el trabajo que desarrollamos, gracias por todo, amigas.

Adriana Espinosa Palmeros

Con mi eterno Agradecimiento:

A Dios, por dame la oportunidad de vivir, de gozar, de sufrir, de reír y de llorar.

A mis padres pero en especial, A ti Mamita, por tus enseñanzas de amor y fortaleza, por tu infinito cariño y dedicación, por dame la mejor herencia que un hijo puede tener: La educación.

A mis hermanos y hermanas, por su apoyo incondicional, por su inigualable alegría, por su invaluable compañía pero sobre todo por su ejemplo de rectitud y honestidad. ¡Gracias manitos!

A Armando, por tanto amor, dedicación, paciencia y apoyo, gracias por cuidarme y por quererme como soy, gracias por todo este tiempo y más aún por todo lo que vendrá. Te quiero de verdad.

A Ady, Ale y July, por sus consejos, por su apoyo, por su compañía. Gracias por ser tan buenas amigas.

A Paul e Israel, por su confianza, por los momentos alegres y por los múltiples regaños, gracias por ser como son, gracias por ser mis amigos.

A ti Pavelito por ser tan honesto, sincero y por ser siempre mi incondicional consejero, gracias por estar siempre ahí, cuando más te necesito.

A ti Moustro por tu alegría de vivir y de ser siempre el mejor, gracias por que me has enseñado mucho, gracias por darme tanto y tan poco, recuerda siempre: *No importa que el Sol se muera...*

A mis profesores de la Universidad, que me describieron una fantasía maravillosa llamada Ingeniería y me ayudaron a conquistarla.

A la Universidad Nacional Autónoma de México y a la Facultad de Ingeniería por brindarme la oportunidad de ser orgullosamente Universitaria.

A todos aquellos que hicieron posible la culminación de esta meta, mil gracias por todo.

Alis

Al escribir estas palabras tal como se me van ocurriendo una a una —como se dan los pasos— quisiera agradecer primero que a nadie A tí mi DIOS: Señor, mi salvador, mi refugio. Al que me ama y me cuida. Bendito seas por darme tanta felicidad y llenura de espíritu. Gracias por tanto, gracias por todo, gracias por mi tesis, una vez más pongo en tus manos cada instante de mi vida para que se haga tu voluntad y no la mía. Gracias por sostenerme hasta aquí. Gracias por todo PADRE ETERNO y gracias señor JESÚS por mi proyecto de tesis y por tú palabra viva.

Sin embargo, este proyecto no sería posible sin ese amor tan bello y grande, a quién dedico esta tesis y mi vida entera: A tí Mamá Sonia: Madre y amiga, gracias por tú amor incomparable, tú inteligencia, tú calidad humana, tu fortaleza de espíritu, tus enseñanzas de fe en DIOS, humildad y honestidad de alma, tus vivos ejemplos, tus consejos y manos cariñosas en los momentos difíciles. Gracias por enseñarme a reirme de la adversidad y aprender de ella. ¡Gracias por esto! ¡Gracias por tanto! Te amo eternamente.

A tí Paul: Amor de mi vida, que me das tanto amor y ganas de hacerlo todo cada vez mejor. Gracias por cuidarme tanto y amarme de verdad. Gracias por todo lo que hemos vivido, por lo que compartimos, por tu música, por lo dado, por seguir a mi lado en mi camino. Y por habitar mi corazón. Gracias por tu inteligencia, apoyo y amor con este proyecto. ¡Te amo como nunca Besoven! Y te buscaré en el gran día.

A mis hermanos y sobrina: Juan Carlo (pollo), Claudia (callita), Rafael (pavellín) y Sonia (sonajita), los amo siempre y los llevo en mis pensamientos día a día, gracias por su apoyo y por quererme como soy.

A mis abuelitos Chano y Lola: Gracias por sus consejos de vida. Quienes apostaron mi talento y a quienes daré la satisfacción del triunfo rotundo. Los quiero muchísimo y agradezco su apoyo sin reservas.

A mi papa Pablo: Eres en mi vida uno de los pilares más importantes, quién me ha legado los cimientos de amor y confianza que necesite para este y los más importantes momentos de mi vida. Te amo papá, gracias por siempre, por tu cariño tan puro y ejemplo de padre valeroso y bueno.

A mis tíos Araceli y Jorge: Gracias por amarme desde el mismo día en que nací. Por recordarme con cariño y bendecirme en el nombre de Jesús. Los quiero mucho muchísimo.

A mis hermanos de sentimientos: Moncho, Bety, Judith, Naty, Dany y Yuri, por ese cariño incondicional y por su gran aportación a mi felicidad. Gracias los amo y amaré toda mi vida.

A mi familia: Gracias a todos por su cariño, a mi bisabuelita Pilar, tíos, tías, cuñados, cuñadas, sobrinos, sobrinas, etc. Que saben que agradezco su amor, apoyo y con quienes disfruto tantos momentos de alegría. Gracias por su distinción.

A los Sres. Rafael, Ceci y Beto: Gracias por recibirme en su casa y apoyarme como si fuese de la familia.

A usted Lic. Chema de los Reyes: Gracias por su aprecio, apoyo y comprensión en mi anhelo culminado.

A tí Sarita: Gracias por tu cariño, consejos y confianza en este y otros proyectos de mi vida. Por el amor a la familia, por tus apoyos económicos y lo más importante tú tiempo. Te quiero.

A mis amigas y amigos : Mara, Marce, Adí, Mel, Alix, Alvarito, Isra, Armando, etc. Gracias por los momentos de alegría, risas y osadías que hemos compartido. Los quiero mucho.

A mis compañeras de tesis: Adí E, Alix y Adí C, por la maravilla de compartir una de las mejores experiencias de la vida. Gracias por el esfuerzo compartido. ¡¡¡ Aplausos !!!

Y de muy especial forma a nuestra máxima casa de estudios
"UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO" y "Facultad de Ingeniería"
 ¡¡¡ Por darnos la maravillosa oportunidad de lograrlo !!!

A mis profesores: Gracias por todo el trabajo hecho, por ser la llave del conocimiento. Gracias por enseñarnos ha usar la inventiva personal y crear todo lo que nos proponemos. ¡Que manera de hacerlo todo bien!. En especial a la Ing. Gloria Mata, Ing. Jose Luis Martínez e Ing. Francisco Rodríguez por hacer de nuestro interés, su interés. Muy especiales gracias.

A todos, pero a todos y cada una de las personas que no fueron mencionados por falta de espacio, que nos apoyaron (por sobre todo), a sus palabras, a su cariño sin el cual no hubiésemos podido concluir esta tesis.
HOMBRES Y MUJERES DE ESTA TALLA LOS NECESITA EL MUNDO, LOS RECLAMA MÉXICO Y LOS EXIGE DIOS

¡¡¡ Gracias por tanto!!!

¡¡¡Gracias por todo!!!

Ale

Página

Prológo I

Capítulo 1 Antecedentes

1.1 Introducción 1
1.2 Problemática actual 2
1.3 Consecuencia de la problemática 3

*Capítulo 2 Fundamentos teóricos del análisis y diseño
estructurado de sistemas*

2.1 Introducción 4
2.2 Utilización de modelos 4
2.3 Diagrama de flujo de datos (DFD) 5
2.4 Modelo conceptual de datos 11
2.5 Representación del diagrama de entidad-relación (E-R) . . . 15
2.6 Normalizar el modelo de datos 18
2.7 Conceptos de base de datos relacional 19
2.8 Normalización de tablas 21
2.9 Diccionario de datos 25
2.10 Diseño estructurado moderno 28

Capítulo 3 Diseño de la red de comunicaciones

3.1 Introducción 35
3.2 Clasificación de las redes 36
3.3 Ventajas de establecer una red 37
3.4 Componentes de red 38

3.5 Medios físicos de conexión de red	40
3.6 Arquitectura de red	42
3.7 Dispositivos de interconectividad	53
3.8 Sistemas operativos de red	55
3.9 Redes LAN	56
3.10 Análisis de las necesidades de intercomunicación de la DGETI Y en particular en el CETIS N°152	61
3.11 Diagnostico de necesidades y recursos	63
3.12 Evaluación y determinación de tecnología para instalar una Red LAN	67
3.13 Especificación de la red con que cuenta el CETIS N°152	71
3.14 Diseño de red LAN en Coordinaciones y Dependencias DGETI	72

Capítulo 4 Análisis y diseño del sistema de control y automatización

4.1 Introducción	74
4.2 Consideraciones de diseño	74
4.3 Diagramas de flujo de datos del SISE	84
4.4 Diccionario de datos	93
4.5 Diagrama Entidad – Relación	99

Capítulo 5 Desarrollo del software

5.1 Introducción	100
5.2 Comparación entre Software's	101
5.3 Análisis y selección del Software para generar el Sistema.	116
5.4 Manejo de datos	129
5.5 Diseño y construcción de la Base de Datos	132

5.6 Diagrama de Flujo de datos	138
5.7 Desarrollo de la Base de Datos	140
5.8 Diseño y construcción del software	144

Capítulo 6 Resultados y Pruebas

6.1 Introducción	157
6.2 Etapa de Pruebas	157

Capítulo 7 Manual del usuario del Sistema de Evaluación (SISE)

Bienvenido	163
Inicializando el Sistema	169
Módulo de Profesores	172
Módulo de Reactivos	181
Módulo de Asignaturas.	214
Módulo de Exámenes	227
Módulo de Claves	235
Módulo de Temarios	243

Capítulo 8 Conclusiones 254

Apéndice A Listado del programa

Apéndice B Listado de Figuras

Apéndice C Listado de Tablas

Glosario

Bibliografía

PRÓLOGO

Dentro de la gama de instituciones educativas que existen en nuestro país, se encuentra el Centro de Estudios Tecnológicos, Industriales y de Servicios (CETIS), el cual ofrece varias carreras para la formación de técnicos profesionistas.

Esta institución cuenta actualmente con un procedimiento manual de selección de evaluaciones el cual es manejado, para todo el plantel, a través de un departamento específico. Este mecanismo presenta para la institución una serie de inconvenientes, siendo uno de los principales la gran pérdida de tiempo, afectando actividades docentes y repercutiendo básicamente en retrasos académicos, administrativos así como en costos adicionales.

El presente trabajo pretende brindar al CETIS, la posibilidad de usar la computadora como una herramienta en el manejo y administración del control de las evaluaciones que realiza, las cuales actualmente se llevan a cabo mediante procesos manuales, diseñando y desarrollando un sistema de software que controle, automatice y permita el manejo de la información con mayor exactitud, rapidez y confiabilidad, así como reducir costos por pérdida de tiempo y en general para el aprovechamiento de todo el plantel.

El sistema tiene como objetivo fundamental asegurar que:

- ❖ Los datos de las diferentes academias con que cuenta el CETIS puedan compartirse entre ellas.
- ❖ El mantenimiento de dicha información sea precisa y consistente.
- ❖ Todos los datos requeridos para las aplicaciones presentes y futuras se encuentren siempre disponibles.
- ❖ El sistema sea presentado en un ambiente amigable mediante el cual el usuario pueda especificar fácilmente una orden y la máquina genere un resultado.

La creación de las aplicaciones de software de uso específico, como es el caso del ambiente gráfico, requieren gran cantidad de recursos tanto en hardware como en software (memoria, almacenamiento masivo, dispositivos periféricos e inclusive programación robusta), por lo que para el buen funcionamiento del sistema es necesario conocer los requisitos impuestos por las políticas propias de la institución además de definir los alcances del mismo, y así disponer de los recursos necesarios para su implantación.

Para la elaboración del sistema que requiere el CETIS fue necesario hacer una revisión de la problemática actual que existe en el plantel con el objeto de poder diseñar un sistema capaz de controlar y automatizar todos los procedimientos, definiendo a su vez el objetivo de la realización del proyecto. El resultado de esta revisión se presenta en el capítulo 1.

El desarrollo del sistema, como se muestra en el capítulo 2, está soportado por un conjunto de conceptos teóricos, los cuales nos permiten la construcción de cada una de las etapas que conforman un sistema.

Se presenta el diseño del sistema en red, con el fin de que pueda ser utilizado en un futuro, esto permitirá que tanto los equipos de cómputo como sus usuarios compartan y manejen grandes cantidades de información de manera óptima así como evitar la duplicidad en la misma. Este estudio se plantea en el capítulo 3.

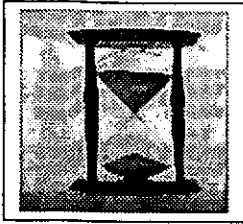
En el capítulo 4 se presenta la explicación del análisis y consideraciones de diseño del sistema de control y automatización

En el capítulo 5 se muestra el desarrollo del software, utilizando un sistema interactivo de administración de bases de datos relacional que permita trabajar en ambiente Windows.

Una vez que el sistema ha sido concluido se realizaron las pruebas correspondientes para verificar que los resultados obtenidos fueron correctos, dichos resultados se pueden observar junto con la depuración del sistema en el capítulo 6.

En el capítulo 7 se incluye el manual del usuario del sistema.

Finalmente se presenta en el capítulo 8, las conclusiones del presente trabajo.



1. ANTECEDENTES

1.1 INTRODUCCION

A través del tiempo y con la evolución del conocimiento, se han creado nuevas técnicas que modifican los procedimientos que el hombre ha desarrollado con anterioridad.

Estas nuevas técnicas vienen a suplir a las anteriores brindando métodos más sencillos y a su vez proporcionando resultados más precisos y menos susceptibles de errores. De esta manera se observa cómo lo utilizado anteriormente pasa a ser un escalón del veloz ascenso del desarrollo humano y de ninguna manera podemos juzgar que las técnicas que antes se utilizaban fueron erróneas, por el contrario, se debe asegurar que fueron éstas las que no dieron las herramientas para construir los nuevos modelos.

La historia del desarrollo de sistemas de cómputo no está exenta de los hechos mencionados anteriormente, por el contrario, debido a la necesidad de automatizar y controlar cada vez más las actividades que el hombre realiza, y siendo la mayoría de estas el objetivo principal del desarrollo de nuevas técnicas, se tiene una especial atención en el desarrollo y mantenimiento de sistemas de cómputo.

Paralelamente y con el surgimiento de la Ingeniería de Programación fueron creándose nuevos modelos para el manejo de la estructura de datos y el desarrollo de las aplicaciones, de manera que conforme el tiempo transcurre, los programas y archivos parecen no ser la mejor técnica para obtener respuestas más rápidas y con un mejor control en el manejo de la información.

Los sistemas basados en esquemas de Bases de Datos ofrecen grandes ventajas para el manejo sencillo de la información, permitiendo el respaldo y la recuperación, garantizando la integridad y seguridad de la misma, utilizando un sin fin de herramientas para su máxima explotación, así como permitir que ésta sea compartida por todos los usuarios.

Las ventajas que nos ofrecen los esquemas de Bases de Datos comparados con los sistemas tradicionales de archivos se presentan en la figura 1.1.

MANEJADOR DE ARCHIVOS	MANEJADOR DE BASES DE DATOS
Redundancia e inconsistencia de datos	Interacción con el administrador de archivos
Dificultad para tener acceso a los datos	Integridad
Aislamiento de datos	Seguridad
Usuarios múltiples son control	Respaldo y recuperación
Problemas de seguridad	Herramientas para la explotación

Tabla 1.1 Ventajas de las Bases de Datos Vs. Archivos Tradicionales

En el CETIS No.152 se imparten un total de 92 asignaturas para las distintas carreras que existen en el plantel, se elaboran diferentes evaluaciones para cada asignatura en los calendarios y periodos de estudio que existen, se cuenta con 3 tipos de exámenes: departamentales, finales y extraordinarios así como con un tipo de examen diagnóstico que no siempre se presenta. Tomando lo anterior y, considerando el proceso de inscripción escolar, el cual se ha incrementado en los últimos años, se ha buscado la óptima utilización de los recursos de la institución. Dadas las características de este proceso, se presentó la necesidad de desarrollar un sistema de cómputo, que facilitara esta tarea a fin de proporcionar al área docente y administrativa un mecanismo sencillo para satisfacer las necesidades académicas.

La idea de crear el sistema de información es estimulado por la necesidad de mejorar los procedimientos de información utilizando nuevas técnicas y recursos.

En fechas recientes las autoridades del plantel solicitaron la realización del sistema, debido al surgimiento de un sin fin de inconvenientes por el crecimiento de las distintas actividades académico-administrativas, originando rezago en la entrega de la documentación requerida por las autoridades, afectando los procedimientos en la administración escolar.

1.2 PROBLEMÁTICA ACTUAL

El problema apremiante es la necesidad de elaborar un sistema de software para el procesamiento, central y administrativo de las evaluaciones que actualmente realiza el CETIS, permitiendo la automatización y el manejo de la información con mayor exactitud, rapidez y confiabilidad, facilitando la tarea a la administración escolar y al personal docente.

El área docente se encarga de elaborar los reactivos de cada una de las evaluaciones que se aplican, y deben de entregarlos al personal administrativo para darle el formato adecuado con los

recursos con que cuenta el plantel e imprimirlos o fotocopiarlos, entregándolos en los distintos periodos de exámenes ordinarios y extraordinarios, para realizar las evaluaciones requeridas a la población estudiantil. Este procedimiento crea contratiempos en los distintos departamentos que conforman la institución, por citar algunos:

- ❖ Legibilidad deficiente de la letra (entregando el documento a mano), lo que origina problemas en la captura de la información.
- ❖ Mal planteamiento de los reactivos.
- ❖ Incompatibilidad de software en donde se capturó la información.
- ❖ Repetición de reactivos respecto a semestres anteriores ya que existen exámenes archivados.
- ❖ Pérdida de tiempo al momento de aplicar la evaluación debido a correcciones de la misma.
- ❖ Demora en la entrega de calificaciones por parte de los profesores.
- ❖ Retraso en el procedimiento debido a la demora en la entrega de actas de calificaciones.

Estos puntos entorpecen otras actividades que realiza la administración escolar ya que su tarea es la de optimizar los recursos del organismo de la institución, estas tareas requieren la participación de gran parte de los sectores de la institución, los alumnos, los profesores, los empleados y las autoridades.

1.3 CONSECUENCIA DE LA PROBLEMATICA

El proceso manual de la selección y elaboración de evaluaciones para la población escolar, es lento y requiere de mucho esfuerzo, produciendo que el resultado de las actividades no sean oportunas para la planeación académica y escolar, además de que representa una gran pérdida de tiempo escolar.



2. FUNDAMENTOS TEORICOS DEL ANALISIS Y DISEÑO ESTRUCTURADO DE SISTEMAS

2.1 INTRODUCCION

El principal objetivo de este capítulo es fundamentar teóricamente un análisis y diseño para desarrollar un software, que cumpla con los requerimientos del usuario.

La capacidad que exhiban los programadores para implementar un sistema de alta calidad y libre de errores depende en gran medida de la naturaleza del diseño; de manera similar, la capacidad de los programadores de mantenimiento para realizar cambios en el sistema después de haberlo puesto en operación depende de la calidad del diseño.

Es en base a lo anterior que la estructuración del sistema debe de estar soportado por un conjunto de conceptos teóricos, abordando metodología de análisis y diseño estructurado que involucra el desarrollo de diversos tipos de modelos tales como es el diagrama de flujo de datos y modelo entidad-relación, que a su vez es el material indispensable para diseñar la arquitectura general de cada una de las etapas que conforman dicho sistema.

El análisis estructurado es en principio transformar sus dos entradas principales: las políticas del usuario y el esquema del proyecto, en una especificación estructurada, lo implica modelar el ambiente del usuario mediante diversos diagramas. El proceso del análisis, en sus etapas primarias, consiste en desarrollar un modelo ambiental y un modelo de comportamiento para conformar el modelo esencial que represente una descripción formal de lo que el nuevo sistema debe hacer, independientemente de la naturaleza de la tecnología que se use para cubrir los requerimientos. Usando como herramientas gráficas el Diagrama de flujo de datos, el modelo conceptual de datos, el diagrama de entidad-relación y en general conceptos de bases de datos relacional; como base para el desarrollo del capítulo.

2.2 UTILIZACION DE MODELOS

El objetivo principal del análisis estructurado es transformar sus dos entradas principales: las políticas del usuario y el esquema del proyecto, en una especificación estructurada, lo implica modelar el ambiente del usuario mediante diversos diagramas.

El proceso del análisis, en sus etapas primarias, consiste en desarrollar un modelo ambiental y un modelo de comportamiento para conformar el modelo esencial que represente una descripción

formal de lo que el nuevo sistema debe hacer, independientemente de la naturaleza de la tecnología que se use para cubrir los requerimientos.

Los modelos se construyen principalmente por tres razones:

- ❖ Para resaltar las características importantes del sistema y minimizar aquellas menos importantes.
- ❖ Para discutir cambios y correcciones a los requerimientos del usuario, a bajo costo y con riesgo mínimo.
- ❖ Para verificar que se entiende el ambiente del usuario, y que se ha documentado de tal manera que los diseñadores y programadores puedan construir el sistema.

Esto se logra al utilizar diagramas gráficos, ya que con ellos se identifican fácilmente los componentes de un sistema y su interfaz. Todos los demás detalles se presentan en documentos textuales como por ejemplo el diccionario de datos.

Asimismo, los diagramas gráficos ofrecen y cumplen con una característica muy importante: la capacidad de mostrar un sistema por partes en forma descendente. Esto quizás no sea fundamental para sistemas pequeños, pues de ellos se puede decir todo lo necesario en pocas páginas, y cualquiera que necesite conocer bien algún aspecto del sistema puede conocerlo en su totalidad. Sin embargo, algunos sistemas como el que nos ocupa, deben subdividirse para un profundo análisis y seguimiento, para representar lo general y lo particular de ellos.

2.3 DIAGRAMA DE FLUJO DE DATOS (DFD)

La herramienta gráfica en la que nos vamos a apoyar es el *Diagrama de Flujo de Datos*, el cual permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por "conductos" y "tanques de almacenamiento" de datos.

Otros nombres que recibe el diagrama de flujo de datos son los siguientes:

- ❖ DFD (Abreviatura que se utilizará en este trabajo).
- ❖ Carta de burbujas.
- ❖ Modelo de proceso.
- ❖ Diagrama de flujo de trabajo.
- ❖ Modelo de función.

El DFD es una de las herramientas más comúnmente usadas, sobre todo por sistemas operacionales en los cuales las funciones del sistema son de gran importancia y son más complejas que los datos que éste maneja.

Los DFD se utilizaron por primera vez en la Ingeniería de Software como explicación para el estudio y diseño de sistemas. A su vez, la notación se tomó prestada de artículos anteriores sobre teoría de gráficas y continúa siendo utilizada por los Ingenieros de Software que trabajan en la implantación directa de modelos de los requerimientos del usuario.

La gran ventaja que presenta el DFD es que prácticamente no requiere explicación; se puede comprender el diagrama con sólo observarlo. La notación es sencilla, clara y, en cierto sentido, intuitivamente obvia. Esto es particularmente importante cuando recordamos que quien lo observará no es el analista, sino el usuario.

Los componentes de un DFD son el *proceso*, el *flujo* y la *entidad*.

El proceso es el primer componente del DFD. Los sinónimos comunes son *Burbuja*, *función* o *transformación*. Se representa como un círculo o un rectángulo con las esquinas redondeadas, como se muestra en la figura 2.1.

Tiene como objetivo mostrar una parte del sistema que transforma entradas en salidas; es decir, permite observar cómo es que una o más entradas se transforman en salidas. Se debe destacar que el proceso se nombra o describe con una sola palabra, frase u oración sencilla. En casi todos los DFD el nombre del proceso denota *lo que hace*.



Figura 2.1 Representación de un PROCESO.

El siguiente componente de un DFD es el *flujo*. Se usa para describir el movimiento de bloques o paquetes de información o sea, un conjunto de datos, de una parte del sistema a otra.

Un *flujo* se representa gráficamente por medio de una flecha que entra o sale de un proceso, tal como se ve en la figura 2.2.

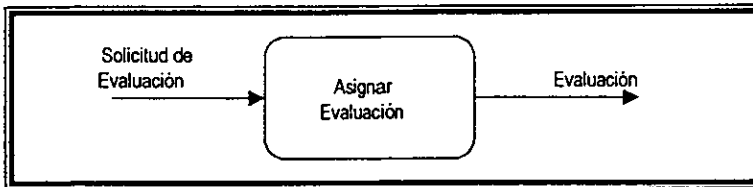


Figura 2.2 Flujo de entrada y flujo de salidas en un proceso.

Por ello, los flujos representan datos en movimiento, mientras que los archivos representan datos en reposo. También hay que notar que las flechas muestran la dirección del flujo, una cabeza de flecha en cualquier extremo (o posiblemente ambos) del flujo indica si los datos se están moviendo hacia adentro de un proceso (o ambas cosas).

Otro componente de un DFD es el *archivo*, el cual se utiliza para modelar una colección de paquetes de datos en reposo.

En la figura 2.3 se representa la notación gráfica del archivo.

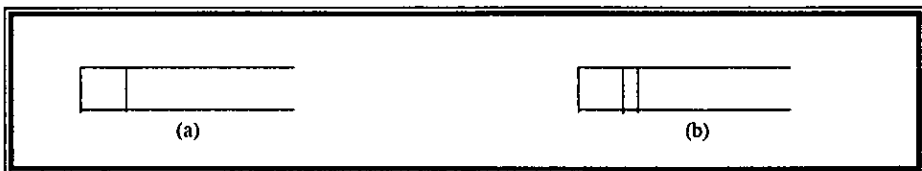


Figura 2.3 Representación gráfica del archivo. (a) Notación para dibujarse una sola vez en una hoja. (b) Notación cuando se utiliza dos o más veces en una sola hoja.

Los archivos se conectan por flujos a los procesos. Así, el contexto en el que se muestra un archivo en un DFD es uno de los siguientes:

- ❖ Un flujo desde un archivo
- ❖ Un flujo hacia un archivo

Normalmente se interpreta un flujo que procede de un sistema como una lectura o un acceso a la información del archivo.

Esto significa específicamente que:

- ❖ Se recupera del archivo un solo paquete de datos; esto es, de hecho, el ejemplo más común del flujo desde un archivo.
- ❖ Se ha recuperado más de un paquete de datos del archivo.
- ❖ Se tiene una porción de un paquete de datos del archivo.
- ❖ Se tienen porciones de más de un paquete de datos del archivo.

Cuando se examinan los flujos que entran y salen de un archivo surgirán muchas preguntas de tipo procedimiento: ¿representa el flujo un solo paquete de datos?, ¿muchos?, ¿porciones de uno o porciones de diversos paquetes?. En algunos casos podemos darnos cuenta simplemente viendo la etiqueta del flujo.

Si el flujo no está etiquetado, significa que todo el paquete de información se está recuperado; si la etiqueta del flujo es la misma que la del archivo significa que se recupera todo un paquete (o múltiples instancias de uno completo); si la etiqueta del flujo es diferente del nombre del archivo, entonces se están recuperando uno o más componentes de uno o más paquetes.

Existe un detalle de tipo procedimiento del cual podemos estar seguros, el archivo es pasivo, y los datos no viajarán a lo largo del flujo a menos que el proceso lo solicite explícitamente.

Existe otro detalle de tipo procedimiento que consideran, por convenio, los sistemas de procesos de datos, el archivo no cambia cuando un paquete se mueve del archivo a lo largo del flujo.

Un flujo hacia adentro habitualmente se describe como una escritura, una actualización o posiblemente una eliminación. Específicamente, sólo puede significar que se tiene una de las situaciones siguientes:

- ❖ Se está guardando uno o más paquetes nuevos en el archivo. Dependiendo de la naturaleza del sistema, los paquetes nuevos pudieran anexarse, es decir, de alguna manera acomodarse para que estén "después" de los paquetes existentes. Esto es a menudo un asunto de la implantación (controlado por el sistema específico de administración de bases de datos).
- ❖ Uno o más paquetes se están borrando o retirando del archivo.
- ❖ Uno o más paquetes se están modificando o cambiando. Esto pudiera traer consigo un cambio de todo un paquete, o (más comúnmente), de sólo una porción de múltiples paquetes.

En todos estos casos es evidente que el archivo cambia como resultado del flujo que ingresa. El proceso (o procesos) conectados con el otro extremo del flujo es el responsable de realizar el cambio al archivo.

Finalmente, el último componente del DFD es una *entidad externa* o *terminador*. Gráficamente se representa como un rectángulo, como se muestra en la figura 2.4.

Los terminadores representan entidades externas con las cuales el sistema se comunica. Comúnmente, un terminador es una persona, un grupo, etc. En algunos casos, un terminador puede ser otro sistema.

Suele ser muy fácil identificar los terminadores en el sistema que se está modelando. A veces el terminador es el usuario, es decir, el usuario suministra datos al sistema o bien los recibe ya procesados.



Figura 2.4 Representación gráfica de la entidad externa. (a) Notación para dibujarse una sola vez en una hoja. (b) Notación cuando se utiliza dos o más veces en una hoja.

En otros casos, el usuario se considera parte del sistema y ayudará a identificar los terminadores relevantes, es decir, de los que reciben información o bien a los que se les entrega ya procesada.

Existen tres factores importantes que hay que tomar en cuenta con relación a las entidades:

- ❖ Son externas al sistema que se está modelando; los flujos que conectan los terminadores a diversos procesos (o archivos) en el sistema representan la interfaz entre él y el mundo externo.
- ❖ Como consecuencia, es evidente que ni el analista, ni el diseñador del sistema están en posibilidad de cambiar los contenidos de un terminador o la manera en que trabaja.
- ❖ Las relaciones que existan entre los terminadores no se muestra en el modelo DFD.

GUIA PARA LA CONSTRUCCION DE UN DFD

Existen reglas adicionales que se requieren para poder utilizar los DFD con éxito. Algunas de estas reglas ayudan para no elaborar DFD erróneos, incompletos o lógicamente inconsistentes. Tienen como finalidad auxiliar en el desarrollo de un DFD que sea grato a la vista y que, por tanto, tenga más probabilidades de que el usuario lo comprenda mejor.

1.- Identificar las entradas y salidas netas.

- ❖ Determinar el contexto del sistema, esto es, las entidades externas e internas del sistema.
- ❖ Buscar los flujos de datos que cruzan el contexto, estos constituirán las entradas y salidas netas.
- ❖ El contexto debe ser suficientemente grande como para incluir todo lo relevante al esfuerzo de desarrollo, pero también lo suficientemente pequeño para no incluir cuestiones irrelevantes.

2.- Llenar el cuerpo del DFD.

- ❖ Concentrarse en los flujos de datos.
- ❖ Averiguar si dentro de los procesos identificados hay flujo de datos, en tal caso separar ese proceso en dos, tres o en los procesos que sean necesarios.
- ❖ Cuestionarse en cada flujo de datos:
 - ❖ ¿Qué se necesita para construirlo?
 - ❖ ¿De dónde provienen sus componentes?
 - ❖ ¿Qué procesos se requieren para ello?
 - ❖ ¿Podemos construir este flujo de datos a partir de los flujos de datos de entrada?

3.- Introducir en el DFD los archivos que representen los depósitos de información que el usuario ha identificado.

4.- Revisar si no hay flujo de datos faltantes, sobrantes o redes desconectadas

5.- Nombrar los flujos de datos.

- ❖ Asignar nombres a todos los flujos de datos.
- ❖ Los flujos que entran o salen de un archivo no requieren nombre en la mayoría de los casos.
- ❖ Evitar nombres ambiguos.
- ❖ No usar homónimos o sinónimos al nombrar un flujo de datos.
- ❖ El nombre del flujo de datos no únicamente debe representar los datos sino también lo que sabemos acerca de ellos.
- ❖ Tener cuidado de no mezclar datos que nada tienen que ver entre si.
- ❖ Si no se encuentra un nombre preciso, dividir el flujo de datos.

6.- Nombrar los procesos.

- ❖ Asegurarse que todos los flujos de datos ya tienen nombre.
- ❖ Tratar de nombrar los procesos usando un verbo y un objeto directo.
- ❖ Un proceso en un DFD puede representar una función que se está llevando a cabo.
- ❖ Los nombres de los procesos deben hacerse en términos de sus entradas y salidas.
- ❖ Evitar ambigüedades o nombres que abarquen parcialmente el proceso.
- ❖ Si se obtienen nombres ambiguos, dividir nuevamente el proceso.
- ❖ No utilizar verbos muy generales como "Procesar", "Hacer", etc. Tampoco objetos directos muy generales como "Información", "Datos", etc.

7.-Asignar un número a los procesos.

- ❖ Como una forma conveniente de referirse a los procesos en un DFD se asigna un número a cada burbuja. No importa mucho cómo se haga esto, de izquierda a derecha, de arriba abajo o de cualquier otra manera servirá, mientras haya consistencia en la forma de aplicar los números. La única situación que hay que tener en cuenta es que el sistema de numeración implicará, para algunos lectores casuales de un DFD, una cierta secuencia de ejecución.

8.- Omitir detalles de manejos de errores.

- ❖ Cuando se detecten rutas de error se recomienda dejarlas indicadas y trabajar en ellas posteriormente.
- ❖ Si el error no requiere descartar algún proceso ya efectuado, ignorarlo.
- ❖ Si el error requiere deshacer algo que ya se hizo, entonces no ignorarlo.

9.- Mostrar flujo de datos y no de control.

- ❖ Verificar que en todos los flujos de datos se esté pasando información, de otra manera, eliminarlos.

- ❖ Verificar que la información en cada flujo de datos se use en el proceso la que llega, de no ser así, eliminarla.

10.- Revisión del DFD.

- ❖ Es imposible que el DFD esté correcto la primera vez.
- ❖ Si el primer DFD dibujado resulta ser el final, seguramente tiene errores.
- ❖ Ejecutar todo este proceso de manera iterativa.
- ❖ Este proceso iterativo dará grandes beneficios en las fases posteriores del ciclo de vida del sistema.

Ahora bien, existen otras reglas respecto a cómo asegurar que el DFD mismo sea consistente, estas son:

- ❖ Evitar procesos infinitos, es decir, aquellos que tienen entradas pero no salidas. Estos procesos también son conocidos como agujeros negros.
- ❖ Evitar los procesos de generación espontánea, que tienen salidas sin tener entradas, por que son sumamente sospechosas y generalmente incorrectas.
- ❖ Estudiar profundamente los flujos y procesos sin nombre, ya que esto indica que puede ser dividido en datos o procesos más elementales.
- ❖ Analizar cuidadosamente los archivos de "sólo lectura" y "sólo escritura".

Cuando el sistema a modelar es intrínsecamente complejo y tiene docenas o incluso cientos de procedimientos que modelar, se recurre a la construcción de un DFD global en una serie de niveles de modo que cada uno proporcione sucesivamente más detalles sobre una porción del nivel anterior.

2.4 MODELO CONCEPTUAL DE DATOS

El modelo conceptual de datos es el primer paso del proceso TOP-DOWN para el desarrollo de una base de datos, se realiza en las etapas de Análisis y Diseño del Ciclo de Desarrollo de un Sistema.

Representa una forma efectiva para integrar y documentar los requerimientos de información de una organización, así como ser independiente del hardware o del software usados para la implementación.

El objetivo del Modelo Conceptual de Datos es desarrollar el Modelo Entidad-Relación (E-R), el cual puede ser utilizado para una base de datos de red, jerárquica o relacional. Los componentes del modelo (E-R) son las entidades y las relaciones.

Las *entidades* representan a las personas, instituciones, grupos, escuelas, etc, de las cuales se necesita conocer información y a las que se les pueden asignar atributos.

Para representar una entidad se utilizan cuadrados o rectángulos de cualquier dimensión con las esquinas redondeadas.

El nombre de la entidad se escribe en mayúsculas y en singular. En ocasiones también se escribe un nombre alternativo o sinónimo, el cual se anota abajo del principal y entre paréntesis.

Finalmente, el nombre de los atributos en minúscula como se muestra en la figura 2.5.

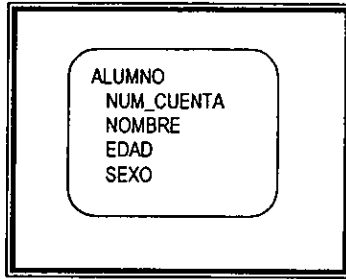


Figura 2.5 Ejemplo de representación de una entidad.

Se debe estar seguro de que la entidad posea múltiples "ocurrencias", "instancias" o "registros", considerando que cada instancia de la entidad tiene valores específicos para cada atributo de la entidad.

Cada instancia debe ser identificada como única de otras instancias de la misma entidad. Un atributo o conjunto de atributos que identifican de manera única a una instancia dentro de una entidad son llamados *identificadores únicos* (UID). Si una entidad no puede tener un identificador único (UID), ésta no puede ser una entidad.

Los atributos que identifican de manera única a una entidad y pertenecen a los UID de las entidades son precedidos por un símbolo de número y asterisco (#*).

El otro componente de un modelo (E-R) es la *relación*, la cual se considera como un evento bidireccional que representa la asociación entre dos entidades, o entre una entidad consigo misma.

La forma de escribir una relación debe contemplar el formato que se presenta en la figura 2.6.



Figura 2.6 Formato para escribir la relación entre dos entidades.

Por ejemplo, si se tuvieran las entidades PROFESOR y MATERIA, su relación estaría dada como

sigue:

- ❖ Cada materia puede ser impartida por uno y solamente un profesor.
- ❖ Cada profesor puede ser asignado a una o más materias.

Es importante destacar que cada dirección de una relación tiene:

- ❖ Un nombre. Por ejemplo: *Impartida por* o *Asignado a*.
- ❖ Una opción. Por ejemplo: *Debe ser* o *Puede ser*.
- ❖ Un grado o cardinalidad. Por ejemplo: *Uno y solamente uno*, o *uno o más*.

Un grado de 0 es etiquetado como *puede ser*.

Para representar la relación se necesita una línea entre dos entidades y los nombres de las relaciones escritas en minúsculas.

La opcionalidad queda esquematizada en la figura 2.7.

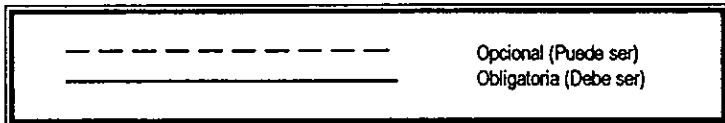


Figura 2.7 Opcionalidad.

El Grado queda ilustrado en la figura 2.8.

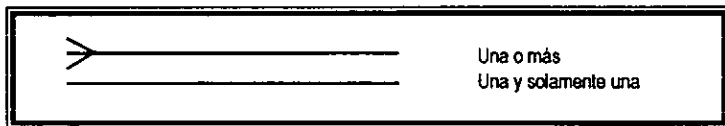


Figura 2.8 El Grado representado por una "pata de gallo" y una línea simple.

En resumen, lo anterior se ejemplifica en la figura 2.9.

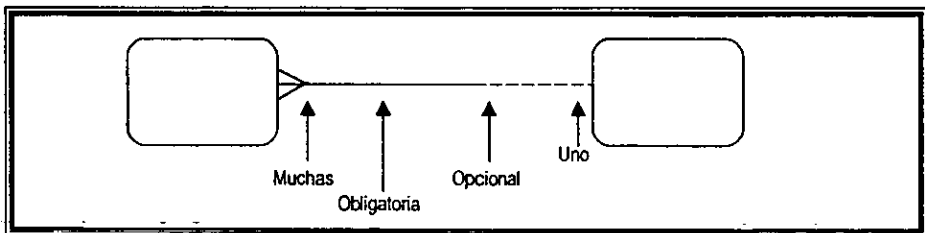


Figura 2.9 Representación gráfica de una relación entre entidades.

Existen tres grados de relación:

- ❖ Relaciones muchos a uno.
- ❖ Relaciones muchos a muchos.
- ❖ Relaciones uno a uno.

Una relación uno a uno (1 : 1) tiene un grado de *uno y solamente uno* en ambas direcciones.

Una relación muchos a uno (M : 1) tiene el grado de *uno o más* en una dirección y el grado de *uno y solamente uno* en otra dirección.

Una relación muchos a muchos (M : M) tiene el grado de *uno o más* en ambas direcciones.

Usualmente son opcionales en ambas direcciones, aunque puede ser opcional también en una sola dirección.

También en este momento se usa una *Matriz de Relaciones* como un apoyo para la recolección inicial de información sobre las relaciones entre una serie de entidades.

Los estándares de una Matriz de Relaciones son:

- ❖ Esta matriz muestra si están relacionadas y en qué forma cada entidad (renglón) con cada entidad (columna) mostrada en la matriz.
- ❖ Todas las entidades están listadas en el lado izquierdo y en la parte superior de la matriz.
- ❖ Si una entidad está relacionada con otra entidad, entonces el nombre de esta relación se muestra en la celda de intersección.
- ❖ Si una entidad no está relacionada con otra entidad, entonces se dibuja una línea en la celda de intersección.
- ❖ Cada relación por encima de la diagonal es el inverso o imagen espejo de la relación por debajo de la línea diagonal.
- ❖ Las relaciones recursivas (o sea, una entidad relacionada consigo misma) son representadas por las celdas en la diagonal.

En la figura 2.10 se representa una matriz de relaciones.

	Entidad 1	Entidad 2	...	Entidad n
Entidad 1	-----	Relación 1-2	...	Relación 1-n
Entidad 2	Relación 2-1	-----	...	-----
...
Entidad n	Relación n-1	-----	...	-----

Figura 2.10 Matriz de relaciones.

2.5 REPRESENTACION DEL DIAGRAMA ENTIDAD RELACION (E-R)

El diagrama (E-R) debe realizarse de tal manera que sea fácil de leer por la gente que necesita trabajar con él. Sus características son:

LIMPIO Y ORDENADO

- ❖ Alinear las cajas de las entidades.
- ❖ Dibujar las líneas de relación como rectas horizontales o verticales.
- ❖ Usar un ángulo de 30° a 60° el cual facilita seguir las líneas de la relación cuando éstas se cruzan.
- ❖ Evitar el uso de muchas líneas paralelas, ya que dificulta el seguimiento.

TEXTO CLARO

- ❖ Escribir el texto con claridad, en forma concisa.
- ❖ Evitar abreviaciones y modismos.
- ❖ Agregar adjetivos para mejorar el entendimiento.
- ❖ Alinear el texto horizontalmente.
- ❖ Poner el nombre de la relación al final de la línea y en lados opuestos de la línea.

FORMAS FACILES DE RECORDAR

- ❖ Hacer el Diagrama E-R fácil de recordar. Que la gente recuerde las formas y los patrones.
- ❖ No se deben dibujar diagramas E-R en una cuadrícula.
- ❖ Compactar en la medida de lo posible las cajas de las entidades para ayudar a la visualización del diagrama.

REGLAS DE FORMATO

- ❖ Tratar de colocar las "patas de gallo" en la parte izquierda para las líneas horizontales y en la parte superior para las líneas verticales.
- ❖ Poner las entidades más volátiles arriba y a la izquierda del diagrama.
- ❖ Poner las entidades menos volátiles abajo y a la derecha del diagrama.

Los *atributos* representan información específica que se necesita conocer o tener acerca de una entidad. Los atributos describen una entidad para calificar, identificar, clasificar, cuantificar o expresar el estado de una entidad. Dicho de otra manera, representa un tipo de descripción o detalle, mas no una instancia.

Los atributos deben tener nombres claros para el usuario y no codificados para el desarrollador. Un atributo debe estar asignado en una sola entidad y deben ser descompuestos hasta su mínimo componente.

También es importante verificar que un atributo no sea derivado o calculado de los valores existentes de otros atributos.

La opcionalidad de atributos puede ser clasificada como sigue:

Atributos obligatorios.

- ❖ Un valor debe ser conocido por cada ocurrencia de la entidad.
- ❖ Marcarlo con un asterisco (*).

Atributos opcionales.

- ❖ Un valor puede ser conocido por cada ocurrencia de la entidad.
- ❖ Marcarlo con una letra minúscula.

Los atributos se pueden *identificar* examinando las notas de entrevistas y realizando preguntas al usuario. Los atributos pueden aparecer en notas de entrevistas como:

- ❖ Frases y palabras descriptivas.
- ❖ Sustantivos.
- ❖ Frases preposicionales.
- ❖ Pronombres y sustantivos posesivos.

Un *identificador único* (UID) es cualquier combinación de atributos y/o relaciones que sirven para identificar en forma única una ocurrencia o instancia de una entidad. Cada ocurrencia de una entidad debe ser identificada de manera única.

En algunas ocasiones el identificador único de una entidad forma parte del identificador único de otra, en estos casos se utiliza la barra UID, como se presenta en la figura 2.11.

Una relación incluida dentro de un UID debe ser obligatoria y *una y solamente una* en la dirección que participa en el UID.

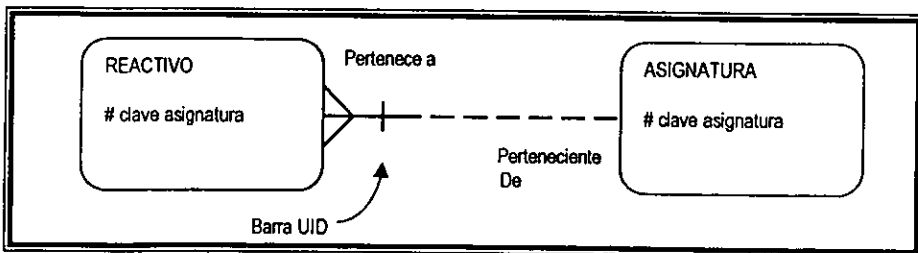


Figura 2.11 La Barra UID indica que la relación con la ASIGNATURA es parte del UID de REACTIVO.

Una entidad puede ser identificada de manera única a través de múltiples relaciones como lo indica la figura 2.12.

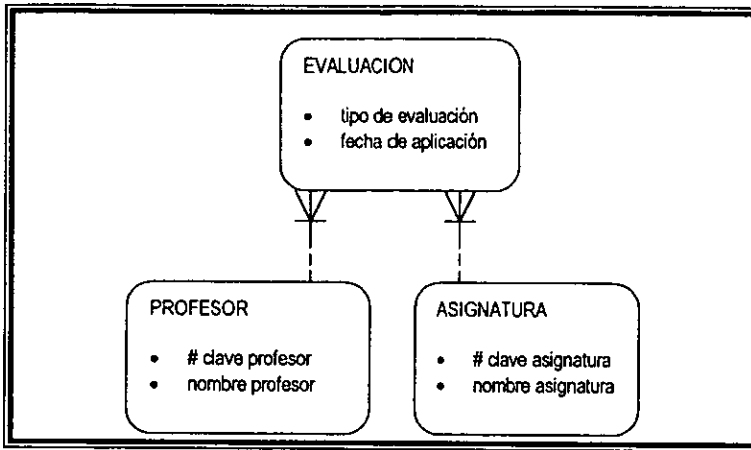


Figura 2.12 La EVALUACION no tiene UID. LA EVALUACION es identificada de manera única por el PROFESOR quien la elaboró, por la ASIGNATURA a la que pertenece, por la fecha de aplicación y por el tipo de evaluación.

Ambas relaciones son obligatorias y una y sólo una, en la dirección incluida en el UID. Una entidad puede tener más de un UID como se ilustra en la figura 2.13.

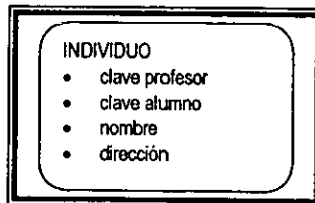


Figura 2.13 Candidatos a UID: "clave profesor", "clave alumno", y "nombre".

Aquí probablemente la combinación de apellido y nombre no es única. En estos casos se elige un candidato UID para ser el UID primario, y los otros para ser los UID secundarios.

Cuando una entidad no ofrezca posibles candidatos para ser UID, es válido crear uno artificial que asuma dicho papel. Esto se ilustra en la figura 2.14.

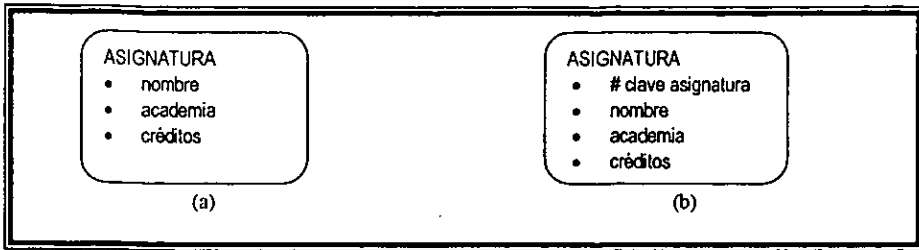


Figura 2.14 En la entidad ASIGNATURA (a) no se aprecia un posible UID. Por lo que fue necesario crear un atributo artificial "clave asignatura" en la entidad ASIGNATURA (b) para habilitarlo como UID.

2.6 NORMALIZAR EL MODELO DE DATOS

Normalizar es un concepto de bases de datos relacional, pero sus principios se aplican al Modelo Conceptual de Datos. Para validar cada atributo se deben utilizar las reglas de normalización a nivel diseño Modelo de Datos, con lo cual se traslada en forma automática a un diseño de bases de datos.

La **primera forma normal** dice que todos los atributos deben tener un solo valor para cada instancia. Se debe verificar que cada atributo tenga un valor único para cada ocurrencia de la entidad, es decir, ningún atributo deberá tener valores repetidos. Si un atributo tiene múltiples valores, se crea una entidad adicional y lo relaciona con la entidad original mediante una relación M:1.

La **segunda forma normal** recomienda que un atributo debe ser dependiente del identificador único completo.

Se debe validar que cada atributo dependa completamente del UID. Cada instancia específica del UID debe determinar una sola instancia de cada atributo. Validar que un atributo no dependa de una sola parte del UID de la entidad. Si un atributo no es dependiente del UID completo, está fuera de lugar y deberá ser movido.

Finalmente, la **tercera forma normal** menciona que ningún atributo no-UID puede ser dependiente de otro atributo no-UID. O sea, se debe validar que cada atributo no-UID no dependa de otro atributo no-UID.

Se debe mover cualquier atributo no-UID que dependa de otro atributo no-UID. Si un atributo depende de otro atributo no-UID, es necesario mover ambos, el atributo dependiente y el atributo del que depende, a una nueva entidad relacionada con la entidad actual.

Cuando algunos atributos se asocian con relaciones M:M (muchos a muchos) se tienen que resolver al sustituir una entidad intersección y dos relaciones M:1 (muchos a uno).

Las relaciones desde una entidad intersección son siempre obligatorias y son muy comunes para representar situaciones de negocios en el mundo real. Las entidades intersección normalmente generan requerimientos para atributos adicionales como el uso de cantidades y fechas. Estas tienden a ser entidades volátiles y de volumen alto.

No olvidar que el UID de una entidad intersección está frecuentemente compuesta por las relaciones entre las entidades que le dieron origen.

Si se resuelven las relaciones M:M al final de la fase de Análisis se puede llegar a tener una entidad intersección sin atributos. Una entidad intersección sin atributos es justamente una lista de referencia cruzada en doble sentido entre las ocurrencias de las entidades.

Una entidad intersección sin atributos es la excepción a la regla de que una entidad debe tener atributos para ser entidad. Existen otros tipos de modelos para representar las relaciones entre entidades, como los que a continuación se presentan:

- ❖ *Modelo jerárquico de datos.* Representación jerárquica de datos como un conjunto de relaciones muchos a uno.
- ❖ *Modelo Recursivo.* En el modelo recursivo se representan relaciones entre una entidad y ella misma.
- ❖ *Modelo de roles.* Este modelo se usa cuando una misma persona puede estar considerada en varias entidades.
- ❖ *Modelo de subtipos.* Usar subtipos para modelar exclusivamente tipos de entidad que tienen atributos o relaciones comunes.

Un supertipo es una entidad que tiene subtipos y puede ser separado en dos o más subtipos mutuamente excluyentes. También puede tener atributos y relaciones compartidas por sus subtipos.

- ❖ *Modelos de relaciones exclusivos.* Para modelar dos o más relaciones mutuamente excluyentes de la misma entidad se usa un arco. Las relaciones de arco frecuentemente tienen los mismos nombres de relación y deben ser o todas obligatorias o todas opcionales. Un arco pertenece a una sola entidad y solamente debe incluir relaciones originadas de esta entidad. Una entidad puede tener múltiples arcos, pero una relación específica solamente debe incluir relaciones originadas de esta entidad.
- ❖ *Modelos de datos de tiempo.* Se utilizan cuando se requiere agregar entidades y relaciones al modelo Entidad – Relación para acomodar datos históricos.

Una entidad intersección se usa frecuentemente para mantener información acerca de las relaciones que cambian con el tiempo.

2.7 CONCEPTOS DE BASE DE DATOS RELACIONAL

Una base de Datos Relacional es aquella que es percibida por el usuario como una colección de relaciones o de tablas de dos dimensiones.

Las tablas de Bases de Datos Relacional son sencillas, debe tener integridad de Datos, es decir, sus datos deben ser precisos y consistentes.

El Instituto Nacional Americano de Estándares (ANSI) ha establecido a SQL como el lenguaje estándar para operar sobre las Bases de Datos Relacionales.

Una Base de Datos Relacional puede soportar un conjunto completo de operaciones relacionales. Las operaciones relacionales manipulan conjuntos de valores de datos. Las tablas pueden ser utilizadas en la creación de otras tablas. Las operaciones relacionales pueden ser anidadas.

Una *Llave Primaria* (PK) es una columna o grupo de columnas que identifican de manera única a cada renglón en una tabla.

Cada tabla debe tener una llave primaria y ésta debe ser única, no pueden cambiar de valor. El UID de una identidad irá de acuerdo con la llave Primaria en su tabla correspondiente.

Una llave primaria que consta de múltiples columnas se llama *Llave Primaria Compuesta*, cuyas columnas deben ser únicas en combinación. Éstas pueden tener duplicados en forma individual, pero en combinación, no se permiten duplicados. Ninguna parte de la Llave Primaria puede ser NULA.

Una tabla puede tener más de una columna o combinación de columnas que pueden servir como la Llave Primaria de la Tabla. Cada una de estas es llamada *Llave Candidata o Alterna*. Todas la Llaves Alternas deben ser únicas y NO NULAS. Los UID secundarios concuerdan con las Llaves Alternas. Los nombres de personas normalmente no son Llaves Alternas por que no se puede garantizar que sean únicas.

Una *Llave Foránea* (FK) o *secundaria*, es una columna o combinación de columnas en una tabla, que se refieren a una llave primaria en la misma o en otra tabla. Las Llaves Foráneas son utilizadas para hacer "JOIN" entre tablas, se basan en los valores de los datos, son puramente lógicas, pueden ser repetidas, nulas y pueden coincidir con un valor de una Llave Primaria existente. Si una Llave Foránea es parte de una llave primaria, la FK pueden ser NULA.

Las restricciones¹ de integridad de datos, definen el estado relacional correcto de la base de datos, aseguran que los usuarios realicen únicamente operaciones en las cuales dejarán a la base de datos en un estado correcto y consistente.

Un dato es inconsistente si existen múltiples copias de un registro y no todas las copias han sido actualizadas. Una base de datos inconsistente puede proveer información incorrecta o contradictoria a los usuarios.

Los requerimientos del usuario también pueden determinar el estado correcto de una base de datos. Estos requerimientos son llamados *Constraints de Integridad de Datos* definidas por el usuario y pueden incluir múltiples columnas y tablas.

¹ Se refiere al término *constraints*, el cual no es muy común o familiar

2.8 NORMALIZACIÓN DE TABLAS

La normalización minimiza la redundancia de los datos y ayuda a identificar entidades, relaciones y tablas mal diseñadas.

Un dato sin normalizar es redundante, ocasionando problemas de integridad, esto es, las transacciones de actualización y borrado pueden no ser consistentes en todas las copias de los datos causando inconsistencia en la base de datos.

Cada valor de una llave no primaria DEBE depender únicamente de la llave completa, solo de ella y no de ningún otro campo.

Las reglas de las formas normales se muestran en la figura 2.15.

REGLA DE LA FORMA NORMAL	DESCRIPCION
Primera Forma Normal (1FN)	La tabla debe tener un solo valor para cada renglón. La tabla no puede contener grupos repetitivos.
Segunda Forma Normal (2FN)	La tabla debe estar en 1FN. Cada columna que no es llave debe ser dependiente de la llave primaria completa.
Tercera Forma Normal (3FN)	La tabla debe estar en 2FN. Una columna que no es llave primaria no debe depender de otra columna no llave.

Figura 2.15 Reglas de las Formas Normales

La tercera Forma Normal es un objetivo normalmente aceptado para un diseño de base de datos para eliminar la redundancia.

Las formas normales posteriores ya no son utilizadas.

Un dato sin normalizar no cumple con ninguna regla de normalización.

Para realizar la *conversión a la primera Forma Normal*, se deben remover los grupos de repetición o vectores de la base de datos y crear una nueva tabla con la PK de la tabla base y el grupo de repetición.

Para llevar acabo la conversión a la segunda Forma Normal, se deben determinar cuáles columnas, que no son llave, no dependen de la llave primaria completa de la tabla, también se deben remover esas columnas de la tabla base y crear una segunda tabla con esas columnas y la(s) columna(s) de la PK de la cual dependen.

Si cada columna no depende de la llave primaria completa, la tabla no está en 2FN.

Cualquier tabla con una llave primaria de una sola columna está automáticamente en la 2FN.

Para hacer la *conversión a la tercera Forma Normal*, se deben determinar que columnas son dependientes de otra columna no llave. Después se remueven esas columnas de la tabla base y al final se crea una segunda tabla con esas columnas no llave de la cual son dependientes.

Una tabla esta en Tercera Forma Normal si una columna no llave no es funcionalmente dependiente de otra columna no llave.

Una columna no llave no puede ser funcionalmente dependiente de otra columna no llave.

El especificar la *integridad referencial* implica considerar que el valor de una columna de llave foránea debe coincidir con un valor existente en otra columna que sea llave primaria (o ser NULO). Usar los constraints de la integridad referencial para especificar como se debe mantener la integridad referencial.

Se recomienda especificar la integridad de datos de borrado para definir qué debe suceder si un renglón que contiene una llave primaria referencial es borrada, así como especificar un constraint de actualización para definir qué debería suceder cuando una llave primaria es actualizada.

En lo que se refiere al diseño de índices, se debe tener presente que un índice es asociado con una sola tabla física y contiene los valores de una o más columnas de esa tabla. Al usar índices se mejora considerablemente el acceso a datos.

Los índices proveen acceso rápido a los renglones de datos y evitan búsquedas en toda la tabla, facilitan uniones (join) entre tablas, aseguran que no existe un valor duplicado si se define como único y son usados automáticamente cuando se referencian en las cláusulas WHERE de las instrucciones de SQL (siempre y cuando la columna no sea modificada).

Un índice concatenado es un índice creado en un grupo de columnas en una sola tabla, también se usan índices para implementar llaves y soportar los requerimientos de acceso de la aplicación.

Los índices únicos se crean para llaves primarias y generalmente los índices no únicos para llaves foráneas. Un índice único hace referencia a una columna o grupo de columnas que tienen valores únicos de la tabla. Los índices no únicos hacen referencia a columnas o grupos de columnas que son únicas en la tabla.

La creación de índices también se considera para llaves alternativas (índices únicos), en columnas no llaves críticas usadas en las cláusulas WHERE y para cualquier llave de búsqueda.

Los índices requieren espacio y generan un sobre carga al momento de hacer actualizaciones. Asimismo, se debe tener en cuenta que bajo ciertas condiciones, los índices no son usados por el RDBMS.

DIAGRAMA DE TRANSICION DE ESTADOS

El diagrama de transición de estados, también conocido como DTE, enfatiza el comportamiento dependiente del tiempo del sistema.

Hasta hace poco, los modelos del comportamiento dependiente del tiempo del sistema importaban sólo para una categoría especial de sistemas conocidos como sistemas de tiempo - real.

Ejemplos de estos sistemas son: control de procesos, sistemas de conmutación telefónica, sistemas de captura de datos de alta velocidad, etc. Algunos de estos sistemas son pasivos, es decir, no buscan controlar el ambiente que los rodea, sino más bien reaccionan a él o capturan datos que le atañen.

Otros sistemas de tiempo real son más activos, o sea que, pretenden mantener el control sobre algún aspecto del ambiente que los rodea.

Los sistemas de este tipo manejan fuentes externas de datos de alta velocidad, y deben proporcionar alguna respuesta y datos de salida de manera suficientemente rápida como para manejar el ambiente externo. Una parte importante de la especificación de tales sistemas es la descripción de *qué sucede cuando*.

Para los sistemas enfocados a cuestiones administrativas, lo tratado anteriormente normalmente no ha sido tan importante. Las entradas pueden llegar al sistema de diferentes fuentes a velocidades relativamente altas, pero habitualmente se pueden detener si el sistema está ocupado haciendo otra cosa.

Generalmente los únicos asuntos que involucran tiempos en este tipo de sistemas son especificaciones de tiempo de tiempo de respuesta.

CONSTRUCCION DEL DIAGRAMA DE TRANSICION DE ESTADOS

Para la construcción de un DTE se puede seguir uno de los siguientes enfoques:

- ❖ Se puede comenzar por identificar todos los posibles estados del sistema y representar cada uno como una caja separada en una hoja de papel. Luego, se pueden explorar todas las conexiones con significado, es decir, los cambios de estado entre las cajas.
- ❖ Como alternativa, se puede comenzar por el estado inicial, y luego metódicamente ir siguiendo un camino hasta él o los estados restantes; luego del o los estados secundarios, proseguir a los terciarios; etc.

El enfoque quedara determinado, en muchos casos, por el usuario con quien se trabaje, sobre todo si él es el único que está familiarizado con el comportamiento dependiente del tiempo del sistema.

Cuando se termine de construir el DTE preliminar, deben seguirse las siguientes reglas para verificar la consistencia:

- ❖ Definir todos los estados. Se observa con cuidado el sistema para detectar si existe algún otro comportamiento observable, o alguna otra condición en la que el sistema pudiera estar, aparte de las que se han identificado.
- ❖ Asegurar que se puedan alcanzar todos los estados, o bien, identificar los estados que no tengan caminos que conduzcan a ellos para eliminarlos.
- ❖ Verificar que se pueda salir de todos los estados. El sistema puede tener uno o más estados finales con múltiples entradas a ellos, pero todos los demás estados deben tener un sucesor.
- ❖ Determinar todas las condiciones posibles a las que el sistema debe responder adecuadamente en cada estado. Uno de los errores más comunes cuando se construye un DTE es el identificar los cambios de estado cuando únicamente ocurren condiciones normales, pero no se especifica el comportamiento del sistema ante condiciones inesperadas.

Por ejemplo, si tomamos en cuenta el modelo de la figura 2.16, se espera que el usuario presione una tecla de función en su terminal para causar un cambio de un estado 1 a un estado 2, y una tecla diferente para ir del estado 2 al 3. Pero ¿qué pasa si el usuario presiona la misma tecla dos veces seguidas? ¿O alguna otra tecla?

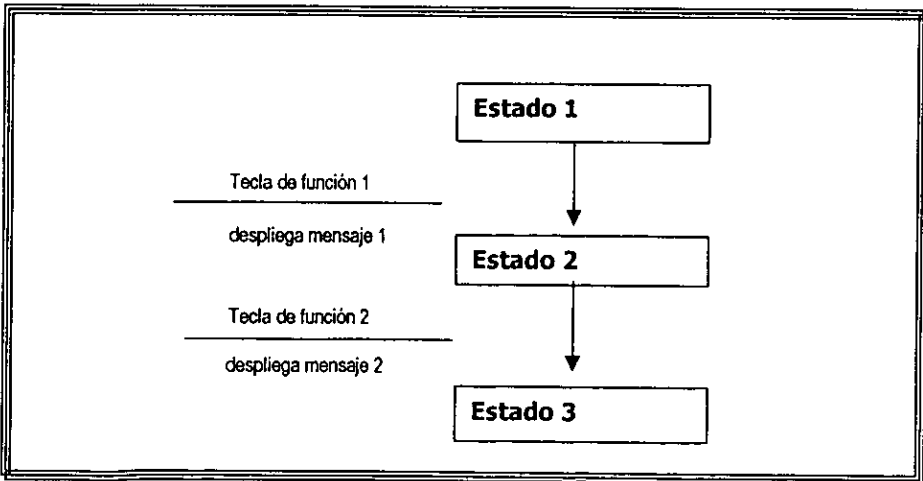


Figura 2.16 DTE incompleto

Si no se especifica el comportamiento del sistema, existe una buena posibilidad de los diseñadores y programadores no lo programen tampoco, y el sistema tendrá un comportamiento impredecible bajo una variedad de circunstancias.

2.9 EL DICCIONARIO DE DATOS

El diccionario de datos es otra herramienta del modelado del sistema; si bien es cierto que no tiene el atractivo gráfico de los DFD, es indispensable.

Sin el diccionario de datos, el modelo de los requerimientos del usuario no puede considerarse completo; todo lo que se tendría es un borrador rudimentario, una "visión del artista" del sistema.

La importancia del diccionario de datos a menudo les pasa de largo a muchas personas, pues no han utilizado algún tipo de diccionario durante muchos años. Es seguro que sin él, llegará el momento en que el usuario lector se enviará y no podrá estar seguro de que entendió los detalles de la aplicación.

El diccionario de datos es un listado organizado de todos los datos pertinentes del sistema con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento común de todas las entradas, salidas, componentes de archivos y cálculos intermedios. El diccionario de datos define los datos haciendo lo siguiente:

- ❖ Describe el significado de los flujos y archivos que se muestran en los DFD.
- ❖ Describe la composición de agregados de paquetes de datos que se mueven a lo largo de los flujos, es decir, paquetes complejos (por ejemplo el domicilio en un cliente), que puede descomponerse en unidades más elementales (como ciudad, estado y código postal).
- ❖ Describe la composición de los paquetes de datos en los archivos.
- ❖ Especifica los valores y unidades relevantes de piezas elementales de información en los flujos de datos y en los archivos de datos.
- ❖ Describe los detalles en las relaciones entre los archivos que se enfatizan en un diagrama de entidad - relación.

El hecho de utilizar una notación para el diccionario de datos representa una necesidad, ya que en la mayoría de los sistemas reales con los que se trabaja, los conjuntos o elementos de datos, serán lo suficientemente complejos como para que se necesite describirlos en términos de otros conceptos.

Los elementos complejos de datos se definen en términos de elementos más sencillos, y los sencillos en términos de los valores y unidades legítimas que pueden asumir.

Existen muchos esquemas de notación comunes utilizados por el analista de sistemas. El que se muestra en la figura 2.17 es de los más comunes y utiliza distintos símbolos sencillos.

=	está compuesto de
+	y
()	optativo (puede estar presente o ausente)
{ }	iteración
{	seleccionar una de varias alternativas
comentario	comentario o descripción
#	identificador (campo clave) para un archivo
	separa opciones alternativas en la construcción

Figura 2.17 Notación del diccionario de datos.

Como se puede observar, los símbolos parecen expresiones matemáticas y pudiera pensarse que serían difíciles de entender. Sin embargo, la notación es bastante fácil de leer.

DEFINICIONES =

La definición de un dato se introduce con el símbolo "=". En este contexto, el "=" se lee: "se define como", o "se compone de", o simplemente "significa".

Por ejemplo la notación $Z=X+Y$ se puede leer de las siguientes formas:

Z esta descompuesto de X y Y
o
Z se define como X y Y

Para definir por completo un dato, se debe incluir lo siguiente:

- ❖ El significado del dato dentro del contexto de la aplicación del usuario. Por lo común se utiliza la notación **** significado o comentario ****.
- ❖ La composición del dato, si se compone de partes elementales con significado.
- ❖ Los valores que puede tomar el dato, si se compone de partes elementales con significado.
- ❖ Los valores que puede tomar el dato, si es un dato elemental que no puede descomponerse más.

Las partes elementales de los datos son aquellas para las cuales ya no existe una descomposición con significado dentro del contexto del ambiente del usuario.

Esto usualmente es una cuestión de aplicación y es algo que se debe explorar cuidadosamente con el usuario. Pero tal vez en algunos ambientes de usuario no se requiere tal descomposición, ni sea relevante, ni tenga significado.

Quando se han identificado los datos elementales, deben introducirse al diccionario de datos. Como se indicó anteriormente, el diccionario de datos debe proporcionar una breve narrativa, encerrada entre caracteres "", que describa el significado del término en el contexto del usuario. Desde luego, habrá términos que se definan solos, es decir, cuyo significado es universal para todos los sistemas de información, o donde el analista pudiera estar de acuerdo en que no se necesita aclarar más.

Por ejemplo apellido paterno del profesor, domicilio, teléfono particular, etc.

Para tales casos se usan la notación "" para indicar "sin comentarios", ya que no se escribe algún texto explicando el significado del dato. Sin embargo, es importante especificar los valores y unidades de medida que pueden tomar.

DATOS OPCIONALES {}

Un dato opcional, como la frase implica, es aquel que puede estar o no presente en un dato compuesto.

Por ejemplo salón de clases = (ubicación) + (nombre del salón) significa que un salón puede ser identificado sólo por su ubicación o bien, sólo por su nombre o bien, por su ubicación y nombre o aunque no es usual por ninguno de los dos.

ITERACIÓN {}

La notación de iteración se usa para indicar la ocurrencia repetida de un componente de un dato. Se lee como "cero o más ocurrencias de".

Por ejemplo dar de alta un profesor = nombre del profesor + domicilio del profesor + {materia} significa que para dar de alta un profesor siempre debe contener un nombre de profesor, un domicilio y también cero o más ocurrencias de una materia. Es decir, un profesor puede estar asignado a una, o dos o más materias.

Quando se quieren especificar límites inferior y/o superior se anotan en los extremos de las llaves; como se indican aquí 1{materia} 5, un alumno se puede inscribir en 1,2,3,4 ó 5 materias.

SELECCIÓN []

La notación de selección indica que un dato consiste en exactamente un elemento de entre un conjunto de opciones alternativas. Las opciones se encierran en corchetes "[]", y se separan por una barra vertical "|".

Por ejemplo status del profesor = [Activo | Inactivo]

ALIAS

Un alias, como el término indica, es una alternativa de nombre para un dato. Esto es una ocurrencia común cuando se trata con diversos grupos de usuarios en diferentes departamentos o ubicaciones geográficas, que consisten en utilizar distintos nombres para decir lo mismo.

El alias se incluye en el diccionario de datos para que esté completo, y se relaciona con el nombre primario u oficial del dato.

Por ejemplo: investigador = *alias del profesor*

En este caso, la composición de investigador no se anota debido a que es un alias de profesor y de esta manera "hereda" la composición de éste último.

En un sistema mediano o grande, el diccionario de datos puede requerir una cantidad considerable de trabajo y tiempo.

Lo más recomendable es hacer uso de la computadora para introducir definiciones al diccionario, verificar que sean completas y consistentes, y producir reportes apropiados.

Se deben tener presentes, además, las siguientes limitaciones posibles:

Estar forzado a limitar los nombres de datos a cierta longitud. Esto probablemente no sea un gran problema, pero podría el usuario insistir en un nombre como *nombre-completo-profesor* y que el paquete de elaboración del diccionario obligue a abreviarlo como *nom-com-pro*.

Pudiera ser que el usuario no permita ciertos caracteres al elaborar el nombre del dato, como el guión (-) por ejemplo.

El diccionario de datos siempre debe estar completo y terminado antes de llevar a cabo la elaboración del pseudocódigo y obviamente del código de los programas. Para lograrlo quizá sea necesario asignar dos o más analistas para su elaboración.

2.10 DISEÑO ESTRUCTURADO MODERNO

La capacidad que exhiban los programadores para implementar un sistema de alta calidad y libre de errores depende en gran medida de la naturaleza del diseño; de manera similar, la capacidad de los programadores de mantenimiento para realizar cambios en el sistema después de haberlo puesto en operación depende de la calidad del diseño.

El campo del diseño estructurado ofrece guías para ayudar al diseñador a determinar los módulos, y sus interconexiones, que mejor dan atención a los requerimientos especificados por el analista.

Las dos reglas más importantes son las referentes al acoplamiento y la cohesión:

La *cohesión* se refiere al grado en el cual los componentes de un módulo (típicamente las instrucciones individuales que conforman un módulo) son necesarios y suficientes para llevar a cabo una sola función bien definida. En la práctica, esto significa que el diseñador debe asegurarse de no fragmentar los procesos esenciales en módulos, y también debe asegurarse de no juntar procesos no relacionados (que se representan por burbujas en DFD) en módulos sin sentido. Los mejores módulos son aquellos que son funcionalmente cohesivos (es decir, módulos en los cuales cada instrucción es necesaria para poder llevar a cabo una sola tarea bien definida). Los peores módulos son los que son coincidentalmente cohesivos (es decir, cuyas instrucciones no tienen una relación significativa entre uno y otro).

El *acoplamiento* es el grado en el cual los módulos se interconectan o se relacionan entre ellos. Entre más fuerte sea el acoplamiento entre módulos en un sistema, más difícil es implantarlo y mantenerlo, pues entonces se necesitará un estudio cuidadoso para la modificación o cambio de algún módulo o módulos. En la práctica esto significa que cada módulo debe tener interfaces sencillas y limpias con otros, y que se debe compartir un número mínimo de datos entre módulos. También significa que un módulo dado no debe modificar la *lógica interna* o los datos de algún otro módulo; lo que se conoce como una conexión patológica.

Además de las características anteriores, el diseñador debe tomar en cuenta otros parámetros para optimizar el sistema. Algunos de ellos son los siguientes:

El *tamaño del módulo*. De ser posible, cada módulo debe ser lo suficientemente pequeño como para caber en una sola página o para que pueda desplegarse en una sola pantalla. Desde luego, a veces no es posible determinar que tan grande va a ser un módulo hasta haberlo escrito, pero las actividades iniciales de diseño a menudo darán al diseñador una buena pista de que el módulo va a ser grande y complejo.

Si es así, debe partirse en uno o más niveles de submódulos. En raras ocasiones, los diseñadores crean módulos que son triviales.

Por ejemplo, módulos que consisten en sólo dos o tres renglones de código. En este caso, pueden juntarse varios en un solo supermódulo mayor.

El *alcance del control* se refiere al número de subordinados inmediatos que un módulo administrador puede llamar. Un módulo no debe poder llamar a más de una media docena de módulos de nivel inferior.

La razón es evitar la complejidad: si el módulo tiene, digamos 25 módulos de nivel inferior, entonces probablemente contendrá tanta lógica compleja de programa (en la forma de declaraciones IF anidadas, o de iteraciones DO-WHILE anidadas, etc.) que nadie lo podrá entender.

La solución es introducir un nivel intermedio de módulos administradores, como haría un administrador de una organización humana si se ve en la necesidad de tratar de supervisar directamente a 25 subordinados inmediatos. Sin embargo, existe una excepción a esto conocida como centro de transacciones.

Si el módulo administrador toma una sola decisión para invocar a uno solo de sus subordinados, entonces su lógica probablemente es bastante sencilla.

La regla del "*alcance del efecto / alcance del control*" sugiere que cualquier módulo afectado por el resultado de alguna decisión debe ser subordinado del módulo que toma la decisión. Es un tanto análogo a la regla de administración que establece que cualquier empleado afectado por los resultados de la decisión de algún administrador, es decir, dentro del alcance del efecto de la decisión debe estar dentro del alcance de control del administrador (o sea, trabajando entre la jerarquía de personas que se reportan con el administrador).

Violar esta regla en un ambiente de diseño estructurado usualmente lleva pasos innecesarios de banderas y condiciones (lo cual incrementa el acoplamiento entre módulos), la toma redundante de decisiones.

DISEÑO DE LA BASE DE DATOS

El diseño de la base de datos se lleva a cabo por medio de las siguientes actividades:

- ❖ Pasar el modelo E-R a tablas relacionales para producir el diseño inicial.
- ❖ Refinar el diseño inicial para producir un diseño completo de la base de datos.

La etapa de diseño de la base de datos produce especificaciones de diseño para una base de datos relacional, incluyendo definiciones para tablas relacionales, índices, vistas y espacio de almacenamiento.

En este punto es necesario documentar cada tabla relacional en un mapa de instancias como el que se muestra en la figura 2.17.

Los tipos válidos de llaves son PK para una columna llave primaria y FK para la columna llave foránea. Se usan sufijos para distinguir entre múltiples columnas FK en una tabla, por ejemplo, FK1 y FK2. Etiquetar múltiples columnas con el mismo sufijo. Se usa NN para una columna que debe ser definida como NO NULA. Se usa U para la columna que debe ser única.

	Columna 1	Columna 2	Columna 3	...	Columna n
Tipo llave					
Nulos/único					
Ejemplo					

Figura 2.18 Mapa de instancias

Si múltiples columnas deben ser únicas en combinación, etiquetarías con un sufijo, por ejemplo U1.

Se etiqueta una columna sencilla PKO como NN, U.

Se etiquetan múltiples columnas PK (compuestas) como NN, U1 o como NN, U1, U.

Para producir el diseño inicial de la base de datos se llevan a cabo los siguientes pasos:

- ❖ Mapear las entidades para las tablas.
- ❖ Mapear atributos para columnas y documentar datos simples.
- ❖ Mapear identificadores únicos a llaves primarias.
- ❖ Mapear relaciones a llaves foráneas.
- ❖ Elegir opciones de arco (si las hay)
- ❖ Elegir opciones de subtipo (si las hay)

MAPEAR LAS ENTIDADES

Mapear la tabla para cada entidad implica crear un mapa de instancias para la nueva tabla. Por ejemplo, para una entidad llamada ASIGNATURA se creará una tabla llamada ASIGNATURA.

El nombre de la tabla debe ser fácil de identificar con el nombre de la entidad. El nombre en plural de una entidad se usa algunas veces porque la tabla debe contener un grupo de renglones. Una entidad simple no es un subtipo o supertipo.

MAPEAR ATRIBUTOS A COLUMNAS

Se mapea cada atributo de la entidad a una columna en su tabla correspondiente. Se establecen los atributos obligatorios para columnas NO NULAS (NN). El nombre de los atributos deben ser cortos pero significativos.

El nombre de las columnas debe ser fácil de identificar en un modelo E-R.

Es importante prevenir al usuario de no usar las palabras reservadas de SQL para nombres de columnas.

Usar abreviaciones consistentes que no causen confusión al usuario y al programador.

Los nombres de columnas cortos o pequeños reducirá, el tiempo requerido para el comando de SQL "parsing".

MAPEAR UIDS A LLAVES PRIMARIAS

Asignar cualquier atributo(s) que sea parte del UID de la entidad a columnas PK, las cuales deben etiquetarse.

Todas las columnas etiquetadas con PK deben etiquetarse también con NN y U.

También se asigna un UID que incluya múltiples atributos a una PK compuesta y se etiquetan estas columnas NN y U1.

Si una entidad incluye una relación, agregar columnas de llaves foráneas para la tabla y señalarlas como parte de la llave primaria.

Se debe escoger un nombre único para cada columna FK, y etiquetar la(s) columna(s) PK, NN y FK.

Si existen múltiples columnas FK en una tabla, usar sufijos para distinguirlos, por ejemplo, FK1 y FK2. Etiquetar múltiples columnas llave con el mismo sufijo.

Las PK compuestas deben ser únicas en combinación y deben ser etiquetadas como U1.

Siempre se deben agregar ejemplos de datos para las columnas FK.

MAPEAR RELACIONES PARA LLAVES FORÁNEAS

Para una relación de entidades M:1 tomar el PK de la tabla (1) y ponerlo en la tabla (M).

Se elige un nombre único para la columna FK y etiquetar la(s) columna(s) FK.

Para las relaciones *debe ser*, se etiqueta la columna como NN y se agregan ejemplos.

Si el PK de la tabla incluye una llave foránea (FK) las columnas FK que soportan la relación, puede ser agregadas en el mapeo de UIDS a llaves primarias.

Para una relación obligatoria 1:1, se colocan las FK como únicas en el mapa de instancias en donde la relación es obligatoria y se usa el constraint NO NULO para forzar la condición obligatoria.

Si una relación 1:1 es opcional en ambas direcciones mover la FK en la tabla en cualquiera de las dos tablas de la relación. La FK debe ser siempre única, pero permitiendo NULOS.

Para una relación recursiva 1:M se agrega una columna FK a la tabla. Esta columna FK debe referenciar valores de la columna PK.

La columna FK hace referencia a un renglón en la misma tabla y su nombre debe reflejar la relación.

Una FK recursiva nunca debe ser NO NULA.

Para una relación recursiva 1:1 se agrega una FK única a la tabla. Esta columna FK debe referenciar un valor de la columna PK.

La combinación de columnas PK y FK siempre debe ser única para asegurar la relación 1:1. Poniendo el PK y FK como únicos (U) se garantiza que la combinación será única.

ESCOGER OPCIONES DE ARCO

Los arcos representan un tipo de llave foránea de alternativa múltiple.

Se debe escoger entre las alternativas de diseño de Arco explícito y de Arco genérico para "mapear" arcos a llaves foráneas.

También se utilizan estos dos tipos de diseño para implementar múltiples llaves foráneas cuando un arco atraviesa un conjunto de relaciones 1:1.

Los arcos solamente pueden atravesar el final de las relaciones que son o todas obligatorias o todas opcionales.

El diseño de Arco Explícito crea una columna de llave foránea para cada relación que incluye el arco. También este diseño soportará llaves foráneas múltiples con diferentes formatos. El software de aplicación debe forzar a una relación de exclusividad entre las llaves foráneas.

El diseño de Arco genérico crea una columna de llave foránea única y una columna para marcar la relación del arco. Como las relaciones son exclusivas, solamente un valor de FK existirá para cada renglón en la tabla.

Si las relaciones debajo del arco son obligatorias, se hacen ambas columnas NO NULAS (NOT NULL). Las llaves foráneas deben compartir el mismo formato para todas las tablas referenciadas. La relación de exclusividad se fuerza automáticamente.

ESCOGER OPCIONES DE SUBTIPOS

Se debe escoger entre las alternativas de diseño de una sola tabla, diseño de tablas separadas e implementación del Arco para "mapear" subtipos a tablas.

No olvidar que los subtipos heredan todos los atributos del supertipo y sus relaciones, que pueden tener atributos y relaciones propias y, por último, que deben ser mutuamente excluyentes.

Con relación al diseño de subtipo en una sola tabla, se crea una sola tabla para el supertipo y se mapean los subtipos dentro de esta tabla. La tabla sencilla contendrá instancias de todos los subtipos.

Los procesos para llevar a cabo este diseño son:

Crear una sola tabla para el supertipo.

Crear una columna para cada uno de los atributos del supertipo.

Crear una columna TIPO para identificar a que subtipo pertenece cada renglón.

Crear una columna para cada uno de los atributos del subtipo.

Crear columnas FK para cada una de las relaciones del supertipo.

Crear columnas FK para cada una de las relaciones del subtipo.

El diseño de subtipo de una sola tabla requiere una nueva columna *tipo* para identificar el subtipo al que pertenece cada renglón.

Se debe usar un diseño de subtipo en una sola tabla cuando hay pocas relaciones y atributos propios. Las ventajas que se tienen son que el acceso al supertipo es directo y que el subtipo puede ser accesado y modificado usando vistas. Pero las desventajas son que los requerimientos del subtipo NO NULO no se pueden forzar a nivel de base de datos y que la lógica de las aplicaciones tendrá que manejar diferentes conjuntos de atributos, dependiendo del TIPO.

En cuanto al **diseño de subtipo en tablas separadas** se debe mapear una tabla para cada subtipo ya que cada una de ellas contendrá solamente instancias de un subtipo.

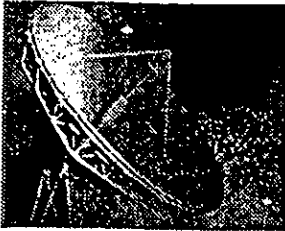
Los pasos para realizar este tipo de diseño son:

- ❖ Crear una tabla para cada subtipo.
- ❖ En cada tabla subtipo, crear columnas para los atributos del subtipo.
- ❖ En cada tabla subtipo, crear columnas para los atributos del supertipo
- ❖ En cada tabla subtipo, crear columnas FK para las relaciones del subtipo.
- ❖ En cada tabla subtipo, crear columnas FK para las relaciones del supertipo.

Se usa un diseño de subtipo de tablas separadas cuando hay muchas relaciones y atributos de subtipos específicos.

Las ventajas de este diseño son que la opcionalidad de los atributos se fuerza a nivel de la base de datos y la lógica de las aplicaciones no requiere de revisión para los subtipos.

Las desventajas son que la consulta al supertipo requiere de un operador de UNION, las vistas que enlazan dos tablas (mediante un join), son solamente de consulta, el código del programa de aplicación debe ser específico para cada tabla individual y el mantenimiento de los UID's de los subtipos es difícil de implementar.



3. DISEÑO DE LA RED DE COMUNICACIONES

3.1 INTRODUCCIÓN

Debido al tremendo impacto que han alcanzado las computadoras y las redes de cómputo en la sociedad durante la última década, a este periodo de la historia se le ha dado el nombre de "era de la informática". Diariamente millones de individuos utilizan la computadora para realizar sus actividades personales y de negocios, lo que la ha hecho casi indispensable. Este tren de vida se acelera más en tanto más personas descubren el potencial de las computadoras y las ventajas que proporcionan las redes de cómputo.

El objetivo principal de una red es la transferencia e intercambio de la información de datos entre computadoras. Ahora, desde una computadora y con una sencilla conexión a una red mundial (como internet) una persona tiene la capacidad de poseer toda la información disponible en el mundo, organizada de tal forma que puede manejarla y emplearla para sus necesidades particulares sin importar el volumen o el lugar donde se encuentre.

Actualmente las empresas líderes de todo tipo ya no se cuestionan si instalan o no una red local o de área amplia o si se conectan o no a Internet, porque esto garantiza que su empresa sobreviva en el futuro próximo. En estos momentos la tecnología de las redes está en pleno desarrollo y aún es un tema relativamente nuevo en México.

El desarrollo del presente capítulo está basado en la importancia de la modernización tecnológica de la Dirección General Estudios Tecnológicos e Industriales (DGETI), particularmente del Centro de Estudios Tecnológicos Industrial y de Servicios N°152 en cuanto a redes de telecomunicaciones se refiere como parte de la nueva política de la DGETI. La nueva política del CETIS N°152 responde al compromiso establecido de enfrentar con firmeza y sentido de futuro el reto de la educación y atender, con hechos las demandas de la sociedad estudiantil.

Para enfrentar adecuadamente los problemas de la educación es necesario desarrollar un esfuerzo tecnológico importante, en donde se requiere que su administración sea la adecuada. El facilitar esta administración es el objetivo del presente capítulo de tesis, ya que se busca que con la instalación de redes locales tanto en la Oficinas Centrales como en las distintas dependencias de la cual forma parte el CETIS N°152, se pueda hacer más eficiente el acceso en tiempo real a las bases de datos de todas las aplicaciones desarrolladas en la DGETI.

La estructura de este capítulo de tesis se conforma de los *Conceptos generales de redes*; en donde se da un panorama general de los conceptos básicos de lo que es la teoría de redes, para que tenga las herramientas que le permitan saber en primer lugar, qué es una red de computadoras, los componentes que las integran y las tecnologías más utilizadas en la implementación de las mismas con el fin de que puedan aplicar estos conceptos en el diseño de las futuras redes.

En el *Análisis de las necesidades de la intercomunicación de la DGETI y en particular el CETIS N°152*, se consideran los problemas que tiene los usuarios con el manejo de su información. Mediante el estudio de estos problemas se plantea una futura instalación de una red de área local en cada una de las instituciones. Una vez definidas las actividades se procede a levantar el inventario del equipo de cómputo existente en el CETIS N°152 para, de esta manera, determinar que equipos conformarán a las redes. Se realiza la evaluación de tecnologías tanto LAN mediante cuadros comparativos y en base a éstas, se selecciona el diseño de la red LAN para el CETIS N°152, de la DGETI, en este parte se plantea el diseño físico de la red LAN se determinan las características de los componentes que las integrarían y las adquisiciones realizadas para su instalación.

3.2 CLASIFICACIÓN DE LAS REDES

Una *red de computadoras* es un sistema de comunicaciones de datos que enlaza dos o más computadoras y dispositivos periféricos. Una red por su tamaño en número de nodos puede ser inicialmente pequeña y después crecer con una organización.

Por su disponibilidad las redes se clasifican en:

- ❖ **Públicas:** Telepac, VSAT, TDMA
- ❖ **Privadas:** Red UNAM, PEMEXPAC, red de la SRE

Por su topología:

- ❖ **Verticales:** También se conocen como redes centralizadas. Aquí el procesamiento de la información se realiza en un centro de cómputo único y los usuarios cuentan con terminales "tontas", a través de la cuales hacen uso del servicio de cómputo centralizado.
- ❖ **Horizontales:** En este tipo de redes de procesamiento se realiza en varios centros de cómputo diseminados geográficamente, las cuales tienen importancia semejante.
- ❖ **Mixtas:** Es una combinación de las dos anteriores.

Atendiendo a su distribución geográfica:

- ❖ **LAN** (Local Area Network [Red de Area Local]) Esta red compuesta por un conjunto de computadoras que se comunican entre sí en un área geográficamente limitada, como puede ser un mismo edificio.
- ❖ **MAN** (Metropolitan Area Network [Red de Area Metropolitana]) Este tipo de red se constituye cuando se conecta un conjunto de redes locales en una misma área geográfica pequeña; por ejemplo, la ciudad de Cuernavaca.
- ❖ **WAN** (Wide Area Network [Red de Area Ancha]) Se constituyen cuando se comunican un conjunto de LAN's o de MAN's en un área geográfica mayor, por ejemplo 2 ciudades.
- ❖ **GAN** (Global Area Network [Red de Area Global]) Son redes muy amplias que pueden ser de tipo nacional o internacional. Ejemplo: Internet.

Por la técnica de conmutación:

- ❖ Una **red de conmutación de circuitos** establece una ruta de comunicación extremo a extremo antes de que se envíen cualquier conjunto de datos. Esta ruta contiene un mensaje especial de señalización enviado por el nodo origen al nodo destino.

- ❖ Una *red de conmutación de mensajes* no establece anticipadamente una ruta entre el que envía y el que recibe. En su lugar, cuando el que envía tiene listo n bloque de datos, éste se almacena en la primera central de conmutación, (es decir, un Procesador de Intercambio de Mensajes, IMP) para expedirse después dándose sólo un salto a la vez.
- ❖ Una *red de conmutación de paquetes* difiere de la red de conmutación de mensaje en que los mensajes grandes se descomponen primero en bloques de tamaño fijo llamados paquetes.

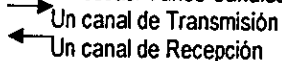
Por su sistema de transmisión:

- ❖ En las *redes de banda base*, la señal se transmite sin modulación. En consecuencia, cada vez que se realiza una transmisión se utiliza todo el ancho de banda del medio. Por ello, deben emplearse técnicas de medio compartido para que éste pueda ser utilizado por múltiples estaciones.

Características: Un solo canal, bajo costo, se modula y demodula la señal, utilizada por los estándares actuales de redes locales.

- ❖ Las *redes de banda ancha* utilizan señales analógicas moduladas. El ancho de banda total del cable se puede dividir, mediante técnicas de multiplexación por división de frecuencias (FDM), en grupos de canales de banda más estrecha. Las ventajas principales de las redes de banda ancha son la posibilidad de integrar todo tipo de señales en un solo medio y la gama de distancias que puede abarcar. Sin embargo, la utilización de módems entraña un coste superior y una planificación e instalación de la red más elaboradas.

Características: Varios Canales Paralelos, multiplexaje por frecuencia



3.3 VENTAJAS DE ESTABLECER UNA RED

A continuación se listan algunas de las aplicaciones más comunes de las LAN dentro de una organización.

Compartición de archivos y programas. Las versiones de red de muchos paquetes populares de software están disponibles, con un ahorro considerablemente de costos cuando se compara con la compra de copias de licencias individuales. Los archivos de programas y datos se almacenan en un servidor de archivos y acceden a ellos muchos usuarios de red con lo que se ahorra espacio en los discos locales de los equipos de cómputo.

Compartición de recursos de red. Los recursos de red incluyen impresoras, módems, cd's, trazadores gráficos y dispositivos de almacenamiento. La red ofrece un enlace de comunicación que permite a los usuarios compartir estos dispositivos.

Compartición de bases de datos. Un sistema gestor de bases de datos es una aplicación ideal para una red. Una característica de red llamada bloqueo de registro permite que múltiples usuarios accedan simultáneamente a un archivo sin corromper los datos. El bloqueo de registro asegura que dos usuarios no editen el mismo registro a la vez.

Expansión económica de la base del PC. Las redes proporcionan una forma económica de aumentar el número de computadoras en una organización. Se pueden enlazar económicas estaciones trabajo sin disco a una red que utilice la unidad de disco duro del servidor para el arranque y el almacenamiento.

Grupos de trabajo. Una red ofrece un medio de crear grupos de usuarios que no se localicen necesariamente en el mismo departamento. Los grupos de trabajo facilitan nuevas estructuras colectivas "planas", donde las personas de diversos y remotos departamentos pertenecen a proyectos de grupos especiales.

Correo electrónico. El correo electrónico permite que los usuarios se comuniquen fácilmente entre ellos. Los mensajes enviados a los "buzones" para que el receptor los lea en el momento oportuno.

Software de grupos y de flujo de trabajo. Se diseñó específicamente el software de grupos y el software de flujo de trabajo para las redes y para aprovechar las ventajas en los sistemas de correo electrónico que ayudan a los usuarios a colaborar en proyectos, programas y procesamiento de documentos.

Administración centralizada. Una red ofrece un camino a los servidores centralizados y a sus datos, junto con otros recursos. Cuando los dispositivos se localizan en un lugar las modernizaciones de hardware, las copias de seguridad de software, el mantenimiento y la protección del sistema son mucho más fáciles de administrar.

3.4 COMPONENTES DE UNA RED

La figura 3.1 ilustra los componentes que integran una red. Una red de computadoras consta tanto de hardware como de software. El hardware incluye a equipos de cómputo, tarjetas de interfaz de red y el cable que las une. Los componentes software incluyen sistemas operativos, protocolos de comunicación y controladores de la tarjeta de la interfaz de red del servidor.

Sistema operativo de red. El sistema operativo de red es el que rige y administra los recursos y lleva todo el control de seguridad de estos. En una red par a par, cada nodo de la red ejecuta un sistema operativo con el soporte de conexión incorporado, el cual permite que los usuarios compartan archivos y periféricos. Normalmente también se incluyen características de seguridad y administración. El sistema operativo de red para una red dedicada, se ejecuta en servidores que ejecutan el software para su comunicación con los clientes.

Servidores. Un servidor es un programa ejecutándose en un equipo de cómputo que presta un servicio en particular a los usuarios de la red, todo esto bajo el control de un sistema operativo. Puede haber los siguientes tipos de servidores:

- ❖ **Servidor de Archivos.** Proporciona servicios de almacenamiento y recuperación de archivos, incluidas las utilidades de seguridad que controlan los derechos de acceso a los archivos.
- ❖ **Servidor de Correo Electrónico.** Ofrece los servicios de correo electrónico de corporación extensa o local y traducción entre distintos sistemas de correo.
- ❖ **Servidor de comunicaciones.** Permite los servicios de conexión en sistemas de computadora central o de minicomputadoras, o en sistemas y redes de computadoras remotas por medio de enlaces de área extensa.
- ❖ **Servidor de Bases de Datos.** Un servidor dedicado que administra las peticiones y respuestas del usuario de base de datos.
- ❖ **Servidor Web.** *W W W (World Wide Web).* Publicación de documentos de hipertexto.

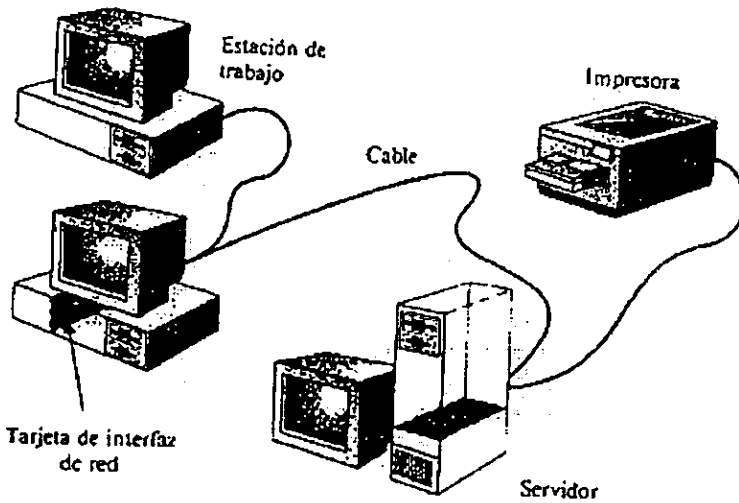


Figura 3.1 Componentes de una red

Tarjetas de la interfaz de red (NIC's). Las tarjetas de la interfaz de red (NIC's, Network Interface Cards) son adaptadores instalados en una computadora que ofrece un punto de conexión de red. Las NIC's permiten empaquetar la información y transmitirla a cierta conexión a la red. Las NICs permiten empaquetar la información y transmitirla a cierta velocidad y de acuerdo con características determinadas de envío. Estas varían según la topología y el protocolo de red que se este utilizando. Cada NIC se diseña para un tipo de red específica. Como Ethernet, Token Ring, FDDI, Arcnet y demás. Operan en el nivel físico de la pila de protocolos de interconexión de sistemas abiertos (OSI, Open Systems Interconnection) y proporcionan un punto de acoplamiento para un tipo específico de cable como los cables coaxiales, de par trenzado o de fibra óptica. El cable de red se une a la parte posterior de la NIC.

Sistemas clientes (nodos o estaciones de trabajo). Sistemas clientes o estaciones de trabajo es un término general que se aplica a las computadoras conectadas a las redes. Estos sistemas se unen a la red por medio de tarjetas de interfaz de red. El sistema operativo que se ejecuta en la estación de trabajo puede incluir el software ya incorporado para soportar las tarjetas, o puede ser necesario cargar el software del cliente. El software del cliente redirecciona las peticiones de red de los usuarios o las aplicaciones al servidor.

Sistema de cableado. El sistema de cableado de red es el medio físico que conecta juntos servidores y estaciones de trabajo. El sistema de cableado es la columna vertebral de cualquier sistema de red, ya que lleva la información de un nodo a otro.

Recursos y periféricos compartidos. Los recursos y periféricos compartidos incluyen dispositivos de almacenamiento unidos al servidor, unidades de disco óptico, impresoras, módems, trazadores gráficos y otros equipos disponibles que utiliza cualquier usuario autorizado de red.

3.5 MEDIOS FÍSICOS DE CONEXIÓN DE RED

Los medios de transmisión soportan la propagación de señales de onda acústica, electromagnética y de luz. Dependiendo del tipo de señal que se tenga que transmitir, es el tipo de medio a utilizar. Para los medios de transmisión se han establecido estándares con los que se rigen los medios, incluyen la velocidad de transmisión de los datos, las características eléctricas y la topología de cableado.

3.5.1 Cable coaxial

El cable coaxial consta de un núcleo de cobre sólido rodeado por un aislante, una especie de combinación entre pantalla y cable de tierra y un revestimiento protector exterior. Los cables coaxiales pueden ser de varios tipos y anchos su principal característica es que pueden transportar una señal eléctrica a mayor distancia entre más grueso es el conductor. El cable grueso suele ser más caro y menos flexible. Por tal razón, cuando tiene que colocarse en instalaciones en donde ya existe canales para cableado o conductos con espacio reducido y, sobre todo limitado en las esquinas o dobles, resulta más conveniente utilizar el cable delgado debido a que las nuevas instalaciones de ductos para cable por lo general son muy costosas. En general se pueden citar como las principales ventajas de este tipo de cable las siguientes:

- ❖ Transmisión de voz, video y datos, compatibilidad con Ethernet y Arcnet, ancho de banda de 10Mbps, distancias hasta de 600 metros sin necesidad de repetidores. buena tolerancia a interferencias debidas a factores ambientales. (RG58U.-Thin (Ethernet delgado), RG11.- Thick (Ethernet grueso), RG62.-Arcnet)

3.5.2 Cable de par trenzado

El cable de par trenzado consta de dos hilos de cobre aislados que se trenzan uno alrededor del otro. Los hilos se encuentran trenzados por pares como se muestra en la figura 3.2 de manera que cada par forma un circuito que puede transmitir datos. Un cable consta de un haz de uno o mas pares trenzados rodeados por un aislante. El par trenzado sin blindaje (UTP, Unshielded Twisted Pair) consta de pares de alambre de cobre trenzados aislados con PVC y es usual en la red telefónica, el par trenzado blindado (STP, Shielded Twisted Pair) se define con un blindaje individual por cada par, mas un blindaje que envuelve a todos los pares y proporciona protección frente a la diafonía, y el par trenzado blindado general (FTP, Foiled Twisted pair) cuenta con un blindaje de aluminio que envuelve a los pares para dar una mayor protección contra las emisiones electromagnéticas del exterior. Precisamente en estos cables, es el trenzado el que previene los problemas de interferencia. Son posibles velocidades de transmisión elevadas (100Mbps) si se ha instalado cable de grado de datos (de categoría 5). El trenzado debe mantenerse durante todo el recorrido del cable entre los puntos extremos de la conexión. El cable de par trenzado se utiliza usualmente en redes con topología Ethernet y Token Ring.

Los cables UTP y STP para redes de tipo Ethernet y Token Ring deben cumplir con las siguientes especificaciones: Tener una impedancia entre 58 y 115 Ohms a 10Mhz, presentar una atenuación máxima de 11dB/110 metros a 10Mhz o una atenuación mínima de 7.2 dB/110 metros a 5 Mhz.

Los cables de par trenzado tienen como principales ventajas:

- ❖ Tecnología conocida, compatibilidad con Ethernet y Token Ring, ancho de banda de 10Mbps a 100Mbps, distancias de hasta 110 metros con cables UTP y de hasta 500 metros en caso de cable STP, buena tolerancia a interferencias debidas a factores ambientales.

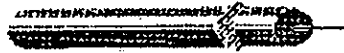
3.5.3 Cable de fibra óptica

El cable de fibra óptica utiliza fotones en la transmisión de las señales digitales. Las señales de las computadoras se envían a través de la fibra óptica mediante la conversión de los 1's y 0's en códigos ópticos. En un extremo de un cable de fibra óptica se sitúa un diodo emisor de luz y un fotodetector percibe el destello de la luz en el otro extremo, convirtiéndolo en una señal eléctrica. Un cable de fibra óptica se fabrica con vidrio puro, que no impone resistencia alguna al paso de los fotones por él. El núcleo óptico del cable de fibra óptica consiste en dióxido de silicio puro.

**Par
Trenzado**



Coaxial



Fibra Óptica

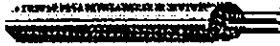


Figura 3.2 Principales medios físicos de conexión

La tabla 3.1 se muestran las características comparadas de los cuatro tipos de cables.

	PAR TRENZADO	COAXIAL DE BANDA BASE	COAXIAL DE BANDA ANCHA	FIBRA ÓPTICA
ANCHO DE BANDA	BAJA	MODERADA	ALTA	MUY ALTA
INSTALACIÓN	SENCILLA	FÁCIL	FÁCIL	DIFÍCIL
LONGITUD	BAJA	MODERADA	ALTA	MUY ALTA
COSTO	BARATA	MODERADA	CARA	MUY CARA
FIABILIDAD DE LA TRANSMISIÓN	BAJA	ALTA	ALTA	MUY ALTA
INTERFERENCIAS	ALTA	MODERADA	BAJA	NINGUNA
SEGURIDAD	BAJA	BAJA	MODERADA	ALTA
TOPOLOGÍA	BUS ESTRELLA ANILLO	BUS	BUS ESTRELLA	ESTRELLA ANILLO

Tabla 3.1 Tabla comparativo de cables

3.6 ARQUITECTURA DE RED

La arquitectura de una red está definida por la topología, el método de acceso al medio y los protocolos de comunicación utilizados. Antes de que cualquier estación de trabajo pueda acceder a los servicios de la red, debe establecer sesiones de comunicación con otros nodos en la red. El método de acceso al medio de una red define cómo una estación de trabajo tiene acceso a los niveles físicos de la red (cable) para transmitir información. Los protocolos son las reglas y procedimientos que los sistemas utilizan para comunicarse unos con otros sobre la red.

3.6.1 TOPOLOGÍA

Se puede considerar una topología de una red como un mapa de la distribución de cables. La topología define cómo se organiza el cable en las estaciones de trabajo individuales y desempeña un papel importante en la decisión que se tome sobre el cable. Se debe considerar la topología de una red cuando se tomen decisiones sobre qué tipo de red instalar. Como se describe aquí, la topología equivale a cómo se instalará el cable a través de los ductos y estructura de un edificio.

Hay dos categorías de diseño de topologías, que dependen del tipo de red, esto es, si la red es una Red de área local (LAN) o una conexión de inter-redes con ruteadores y enlaces de telecomunicaciones (Red de área extensa (WAN)). Las topologías LAN son fundamentalmente las divisiones en subredes de una interred.

En el mercado actual existe una gran variedad de Topologías Físicas, para entender como funcionan todas estas, es importante conocer como funciona lógica y físicamente los tipos básicos, y con base en esto, entender las características que cualquier topología en el mercado pueda ofrecer.

Existen cuatro tipos básicos de topologías: Estrella, Bus, Anillo, Árbol, las cuales se muestran en la figura 3.3.

Topología Estrella. En este tipo de conexión, el elemento central es el servidor con sus Periféricos. Se mantiene preguntando constantemente a cada estación de trabajo mediante comunicación exclusiva y por turno si desea transmitir información, y en caso afirmativo, la atiende y al terminar prosigue con otra su interrogatoria permanente.

Para este caso de pregunta a una –respuesta- pregunta a la siguiente, la regla de comunicación se le denomina Protocolo POLLING (poleo), y es comúnmente empleada en la "minis".

Topología de Bus. Esta conexión se considera que es la más sencilla de todas, donde las Estaciones de Trabajo y el Servidor, están enlazadas por un solo cable (coaxial o par trenzado), y la información viaja en ambos sentidos, por lo que es necesario prevenir colisiones. Por ello el protocolo apropiado es el CSMA/CD (Carrier Sense Múltiple Acces / Colisión Detection). Con este protocolo, cualquier nodo trasmite y espera a que se le confirme que la información fue recibida correctamente, de otra forma, se detecta la posible colisión, espera un tiempo a que el canal esté desocupado y la información se transmite nuevamente.

Topología de Anillo. En esta conexión, la información viaja ordenadamente en un solo sentido a través de un solo cable, describiendo un ángulo de 360° en cuyo anillo imaginario, están conectadas en serie las estaciones de trabajo y al SERVER.

Una señal llamada Token (receptáculo, a modo de estafeta) va circulando por la red y pasando por cada estación, si la primera resultó ser la solicitante, previa identificación entrega la información, de

lo contrario la deposita en "sobre cerrado" para que ésta a su vez la envíe a la siguiente, llevando la consigna de entregarla hasta identificar a la solicitante. Cada estación de paso, cuando más, colecta información adicional enviándola a la siguiente y así se le pasa la señal cerrando ciclos "circulares" por ello el protocolo apropiado para este caso se conoce como Token Passing.

Topología de Árbol. Desde el punto de vista físico, esta conexión como se dijo anteriormente, es combinada y es una opción más para implementar REDES, según las necesidades del usuario. Trabaja el protocolo Token Passing, tarjeta Arcnet y repetidores tanto pasivos como activos.

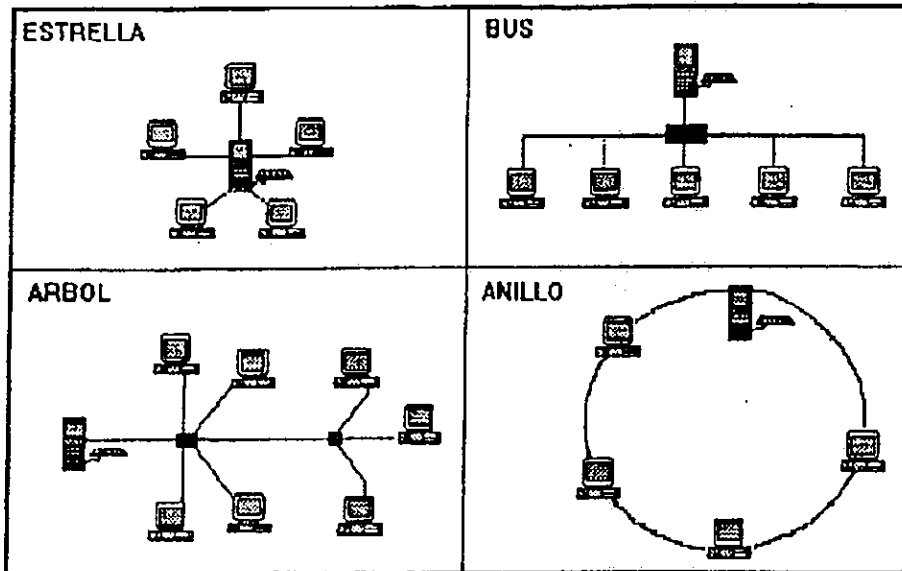


Figura 3.3 Topologías físicas básicas

Anillo configurado en estrella. Una red en anillo donde se pasan las señales de una estación a otra en círculo. La topología física constituye una estrella en la que las estaciones de trabajo se ramifican desde los concentradores.

Configuración estrella / bus. Una red que tiene grupos de estaciones de trabajo configurados en estrella conectados con cables de conexión largos de bus lineales.

Una inter-red consta de LAN's departamentales o de estaciones de trabajo que se interconectan con puentes y ruteadores. En un entorno local, tal como un edificio, frecuentemente se utiliza un cable soporte (backbone), pero para construir redes de área extensa se utilizan los servicios públicos, como aquellos ofrecidos por las compañías telefónicas. Las tres topologías principales, las cuales se discuten a continuación.

Red soporte. Típicamente encontrada en entornos de oficina o campus en los que los departamentos o edificios se interconectan a través de los cables soportes. Los puentes o ruteadores gobiernan el flujo de tráfico entre las subredes unidas y el soporte.

Red de malla. Los ruteadores se interconectan con otros ruteadores. La topología se puede configurar localmente, pero frecuentemente se encuentra en redes de área metropolitana o extensa que conectan oficinas remotas mediante enlaces de telecomunicaciones. Se utilizan los ruteadores para elegir el trayecto mejor y más eficiente de la fuente al destino a través de la malla.

Estrella entrelazada. Este es un paradigma de nueva topología para los sistemas de cableado estructurado en entornos de edificios o de campus. Los concentradores de departamentos se cablean en estrella a un concentrador central que gobierna al tráfico entre concentradores.

3.6.2 MÉTODOS DE ACCESO AL MEDIO

El método de acceso al medio describe cómo una estación de trabajo tiene acceso al sistema de cables. A continuación se mencionan los métodos de acceso más utilizados.

Acceso múltiple con detección de portadora / detección de colisión (CSMA /CD, Carrier Sense Multiple Access / Collision Detection). La detección de portadora implica que los nodos de la red esperan un tono de portadora en el cable y envían información cuando otros dispositivos no transmiten. Acceso múltiple significa que muchos dispositivos comparten el mismo cable. Una estación de trabajo transmite una señal y cada uno de los otros nodos de la red la oye, pero solo el nodo direccionado presta atención. Si uno o más dispositivos detectan que la red está libre e intentan el acceso simultáneo (contención), causarán una colisión. En este caso, cada estación debe retirarse y esperar un cierto tiempo antes de intentar la retransmisión.

Acceso múltiple con detección de portadora /Evitación de colisiones (CSMA /CA, Carrier Sense Multiple Access / Collision Avoidance). Este método de acceso es una variación del método CSMA /CD. Los nodos estiman cuando puede ocurrir una colisión y evitan la transmisión durante ese periodo. La implementación de este método es más barata, ya que no se requiere circuitería de detección de portadora, sin embargo, impone un retardo mayor y puede bajar el rendimiento de la red.

Método de acceso de prioridad bajo demanda. Este es un método nuevo de acceso para LAN a 100Mbps. Con el método de prioridad bajo demanda, las estaciones de trabajo pueden recibir al mismo tiempo que transmitir. Esto es posible gracias a la utilización de los cuatro pares del cable de par trenzado y del cuarteto de señalización. La prioridad bajo demanda aprovecha el diseño estructurado del cableado y cede el control de acceso a la red a un concentrador (hub) central, en lugar de confiar en que cada estación de trabajo individual determine cuándo puede acceder al cable.

Paso de testigo (Token Passing). Un testigo es un paquete especial de las redes *Arcnet*, *Token Ring* e *Interfaz de datos distribuidos por fibra (FDDI, Fiber Distributed Data Interface)*, que controla el acceso a la red. El nodo que toma el control del paquete testigo tiene derecho a comunicarse en la red. Al contrario que en las redes basadas en la contención (como *Ethernet*), las estaciones de trabajo no intentan el acceso simultáneo a la red. Sólo transmite la estación que está en posesión del testigo. El testigo se pasa alrededor de la red de una estación a la siguiente, lo que garantiza que todas las estaciones tendrán la misma oportunidad de transmitir y que sólo un paquete viajará a la vez en la red.

Por pofeo. Este método de acceso se caracteriza por contar con un dispositivo controlador central, que es una computadora inteligente, como un servidor. Pasa lista a cada nodo en una secuencia predefinida solicitando acceso a la red. Si tal solicitud se realiza, el mensaje se transmite; de lo contrario, el dispositivo central se mueve a pasar lista al siguiente nodo.

Las redes también pueden ser reconocidas como:

❖ Ethernet

Esta red fue desarrollada por Xerox Corporation para enlazar un grupo de microcomputadoras que estaban distribuidos por los laboratorios de investigación de Palo Alto California, para poder intercambiar programas y datos, así como compartir los periféricos.

En un principio, se creó para ser utilizada con cable coaxial de banda base, aunque actualmente se pueden utilizar otros tipos de cable. Si se utiliza cable coaxial grueso, se pueden tener hasta cuatro tramos de cable (unidos con repetidores) y los computadores se conectan por medio de transceptores. Se pueden conectar computadores entre tramos únicamente, con una máximo de 100 estaciones en cada tramo. Si se utiliza cable coaxial fino, no es necesario utilizar coaxial fino, no es necesario utilizar transceptores, pudiéndose conectar el cable al computador por medio de una conexión BNC en forma de T.

El número máximo de tramos es de cinco, y la longitud máxima de cada tramo es, aproximadamente, de un tercio de la longitud máxima conseguida con el cable coaxial grueso(550 metros). Así mismo, el número máximo de estaciones es de 30 por cada uno de los tres tramos en los que se pueden conectar computadores. Los datos se transmiten a una velocidad de 10 megabaudios (bytes por segundo) a una distancia máxima de dos Kilómetros.

Utiliza la topología en bus con protocolo de contienda CSMA/CD (acceso múltiple por detección de portadora con detección de colisiones), es decir cualquier estación puede intentar transmitir en cualquier momento, pero como todas utilizan un canal único, sólo una estación puede transmitir datos simultáneamente. El tamaño del bloque de datos puede oscilar desde 72 hasta 1526 bytes (con un tamaño normal de 256 bytes). Todas las estaciones tienen asignada una dirección de 48 bytes, que permite que cuando se cambia de lugar una estación no haya posibilidad de conflictos, y por tanto se puede reconfigurar completamente la red local con unos mínimos cambios en el sistema operativo.

❖ Arcnet

Es un desarrollo de Datapoint, este sistema apareció en el mercado a mediados de los 70's y es un esquema de bus de transmisión de señales codificadas.

Como ARCNET es anterior a la aparición de estándares de bus de señales, los sistemas basados en ARCNET observan algunas inconsistencias con el resto de los productos de la industria de las comunicaciones de datos, como otros esquemas con bus de señales que se desarrollaron antes de la promulgación del estándar 802.4 de la IEEE. Y no es un estándar.

❖ Token Ring

Como su nombre lo indica emplea una topología de anillo y el método de acceso con transmisión de señales. Comúnmente, las estaciones de trabajo se conectan con par trenzado blindado o no blindado, hacia un concentrador de conexiones llamado unidad de acceso a multiestaciones o MAU (unidad de acceso multiestación) que son aparatos confiables que además facilitan la instalación de la red, así como su mantenimiento.

La red original Token Ring operaba en la transmisión de sus datos a una velocidad de 4 Mbps con un máximo de 100 metros del concentrador de conexiones a una computadora y 72 estaciones que usaban cable UTP especial de IBM. Más tarde en 1989, se extendió hasta 16 Mbps si se utiliza fibra óptica. Cuando se usa par trenzado blindado (STP) se puede construir LAN mayores de hasta 260 estaciones. Una dificultad que compartían los fabricantes de hardware para redes Token Ring en común con IBM, era que el precio de lista de una tarjeta de interfase ordinaria era aproximadamente el doble de una tarjeta de interfase Ethernet, además, en la versión de 16 Mbps se requiere cable duplex trenzado aislado, lo que eleva aún más los costos de instalación.

3.6.3 PROTOCOLOS DE COMUNICACIÓN

Los protocolos de comunicación son las reglas y procedimientos utilizados en una red para comunicarse entre los nodos que conforman la red. Los protocolos controlan dos niveles de comunicación diferentes. Los protocolos de nivel superior definen cómo se comunican las aplicaciones y los protocolos de nivel inferior cómo se transmiten las señales por un cable. Cuando se definen y publican los protocolos de red, los fabricantes pueden diseñar y fabricar fácilmente productos de red que trabajen en sistemas de múltiples vendedores.

Modelo de referencia ISO-OSI

La existencia de diferentes arquitecturas de red representaba serios problemas en los sistemas de red, pues no permitía la interconexión de componentes heterogéneos. Para atacar este problema la ISO (International Estándar Organization) desarrolló un modelo denominado OSI (Open System Interconnection [Interconexión de Sistemas Abiertos]).

El modelo para la interconexión de sistemas abiertos, de sus siglas en español ISA se ha convertido en una referencia obligada para todo lo relacionado con la intercomunicación de computadoras. Como lo indica su nombre es un modelo que tiene como objetivo principal lograr la interconexión de sistemas abiertos, ello es posible mediante la definición de un conjunto de mecanismos que hacen posible esta interconexión. Se busca además proteger a los consumidores contra monopolios por parte de los fabricantes de una línea de productos.

El modelo OSI estructura en siete capas, mostrada en la figura 3.4 es el fenómeno global de la comunicación, es un marco hoy en día obligado y universalmente aceptado. A continuación se analizan una por una las diferentes capas, estudiando sus funciones y características.

Capa 1: FÍSICA

Esta capa engloba los medios mecánicos, eléctricos, funcionales y de procedimiento para activar, mantener y concluir una conexión física que permita la transmisión de bit entre 2 entes de enlace de datos (computadoras). Es la encargada de la activación y desactivación física de la conexión.

Capa 2: ENLACE

Un enlace de datos se establece siempre entre dos puntos físicos de conexión del sistema. Esta capa se encarga de proporcionar los elementos necesarios que permiten la transmisión de información, a través de un enlace de datos que sea confiable; también proporciona servicios como: administración de la trama de información, control de flujo de datos, control de secuencia y aviso en caso de errores no recuperables. En el caso de una red de datos de conmutación de paquetes, el nivel de enlace es responsable de la transferencia fiable de cada paquete al nivel de la red. La CCITT (Comité Consultivo Internacional Telegráfico y Telefónico, de sus siglas en francés) ha definido dentro de la recomendación X.25 un subconjunto del protocolo HDLC (High Level Data Link Control) como protocolo de la capa de enlace.

Capa 3: RED

El objetivo de esta capa es proporcionar los servicios de ruteo necesarios para que la información fluya entre los diferentes entes del nivel de red. Busca establecer, mantener y liberar las comunicaciones o conexiones necesarias para la transferencia de los paquetes de datos. Un servicio muy importante en las redes de conmutación de paquetes es el denominado llamada virtual que permite establecer a través de la red un gran número de comunicaciones simultáneas. Esta capa enmascara todas las particularidades del medio real de transferencia.

Es el responsable del encaminamiento de los paquetes de datos a través de la red. Cada vez que un paquete llega a un nodo, el nivel tres de ese nodo deberá seleccionar el mejor enlace de datos por el que envíe la información.

Las unidades de datos de esta capa son los paquetes de datos que deberán ir provistos de la dirección de destino. Además, son funciones de esta capa la definición de la estructura de datos de los paquetes, las técnicas de corrección de errores, la entrega en secuencia correcta al nivel de transporte de los paquetes recibidos, así como otras de reiniciación y control de flujo.

Capa 4: TRANSPORTE

Esta capa se encarga de proporcionar el servicio de transporte de la información entre puntos finales de la red. Proporciona 3 tipos de servicio: Establecimiento de la conexión, transferencia de la información y terminación de la conexión. Es responsable de una transferencia de datos transparente entre dos entidades de la capa de sesión, liberando a dichas entidades de todo lo referente a la forma de llevar a cabo dicho transporte. Los protocolos que maneja esta capa suelen llamarse *protocolos end-to-end*, o protocolos entre puntos finales, debido a que esta capa se encarga de realizar una conexión lógica entre dos estaciones de transporte de los sistemas informáticos que quieren comunicarse, independientemente de donde se encuentren éstos. Esta capa puede multiplexar varias conexiones de transporte dentro de una única conexión de red, o puede por el contrario, repartir una conexión de transporte entre varias conexiones de red.

Capa 5: SESIÓN

Se ocupa del establecimiento y terminación de corrientes de datos entre dos nodos de la red. Cada vez que dos entes de sistemas distintos desean establecer una comunicación se establece una sesión; la sesión asegura el diálogo entre ellos.

El control de diálogo busca asegurar que los mensajes lleguen a su destino en el orden indicados. Su función es establecer y gestionar un camino de comunicación entre dos procesos del nivel de aplicación.

La información que se envía se fracciona en pedazos y se generan unos puntos de sincronización. En caso de interrumpirse la sesión por alguna falla en la comunicación, los datos pueden ser recuperados y se conoce con precisión por ambos interlocutores hasta que punto de sincronización la comunicación fue correcta. Al reanudarse la sesión no será necesario transmitir de nuevo toda la información, sino solamente a partir del punto donde se quedó el último paquete de información válido. En una sesión hay un diálogo entre máquinas, entre procesos y el protocolo debe regular quién "habla" cuándo y por cuánto tiempo. Estas reglas necesitan ser acordadas cuando la sesión comienza. Esta capa también es responsable de dirigir el diálogo entre las entidades de la capa de presentación.

Capa 6: PRESENTACIÓN

Proporciona servicios a los entes de esta aplicación orientados a interpretar la información intercambiada. Algunos de estos servicios son: Conversión de códigos de datos, formato de los datos y órdenes de control, forma de transferir la información y empaquetamiento. Esta capa se ocupa de la representación de los datos usados por los procesos de aplicación del nivel siete. Por lo tanto, si es necesario, realizará la transformación de los datos que reciba de o para el nivel de aplicación. Esto en el caso de que el proceso originador y el receptor tuvieran versiones de datos sintácticamente diferentes, pero también puede darse el caso de que, para una determinada aplicación distribuida exista un conjunto de caracteres normalizados diferentes de los del originador y el receptor, en cuyo caso, las capas de presentación respectivos deberán de hacer las transformaciones necesarias. Otra función que se puede encargar a la capa seis, es la de velar por la seguridad de los datos, siendo responsables de la encriptación de mensajes confidenciales antes de su transmisión. La función inversa será realizada por la capa de presentación del sistema receptor.

Capa 7: APLICACIÓN

Proporciona el soporte necesario para que las aplicaciones que así lo requieran puedan acceder a la red. Esta capa se preocupa de proporcionar un conjunto de servicios distribuidos a los procesos de aplicación de los usuarios. El usuario se comunicará directamente con esta capa a través de la correspondiente interfase o agente de usuario. Actualmente se están desarrollando una serie de normas y recomendaciones tendientes a tipificar cada uno de estos servicios o aplicaciones distribuidas.

Entre los más conocidos podemos citar: Servicio de mensajería (correo electrónico), servicio de almacenamiento y recuperación de documentos, servicio de directorio, etc.

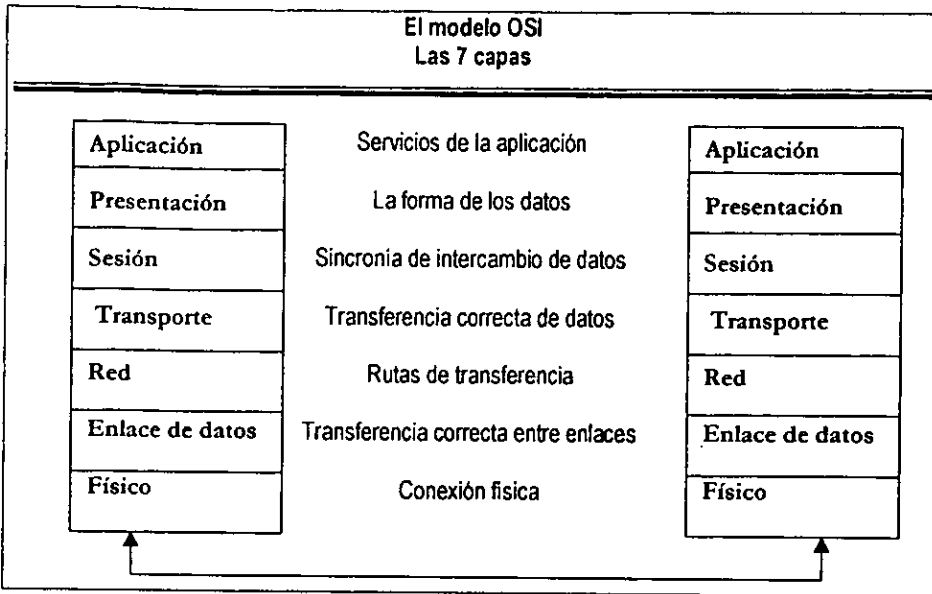


Figura 3.4 Modelo OSI

3.6.3.1 TCP / IP

Los objetivos de desarrollo para el grupo de protocolos TCP / IP (Transmission Control Protocol / Internet Protocol) fueron el permitir comunicaciones entre varios sistemas independientes y multiprovedores. En 1983, los protocolos TCP / IP se convirtieron en el mecanismo de transporte oficial para la red del departamento de los Estados Unidos DOD (Department of Defense), que evolucionó en un sistema de redes interconectadas expandidas por todo el globo terráqueo (Internet).

Las principales características de la arquitectura TCP / IP están implementadas bajo los siguientes objetivos:

- ❖ Protocolos sin conexión (cada paquete se transmite independientemente uno de otro).
- ❖ Rutas dinámicas.
- ❖ Protocolos de transporte con funciones de seguridad.
- ❖ Un conjunto de programas de aplicaciones comunes.

Protocolo de control de transmisión (TCP, Transmisión Control Protocol)

Se desarrolló el protocolo TCP original como un modo de interconectar redes con el uso de muchos tipos diferentes de métodos de transmisión. Para acomodarse a esos medios, se creó el concepto de pasarela (más tarde llamada roteador) en el que los paquetes de una red se encapsulaban en un paquete que contenía la dirección de la otra pasarela. El paquete podría ser reempaquetado y direccionado a varias pasarelas, antes de alcanzar su destino final. Se utilizó este método de encapsulación por varias razones, pero la más importante de ellas es que los diseñadores no querían que los propietarios de diversas redes alterasen sus esquemas de interconexión de red. Se asumió que cada red implementaría sus propias técnicas de comunicación.

Protocolo Internet (IP, Internet Protocol)

IP es un protocolo de comunicación sin conexión que por si mismo proporciona un servicio de datagramas. Los datagramas son paquetes independientes de información, que se envían a través de los ruteadores en función de su dirección y a la información de la tabla de encadenamiento contenida en los ruteadores. Los datagramas se pueden direccionar a un único nodo o a múltiples nodos. No hay control de flujo, reconocimiento de recepción, comprobación de error ni secuenciamiento. Los datagramas pueden atravesar trayectos diferentes para llegar al destino y así llegar fuera de secuencia. La estación receptora es responsable del resecuenciamiento y de determinar si se han perdido los paquetes. IP maneja la congestión simplemente con el descarte de paquetes. Los protocolos del nivel superior cuidan el resecuenciamiento y el manejo de errores, de modo que no lo tiene que hacer IP. Por eso, IP es rápido y eficiente, y muy adecuado para las redes y los sistemas de telecomunicaciones modernos, que ya proporcionan un servicio relativamente fiable. IP trabaja con diversas redes de área local y extensa.

Direccionamiento IP

Cada nodo en una red TCP/IP requiere una dirección numérica de cuatro bytes (32 bits) que identifica una red y un anfitrión local (host) o nodo de red. Esta dirección se compone de cuatro números menores a 255 (FF en Hexadecimal) separados por puntos, por ejemplo, 191.31.140.115 en la mayoría de los casos, el administrador de la red establece esas direcciones cuando instala nuevas estaciones de trabajo, sin embargo, en algunos casos es posible, para una estación de trabajo, consultar un servidor para alguna asignación dinámica de dirección cuando arranca.

Como ya se mencionó anteriormente, TCP establece sesiones orientadas a la conexión las cuales requieren una fase de establecimiento, una fase de desconexión y mucha supervisión y El modelo de la arquitectura TCP / IP se muestra en la figura 3.5.

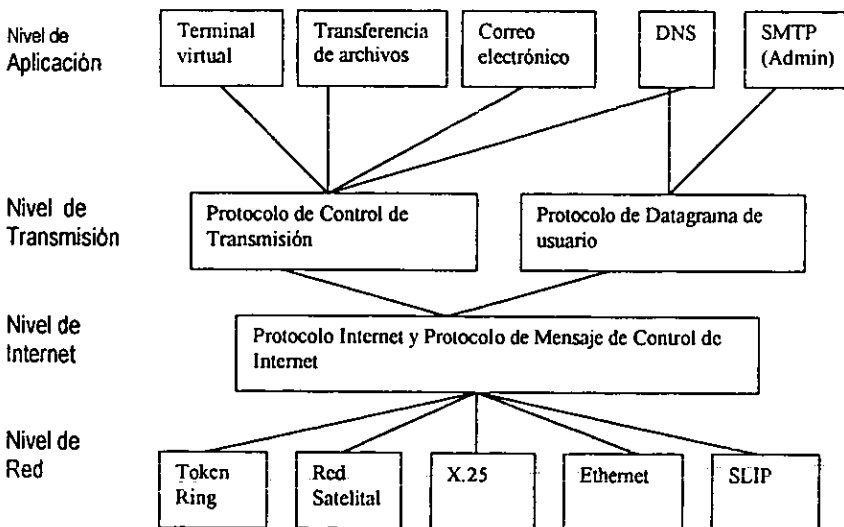


Figura 3.5 Arquitectura de TCP / IP

3.6.3.2 Relación entre la arquitectura del modelo OSI y la de TCP / IP

La relación que existe entre las arquitecturas OSI y TCP / IP se dan la tabla 3.2. La diferencia entre estas arquitecturas se da primeramente por la filosofía de las capas arriba del nivel de transporte. Aquí, el modelo OSI cuenta con dos capas adicionales, la capa de sesión y la capa de presentación. Por otro lado, la arquitectura TCP / IP ve las funciones de las capas de sesión y presentación como parte del nivel de aplicación. El requerimiento para la conectividad universal son soportados por el protocolo Internet (IP): En el nivel de TCP / IP hay sólo un protocolo al cual todos los servidores y nodos participantes pueden entender. El protocolo de Mensajes de Control Internet (ICMP), transporta información de errores y de diagnóstico para el IP.

TCP / IP	MODELO OSI
Aplicación	Aplicación Presentación
TCP	Sesión Transporte
IP	Red
Red	Enlace Física

Tabla 3.2 Relación entre el modelo OSI y el conjunto de protocolos TCP / IP

3.7 DISPOSITIVOS DE INTERCONECTIVIDAD

Los dispositivos de interconectividad de red son aquellos que permiten que una red crezca y se clasifican en cuatro grupos según su correspondencia a capas específicas del modelo OSI. Esta correspondencia se muestra en la figura 3.6.

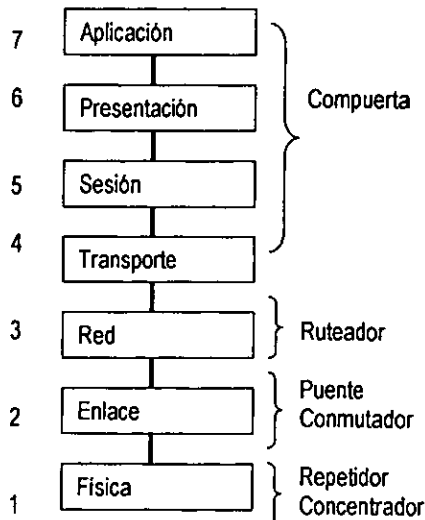


Figura 3.6 Dispositivos de interconectividad y el modelo OSI.

3.7.1 Repetidores (Repeaters)

Los repetidores operan en la capa física (capa 1 de OSI) e interconectan a dos segmentos de la misma red que estén geográficamente separados. Un repetidor proporciona servicios de regeneración de señales. Cuando una señal eléctrica pasa por medio de transmisión (por ejemplo, cable coaxial), la señal se degenera en proporción directa a la longitud e impedancia del cable. Esta pérdida de señal se conoce como atenuación. Un repetidor conecta dos redes idénticas, por ejemplo Ethernet y las protege contra atenuaciones. El repetidor amplifica la señal recibida por un segmento de cable y retransmite (repite) la misma señal hacia otro segmento de cable. El repetidor no realiza filtrado de paquetes, por tanto, ambas redes comparten el mismo ancho de banda.

3.7.2 Concentradores (Hubs)

Los concentradores son dispositivos que trabajan en la capa física (capa 1 de OSI) y en su forma más simple, un concentrador es un dispositivo que centraliza la conexión física y lógica de las estaciones de trabajo, los concentradores se diseñan específicamente para cada tipo de red (Ethernet, Token Ring).

3.7.3 Puentes (Bridges)

Los puentes son dispositivos que operan en la capa de enlace de datos (capa 2 de OSI) e interconectan a dos redes de tal manera que los usuarios piensen que están conectados a una sola red. Los puentes son más eficientes que los repetidores en esta tarea, debido a que los primeros envían solamente los datos que son necesarios para la otra red. Esto reduce bastante la cantidad de tráfico que se transmite entre las dos redes, pero requiere que ambas sean del mismo tipo (por ejemplo, Ethernet con Ethernet).

3.7.4 Conmutadores (Switches)

En una red estándar, cada nodo sigue un esquema de control de acceso a medios (MAC), como Ethernet o Token Ring, que permite compartir el mismo ancho de banda. Entre más nodos hay en una LAN, será menor la cantidad de tiempo que cada uno de ellos va a necesitar para las transmisiones. Un conmutador aísla y cataliza los datos, de modo que cada nodo tiene acceso a un mayor ancho de banda y una mejora radical en el desempeño, éste dispositivo sigue la misma filosofía del puente. Un conmutador es la manera más sencilla y más económica para mejorar el desempeño de una red muy ocupada, dado que permite el aumento del ancho de banda en cada puerto, así como la creación de redes virtuales (VLAN). El conmutador opera en la capa de enlace de datos (capa 2 de OSI).

3.7.5 Ruteadores (Routers)

Los ruteadores son dispositivos que operan en la capa de red (capa 3 de OSI) y pueden interconectar a dos redes diferentes (por ejemplo Ethernet con Token Ring). Sus capacidades van mucho más allá de las de un puente. Debido a que los ruteadores no son transparentes, los usuarios que desean tener acceso a los recursos o a otra red, deben dar una dirección de destino. Esta dirección es igual a la dirección de Internet. Los ruteadores utilizan la dirección Internet para encontrar ese destino siempre que esté en la red. Ya que los ruteadores realmente "hablan" con los dispositivos en esa red. Esto hace que los ruteadores sean más costosos, más complejos de establecer y mantener, y más lentos para establecer la conexión que los puentes. Sin embargo, son el producto de interconexión a utilizar cuando el requerimiento es establecer conexiones de extremo

en cualquier parte de la red, independientemente del tipo de red (por ejemplo, X.25, Ethernet o Token Ring).

3.7.6 Compuertas (Gateways)

Son dispositivos que operan en la capa de aplicación (capa 7 de OSI) e interconectan a redes de diferentes tipos. Realizan la interconexión de la manera más directa posible. Literalmente convierten la salida de los dispositivos de una red en el protocolo que entienden los dispositivos de la otra red. A diferencia de los ruteadores, los gateways de hecho traducen un protocolo a otro. Los ruteadores agregan otro nivel de información de dirección de datos, de modo que estos se pueden enviar a otra red. Cuando se utiliza un gateways para interconectar diferentes redes, los dispositivos de una red se comunican con los de otra como si fueran de la misma clase.

3.7.7 Módems

Los módems (moduladores / demoduladores) son equipos de comunicación de datos (DCE, Data Communication Equipment) que proporcionan conexiones para las computadoras sobre la red telefónica pública conmutada (PSTN, Public Switched Telephone Network). Los módems convierten (modulan) las señales digitales a lo largo de las líneas telefónicas. Un módem en el otro extremo del enlace demodula la señal convirtiéndola de nuevo a bits digitales.

3.8 SISTEMAS OPERATIVOS DE RED

El sistema operativo y los protocolos que proporcionan servicios de comunicación y de red, definen el entorno de una red. Hay dos tipos básicos de sistemas operativos de red:

- ❖ **Par a par.** Este es un sistema operativo que permite a los usuarios compartir recursos en sus computadoras y acceder a los recursos compartidos en otras computadoras. Par a par implica que todos los sistemas tienen el mismo rango en la red. Ningún sistema es "esclavo" de otro.
- ❖ **Servidor Dedicado.** En un sistema operativo de servidor dedicado, una o más computadoras (servidores), regulan la comunicación entre los equipos que requieren de los servicios (clientes). A diferencia de las redes par a par, el funcionamiento de los clientes, depende de sus permisos de acceso al servidor.

El sistema operativo de la red se engloba en dos componentes básicos. El sistema operativo de red del servidor mismo y el sistema de la estación de trabajo. El sistema operativo del servidor de red se ejecuta dentro de la máquina del servidor y procesa todos los servicios. El sistema operativo de la red normalmente es proporcionado por el fabricante de equipo original (OEM, Original Equipment Manufacturer), los componentes de la estación de trabajo se ejecutan en esta, establecen la conexión de la red, y el servidor y controlan el flujo de las comunicaciones. El sistema operativo del servidor de red se puede dividir en cinco subsistemas básicos: el núcleo de control (Control Kernel), las interfaces de la red, los sistemas de archivo, las extensiones del sistema y los servicios del sistema.

El control kernel o el núcleo de control es el corazón del sistema operativo, el cual coordina los diferentes procesos de los otros subsistemas.

De manera central, en el diseño del kernel están los procesos que optimizan el acceso a los servicios para la actividad del usuario. El kernel puede distribuir la actividad del usuario tan uniformemente como sea posible a través de los sistemas de disco y de cualquier dispositivo de entrada / salida, de tal manera que no se favorece a un usuario o grupo de usuarios obteniendo un mejor funcionamiento. El kernel también es responsable de mantener la información de estado de muchos procesos, es un componente de las facilidades de administración de la red. El reporte de error, la inicialización del servicio y la terminación del servicio, comúnmente se rigen por los servicios del kernel.

Las interfaces de red apoyan las tecnologías que son la implantación real del medio de la red. En los sistemas operativos de red más completos, las interfaces de red pueden cargarse y descargarse en forma dinámica, y se pueden instalar, simultáneamente, múltiples interfaces de diferentes tipos y marcas. Los componentes de la interface de red también manejan los protocolos de bajo nivel de la red y proporcionan el traslado básico entre estos protocolos cuando se requieren servicios de puenteo.

Los sistemas de archivo (file server) son los mecanismos mediante los cuales, se organizan, almacenan y recuperan los datos, a partir de los sistemas de almacenamiento disponibles para el sistema operativo de red. Estos sistemas pueden ser subsistemas de alta velocidad, tales como discos duros o discos de RAM, o dispositivos de archivo con frecuencia se instrumentan con el concepto de aplicabilidad universal, es decir, el sistema de archivos puede presentarse como compatible con cualquier expectativa de aplicación de protocolos de entrada / salida del archivo.

Las extensiones del sistema operativo de red definen lo "abierto" del sistema. Las extensiones que comúnmente se ofrecen en los sistemas operativos de red, por lo general son manejadores de productos de alto nivel que efectúan operaciones, tales como el traslado entre los protocolos de acceso de archivos que requieren los diferentes sistemas operativos de usuarios o estaciones.

Los servicios de sistemas de red pueden ser servicios de almacenar y dirigir al nivel de sistema, tales como enfilear protocolos o subsistemas de contabilidad de recursos. Las características de seguridad y confiabilidad con frecuencia se implantan en los servicios del sistema de red para asegurar que proporcionen un nivel de sistema confiable. Para las estaciones de trabajo, los servicios del sistema operativo de red atrapan o capturan las llamadas desde la estación de trabajo y luego las dirigen hacia un recurso de la red.

3.9 REDES LAN

En los últimos veinte años, la industria de comunicaciones ha concentrado su atención en sistemas que transportan datos a largas distancias. La industria de las redes de gran cobertura WAN (Wide Area Networks) ha madurado y es hoy por hoy un sector estable. Las redes locales LAN (Local Area Network) se basan en una tecnología que empezó a adquirir interés a mediados de los setenta y es en la actualidad uno de los sectores de más rápido crecimiento dentro de la industria de comunicación de datos.

3.9.1 REDES DE ÁREA LOCAL (LAN)

Las principales características o atributos de una red local son:

- ❖ Las conexiones entre las estaciones de trabajo suelen tener longitudes comprendidas entre algunos metros y dos kilómetros.
- ❖ Una red local transmite datos entre estaciones de usuario y computadoras (aunque algunas redes pueden transportar también imágenes y sonido).
- ❖ La capacidad de transmisión de una red local suele ser mayor que la de una red extensa: las velocidades de transmisión suelen estar comprendidas entre 1 Mbps a 100 Mbps.
- ❖ El canal de la red local suele ser propiedad de la misma organización que utiliza la red. Por lo general, las compañías telefónicas no intervienen en su propiedad ni en su administración.
- ❖ La tasa de errores de una red local suele ser considerablemente menor que la del canal de telecomunicaciones orientado a redes extensas.

3.9.1.1 Estándares para redes locales

Un estándar es un conjunto de lineamientos que todos los fabricantes están dispuestos a cumplir. En el mundo de la computación, cuando se establece un estándar y un fabricante lo cumple se dice que su producto es compatible. De esta manera, los fabricantes pueden desarrollar productos de red que pueden desempeñarse con otros productos que a su vez también lo sean. Con esto no importa la marca de un producto. Mientras se apege al estándar podrá comunicarse perfectamente con todos los que lo sigan.

Para las redes locales, organizaciones tales como el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, Institute of Electrical and Electronic Engineers), principalmente en el proyecto 802, desarrollan estándares de comunicación. Este proyecto, en particular, tiene como finalidad establecer el procedimiento para lograr la comunicación entre los nodos de una red. Entre los estándares más utilizados se encuentran el 802.3, el 802.5, el 802.12 y el FDDI.

3.9.1.2 IEEE 802.3 (Ethernet)

El sistema de red Ethernet fue originalmente creado por Xerox, pero desarrollado conjuntamente como una norma en 1980 por Digital Equipment Corporation, Intel y Xerox. Esta norma se conoció como DIX Ethernet, haciendo referencia a los nombres de quienes lo habían desarrollado. La norma 802.3 del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) define una red similar, aunque ligeramente diferente, que utiliza un formato alternativo de trama (una trama constituye la estructura de codificación de un flujo de bits transmitidos a través de un enlace). Ethernet presenta un rendimiento de 1, 10 y 100 Mbps, y utiliza un método de acceso sensible a la señal portadora, mediante el que las estaciones de trabajo comparten el canal de comunicación de la red, pero solo una de ellas puede utilizarlo en un momento dado. El CSMA/CD se utiliza para arbitrar el acceso medio. El comité 802.3 del IEEE es el responsable de la definición del nivel físico en la pila de protocolos OSI. Este nivel se divide en dos subniveles denominados de control de acceso al medio (MAC) y subnivel de enlace de datos (Data-Link). Todas las adaptaciones de la norma 802.3 del IEEE presentan una velocidad de transmisión de 10 Mbps, con la excepción de 1Base-5 y 100Base-T, que permite la transmisión a 1 Mbps y 100 Mbps respectivamente. Pueden conectarse hasta 8.000 estaciones de trabajo en una única red de área local (LAN). En la siguiente lista se relacionan todas las topologías. Téngase en cuenta que el primer número del nombre se refiere a la velocidad

en Mbps, y el último a los metros que admite un segmento (multiplicados por 100). Base hace referencia a la banda base y Broad a banda ancha.

- ❖ 10 Base-5. Cable coaxial con longitud máxima de segmento de 500 metros; utiliza métodos de transmisión en banda base.
- ❖ 10 Base-2. Cable coaxial (RG-58 A/U) con longitud máxima de segmento de 185 metros; utiliza métodos de transmisión de banda base.
- ❖ 10 Base-T. Cable de par trenzado con longitud máxima de segmento de 100 metros.
- ❖ 1 Base-5. Cable de par trenzado con longitud máxima de segmento de 500 metros y velocidad de transmisión de hasta 1 Mbps.
- ❖ 10 Broad-36. Cable coaxial (RG-59 A/U CATV) con longitud máxima de segmento de 3,600 metros; utiliza métodos de transmisión en banda ancha.
- ❖ 10 Base-F. Sirve de soporte a redes soporte de fibra óptica de hasta 4 kilómetros con transmisión a 10 Mbps. LA EIA/TIA (Electronic Industries Association / Telecommunications Industries Association) ha adoptado este cable para establecer conexiones cruzadas entre los edificios de un campus en su normativa de cableado para edificios comerciales.
- ❖ 100 Base-T. Una nueva norma de Ethernet que presenta un rendimiento de 100 Mbps y utiliza el método de acceso CSMA/CD sobre configuraciones jerárquicas de cableado de par trenzado, también se le denomina Ethernet rápida (Fast Ethernet).

La topología de la redes Ethernet 802.3 consiste en un bus lineal que utiliza el método de acceso CSMA/CD. En las realizaciones sobre cableado coaxial, las estaciones de trabajo se conectan en serie conectando los segmentos de cable, denominado línea troncal. La versión de cable trenzado de Ethernet (10 Base-T) adopta una topología física en estrella, en la que el cable dirigido hacia cada estación es una rama que parte de un concentrador central de cableado.

Formatos de trama

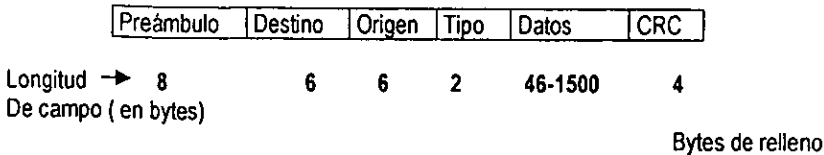
Una trama Ethernet representa la estructura de un paquete de datos enviados a través de una red Ethernet. Describe la posición de las cabeceras, bits de datos y la carga útil de información del paquete. Comprender los tipos de trama es importante si se desea conectar un analizador de protocolos a una red para realizar una supervisión del tráfico de la misma. Es posible descubrir ciertos problemas de red observando el contenido de los paquetes reuniendo estadísticas al respecto.

Existen cuatro tipos de trama en Ethernet:

- ❖ Ethernet II. El tipo de trama original de Ethernet asigna una única cabecera al paquete, el utilizado en las redes Apple Talk Phase I y las redes conectadas a sistemas DEC o a computadoras que utilizan el protocolo TCP/IP.
- ❖ Ethernet 802.3. El tipo de trama utilizado genéricamente en redes Netware de Novell.
- ❖ Ethernet 802.2. El tipo de trama utilizado por defecto en redes Netware 4.x de Novell.
- ❖ Ethernet SNAP. El tipo de trama utilizado en redes Apple Talk Phase II.

En la parte superior de la figura 3.7, se muestra la trama original de Ethernet II, y en la parte inferior la trama IEEE 802.3. Los campos más importantes de las tramas se describen a continuación.

Trama Ethernet original



Ethernet (IEEE 802.3)

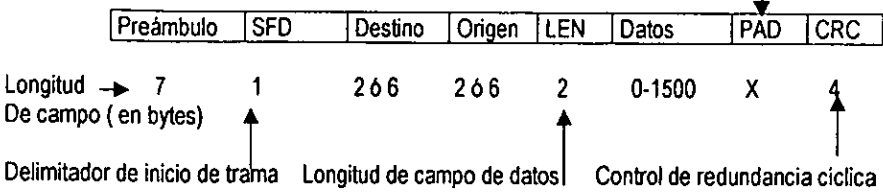


Figura 3.7 Trama de Ethernet

- ❖ Preámbulo. Este campo señala el comienzo de la trama
- ❖ Delimitador de inicio de trama (SFD, Star Frame Delimiter). Este campo proporciona un campo adicional que indica el comienzo de la trama Ethernet IEEE 802.3
- ❖ Destino y Origen. Estos campos mantienen la dirección original y destino.
- ❖ Longitud (LEN) del campo de datos. Este campo indica la longitud de la porción de datos de la trama.
- ❖ Control de redundancia cíclica (CRC, Cyclical Redundancy Checksum). Este campo mantiene un valor calculado por el emisor. El receptor realiza el mismo cálculo para ver si coincide con el valor del campo CRC. Si no es así, se considera que la trama se ha corrompido y se retransmite de nuevo.

3.9.1.2 Ethernet 10BASE_T (Par trenzado)

10 BASE-T ofrece la mayoría de las ventajas de Ethernet sin las restricciones y el costo que impone el cable coaxial. Además, la topología en estrella o distribuida permite la conexión de grupos de estaciones de trabajo departamentales o situadas en otras zonas. Es por esto que este estándar es el más utilizado.

Parte de las especificaciones 10BASE-T es compatible con otras normas 802.3 del IEEE de modo que es sencillo realizar una transición de un medio a otro. Es posible mantener las mismas tarjetas Ethernet al pasar del cable coaxial a cable par trenzado. Además, pueden añadirse líneas troncales de par trenzado a las existentes, gracias a repetidores que admiten la conexión de líneas troncales de cable coaxial, fibra óptica y par trenzado.

La especificación 10BASE-T incluye una utilidad de verificación de cableado denominada verificación de integridad del enlace. Gracias a esta utilidad el sistema puede realizar controles constantes del cable de par trenzado en busca de hilos abiertos o cortes del cable. Se realiza la supervisión desde un punto central.

En una red básica Ethernet 10BASE-T, las estaciones de trabajo se conectan a un concentrador central que actúa como repetidor. Cuando llega la señal procedente de una estación de trabajo, el concentrador la difunde hacia todas las líneas de salida. Es posible realizar conexiones entre distintos concentradores para obtener una configuración jerárquica. Las estaciones de trabajo están conectadas a través de un par trenzado que no puede exceder 100 metros de longitud. En una red pueden existir hasta 1024 estaciones sin necesidad de utilizar puentes.

Las conexiones 10BASE-T utilizan cable categoría 3 y categoría 5, las cuales permiten un crecimiento futuro a tecnologías de transmisión más rápidas, como 100 Mbps.

Los componentes que se describen a continuación forman parte típica de las redes 10BASE-T.

- ❖ *Tarjeta de interfaz de red.* La tarjeta Ethernet deberá incluir un conector tipo RJ-45 10BASE-T. Si la tarjeta se instala en una estación de trabajo sin disco, hay que añadir una PROM de arranque remoto.
- ❖ *Concentrador (HUB).* El concentrador dispone a menudo de 12 ó 24 puertos. Normalmente dispone de un puerto de conexión a redes con soporte de cable coaxial o de fibra óptica y tiene la capacidad de comunicarse en cascada con otros concentradores.
- ❖ *Cable de par trenzado.* 10BASE-T utiliza el cable de par trenzado con conectores RJ-45 de hasta 100 metros de longitud. Puede adquirirse cable a granel y conectores aparte para construir segmentos de distintas longitudes según sus necesidades.
- ❖ *Conector RJ-45.* Se debe instalar un conector de este tipo en cada extremo de los cables. Los pines 1 y 2 son "emisores" y el 3 y 6 "receptores". En el caso de un cable que vaya a conectar una estación de trabajo con el concentrador, las conexiones de los pines son uno a uno, para conectar dos concentradores, cada par debe estar cruzado de modo que el transmisor en un extremo se conecta con el receptor en el otro.

Las especificaciones y limitaciones de la norma 10BASE-T se relacionan a continuación:

- ❖ Se debe utilizar cable de par trenzado sin blindaje de categoría 3, 4 ó 5.
- ❖ Requiere de conector RJ-45 en cada extremo de los cables.
- ❖ La longitud del cable entre la estación de trabajo y el concentrador no debe exceder los 100 metros.
- ❖ Un concentrador normalmente conecta 12 ó 24 puertos.
- ❖ Pueden conectarse los concentradores en cascada para aumentar el número de estaciones de red.
- ❖ Los concentradores pueden conectarse a redes soporte de cable coaxial o fibra óptica con objeto de formar partes de redes Ethernet extensas.
- ❖ Pueden existir hasta 1024 estaciones en una red sin necesidad de utilizar puentes.

3.9.2.1 Tecnologías de Telecomunicaciones

Las tecnologías más utilizadas para la transmisión de datos a niveles locales, nacionales e internacionales son sin duda alguna, el X.25, el Frame Relay, ATM, ISDN y VSAT. Dichas tecnologías están siendo usadas cada día más por los operadores públicos para ofrecer servicios de alta y baja velocidad, que buscan satisfacer las necesidades de interconexión de datos y redes de área amplia, así como la transmisión de voz, imágenes y video.

3.9.2.2 X.25

X.25 es un servicio de conmutación de paquetes que tradicionalmente se usa para la conexión de terminales remotos a sistemas anfitriones (host). El servicio proporciona conexiones cualquiera a cualquiera para usuarios simultáneos.

3.9.2.3 Frame relay

Frame Relay constituye un método de comunicación orientado a paquetes para la conexión de sistemas informáticos. Se utiliza principalmente para la interconexión de redes de área local (LAN) y redes de área extensa (WAN) sobre redes públicas y privadas.

3.9.2.4 Modo de Transferencia Asíncrono (ATM, Asynchronous Transfer Mode)

ATM permite la consolidación de múltiples señales en un solo canal (multiplexión) de diversos servicios tales como voz, video y datos, a muy alta velocidad.

3.9.2.5 Red Digital de Servicios Integrados (ISDN, Integrated Services Digital Network)

ISDN se declara a menudo como la interfaz pública para telefonía y telecomunicaciones del futuro. Integra datos, voz y señales de video en una línea digital (en oposición a la analógica) telefónica. El punto importante es que se lleva servicios digitales a todo tipo de casas u oficinas.

3.9.2.6 Terminales de Apertura muy Pequeña (VSAT, Very Small Aperture Terminal)

Una red VSAT es un sistema de comunicaciones de datos que usa tecnología satelital utilizada para una gran variedad de aplicaciones en el campo de las telecomunicaciones, que incluye las comunicaciones de datos interactivas y por lotes en diversos protocolos, operación de redes de conmutación de paquetes, servicios de voz, transmisión de datos, videos y operación de una red vasta área para permitir a muchas localidades remotas comunicarse en forma bidireccional (con excepción de video) con una oficina encargada de centralizar y procesar la información.

3.10 "ANÁLISIS DE LAS NECESIDADES DE INTERCOMUNICACIÓN DE LA DGETI Y EN PARTICULAR EN EL CETIS N°152"

En esta parte se hace el análisis de la situación actual de La DGETI y el CETIS N°152 para identificar problemas de intercomunicación y encontrar una posible solución. Para llevar a cabo esta solución es necesario conocer con que recursos se cuenta y cuáles son los planes que dicha institución tiene a futuro y así poder determinar la tecnología que cubra sus necesidades.

PLANTEAMIENTO DEL PROBLEMA

La Dirección de Educación Tecnológica Industrial (DGETI), como un órgano coordinador del CETIS N°152, tiene como objetivo ofrecer a la población estudiantil bachillerato y carreras técnicas que se imparten con una duración de seis semestres, al cabo de las cuales el alumno obtendrá el Certificado de Técnico Profesional, que lo acredita con el título de Técnico profesional en el área de su elección o Certificado de bachillerato con la opción de continuar su preparación a nivel licenciatura.

Este compromiso de la Institución, requiere toda una estructura jerárquica de cada unos de los Departamentos existentes, y además de sistemas de cómputo que brinden la posibilidad de usar la computadora como herramienta en el manejo y administración de toda la información.

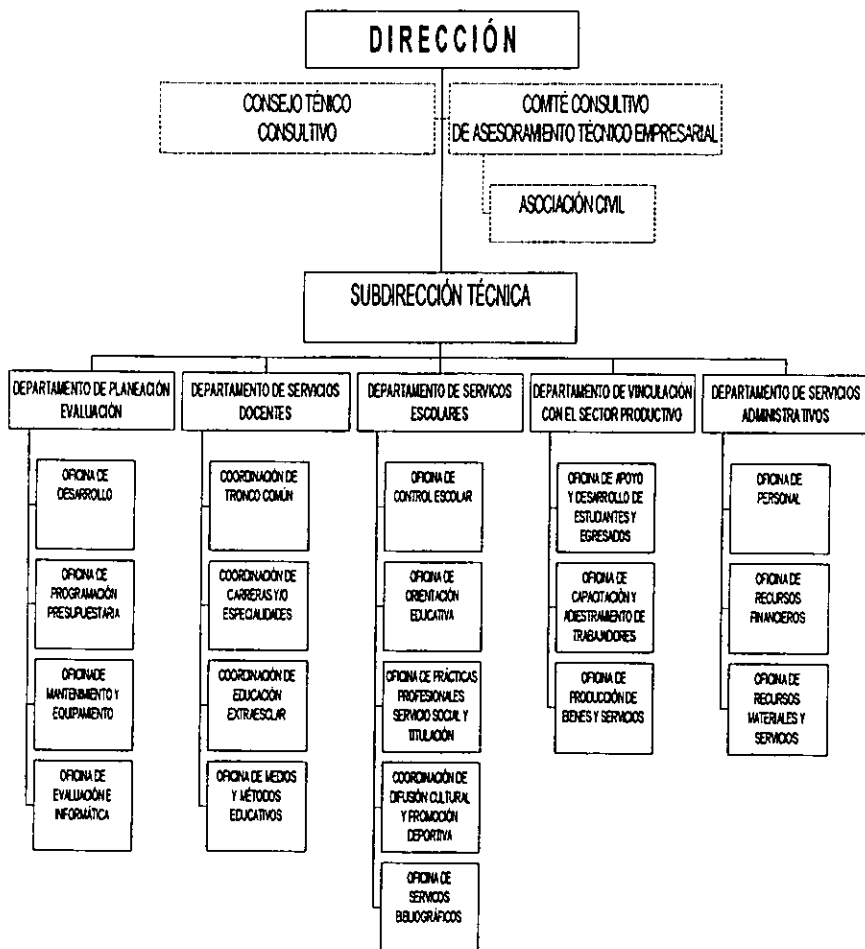
La creación de las aplicaciones de software de uso específico, como es el caso del SISE (Sistema de Evaluaciones), requieren una cantidad de recursos tanto en hardware como en software, por lo que para el buen funcionamiento del sistema es necesario conocer los requisitos impuestos por las políticas propias de la institución, además de definir los alcances del mismo, y así disponer de los recursos necesarios para su implantación.

Actualmente, los sistemas de cómputo que operan en la DGETI y sus dependencias, no permiten a los usuarios compartir los datos que se encuentran registrados en los sistemas los diferentes CETIS, así como los diferentes recursos informáticos con los que cuenta cada una de ellas. Estos recursos requieren de una permanente transmisión de información para mantener actualizadas sus bases de datos. Dicha transferencia de datos se lleva acabo por medio de comunicaciones costosas y poco eficientes, ocasionando el retraso en la actualización de la información y su correspondiente procesamiento.

Asimismo, la DGETI está proponiendo proyectos que requieran modernizar los canales de comunicación existentes para la consulta e intercambio de información entre los diversos puntos donde se localizan las Instituciones coordinadoras y sus dependencias, razón por la cual es necesario contar con medios de comunicación de capacidad suficiente para transmitir datos a gran velocidad y bajo costo.

Problema: No existe un medio de comunicaciones eficaz que permita compartir la información de las diversas bases de datos y recursos informáticos con que cuenta la DGETI, además de que no se cuenta con herramientas que faciliten la transmisión de información, como el trabajo en grupos, correo electrónico, etc.

DIAGRAMA DE ORGANIZACIÓN DEL PLANTEL



HIPÓTESIS

Dentro del actual sistema de procesamiento de datos con que cuenta la DGETI, uno de los principales problemas que se presenta es que la actualización de bases de datos y la transmisión de información no es muy segura, esto se debe a que la mayoría de la información se maneja manualmente, además de que se cuenta con bases de datos aisladas que manejan los distintos departamentos, sin satisfacer los requerimientos de rapidez y consistencia de la información.

Otro de los aspectos que afectan directamente a los usuarios de los equipos es el hecho de no poder contar con suficientes dispositivos periféricos que puedan satisfacer las demandas que de ellos se tienen, como es el caso de las impresoras donde cada usuario tiene que trasladar su información via disco flexible hasta donde se encuentra el equipo que tiene conectada una impresora.

Esto trae como consecuencia información errónea, duplicidad de datos, tiempo extraordinario en trámites y el no poder contar con herramientas que faciliten la transmisión de información. En base a las necesidades de comunicaciones se plantea la siguiente hipótesis:

"Si se instalan redes de área local en cada una de las áreas de la DGETI, con el objeto de que varios usuarios puedan tener acceso a diferentes recursos informáticos y fuentes de información desde su estación de trabajo y estas redes se interconectan por medio de enlaces de alta velocidad para crear una red de área amplia (WAN); entonces tanto las Coordinaciones y sus dependencias estarían intercomunicadas con mecanismos eficientes de transmisión de datos y se dotaría a cada área de mejores herramientas para el cumplimiento de sus funciones".

3.11 DIAGNÓSTICO DE NECESIDADES Y RECURSOS

Para poder identificar de una manera clara las necesidades que debe satisfacer la instalación de la posible red LAN, es necesario conocer antes los requerimientos operativos de cada departamento y la forma como realizan las operaciones hasta ese momento. Para llevar a cabo este análisis, se elaboró un cuestionario del cual se propone su aplicación tomando como base el CETIS N°152, para que nos de a conocer las necesidades más urgentes en cuanto a manejo de información en cada gerencia.

CUESTIONARIO DE DIAGNOSTICO DE NECESIDADES DE COMUNICACIONES

Coordinación: _____

Institución: _____

1.- ¿Es constante el intercambio de archivos con personas de su misma Institución o con otras? _____

2.- Este intercambio ¿cómo se lleva a cabo? _____

3.- ¿Necesita acceder a bases de datos o programas que no residen en su equipo de computo? _____

4.- ¿De qué manera realiza sus tareas de impresión?

- a) Impresora conectada a otra PC b) Impresora local c) No requiero acceso a Impresora

5.- Del 100% de la información que genera, ¿qué porcentaje es de uso de la Institución a la que pertenece y cuanto de otras Instituciones? _____

6.- De la información que recibe para su procesamiento, ¿qué porcentaje proviene de su Institución y cuánto de otras gerencias? _____

7.- ¿Intercambia información con las Instituciones regionales? Si la respuesta es no, omita la última pregunta.

Si _____ ¿cada cuándo? _____

No _____

8.- ¿Cómo se lleva a cabo el intercambio de información con las Instituciones regionales? _____

¿Es rápido? _____

También puede servirnos el siguiente formato para identificar los bienes informáticos con los que cuenta la DGETI para la implantación de redes y el número de usuarios por equipo. Para ello se requiere de enviar a todas las gerencias tanto a nivel Coordinación, como Institución, donde además se incluya información acerca del software que se maneja en los diferentes equipos.

CENTRO DE ESTUDIOS TECNOLOGICOS Industrial y de Servicios N° 152

FORMATO DE CONTROL DE BIENES INFORMATICOS

DATOS DEL USUARIO				
NOMBRE DEL USUARIO				R.F.C:
CARGO				
COORDINACIÓN				
INSTITUCIÓN				
JEFATURA				
UBICACIÓN FÍSICA				
EQUIPO DE COMPUTO				
MARCA Y MODELO			MEMORIA RAM	
PROCESADOR			DISCO DURO	
COMPONENTES	MOUSE	MONITOR	TECLADO	CPU
N° SERIE				
CARACTERÍSTICAS				
OBSERVACIONES				
PROGRAMAS DE COMPUTO INSTALADOS				
NOMBRE	VERSIÓN	IDIOMA	N° DE LICENCIA	
EQUIPO PERIFÉRICO (Unidades de cinta, Módem, Impresoras, etc)				
EQUIPO	N° SERIE	MARCA	MODELO	CARACTERÍSTICAS

Lo que se pretende con la aplicación de este cuestionario y el formato es recabar la información, de que por lo menos todos los mandos medios y superiores cuenten cada uno con un equipo de cómputo, ya que es un requerimiento importante dentro del manejo de información; el hecho de que por falta de equipo de cómputo éstos se tengan que compartir entre varios usuarios, hace que la información que genera cada uno de ellos esté sin proteger de los demás usuarios.

Objetivo:

Contar con un medio de telecomunicación eficiente, que permita el intercambio y consulta de la información generada en los sistemas del CETIS N°152, mediante el uso de redes de área local enlazadas en una red de área amplia que contenga a todas las Instituciones, dependientes de la DGETI.

Asimismo, que permita la agilización del correo electrónico, el acceso al nodo Internet de la DGETI y hacia las Instituciones dependientes, por medio de enlaces de telecomunicaciones de alta velocidad.

Justificación.

Con el fin de contar con la infraestructura de telecomunicaciones requerida por la DGETI para comunicar las dependencias Institucionales, se requiere implementar la tecnología informática de telecomunicaciones de vanguardia por medio de la instalación de redes de área local y de área amplia, lo que facilitará la comunicación y compartición de recursos e información entre todas los departamentos de acuerdo a los planes de sistematización de la DGETI.

3.12 EVALUACIÓN Y DETERMINACIÓN DE TECNOLOGÍA PARA INSTALAR UNA LAN

En el principio de este capítulo se explicaron los tipos de redes que existen y sus componentes principales, ahora se tomarán las características de cada uno de los elementos a considerar para la implementación de las redes de área local LAN y se irán resumiendo en cuadro para su análisis más rápido y simplificado.

Una parte fundamental en el diseño de una red LAN es definir el estándar que se va a implementar, para esto, se llevó a cabo el estudio de los estándares de Ethernet, Token Ring y FDDI que son los más comúnmente usados y que definen la implementación de una red de área local. Analizando estos estándares, se pudieron definir sus características principales reuniéndolas en la tabla 3.3 en donde se aprecian claramente las diferencias entre cada una de ellas.

	ETHERNET	TOKEN RING	FDDI
Velocidad de transferencia de datos	10 Mbps	4 ó 16 Mbps	100 Mbps
Topología	Bus físico. Bus lógico / estrella física	Anillo con estrella	Anillo doble
Método de acceso	CSMA/CD	Token Passing	Token Passing
Especificación del medio	Coaxial grueso, Coaxial Delgado, par trenzado, fibra óptica	Par trenzado blindado y no blindado, fibra óptica	Fibra óptica
Longitud máxima de la red (sin puentes)	500 m	1000 m	200 km (100 km por anillo)
Distancia máxima entre estaciones de trabajo	500 m	100 m	Aprox. 2 km (-11 dB perdidos entre nodos)
Tamaño máximo del paquete	Aprox. 1,5 k	Aprox. 4 ó 18 k	Aprox. 4,5 k
Número máximo de estaciones de trabajo	1024	260 usando par trenzado blindado, 72 con par trenzado sin blindaje	500 (1000 conexiones) para la configuración por default

Tabla 3.3 Análisis de Red LAN

Estándares de redes LAN cada estándar admite o más topologías específicas y dependiendo de la topología es el medio de transmisión que se debe utilizar. En la tabla 3.4, se muestran las topologías existentes para la implementación de una red y la relación que guarda ésta con los medios de transmisión.

Medio	Topología Física		
	Bus	Anillo	Estrella
Cable de par trenzado			X
Cable coaxial banda base	X	X	
Cable coaxial banda ancha	X		
Cable de fibra óptica		X	

Tabla 3.4 Relación entre las topologías y los medios de transmisión

Para complementar la información obtenida de la tabla 3.4, se presentan en la tabla 3.5 las principales características de los diferentes medios de transmisión.

	CABLE COAXIAL	PAR TRENZADO	FIBRA OPTICA
Transmisión	Voz, video y datos	Voz y datos	Voz, video y datos
Instalación	Fácil	Fácil y rápida	Delicada
Compatibilidad	Ethernet y Arcnet	Ethernet, Token Ring y Starlan	Ethernet, Token ring y FDDI
Ancho de banda	10 Mbps	10 ó 100 Mbps	100 ó 200 Mbps
Distancia máxima sin repetidores	600 m	110 m con UTP 500 m con STP	2000 m
Tolerancia a interferencias	Buena	Buena	Excelente
Costo	Bajo	Bajo	Alto

Tabla 3.5 Características de los medios de transmisión

Elección del estándar para las redes locales

Para poder hacer una buena elección del estándar a utilizar en las redes locales, debemos conocer primeramente los requerimientos de la empresa para el mismo, los cuales se listan a continuación:

- ❖ Velocidad de transmisión de al menos 10 Mbps
- ❖ Fácil de instalar
- ❖ Que permita que se añadan nuevos nodos de forma sencilla
- ❖ Que permita la migración a una velocidad mayor
- ❖ Que tenga una buena administración de los recursos de hardware
- ❖ Que tenga un costo accesible

De acuerdo a estos requerimientos y al evaluar los cuadros anteriores en donde se comparan las diversas tecnologías de redes LAN más utilizadas, se determinó que el estándar a utilizar en las redes LAN sería el Ethernet. En la tabla 3.6 se resumen las características de las especificaciones más utilizadas.

	10Base-2	10Base-5	10Base-T
Velocidad de transmisión	10 Mbps	10 Mbps	10 Mbps
Nº de estaciones	1024	1024	1024
Longitud del cable	200 m	500 m	100 m/nodo
Topología física	Bus	Bus	Estrella
Tipo de cable	RG58 50ohms coax	RG59 Ohms coax.	UTP 24 AWG
Tipo de conector	BNC	DB15 AUI	RJ-45
Tipo de transceptor	In/Ext	Externo	In/Ext

Tabla 3.6 Especificaciones Ethernet

De acuerdo a las características presentadas en la tabla 3.6, la especificación elegida fue la 10Base-T porque utiliza topología de tipo estrella, lo que permite que en caso de que ocurra algún problema con una de las estaciones de trabajo, sólo la estación donde se presentó el problema será la que quede desconectada de la red sin afectar a las demás estaciones, también por su velocidad, precio medio, gran estandarización, facilidad de cableado y sobre todo la posibilidad de administrar los recursos hardware más fácilmente.

Otra de las ventajas del estándar Ethernet 10Base-T es que utiliza como medio de transmisión cable de par trenzado sin blindaje dado que es la mejor elección para las conexiones de red en un mismo edificio y se utiliza cableado UTP que se adapte a la especificación denominada categoría 5, podrá transportar los datos a una velocidad de 10 a 100 Mbps, además de satisfacer las necesidades de transmisión de datos requeridas por más de diez años (según las especificaciones de los fabricantes) y de ser más fácil de manipular que los otros cables.

La figura 3.7 ilustra las características descritas anteriormente para las redes LAN Ethernet 10Base-T que se podrían instalar.

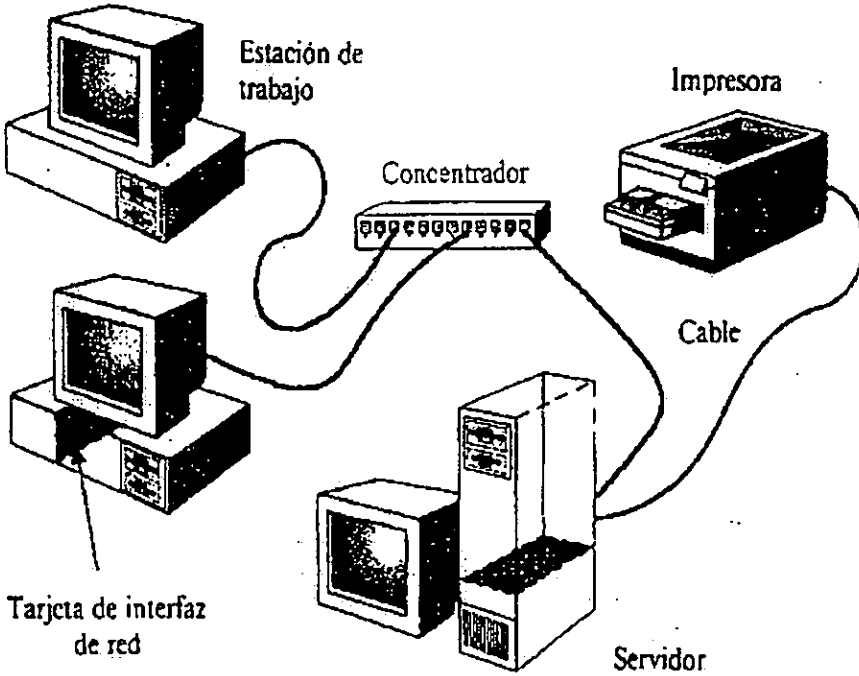


Figura 3.7 Red LAN Ethernet 10Base-T

Elección del estándar para las redes locales

La elección de la tecnología de telecomunicaciones para la red de área amplia de la DGETI también debe considerar los requerimientos que para la misma tiene la institución tales como:

- ❖ Medios de comunicación eficientes para la transferencia de información.
- ❖ Enlaces de alta velocidad entre los nodos de la red.
- ❖ Que tenga poco retraso en las transmisiones
- ❖ Redundancia para evitar caídas de la red.
- ❖ Que permita el aumento del ancho de banda en los enlaces.
- ❖ Soporte de diferentes tipos de tráfico.
- ❖ Incorporación de nuevas terminales en cada nodo de la red.
- ❖ Soporte de nuevas tecnologías.
- ❖ Que permita el monitoreo de la red.
- ❖ Costo accesible.

3.13 ESPECIFICACIÓN DE LA RED DE COMPUTADORAS CON QUE CUENTA EL CETIS N°152

El diseño de esta red es la parte más importante del capítulo ya que hace uso de los conceptos de generales de redes, además de la utilización del análisis previo de la situación de la Institución, para establecer las características con las que debían contar las redes instaladas en la DGETI y la forma de implementarse, para que de esta forma se puedan alcanzar las metas que dicha Institución tiene a futuro.

Actualmente, cada plantel, de los 32 que conforman la DGETI, cuentan con la instalación de recientes redes de área local (LAN), cuyas características de equipo en la mayoría de los casos, son o pueden ser actualizadas, para ser utilizadas por otros sistemas que automatizen procesos de un área de trabajo específica; como el desarrollo del SISE (Sistema de Evaluación, que es el proyecto desarrollado).

Anteriormente todos los procesos de la Secretaría Escolar se realizaban de manera manual; al introducir equipo de cómputo para automatizar las rutinas del trabajo, el software que se utilizaba era para plataformas monousuarios.

El uso de las redes de cómputo instaladas en cumplimiento de todas las actividades referentes a los procesos educativos y a los diferentes sistemas que maneja la institución, dan pauta a crear sistemas de cómputo que pueden ser diseñados y explotados de manera tal que puedan ser doblemente útiles en las actividades encomendadas de las instituciones.

Para los diferentes CETIS, la propuesta de hardware que se hace en este trabajo pretende plantear y diseñar el sistema en un ambiente preferencial de red, de manera que todas las instituciones incorporadas en la red puedan tener acceso directo a la información, pero sin descuidar que el sistema en desarrollo puede utilizar la misma infraestructura de cada uno de las instituciones, con el fin de aprovechar lo que se tiene, anotando a caso algunos costos, para que nuestro sistema sea de utilidad y se ponga en marcha sin más retrasos.

Descripción del edificio del CETIS N°152:

El edificio del CETIS, es un edificio de 4 pisos incluyendo el sótano. El edificio cuenta con total de 45 equipos de cómputo y 5 impresoras repartidas entre los pisos, además de 2 servidores, que tienen instalado el sistema operativo Windows NT y Netware de Novell 2.1. En otro de los pisos se planeo instalar una segunda red, la cual tendrá la capacidad de acceder a el servidor con Windows NT y a los de Netware.

Para la instalación de la red local, es necesario contar con los componentes de hardware adecuados, por lo que además del equipo de cómputo e impresoras con los que ya se cuentan en el edificio, se debe adquirir el equipo restante como tarjetas de red, concentradores, cable, canaletas, rosetas, conectores, etc. Los requerimientos de debe cumplir este equipo se presentan a continuación.

Un proyecto de red local involucra varias actividades todas interrelacionadas. Se han analizado preferentemente los aspectos técnicos, y en ese sentido, se dan continuación las lista de elementos que es necesario contenga cualquier red local.

A continuación se listan los estándares en un primer nivel físico. (Los componentes son los que se listan o superiores).

SERVER

PENTIUM II MMX "IBM"
Memoria 64 MB en RAM (2 DIM 32MB)
Velocidad 233 MHZ
Disco Duro IDE 4.3GB y Drive 3 ½
CD-ROM 24 velocidades y Unidad de cinta 8mm
Tiempo de acceso en los discos duros
Impresora (s) conectada (s)

ESTACIÓN DE TRABAJO

PENTIUM MMX
Memoria 32 MB en RAM
Velocidad 200MHZ
Drive de 3 ½
Monitor UVGA de 14"
Disco duro de 2.4 GB

**SISTEMA OPERATIVO DE RED
WINDOWS NT****SOFTWARE DE APLICACIÓN**

OFFICE 97, Base de datos,
Lenguajes de programación,
Manejo concurrente no necesario
Desarrollo de sistemas
Ambiente
Tiempo estimado de desarrollo

TARJETAS DE RED

El server Ethernet
Estaciones de trabajo Ethernet
Ambiente WINDOWS NT

3.14 DISEÑO DE RED LAN EN COORDINACIONES Y DEPENDENCIAS DE LA DGETI

REQUERIMIENTOS DE HARDWARE

❖ *Servidores para los CETIS.*

Procesador Intel Pentium II o compatible.

Velocidad mínima de 233 MHZ.

Caché interno de 512Kb

Memoria principal ECC de 128 MB en DIMM's expandible a 256 MB

Controlador de disco Ultra SCSI.

Disco duro de 6.0 GB al menos SCSI con velocidad de acceso no mayor a 11ms.

Tarjeta controladora de video Super VGA con 4MB de memoria mínimo, con capacidad de manejar 256 colores a una resolución de 1024 por 768 pixels.

Monitor de 1024 colores de 14"

Unidad de CD_ROM 12x mínima, ATAPI.

Tarjeta de red PCI Ethernet 10 / 100 Base-T con conector RJ-45.

Unidad de respaldo en cinta interna DAT SCSI de 8GB en 4mm.

Unidad interna de disco flexible de 1.44 MB.

Teclado en español de 102 teclas para Windows.

Mouse de dos botones con conector mini-din.

Gabinete tipo torre.

Ranuras de expansión: cuatro ISA y dos PCI libres mínimo después de configurar.

Bahías de expansión internas para disco duro: 2 de 5.25"

Fuente de poder mínima de 250 watts

Un puerto paralelo bi-direccional, un puerto serial 16550 UART y un puerto USB.

❖ *Tarjetas de red Ethernet.*

Tarjeta tipo 10Base-T de 16 bits con entrada para cable tipo para trenzado. Conexión a computadoras con arquitectura IBM PC, XT o AT.

Conector UTP RJ-45. Estándar IEEE 802.3 10Base-T.

Velocidad de transferencia 10Mbps.

Mínimo de 16KB de memoria RAM en la tarjeta para buffer de transmisión de paquetes.

Soporte de software para productos Novell Netware 4.x, Windows para trabajo en grupos. Windows NT 4.x y Windows 95.

❖ *Concentrador para red Ethernet.*

Concentrador Ethernet 10Base-T con un número de conexiones mayor o igual a doce, tipo RJ-45.

Un puerto AUI. Estándar IEEE 802.3 10Base-T.

Capacidad de conectarse con otros concentradores.

Velocidad de transferencia de 10Mbps.

Luces indicadoras del funcionamiento de cada puerto.

Módulo SMTP para administración remota.

- ❖ Tipo de cable: Cable UTP nivel 5 de cuatro pares.
- ❖ Conectores: Plug RJ-45 nivel 5 de ocho conductores.
- ❖ Canaleta: De acuerdo al requerimiento de la instalación.
- ❖ Rosetas de red: Tipo Jack RJ-45 nivel 5 de ocho conductores.
- ❖ Placas: Placa de red compatible con el tipo de roseta mencionado anteriormente.
- ❖ Cajas: Cajas de soporte compatibles con la placa y roseta anteriormente definidas.
- ❖ No breake c/ regulador integrado. Corriente de salida 450/280 (VA / W) mínimo. Voltaje de salida de 120V AC, a 60Hz. Frecuencia de entrada y corriente 60Hz / 300 W. Voltaje bajo de línea a batería-nominal 150V AC, voltaje alto de batería a línea nominal 143V AC. Tiempo de respaldo media carga / carga completa 17/5 min. Tiempo típico de transferencia 2-4 mseg. Velocidad de recarga de 4-6 hrs. Tomas de corriente alterna 4 mínimo. Conector para redes DB-9.
- ❖ Sistema operativo. Debido a estándares en la DGETI, el sistema operativo que tendrán todos los servidores de las Instituciones será Windows NT 4.0

3.14.1 Configuración de clientes de red

De la misma forma que se busca que todas las redes instaladas tengan las mismas características técnicas, también se ha establecido que la configuración de los clientes sea la misma en todos los equipos. Las características de esta configuración se dan a continuación:

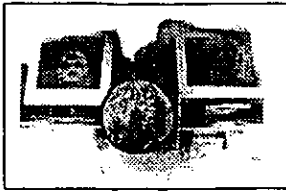
Tarjetas: Direcciones E/S: 0300 / 031F
 IRQ: 10

Clientes: Para red Microsoft con dominio Servidor NT
 Novell NetWare client 32

Protocolos: IPX 32-BIT Protocol for Novell NetWare Client 32
 NetBEUI
 Compatible con IPX / SPX
 TCP / IP

Cuando se instalen los equipos en red Microsoft, a cada uno de ellos se le tiene que asignar un nombre para la PC, el grupo de trabajo al que pertenece y la descripción del equipo. La creación de grupos de trabajo permite que se tenga un mejor control sobre la red y que la administración de la misma sea más sencilla.

Para dar de alta la impresión en los equipos que no cuentan con impresora, se configurará ésta como compartida en los equipos que disponen de una impresora conectada en forma física, para después direccionar los equipos que no cuenten con este recurso mediante una conexión lógica hacia aquellos equipos que tienen la impresora compartida.



4. ANALISIS Y DISEÑO DEL SISTEMA DE CONTROL Y AUTOMATIZACIÓN DE EVALUACIONES

4.1 INTRODUCCION

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

El objetivo del diseñador es producir un modelo o representación de una entidad que será construida más adelante. El proceso por el cual se desarrolla el modelo combina: intuición y criterios basándose en la experiencia de construir entidades similares, un conjunto de principios y/o heurísticas que guían la forma en la que se desarrolla el modelo, un conjunto de criterios que facilitan discernir sobre la calidad y un proceso de iteración que conduce finalmente a una representación del diseño final.

Este capítulo presenta el proceso de diseño para el Sistema de Control y Automatización de Evaluaciones (SISE) mostrando primero las consideraciones que se deben tomar en cuenta para el diseño de Sistemas. Así mismo, se muestra las herramientas de diseño de Sistemas aplicadas al SISE que sirvieron de guía para el desarrollo del presente proyecto, dichas herramientas (las cuales se expusieron teóricamente en el Capítulo 2) muestran las características técnicas de la construcción del SISE, así, se veremos el Diagrama de Flujo de Datos, el Diccionario de Datos y el Modelo Entidad - Relación.

4.2 CONSIDERACIONES DE DISEÑO

ESPECIFICACION DE REQUERIMIENTOS

En esta etapa se plantea el problema que se desea resolver y los objetivos y metas a alcanzar.

En el primer capítulo mencionamos la importancia de contar con un sistema de información distribuido, flexible, seguro y con un GUI (Graphical User Interface) sencillo y fácil de manejar para el control de los exámenes que se aplican a los alumnos del Cetis No. 152. Estas son las principales características que se deben tomar en cuenta para la creación del sistema.

El sistema debe permitir el manejo y manipulación de los datos, así como la creación de reportes (exámenes) en un formato fácil de manipular y desplegar, como lo es el formato de las pantallas windows. Este último brindará la opción al usuario de ver los documentos tal y como se imprimirán para su aplicación a los estudiantes.

El sistema debe contemplar aspectos de seguridad para resguardar la integridad de la información almacenada en las máquinas administradas además debe ser lo suficientemente modular para permitir una mayor funcionalidad de manera simple y congruente con la forma de operar del sistema.

El objetivo fundamental en el diseño de un sistema de información es asegurar que éste proporcione ventajas competitivas a la actividad de la empresa para el que fue desarrollado. El diseño tiene que ajustarse a la forma que trabaja la empresa así como a sus políticas y procedimientos ya establecidos, las necesidades de la organización deben de servir de guía a todas las necesidades de diseño.

Se dice que un sistema de información satisface las necesidades del usuario si cumple con lo siguiente:

- ❖ Realiza en forma apropiada los procedimientos.
- ❖ Presenta información e instrucciones en forma aceptable y efectiva.
- ❖ Produce resultados exactos.
- ❖ Proporciona una interface y métodos de interacción aceptables.
- ❖ Es percibido por los usuarios como un sistema confiable.

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*.

El diseño de sistemas tiene dos etapas: el diseño lógico y la construcción física del sistema. Cuando el analista formula el Diseño lógico, muestra las especificaciones detalladas del nuevo sistema, es decir, aquellas que lo caracterizan: salidas, entradas, bases de datos y los procedimientos, todo en una forma que satisfaga los requerimientos del proyecto, el conjunto formado por todas esas características recibe el nombre de *especificaciones del diseño del Sistema*. La Construcción física produce el *software*. (Figura 4.1)

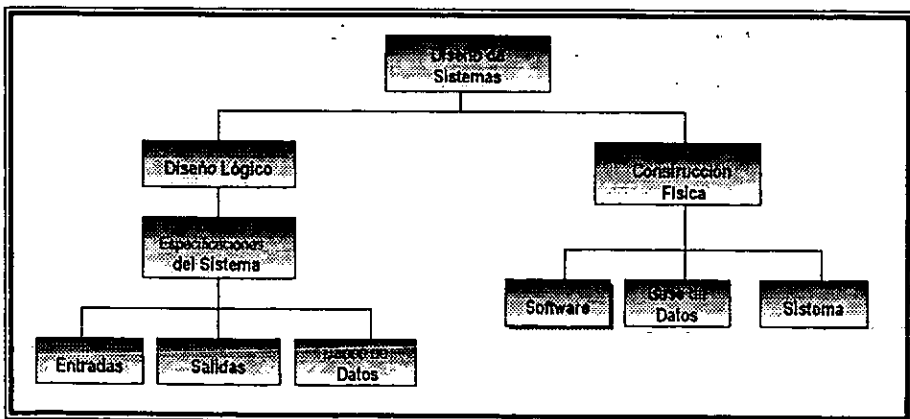


Figura 4.1 Elementos del Diseño de Sistemas.

DISEÑO DE LAS SALIDAS

La salida es la información que reciben los usuarios del sistema de información. Los usuarios confían en las salidas para la realización de sus tareas; y con frecuencia, juzgan el mérito del sistema exclusivamente por sus salidas. Puesto que una salida útil es esencial para lograr la aceptación y el uso del sistema de información, el analista de sistemas tiene varios objetivos que alcanzar cuando diseña una salida. Los objetivos de una salida son seis, tal y como se muestra en la figura 4.2.

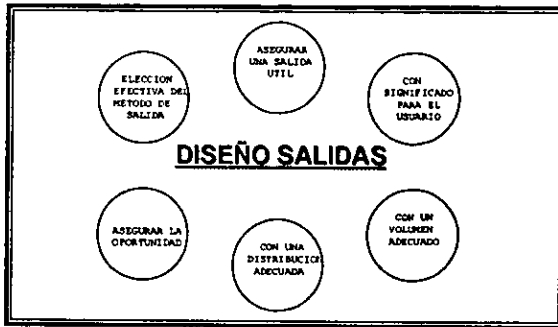


Figura 4.2 Seis objetivos en el Diseño de Salidas.

- ❖ **Diseño de la salida para satisfacer el objetivo planteado.**
Toda salida debe contar con un propósito explícito. No es suficiente que se presente a los usuarios un reporte o una pantalla sólo porque tecnológicamente es posible hacerlo. Durante la fase del análisis de determinación de los requerimientos de información, el analista de sistemas identifica los propósitos a satisfacer; y con base en tales propósitos se diseña la salida.
- ❖ **Diseño de la salida para adaptarse al usuario.**
Es difícil personalizar la salida con un gran sistema de información que atiende a numerosos usuarios con diferentes propósitos. Con base en entrevistas, observaciones, consideraciones de costo y, tal vez, prototipos, será posible diseñar salidas que se apeguen a la mayoría, sino a todas las necesidades de los usuarios y sus preferencias.
- ❖ **Proveer la cantidad adecuada de información.**
Parte de la tarea del diseño de la salida es decidir qué cantidad de información es correcta para los usuarios, ya que los requerimientos de información cambian de manera continua. Puede ser apropiado presentar un subconjunto de tal información y luego proporcionar un camino para el usuario tenga información adicional de manera sencilla.
- ❖ **Asegúrese de que la salida esté disponible donde se requiere.**
El incremento de las salidas en línea (on-line), desplegadas en pantalla y que son accesibles de manera individual, han resuelto parte del problema de la distribución de la información, pero una distribución apropiada todavía es un importante objetivo para el analista de sistemas. Para ser útil y aprovechada, la salida debe presentarse al usuario adecuado. No importa qué tan bien se diseñen los reportes, si éstos no los ven los tomadores de decisiones pertinentes; carecerían de valor.

- ❖ Proporcionando oportunamente la salida.
Una queja común de los usuarios es que no reciben de manera oportuna la información para la toma de decisiones. No únicamente se debe estar consciente de quién recibe la salida, sino también se debe de considerar la puntualidad de su distribución. Una presentación a tiempo puede llegar a ser decisiva para la operación de la empresa.
- ❖ Elección del método correcto de salida.
Para mucha gente, la salida se asocia con la idea de voluminosos bultos de papel de computadoras, pero esto ha cambiado últimamente. Con el movimiento hacia los sistemas en línea, la mayoría de la información se despliega en pantalla. El analista debe evaluar las ventajas involucradas al elegir un método de salida. Los costos difieren, así como la flexibilidad, vida media, distribución, almacenamiento y posibilidades de acceso y transporte; y, finalmente, el impacto global hacia el usuario.

DISEÑO DE ENTRADAS

La calidad de la salida del sistema está determinada por la calidad de su acceso o entrada. Durante el diseño de las formas de entrada y las pantallas es vital tener en mente esta relación decisiva, las entradas pobres pone en entredicho la calidad del sistema completo.

Un buen diseño de los formatos y las pantallas de entrada debe satisfacer los objetivos de eficacia, precisión, facilidad de uso, consistencia, sencillez y atracción, tal y como se plantea en la figura 4.3.

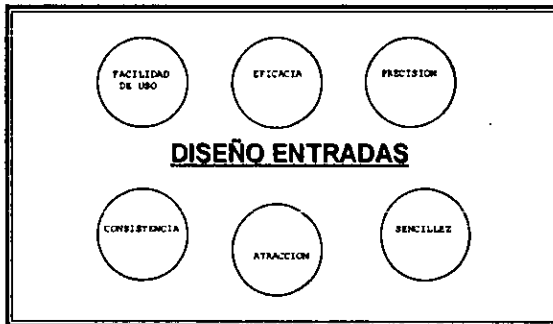


FIGURA 4.3 Seis objetivos en el Diseño de Entradas.

La eficacia significa que las formas y las pantallas de entrada satisfagan propósitos específicos del sistema de información de la administración, mientras que la precisión se refiere a un diseño tal que asegure una realización satisfactoria. La factibilidad de uso implica que tanto las formas como las pantallas serán explícitas y no requerirán de tiempo adicional para descifrarse. La consistencia, en este caso, significa que las formas y las pantallas ordenen los datos de manera similar de una aplicación a otra, mientras que la sencillez se refiere a mantener en un mínimo los elementos indispensables que centren la atención del usuario. La atracción implica que el usuario disfrutará del uso o tránsito a través de las formas y las pantallas cuyos diseños les sean más atractivos.

LINEAMIENTOS PARA EL DISEÑO DE PANTALLAS

A continuación se presentan cuatro lineamientos para el diseño de pantallas, que si bien son importantes, no son exhaustivos.

1. - Mantenga la pantalla sencilla.
2. - Mantenga una presentación consistente.
3. - Facilite los movimientos del usuario entre pantallas.
4. - Cree pantallas atractivas.

Debe mantenerse la sencillez de la pantalla

El primer lineamiento para un buen diseño de pantallas es mantener la sencillez de la pantalla. La pantalla debe mostrar sólo lo que es necesario para la acción particular que se lleva a cabo.

La figura 4.4 muestra las tres secciones que son útiles para simplificar la pantalla. La parte superior de la pantalla contiene la sección del encabezado, parte de la cual se encuentra programada para indicar al usuario dónde se encuentra dentro de la aplicación o paquete.

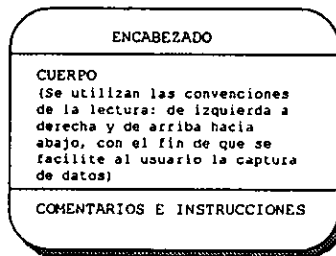


Figura 4.4 Tres secciones de una pantalla

En la segunda parte o cuerpo debe proporcionarse al usuario la definición de los campos indicando la dimensión previsible del dato para cada campo de la pantalla. Esto puede lograrse a través de diferentes características de diseño: video inverso que resalte la longitud precisa de cada campo; recuadros que indiquen cuáles datos cabrían, llaves, paréntesis o diagonales que indiquen el comienzo y el final de los campos; o bien, instrucciones al usuario para que éste capture sus datos de manera directa sobre un área de símbolos distintivos, que reemplazarían de manera automática los datos.

La tercera sección de la pantalla se denomina sección de "Comentarios o Instrucciones". Esta sección puede contener un menú conciso de órdenes que recuerde al usuario las funciones básicas del sistema, tales como el cambio de pantalla, o funciones tales como la grabación de registros o la conclusión de la sesión de captura. La inclusión de tales funciones básicas permite que los usuarios sin experiencia se sientan más seguros de su habilidad para operar la computadora sin llegar a ocasionar errores fatales.

Conservar consistencia en las pantallas

El segundo lineamiento para un buen diseño de pantalla es el mantenimiento de una imagen consistente. La consistencia de la pantalla se mantiene, si la información se localiza en la misma área cada vez que se acceda una nueva pantalla. La información que tenga alguna relación lógica entre sí, debe presentarse en forma agrupada: el nombre y apellidos van juntos, más no el nombre y el código postal.

Facilidad de movimiento

El tercer lineamiento para un buen diseño de pantalla es la factibilidad de desplazarse con facilidad entre una pantalla y otra. Un método comúnmente utilizado, es hacer sentir al usuario que se mueve físicamente hacia una nueva pantalla.

Diseño de una pantalla atractiva

Si el usuario se siente atraído por la pantalla, es muy probable que sea más productivo, requiera de menos supervisión y cometa menos errores. Las pantallas deben atraer al usuario y mantener su atención. Esto se logra con el uso de espacios abiertos que rodeen los campos de captura de datos, de tal forma que la pantalla no se vea sobrecargada, nunca se debe saturar una pantalla, siempre será mejor utilizar pantallas múltiples, que amontonar todo en una sola. Al crear pantallas que de primera instancia son fáciles de recordar, atraerá tanto a los usuarios inexpertos como a los expertos. Si la pantalla requiere ser compleja, tome muy en cuenta la presentación, separe la información por categorías, por medio de líneas formadas por puntos, diagonales, comas, signos, etc.

EL USO DE IMÁGENES EN EL DISEÑO DE PANTALLAS

Las imágenes (iconos) son representaciones pictóricas en la pantalla, las cuales simbolizan ciertas acciones de cómputo que los usuarios pueden seleccionar mediante un ratón, el teclado, un lápiz óptico o una palanca de comando. Las imágenes hacen funciones similares a las palabras y pueden reemplazarlas en menús numerosos, ya que su significado se comprende con mayor rapidez que el de las palabras.

Existen ciertos lineamientos para el diseño efectivo de imágenes. Las imágenes deben ser fácilmente reconocibles, de tal forma que el usuario no requiera dominar un nuevo vocabulario. Existen numerosas imágenes conocidas por la mayoría de los usuarios. Un usuario puede señalar un archivero, "sacar" una imagen de un folder, "copiar" una hoja de papel y "tirar" la imagen al cesto de basura. Tanto los diseñadores como los usuarios, ahorran tiempo al utilizar imágenes estándar.

Utilice de manera consistente las imágenes que a todo lo largo de la aplicación deben aparecer juntas. Esto asegura continuidad y comprensión.

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:

- ❖ Asegurar que solo los usuarios autorizados tengan acceso al sistema.
- ❖ Garantizar que las transacciones sean aceptables.
- ❖ Validar los datos para comprobar su exactitud.
- ❖ Determinar si se han omitido datos que son necesarios.

DISEÑO DE ARCHIVOS Y DE BASE DE DATOS

Algunas personas consideran el almacenamiento de los datos como la esencia del sistema de información. Los objetivos generales del diseño de la organización del almacenamiento de los datos se muestran en la figura 4.5.



Figura 4.5 Cinco objetivos en el diseño en el almacenamiento de datos

Primero, los datos deben estar disponibles para cuando el usuario desee usarlos. Segundo, los datos deben ser precisos y consistentes (deben poseer una integridad). Más allá de esto, dentro de los objetivos se incluye un almacenamiento eficiente de los datos, así como su actualización y grabado eficientes. Finalmente, es necesario que el acceso a la información tenga un propósito. La información obtenida de los datos almacenados debe contar con un formato útil que facilite la administración, la planeación, el control o la toma de decisiones.

ARCHIVOS Y BASES DE DATOS CONVENCIONALES

En un sistema de información basado en computadora se cuenta con dos enfoques para el almacenamiento de los datos. El primer método consiste en almacenar los datos en archivos individuales, exclusivos para una aplicación en particular.

El segundo enfoque para el almacenamiento de datos en un sistema basado en computadora, involucra la elaboración de una base de datos. Una base de datos es un almacenamiento de datos formalmente definido, controlado centralmente para servir a múltiples y diferentes aplicaciones.

Archivos

Los archivos convencionales, sin duda alguna, permanecen como una manera práctica de almacenar los datos de ciertas (sino de todas) las aplicaciones. Un archivo convencional se puede diseñar y elaborar de manera rápida, reduciendo los problemas de disponibilidad de datos y de seguridad. Cuando el diseño de los archivos se realiza de manera cuidadosa, toda la información necesaria queda incluida y se reduce el riesgo de omitir datos de manera accidental.

La velocidad de procesamiento es una ventaja para el uso de archivos. Hay posibilidad de elegir una técnica óptima para el procesamiento de los archivos de una aplicación sencilla, pero llega a ser imposible alcanzar un diseño óptimo para tareas muy variadas.

El uso de archivos individuales tiene diversas consecuencias. Uno de los principales problemas de los archivos es la falta de potencial para evolucionar; con frecuencia, los archivos se diseñan con base en las necesidades inmediatas. El rediseño de archivos, implica a menudo que los programas que los accesan, deban redactarse nuevamente de manera acorde. Esto implica un incremento en el tiempo de programación para el archivo, en el desarrollo y mantenimiento de programas.

Bases de Datos

Las bases de datos no son meramente una colección de archivos. Más bien, una base de datos es una fuente central de datos significativos, los cuales son compartidos por numerosos usuarios para diversas aplicaciones. La esencia de una base de datos es el Sistema Administrador de la Base de Datos (DBMS: Database Management System), el cual permite la creación, modificación y actualización de la base de datos; la recuperación de los datos y la emisión de reportes. A la persona responsable de asegurar que la base de datos satisfaga los objetivos programados se le denomina administrador de la base de datos.

Los objetivos de eficacia de la base de datos son:

- ❖ Asegurar que los datos puedan ser compartidos por los usuarios, para una variedad de aplicaciones.
- ❖ Que el mantenimiento de los datos sea preciso y consistente.
- ❖ Asegurar que todos los datos requeridos para las aplicaciones presentes y futuras se encuentren siempre disponibles.
- ❖ Permitir que la base de datos evolucione y se adapte a las necesidades crecientes de los usuarios.
- ❖ Permitir que los usuarios desarrollen su propia visión de los datos, sin preocuparse por la manera en que los datos se encuentren almacenados físicamente.

Una desventaja del enfoque de base de datos es que todos los datos se almacenan en un solo lugar; y en consecuencia, los datos son más vulnerables a accidentes y requerirán de un respaldo completo. Existe el riesgo de que quien administra la base de datos se convierta en el único privilegiado o habilitado para estar cerca de los datos. Y los procedimientos burocráticos requeridos para modificar o para actualizar la base de datos pueden llegar a ser insuperables.

Otras desventajas para la administración de los datos como recurso se presenta al intentar satisfacer dos objetivos de eficiencia:

- ❖ Reducción del tiempo requerido para insertar, actualizar, eliminar y recuperar los datos en tiempos tolerables.
- ❖ Mantenimiento del costo del almacenamiento de datos en una cantidad razonable.

Es necesario recordar que una base de datos no puede optimizar la recuperación de los datos para una aplicación especial, ya que deberá compartirse con numerosos usuarios y con varias aplicaciones. Además, se puede llegar a requerir de cierto software adicional para la administración de la base de datos (DBMS) y, ocasionalmente, también se necesitará una computadora de mayor capacidad.

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 los siguientes:

PROCEDIMIENTO PARA ENTRADA DE DATOS:

Métodos para la captura de datos de las transacciones y su ingreso al sistema de información.

PROCEDIMIENTO 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 la impresora se atasca durante la impresión de una proceso.

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 o partes de las bases de datos?

Estos procedimientos deben formularse por escrito y formar parte de la documentación del sistema.

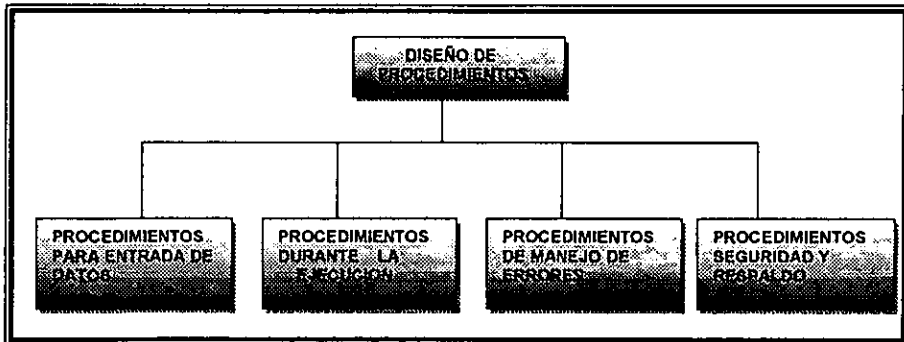


Figura 4.6. Diseño de procedimientos.

DISEÑO DE ESPECIFICACIONES PARA PROGRAMAS

Las especificaciones para programas son por si mismas un diseño. Describen como transformar las especificaciones del diseño del sistema; salidas, entradas, archivos, procesamiento y otras; en software de computadora.

El diseño del software 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 puedan realizar en forma eficiente y con un mínimo de interrupción en el diseño del sistema.

CARPETA DE DESCRIPCIÓN DEL DISEÑO DEL SISTEMA

Cuando 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 está 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 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:

Diagramas 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.

4.3 DIAGRAMA DE FLUJO DE DATOS DEL SISE

El Diagrama de Flujo de Datos del SISE tiene como objetivo mostrar en forma las propuestas tanto del usuario como del analista para mejorar los procedimientos manuales con el fin de delimitar la frontera hombre - máquina. El DFD de Contexto se muestra en la figura 4.7.

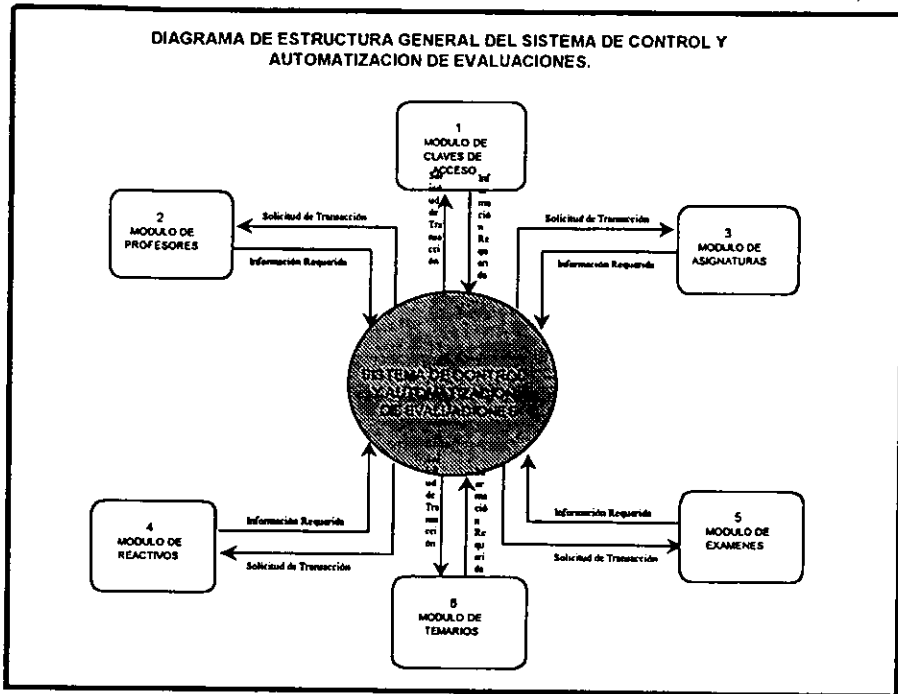


Figura 4.7 Diagrama de Contexto SISE.

En este DFD se identifican seis unidades o módulos: Módulo de Claves de Acceso, Módulo de Profesores, Módulo de Asignaturas, Módulo de Reactivos, Módulo de Exámenes y Módulo de Temarios. Así mismo se visualizan los flujos de datos que los diferentes módulos reciben y entregan al sistema.

Después de esquematizar las principales entidades y flujos se procede describir lo relacionado con cada uno de los procesos, es decir, cuales flujos de datos recibe, cuales emite y su significado.

4.3.1 MODULO DE CLAVES DE ACCESO

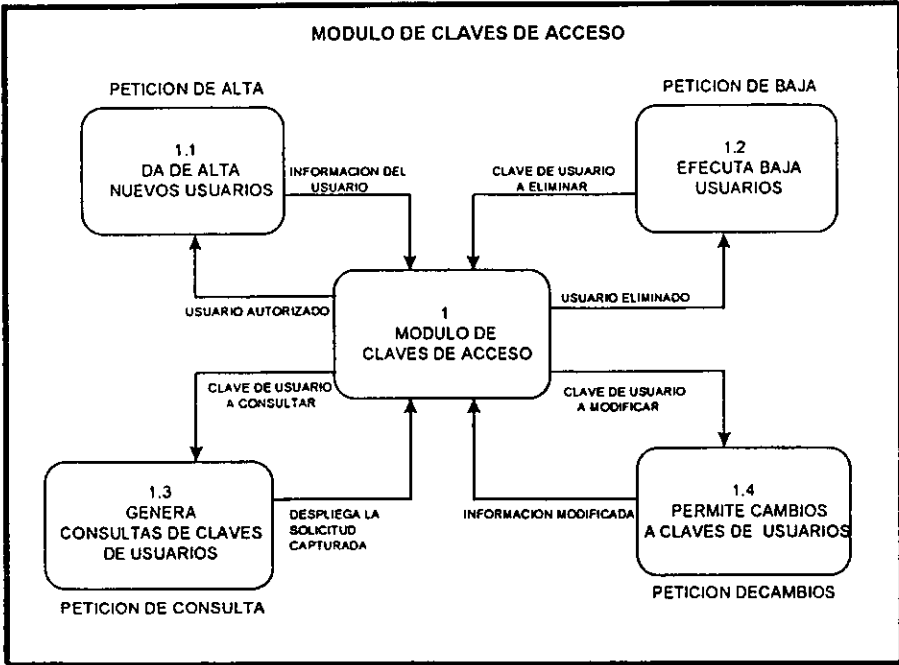


Figura 4.8 DFD Módulo de Claves de Acceso.

El Módulo de Claves de Acceso engloba una serie de acciones en donde el Administrador del sistema puede manipular la cantidad de usuarios autorizados para utilizar el sistema, por medio de éste el Administrador realiza la Captura de nuevos Usuarios, la Eliminación de aquellos usuarios que ya no estén autorizados, los Cambios de Claves de Usuarios así como la Consulta de Claves.

Si se trata de una *Baja de Claves* así como un *Cambio de Claves* se llevan a cabo primero la consulta o visualización de los datos actuales para posteriormente permitir ya sea la eliminación o la modificación de dicho registro.

Si se trata de una *Consulta de Claves* se elaboran las consultas de acuerdo únicamente a la Clave a consultar.

4.3.2 MODULO DE PROFESORES

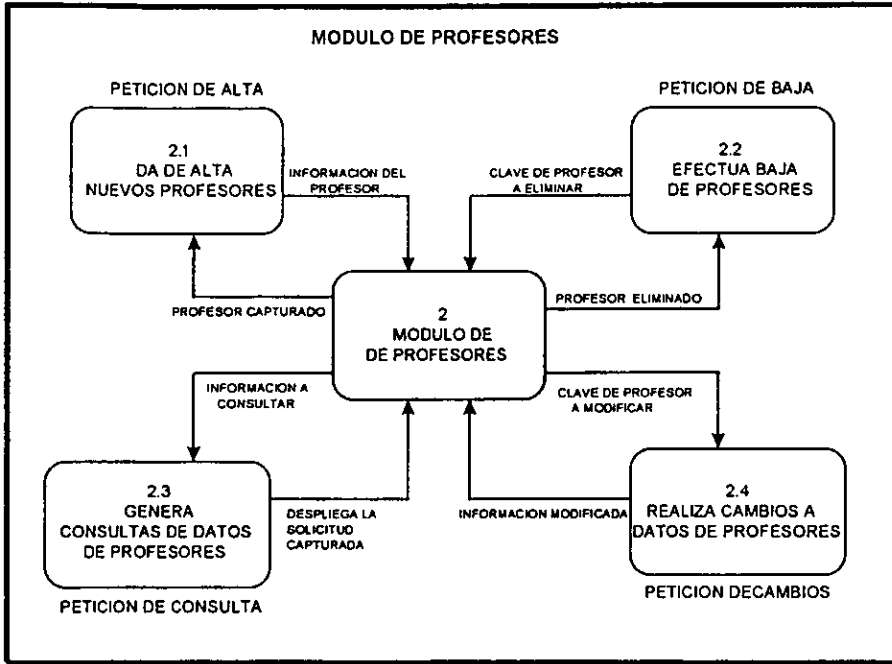


Figura 4.9 DFD Módulo de Profesores.

El Módulo de Profesores abarca las siguientes acciones: *Alta Profesor*, que le permite al Coordinador de las Academias o bien al Administrador del Sistema realizar la captura de nuevos profesores en el sistema. *Baja Profesor*, por medio de la cual se puede depurar la cantidad de profesores registrados. *Cambios Profesor*, efectúa las modificaciones a los datos de los profesores registrados en el sistema.

Si se trata de una *Baja de Profesores* así como un *Cambio de Profesores* se llevan a cabo primero la consulta o visualización de los datos actuales para posteriormente permitir ya sea la eliminación o la modificación de dicho registro.

Si se trata de una *Consulta de Profesores* se efectúan las consultas de acuerdo a:

- ❖ Consultas por Clave Profesor.
- ❖ Consultas por Nombre Profesor.

El DFD de las Consultas Profesor se visualiza en la figura 4.10.

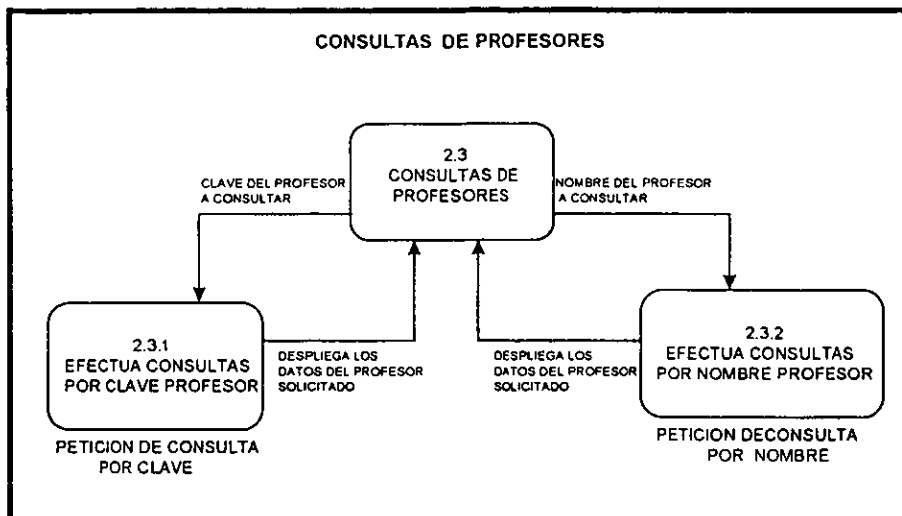


Figura 4.10 DFD Consultas de Profesores.

4.3.3 MODULO DE ASIGNATURAS

El Diagrama de Flujo de Datos del Módulo de Asignaturas (figura 4.11) nos permite visualizar las acciones que se pueden realizar para las Asignaturas en el SISE. El Coordinador de las Academias o bien el Administrador del Sistema pueden realizar la captura de las nuevas Asignaturas de las carreras que existen en el CETIS por medio de las *Altas de Asignaturas*. El proceso *Bajas de Asignaturas* realiza la eliminación de las Asignaturas que se den de baja de los diferentes planes de estudios con que cuenta el CETIS. Dentro de *Cambios de Asignaturas* se pueden realizar las modificaciones a los datos de las Asignaturas registradas en el Sistema de Control y Automatización de Evaluaciones.

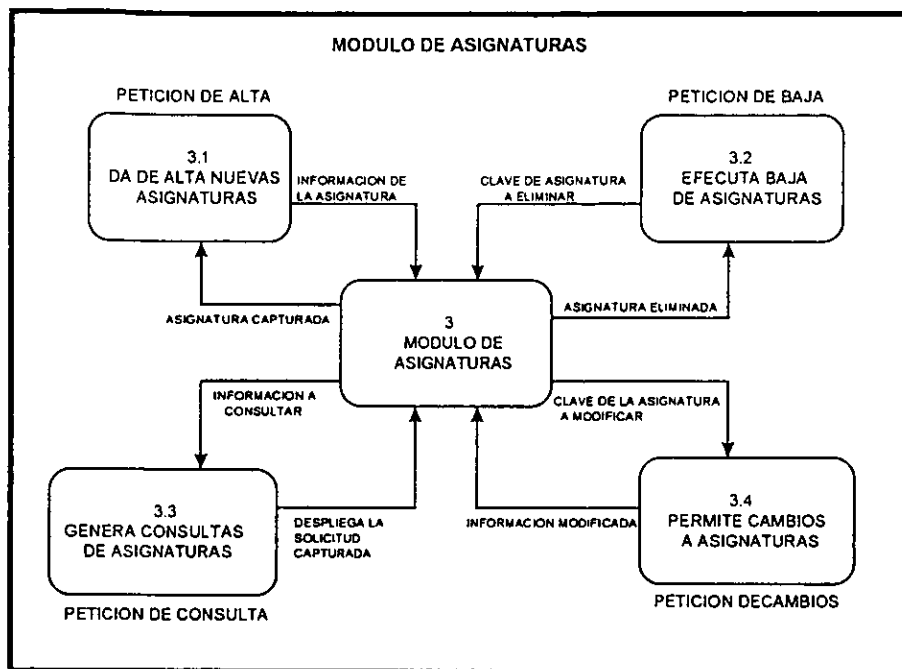


Figura 4.11 DFD Módulo de Asignaturas.

En la sección correspondiente a las *Consultas de Asignaturas* se pueden visualizar los datos generales de las Asignaturas que se imparten en el CETIS 152. Existen diferentes medios de Consultas de Asignaturas como se puede ver en la figura 4.12.

- ❖ Consultas por Clave de Asignatura.
- ❖ Consultas por Nombre de la Asignatura (en donde se puede elegir del índice de Asignaturas existentes)
- ❖ Consultas por Academia (en donde se puede elegir del índice de Academias existentes)

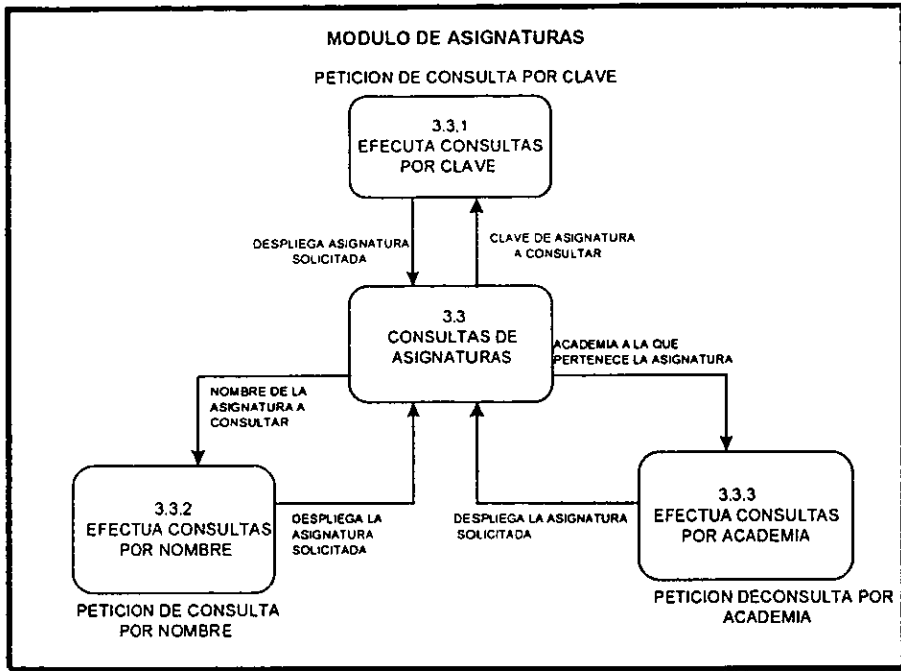


Figura 4.12 DFD Consultas Asignaturas.

4.3.4 MODULO DE REACTIVOS

El módulo de Reactivos es la parte donde el Sistema permite al usuario del Sistema (a excepción de los *Invitados*) a capturar, eliminar, modificar o consultar los diferentes tipos de reactivos que se utilizan para las evaluaciones dentro del Cetis 152. Dichos tipos de reactivos o preguntas se caracterizan por el tipo de respuesta correspondiente y son los siguientes:

- ❖ Tipo Respuesta Simple Breve
- ❖ Tipo Respuesta Abierta
- ❖ Tipo Falso ó Verdadero
- ❖ Tipo Opción Múltiple
- ❖ Tipo Relación de Columnas
- ❖ Tipo Solución a Problemas
- ❖ Tipo Localización ó Identificación.

Cabe señalar que estos tipos serán detallados con mayor énfasis en el siguiente capítulo.

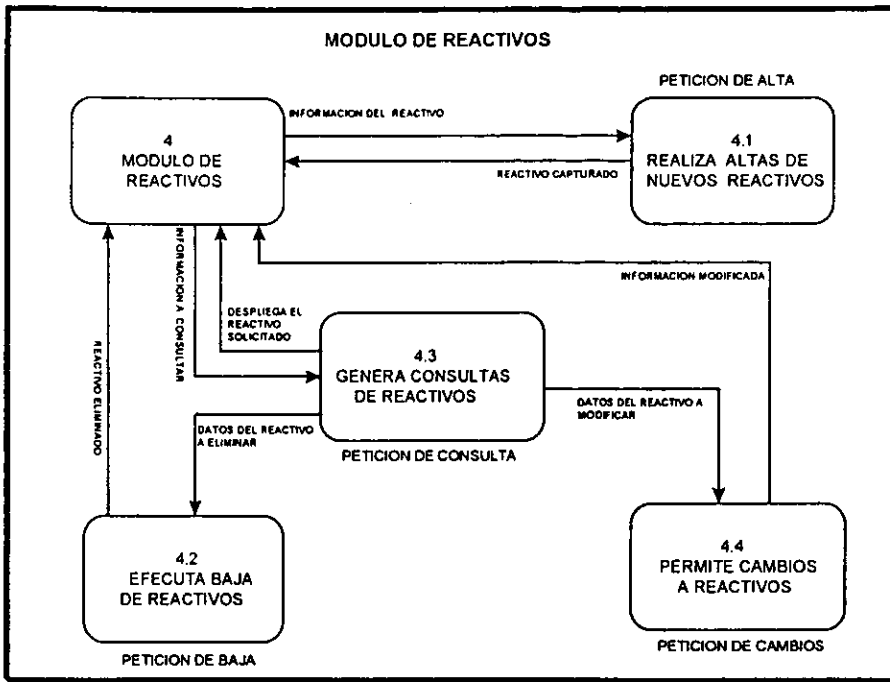


Figura 4.13 DFD Módulo de Reactivos.

Como se puede observar en el diagrama anterior, para la realización de una *Baja de Reactivos* o para hacer *Cambios de Reactivos* se debe primero de realizar una *Consulta de Reactivos* debido a que las consultas se realizan por grupos de reactivos y no por reactivos individuales.

Existen diferentes opciones para realizar una *Consulta de Reactivos* partiendo siempre del Tipo al que pertenezca el reactivo a consultar:

- ❖ Por Asignatura
- ❖ Por Nivel de Dificultad (diagnóstico, parcial, final o extraordinario)
- ❖ Por Fecha de Creación
- ❖ Por Clave del Profesor que lo capturó

O bien se puede realizar la combinación de una o varias de las opciones anteriores para reducir el número del conjunto de reactivos seleccionados.

Para observar mejor lo anterior se presenta la figura 4.14.

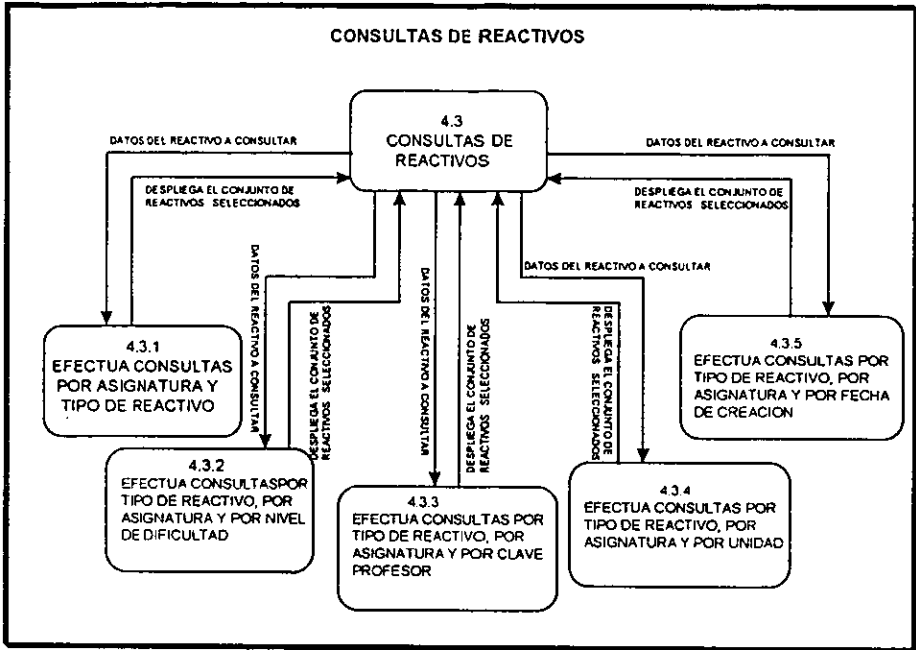


Figura 4.14 DFD Consultas de Reactivos.

4.3.5 MODULO DE EXAMENES

Este módulo está enfocado básicamente a la elaboración, consulta, eliminación y modificación de exámenes.

Para la realización de exámenes (*Altas*) debemos de realizar primero la consulta y selección de los reactivos correspondientes a la Asignatura y nivel de dificultad a los que pertenezca el examen. En esta sección del módulo podemos guardar el examen creado como documento dentro de la Base de Datos para su posterior consulta o modificación.

Para la *Consulta* de exámenes debemos de determinar los datos característicos del examen a consultar, como son, la asignatura, el nivel de dificultad, el profesor que lo elaboró, la fecha de

creación o alta, etc. en donde el sistema nos mostrará un conjunto de exámenes que cumplen con dichas características, los cuales se podrán consultar uno a uno.

Para la *Modificación y Eliminación* de exámenes debemos de consultar primero el examen. Una vez teniendo en la pantalla el examen seleccionado se podrán realizar las modificaciones necesarias y grabar el documento modificado, o bien, eliminarlo.

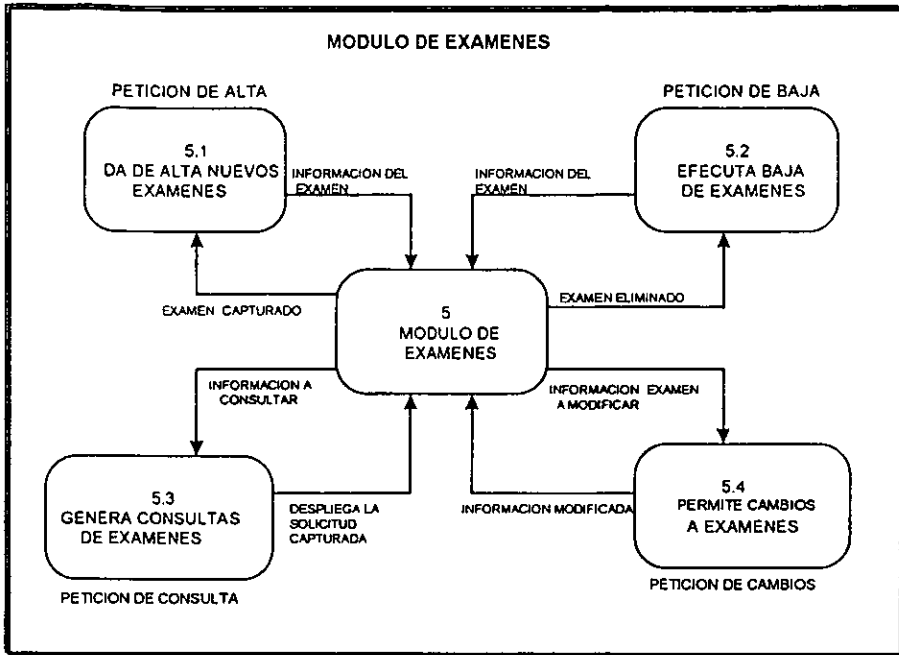


Figura 4.15 DFD Módulo de Exámenes.

4.3.6 MODULO DE TEMARIOS

En este módulo se pueden realizar las operaciones para los Temarios de las Asignaturas grabadas en la Base de Datos. Las operaciones que se podrán realizar, como en todos los módulos son: Altas, Bajas, Modificación y Consulta de Temarios.

Todas las operaciones requieren de la clave de la Asignatura a la cual pertenece el Temario, en caso de que no exista la Asignatura el Temario no podrá darse de *Alta*. En la sección de *Cambios* se podrán realizar las modificaciones que pudieran surgir a los Temarios de un periodo a otro, así mismo, en la sección de *Bajas* se podrán eliminar los Temarios de las Asignaturas que dejen de pertenecer a los planes de estudios con que cuenta el Cetis 152.

Para la elaboración de reactivos se requiere conocer la unidad o unidades del Temario a las que pertenece por lo que las *Consultas* a los Temarios se utilizarán para este fin.

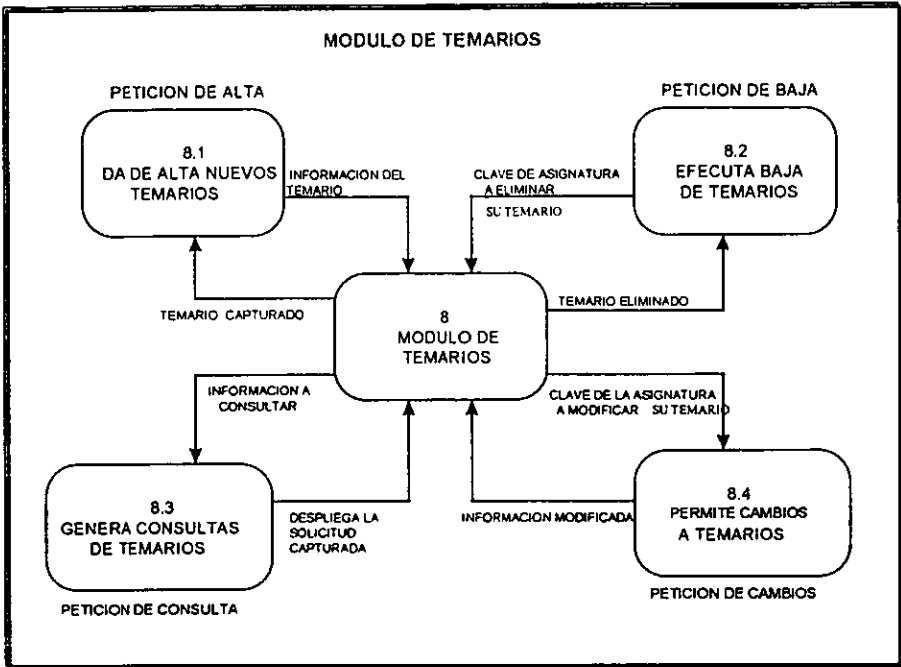


Figura 4.16 DFD Módulo de Temarios.

4.4 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 las entradas, salidas y componentes de almacenamiento. El empleo del diccionario de datos reduce ambigüedades y especifica detalles.

La siguiente figura muestra las tablas con sus campos, así como las características que conforman la Base de Datos de SISE.

TABLA DE PROFESORES (fig. 4.1)

Almacenará los principales datos pertenecientes a los profesores que laboran en el Cetis 152, así como a todos los usuarios que utilicen este sistema.

TABLA DE ASIGNATURAS (fig. 4.2)

Esta tabla contendrá los datos básicos de las asignaturas que se imparten en las carreras con que cuenta el Cetis 152, esta tabla nos permitirá acceder y dar mantenimiento a dichas asignaturas mediante el SISE.

TABLA DE TEMARIOS (fig. 4.3)

Almacena la información correspondiente a los Temarios de las asignaturas dadas de alta dentro del sistema, esta información nos servirá como ayuda auxiliar para la obtención de las unidades que conforman los Temarios para cada una de las Asignaturas que se imparten en el Cetis 152. Esta tabla nos permitirá dar mantenimiento a dichos Temarios en cualquier momento.

TABLA DE REACTIVOS (TIPO 1,2 Y 3) (fig.4.4)

Aquí se guardarán los reactivos pertenecientes a los tres primeros tipos (Tipo Respuesta Simple Breve, Tipo Respuesta Abierta y Tipo Falso ó Verdadero) así como los datos que los caracterizan. Esta tabla nos permitirá obtener los reactivos las veces que sea necesario de una forma sencilla.

TABLA DE REACTIVOS (TIPO 4) (fig.4.5)

Almacenará los datos relacionados con los reactivos tipo Opción Múltiple.

TABLA DE REACTIVOS (TIPO 5) (fig.4.6)

Almacenará los datos relacionados con los reactivos tipo Relación de Columnas.

TABLA DE REACTIVOS (TIPO 6) (fig.4.7)

Almacenará los datos relacionados con los reactivos tipo Localización ó Identificación.

TABLA DE REACTIVOS (TIPO 7) (fig.4.8)

Almacenará los datos relacionados con los reactivos tipo Solución a Problemas.

TABLA DE EXAMENES (fig.4.9)

Esta Tabla nos permitirá almacenar dentro de la Base de Datos los exámenes que se realicen, además de que nos dará la facilidad para poder consultar, modificar o eliminar exámenes elaborados anteriormente. Esta tabla contiene los datos básicos para poder identificar un examen o un conjunto de ellos con facilidad.

TABLA DE CLAVES (fig.4.10)

La Tabla de Claves de la Base de Datos contendrá, como su nombre nos dice, las Claves de identificación de los usuarios que estén dados de Alta en el sistema, así como sus passwords asignados, los cuales servirán para el acceso al SISE. Esta tabla nos permitirá resguardar la seguridad de la información de la Base de Datos ya que aunado a la Clave y el Password se contará con un nivel de seguridad asignado al usuario dependiendo del puesto que ocupe dentro de la institución el cual le permitirá determinados privilegios dentro del sistema (los niveles de seguridad así como los privilegios asignados se detallan con mayor profundidad en el siguiente capítulo).

TABLA DE PASO (fig.4.11)

La tabla de Paso, como su nombre lo indica, nos servirá únicamente como puente para la creación de un examen en particular. Durante el proceso de Atlas de Exámenes se elegirán los reactivos que conformarán el nuevo examen, dichos reactivos, así como los datos característicos del examen se almacenarán temporalmente en esta tabla, la cual se destruirá en el momento en que se grabe la información dentro de la tabla de Exámenes, quedando disponible para un nuevo examen.

TABLA DE PROFESORES

NOMBRE CAMPO	TIPO	LONGITUD	LLAVE	DESCRIPCION
CLAVEPROFE	NUMERICO	02	SI	Es la clave de identificación del Profesor.
NOMBPROFE	TEXTO	30	NO	Es el nombre(s) del Profesor
APEPPROFE	TEXTO	30	NO	Es el apellido paterno del Profesor
APEMPROFE	TEXTO	30	NO	Es el apellido materno del Profesor
CALLEPROFE	TEXTO	30	NO	Es la calle y número del domicilio del Profesor
COLONPROFE	TEXTO	30	NO	Es la colonia del domicilio particular del Profesor
CIUDAPROFE	TEXTO	30	NO	Es la ciudad del domicilio del Profesor
CPOSTPROFE	NUMERICO	05	NO	Es el código postal del domicilio del Profesor
TELEFPROFE	TEXTO	13	NO	Es el número telefónico del Profesor
EDOCIPROFE	TEXTO	10	NO	Es el estado civil del Profesor
PERFLPROFE	TEXTO	20	NO	Es la formación académica que tiene el Profesor
STATUPROFE	TEXTO	08	NO	Es el status (activo/inactivo) que tiene el Profesor
PUESTPROFE	TEXTO	20	NO	Es el puesto que ocupa el Profesor en el Cetis
NIVELPROFE	NUMERICO	01	NO	Es el nivel de usuario que tiene el Profesor para uso del Sistema.
AREAPROFE	TEXTO	30	NO	Es el área a la que pertenece el Profesor

Tabla 4.1 Tabla de profesores

TABLA DE ASIGNATURAS

NOMBRE CAMPO	TIPO	LONGITUD	LLAVE	DESCRIPCION
CLAVEASIGN	NUMERICO	03	SI	Es la clave de identificación de la Asignatura
NOMBASIGN	TEXTO	30	NO	Es el nombre de la Asignatura
TIPOASIGN	TEXTO	16	NO	Es la categoría a la que pertenece la Asignatura (Teórica, Práctica o Teórica - Práctica)
CARREASIGN	TEXTO	03	NO	Es la carrera a la que pertenece la Asignatura
ACADEASIGN	TEXTO	25	NO	Es la academia a la que pertenece la Asignatura
SEMESASIGN	NUMERICO	02	NO	Es el semestre en el que se imparte la Asignatura
CREDIASIGN	NUMERICO	02	NO	Es el número de créditos que tiene la Asignatura
PERIODOASIGN	TEXTO	20	NO	Es el período en el que se imparte la Asignatura

Tabla 4.2 Tabla de Asignaturas

TABLA DE TEMARIOS

NOMBRE CAMPO	TIPO	LONGITUD	LLAVE	DESCRIPCION
CLAVEASIGN	NUMERICO	03	SI	Es la Clave de la Asignatura a la que pertenece el Temario
FECHATEMAR	TEXTO	11	NO	Es la fecha de Alta o Actualización del Temario
TEXTOTEMAR	MEMO	Hasta 65,535 caracteres	NO	Es el Temario de la Asignatura

Tabla 4.3 Tabla de Temarios

TABLA DE REACTIVOS (TIPO 1, 2 y 3)

NOMBRE CAMPO	TIPO	LONGITUD	LLAVE	DESCRIPCION
NUMREACTIVO	AUTONUMERICO	03	NO	Es el número consecutivo del Reactivo dentro de la Tabla
CLAVEASIGN	NUMERICO	03	NO	Es la clave de la Asignatura a la que pertenece el Reactivo
TIPOREACTIVO	NUMERICO	02	NO	Es el Tipo al que pertenece el Reactivo (Ver tabla de Tipos)
EXAMENDIAG	SI/NO	02	NO	Es el nivel de dificultad más bajo (examen diagnóstico)
EXAMENPARC	SI/NO	02	NO	Es el nivel de dificultad para un examen parcial
EXAMENFINAL	SI/NO	02	NO	Es el nivel de dificultad para un examen final
EXAMENEXT	SI/NO	02	NO	Es el nivel de dificultad más alto (examen extraordinario)
UNIDAD	NUMERICO	02	NO	Es la Unidad del Temario a la que pertenece el Reactivo
FECHAREAC	TEXTO	10	NO	Es la fecha del día de alta del Reactivo
CLAVEPROFE	NUMERICO	02	NO	Es la clave del profesor que dio de alta o realizó la última modificación al Reactivo
TEXTOREAC	MEMO	Hasta 65,535 caracteres	NO	Es el Texto o Pregunta del Reactivo
AUTOR	TEXTO	30	NO	Es el Autor del libro del cual se obtuvo el Reactivo, o bien, es el nombre del Profesor que lo dio de alta (campo no obligatorio)
TITULO	TEXTO	30	NO	Es el Título del libro del cual se obtuvo el Reactivo (campo no obligatorio)
EDITORIAL	TEXTO	20	NO	Es la Editorial del libro del cual se obtuvo el Reactivo (campo no obligatorio)

Tabla 4.4 Tabla de reactivos tipos del 1 al 3

TABLA DE REACTIVOS (TIPO 4)

NOMBRE CAMPO	TIPO	LONGITUD	LLAVE	DESCRIPCION
NUMREACTIVO	AUTONUMERICO	03	NO	Es el número consecutivo del Reactivo dentro de la Tabla
CLAVEASIGN	NUMERICO	03	NO	Es la clave de la Asignatura a la que pertenece el Reactivo
TIPOREACTIVO	NUMERICO	02	NO	Es el Tipo al que pertenece el Reactivo (Ver tabla de Tipos)
EXAMENDIAG	SI/NO	02	NO	Es el nivel de dificultad más bajo (examen diagnóstico)
EXAMENPARC	SI/NO	02	NO	Es el nivel de dificultad para un examen parcial
EXAMENFINAL	SI/NO	02	NO	Es el nivel de dificultad para un examen final
EXAMENEXT	SI/NO	02	NO	Es el nivel de dificultad más alto (examen extraordinario)
UNIDAD	NUMERICO	02	NO	Es la Unidad del Temario a la que pertenece el Reactivo
FECHAREAC	TEXTO	10	NO	Es la fecha del día de alta del Reactivo
CLAVEPROFE	NUMERICO	02	NO	Es la clave del profesor que dio de alta o realizó la última modificación al Reactivo
TEXTOREAC	MEMO	Hasta 65,535 caracteres	NO	Es el Texto o Pregunta del Reactivo
AUTOR	TEXTO	30	NO	Es el Autor del libro del cual se obtuvo el Reactivo, o bien, es el nombre del Profesor que lo dio de alta (campo no obligatorio)
TITULO	TEXTO	30	NO	Es el Título del libro del cual se obtuvo el Reactivo (campo no obligatorio)
EDITORIAL	TEXTO	20	NO	Es la Editorial del libro del cual se obtuvo el Reactivo (campo no obligatorio)

Tabla 4.5 Tabla de reactivos tipo 4

TABLA DE REACTIVOS (TIPO 5)

NOMBRE CAMPO	TIPO	LONGITUD	LLAVE	DESCRIPCION
NUMREACTIVO	AUTONUMERICO	03	NO	Es el número consecutivo del Reactivo dentro de la Tabla
CLAVEASIGN	NUMERICO	03	NO	Es la clave de la Asignatura a la que pertenece el Reactivo
TIPOREACTIVO	NUMERICO	02	NO	Es el Tipo al que pertenece el Reactivo (Ver tabla de Tipos)
EXAMENDIAG	SI/NO	02	NO	Es el nivel de dificultad más bajo (examen diagnóstico)
EXAMENPARC	SI/NO	02	NO	Es el nivel de dificultad para un examen parcial
EXAMENFINAL	SI/NO	02	NO	Es el nivel de dificultad para un examen final
EXAMENEXT	SI/NO	02	NO	Es el nivel de dificultad más alto (examen extraordinario)
UNIDAD	NUMERICO	02	NO	Es la Unidad del Temario a la que pertenece el Reactivo
FECHAREAC	TEXTO	10	NO	Es la fecha del día de alta del Reactivo
CLAVEPROFE	NUMERICO	02	NO	Es la clave del profesor que dio de alta o realizó la última modificación al Reactivo
TEXTOREAC	MEMO	Hasta 65,535 caracteres	NO	Es el Texto o Pregunta del Reactivo
AUTOR	TEXTO	30	NO	Es el Autor del libro del cual se obtuvo el Reactivo, o bien, es el nombre del Profesor que lo dio de alta (campo no obligatorio)
TITULO	TEXTO	30	NO	Es el Título del libro del cual se obtuvo el Reactivo (campo no obligatorio)
EDITORIAL	TEXTO	20	NO	Es la Editorial del libro del cual se obtuvo el Reactivo (campo no obligatorio)

Tabla 4.6 Tabla de reactivos tipo 5

TABLA DE REACTIVOS (TIPO 6)

NOMBRE CAMPO	TIPO	LONGITUD	LLAVE	DESCRIPCION
NUMREACTIVO	AUTONUMERICO	03	NO	Es el número consecutivo del Reactivo dentro de la Tabla
CLAVEASIGN	NUMERICO	03	NO	Es la clave de la Asignatura a la que pertenece el Reactivo
TIPOREACTIVO	NUMERICO	02	NO	Es el Tipo al que pertenece el Reactivo (Ver tabla de Tipos)
EXAMENDIAG	SI/NO	02	NO	Es el nivel de dificultad más bajo (examen diagnóstico)
EXAMENPARC	SI/NO	02	NO	Es el nivel de dificultad para un examen parcial
EXAMENFINAL	SI/NO	02	NO	Es el nivel de dificultad para un examen final
EXAMENEXT	SI/NO	02	NO	Es el nivel de dificultad más alto (examen extraordinario)
UNIDAD	NUMERICO	02	NO	Es la Unidad del Temario a la que pertenece el Reactivo
FECHAREAC	TEXTO	10	NO	Es la fecha del día de alta del Reactivo
CLAVEPROFE	NUMERICO	02	NO	Es la clave del profesor que dio de alta o realizó la última modificación al Reactivo
TEXTOREAC	MEMO	Hasta 65,535 caracteres	NO	Es el Texto o Pregunta del Reactivo
AUTOR	TEXTO	30	NO	Es el Autor del libro del cual se obtuvo el Reactivo, o bien, es el nombre del Profesor que lo dio de alta (campo no obligatorio)
TITULO	TEXTO	30	NO	Es el Título del libro del cual se obtuvo el Reactivo (campo no obligatorio)
EDITORIAL	TEXTO	20	NO	Es la Editorial del libro del cual se obtuvo el Reactivo (campo no obligatorio)
RUTAIMAGEN	TEXTO	255	NO	Es la ruta donde se encuentra la imagen 1
RUTAECUACION	TEXTO	255	No	Es la ruta donde se encuentra la imagen 2

Tabla 4.7 Tabla de reactivos tipo 6

TABLA DE REACTIVOS (TIPO 7)

NOMBRE CAMPO	TIPO	LONGITUD	LLAVE	DESCRIPCION
NUMREACTIVO	AUTONUMERICO	03	NO	Es el número consecutivo del Reactivo dentro de la Tabla
CLAVEASIGN	NUMERICO	03	NO	Es la clave de la Asignatura a la que pertenece el Reactivo
TIPOREACTIVO	NUMERICO	02	NO	Es el Tipo al que pertenece el Reactivo (Ver tabla de Tipos)
EXAMENDIAG	SI/NO	02	NO	Es el nivel de dificultad más bajo (examen diagnóstico)
EXAMENPARC	SI/NO	02	NO	Es el nivel de dificultad para un examen parcial
EXAMENFINAL	SI/NO	02	NO	Es el nivel de dificultad para un examen final
EXAMENEXT	SI/NO	02	NO	Es el nivel de dificultad más alto (examen extraordinario)
UNIDAD	NUMERICO	02	NO	Es la Unidad del Temario a la que pertenece el Reactivo
FECHAREAC	TEXTO	10	NO	Es la fecha del día de alta del Reactivo
CLAVEPROFE	NUMERICO	02	NO	Es la clave del profesor que dio de alta o realizó la última modificación al Reactivo
TEXTOREAC	MEMO	Hasta 65,535 caracteres	NO	Es el Texto o Pregunta del Reactivo
AUTOR	TEXTO	30	NO	Es el Autor del libro del cual se obtuvo el Reactivo, o bien, es el nombre del Profesor que lo dio de alta (campo no obligatorio)
TITULO	TEXTO	30	NO	Es el Título del libro del cual se obtuvo el Reactivo (campo no obligatorio)
EDITORIAL	TEXTO	20	NO	Es la Editorial del libro del cual se obtuvo el Reactivo (campo no obligatorio)
RUTAIMAGEN	TEXTO	255	NO	Es la ruta donde se encuentra la imagen 1
RUTAECUACION	TEXTO	255	No	Es la ruta donde se encuentra la imagen 2

Tabla 4.8 Tabla de reactivos tipo 7

TABLA DE EXAMENES

NOMBRE CAMPO	TIPO	LONGITUD	LLAVE	DESCRIPCION
CLAVEPROFE	NUMERICO	02	NO	Es la Clave del Profesor que elaboró el examen
CLAVEASIGN	NUMERICO	03	NO	Es la Clave de la Asignatura a la que pertenece el Examen
TIPOEXAMEN	TEXTO	15	NO	Es el Tipo al que pertenece el Examen (Diagnóstico, Parcial, Final o Extraordinario)
FECHACREA	TEXTO	11	NO	Es la Fecha de elaboración del Examen
FECHAPLICA	TEXTO	11	NO	Es la fecha de Aplicación del Examen (campo no obligatorio)
EXAMEN	TEXTO	50	NO	Es la ruta donde se encuentra el documento del Examen

Tabla 4.9 Tabla de Exámenes

TABLA DE CLAVES

NOMBRE CAMPO	TIPO	LONGITUD	LLAVE	DESCRIPCION
CLAVEUSUARIO	NUMERICO	02	SI	Es la clave de identificación del Usuario
PASSWUSUARIO	TEXTO	10	NO	Es la contraseña o password personal del Usuario
NIVELUSUARIO	NUMERICO	01	NO	Es el nivel de seguridad del Usuario (1-Administrador, 2-Coordinador, 3-Usuario ó 4-Invitado)

Tabla 4.10 Tabla de Claves

TABLA DE PASO

NOMBRE CAMPO	TIPO	LONGITUD	LLAVE	DESCRIPCION
NUMBLOQUE	NUMERICO	03	SI	Número de Bloque o sección del Examen
INSTRBLOQUE	TEXTO	150	NO	Instrucciones del bloque o Sección del Examen
NUMREACTIVO	NUMERICO	03	SI	Número consecutivo del reactivo dentro del Bloque o sección
TIPOREACTIVO	NUMERICO	02	NO	Es el tipo al que pertenece el Reactivo (ver tabla de Tipos)
TEXTREACTIVO	MEMO	Hasta 65,535 caracteres	NO	Texto del Reactivo
IMA1REACTIVO	TEXTO	250	NO	Imagen 1 del Reactivo (No necesaria para algunos tipos)
IMA2REACTIVO	TEXTO	250	NO	Imagen 2 del Reactivo (No necesaria para algunos tipos)
NOMBPROFE	TEXTO	35	NO	Nombre del profesor que está elaborando el examen
NOMBASIGN	TEXTO	30	NO	Nombre de la Asignatura a la que pertenece el Examen
TIPOEXAMEN	TEXTO	20	NO	Es el Tipo de Examen (Diagnóstico, Parcial, Final, Extraordinario)
FECHAPLICA	TEXTO	11	NO	Es la fecha en que se va a aplicar el Examen (campo no obligatorio)

Tabla 4.11 Tabla de Paso

4.5 DIAGRAMA ENTIDAD - RELACION

Es importante determinar la relación que existe entre los datos del sistema, por lo cual se determina un diagrama estructurado de información que nos muestra ésta y las entidades más importantes.

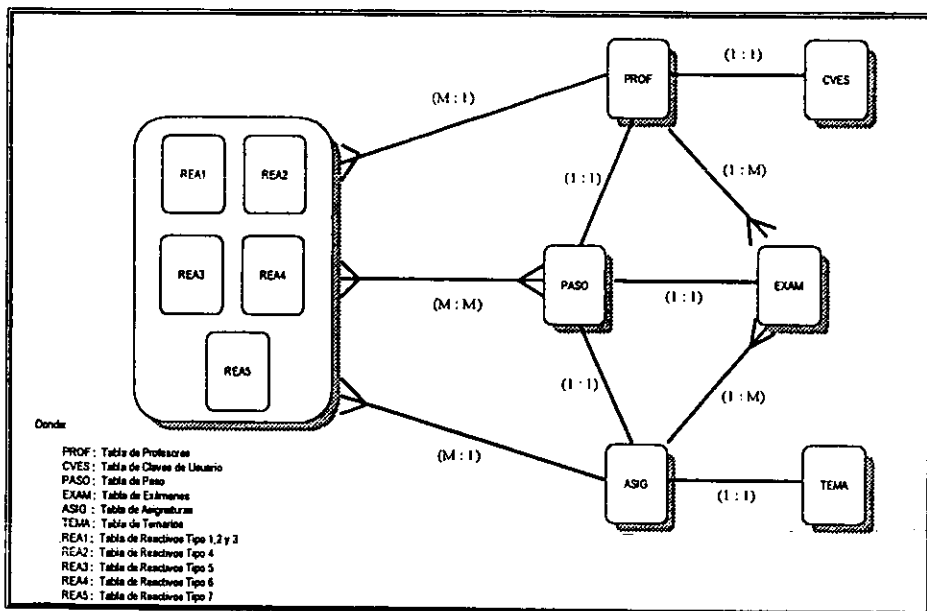
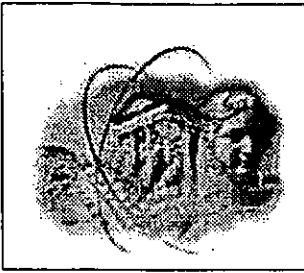


Figura 4.12 Diagrama Entidad - Relación. Base de Datos del SISE.



5. DESARROLLO DEL SOFTWARE

5.1 INTRODUCCION

Este capítulo conjunta una gran variedad de conocimientos que forman parte importante de la informática contemporánea que nos permitieron decidir (de una serie de comparaciones de distintos lenguajes) la selección del software más adecuado para el desarrollo del Sistema de Control y Automatización de Evaluaciones para el Cetus 152, 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 más apegado a sus requerimientos. Uno de los lenguajes visuales más representativos es Delphi.

Delphi es uno de los principales lenguajes de programación que admite el concepto de programación orientada a eventos; tiene 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. Se puede decir que es un estilo de programación adaptada a las interfaces gráficas del usuario.

Por esta y por muchas otras razones puede ser de gran utilidad en el desarrollo de este proyecto.

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 Institución que merece ser debidamente organizada con el fin de maximizar su valor.

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 realizado en Delphi versión 3.0 y el manejador de bases de datos en Access (resultado de un estudio que se realizó en este apartado), debido a que la programación en ese lenguaje es fácil, el código generado puede ser reutilizable proporcionando flexibilidad y un gran potencial.

En términos generales, en este capítulo se describirá el estudio que se realizó para seleccionar el software y el manejador de la base de datos más adecuados a nuestras necesidades. Describiendo en páginas posteriores el desarrollo del sistema.

5.2 COMPARACION ENTRE SOFTWARE'S

El objetivo de un sistema de base de datos es simplificar y facilitar el acceso a los datos. 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:

- * **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 fisico y el diseño lógico de datos.
- * Nivel alto en la manipulación de datos.

A continuación se detalla un resumen de los principales manejadores de bases de datos que se utilizan hoy en día así como una serie de elementos que nos permiten comparar y decidir la mejor opción de todos los productos que tiene mayor demanda en el mercado.

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 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.

FOXPRO

La interface de FoxPro incluye menús presentados junto con una ventanilla de comandos, 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 anexar otras pantallas a ella, lo cual ahorra tiempo una vez formulada una biblioteca de pantallas genéricas.

El Funcionamiento de FoxPro es magnífica, 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.

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.

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.

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 OLE¹, 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).

¹ OLEContainer (OLE es la abreviatura de Object Linking and Embedding, que significa vinculación e incrustación de objetos).

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. 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 5.1, Tabla 5.2, Tabla 5.3, Tabla 5.4, Tabla 5.5, Tabla 5.6, Tabla 5.7, Tabla 5.8, Tabla 5.9, Tabla 5.10, Tabla 5.11 y Tabla 5.12

PRODUCTO	COMPANÍA
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

Tiempo de procesamiento en Batch	C	F	I	P	A
bueno	X				
regular		X		X	X
malo			X		

Tabla 5.1. Evaluación de Tiempo de Procesamiento en Batch

Tiempo importación y generación de índices	C	F	I	P	A
bueno		X		X	
regular	X				X
malo			X		

Tabla 5.2. Evaluación de Tiempo Importación y generación de índices

Soporte de Núm Máximos de archivos abiertos	C	F	I	P	A
bueno			X		
regular	X	X		X	X
malo					

Tabla 5.3 Evaluación de Soporte de Núm Máximo de archivos abiertos

Facilidad de Uso	C	F	I	P	A
bueno		X		X	X
regular	X		X		
malo					

Tabla 5.4 Evaluación de Facilidad de Uso

Proporciona Seguridad	C	F	I	P	A
bueno		X	X	X	X
regular	X				
malo					

Tabla 5.5 Evaluación que Proporciona Seguridad

Integridad en los Datos	C	F	I	P	A
bueno	X			X	X
regular		X	X		
malo					

Tabla 5.6 Evaluación de la Integridad de los Datos

Presentación de Calidad en los Reportes y Formas	C	F	I	P	A
bueno					X
regular	X	X	X	X	
malo					

Tabla 5.7 Evaluación de la Presentación de Calidad en los Reportes y Formas

Facilidad de Programación	C	F	I	P	A
bueno					X
regular	X	X		X	
malo			X		

Tabla 5.8 Evaluación en la Facilidad de Programación

Compartición de los datos con otras bases de datos	C	F	I	P	A
bueno					X
regular	X	X	X	X	
malo					

Tabla 5.9 Evaluación en la Compartición de los datos con otras bases de datos

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

Tabla 5.10 Evaluación en la Diversidad en Tipos de Datos

Manejo de Gráficos para formulario	C	F	I	P	A
bueno					
regular					X
malo	X	X	X	X	

Tabla 5.11 Evaluación en el Manejo de Gráficos 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 5.12)

CARACTERÍSTICAS	C	F	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 5.12 Evaluación para la Puntuación final

PUNTUACIÓN DE LA COMPARACIÓN

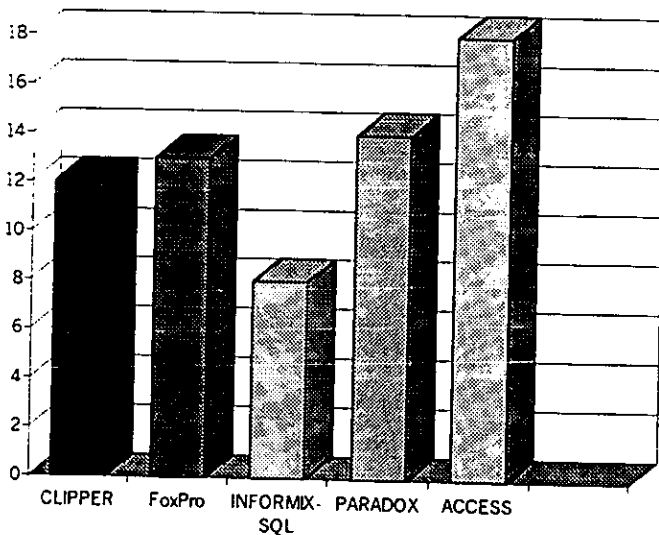


Tabla 5.13. Puntuación de la Comparación

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:

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

En la figura 5.14 se muestra la ventana de la base de Datos con sus respectivos elementos que la forman.

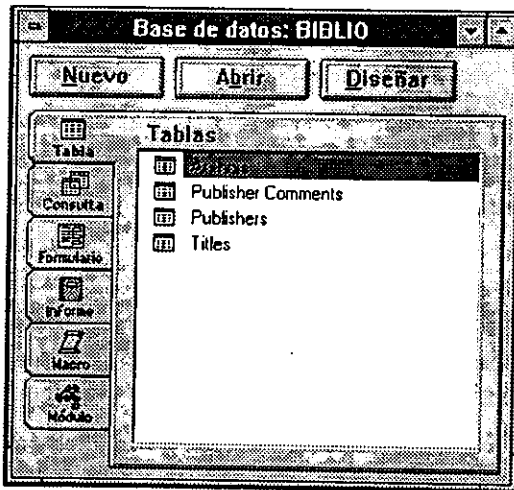


Figura 5.14 Muestra la ventana de Una Base de Datos de Access

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	CONCEPTO
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, la tabla que almacena registros de profesores tal vez se le quieran agregar campos como nombre, dirección y el número telefónico de los maestros, etc.

Cuando se declaran campos es necesario especificar:

- * Nombre del Campo.
- * Tipo de Datos.
- * 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 5.15 se muestra una tabla que diseñó Access.

Tabla: Publishers		
Nombre de 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	
Tamaño del campo	50
Formato	
Máscara de entrada	
Título	
Valor predeterminado	
Regla de validación	
Texto de validación	
Requerido	No
Permitir longitud cero	No
Indexado	No

El mismo número de caracteres que puede introducir en el campo. Lo

Figura 5.15 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.
Númérico	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
Sí/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.

5.3 ANÁLISIS Y SELECCIÓN DEL SOFTWARE PARA GENERAR EL SISTEMA

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 la 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". 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.

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 5.16

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. Pero tendremos que seguir trabajando con las herramientas disponibles, muchas de las cuales ya han comenzado su transición a la "Programación Visual".

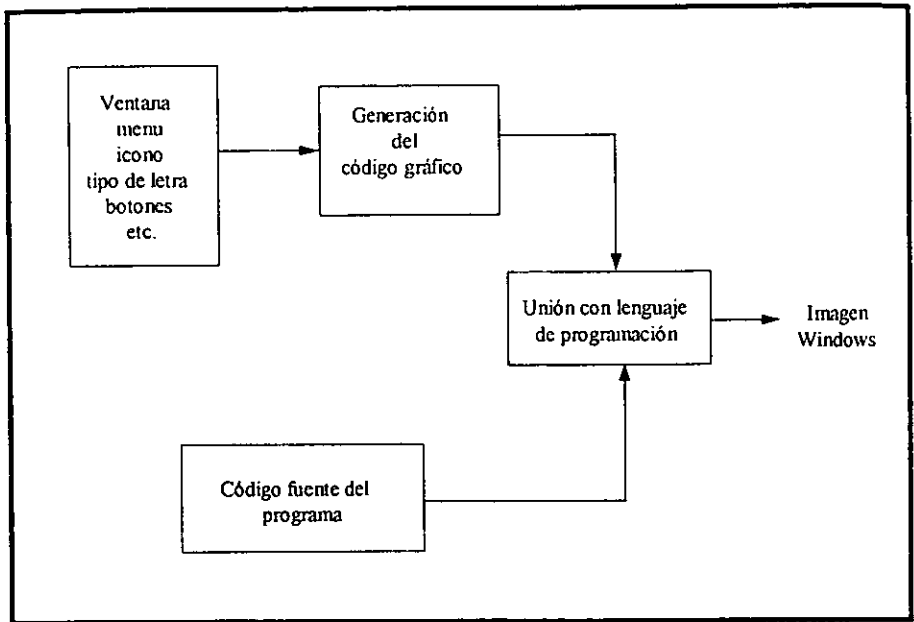


Figura 5.16 Proceso para realizar una aplicación con imagen Windows.

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 - 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.

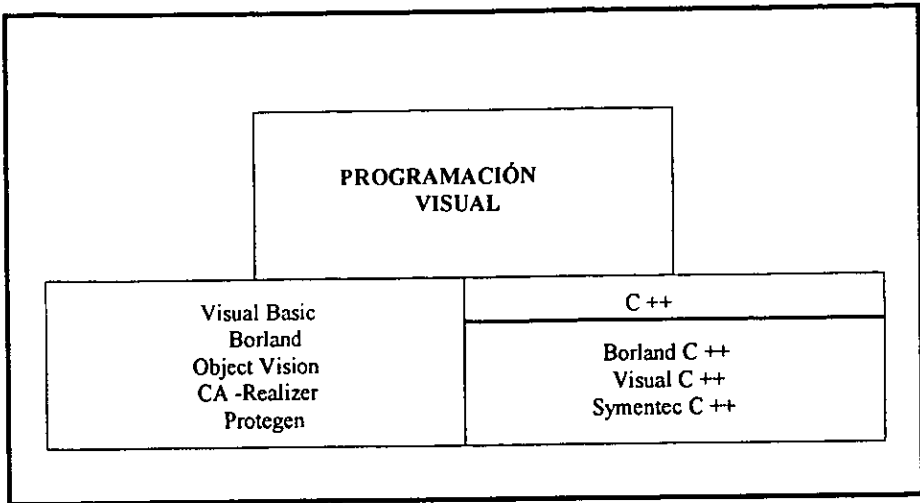


Figura 5.17 Paquetes y lenguajes que manejan programación "Visual".

Con la aparición de los últimos productos de desarrollo de aplicaciones bajo entornos gráficos es necesario dar 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.

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 GUIs² que se comercializan actualmente.

² *Graphical user interface*. Interface de salida que permite la visualización de imágenes, gráficas, etc.

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 o los nuevos sistemas. El segundo camino se simplifica al utilizar las mencionadas herramientas de "Programación Visual".

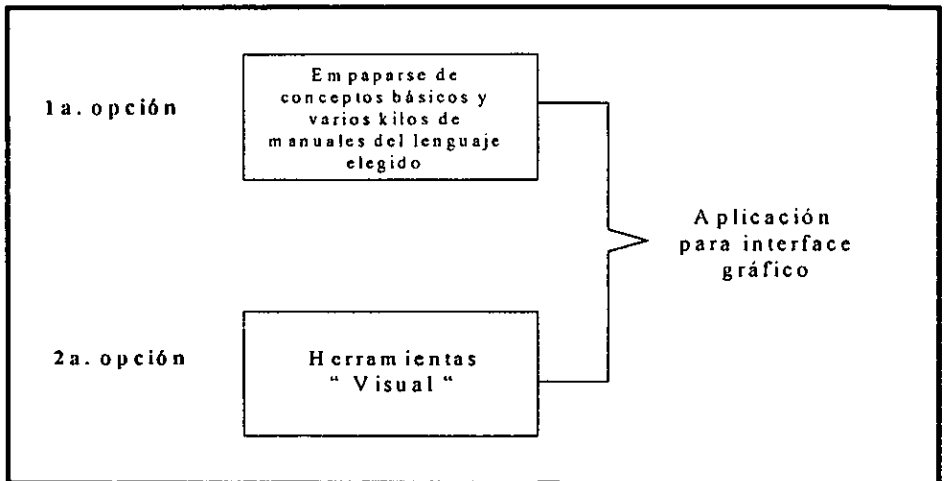


Figura 5.18 Desarrollo de una aplicación para una interface gráfica.

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. El poder, flexibilidad y velocidad de Visual Basic está ahora a la par con el lenguaje C.

Visual Basic es altamente interactivo, manejando un lenguaje de programación, con el cuál 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 para aplicaciones basadas en ventanas; trabaja con las bases de datos en una interacción; y Objetos Ligados y Empotrados (OLE).

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* (depurador).

Ventajas

- * Es un sistema productivo para crear soluciones en Windows.
- * Acceso a Bases de Datos.
- * Manejo de Multimedia.
- * 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.
- * 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 (*VCL, Biblioteca de componentes visuales*), 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.

Un programador medio necesita meses para tener una velocidad de programación para Windows usando el lenguaje C, pero con Delphi se pueden desarrollar aplicaciones para Windows con solo algunas horas de familiarizarse con el lenguaje tanto del paquete como de programación Pascal.

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.

Las características más sobresalientes de Delphi son las siguientes:

Arquitectura de la base de datos y conectividad. La arquitectura de la base de datos ha sido completamente reestructurada para ofrecer un enfoque multinivel, en lugar del enfoque tradicional cliente/servidor. Esto le permite crear una aplicación cliente muy ligera. Se proporciona soporte para el acceso a bases de datos Access (una característica muy solicitada), para ayudar en la migración de aplicaciones de Visual Basic a Delphi.

Aplicaciones Web. Se pueden crear aplicaciones Web, tanto cliente como servidor. Esto da a Delphi una ventaja clara en la batalla por el mercado de las herramientas de desarrollo intranet.

Paquetes. Ahora tiene la libertad de escoger entre incluir la biblioteca de tiempo de ejecución en el ejecutable, o dejarla como un DLL separada. Esta característica permite al desarrollador independiente incluir toda la información en el archivo .EXE para que sea más fácil su distribución y deja que los desarrolladores empresariales distribuyan una biblioteca estándar de tiempo de ejecución a todos los escritorios, liberando solo pequeños .EXE's

Mejoras al Editor y al Depurador. El nuevo editor tiene varias características nuevas, como la de completar automáticamente el código y un experto en parámetros del código.

Existen tres versiones diferentes de Delphi: Delphi Desktop, Delphi Developer y Delphi Cliente/Servidor (CS). Estas versiones de Delphi ofrecen diferentes niveles de conectividad con el mundo exterior. Delphi Desktop viene con la capacidad de conectarse con dBase y Paradox a través del dispositivo de base de datos de Borlan. La edición Developer contiene conectividad ODBC (para conectarse cualquier fuente de datos con un manejador ODBC), y la edición CS viene con vínculos SQL /SQL Links). Este producto le da manejadores de alta velocidad de 32 bits, para conectarse con servidores SQL de base de datos como Sybase y Oracle.

La edición CS se diseñó para competir con otros productos de desarrollo de aplicaciones cliente/servidor. Sus principales competidores son PowerBuilder y por supuesto Visual Basic Enterprise Edition. Estos productos típicamente se utilizan en un ambiente empresarial para desarrollar interfaces frontales basados en Windows para bases de datos basados en redes.

Otra excelente característica de Delphi CS es el generador visual de consultas (Visual Query Builder), es parecido a utilizar el Query o Access de Microsoft para realizar consultas de datos mediante arrastrar y soltar campos en una ventana y jugar a conectar los puntos.

La última particularidad en verdad notable de Delphi CS es ReportSmith es una herramienta de generación de reportes. En la edición CS de Delphi se tiene la edición SQL de ReportSmith, la cual puede conectarse en forma directa con bases de datos SQL, así como a bases locales de datos Paradox, dBase y otros formatos de bases de datos como Access.

Delphi CS fue creado para competir con otros constructores GUI de base de datos de proceso frontal para nivel corporativo. Proporciona las mismas (si no es que mejores) características que otros productos más caros.

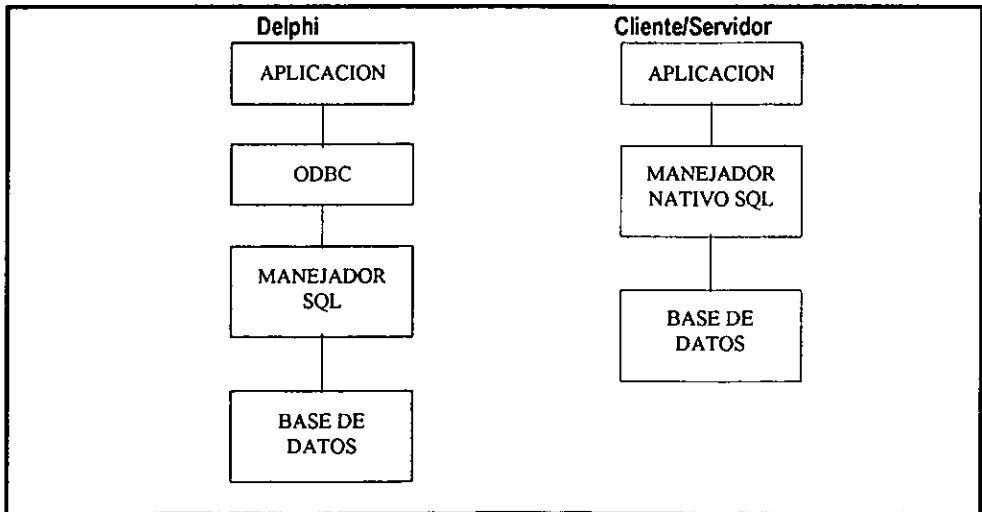


Figura. 5.19 Acceso a la base de datos nativas con Delphi CS

Este lenguaje esta disponible para programadores de MS-DOS y Windows. Delphi 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.

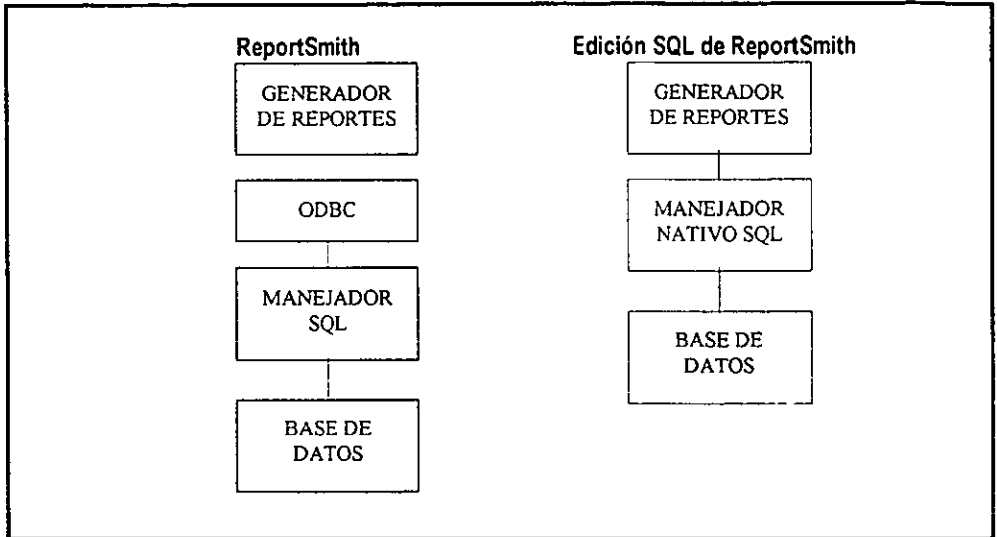


Figura. 5.20 Acceso a servidores nativos SQL con la edición SQL de Report Smith

Los requerimientos para su instalación se muestran en el siguiente recuadro:

DELPHI	
REQUERIMIENTOS:	Procesador 80486 ó superior 44MB de espacio disponible Mouse Windows 95, Windows NT 3.51 8MB de RAM (Preferible 16MB)
COMENTARIOS:	Delphi 3.0 es un lenguaje que apoya totalmente las propiedades de la programación orientada a objetos con un mínimo de programación y utiliza una comunicación más sencilla con otras aplicaciones de Windows, aparte de que se puede controlar varios tipos de bases de datos.

Figura 5.21 Ficha Técnica de Visual Basic.

Base de datos cliente/servidor que se pueden utilizar en Delphi

Al desarrollar aplicaciones Delphi probablemente se elegirá Paradox, Access o dBase, porque estas están fuertemente integradas a Delphi. Asimismo, pueden crearse fácilmente objetos de base de datos para dBase y Delphi, desde el Database Desktop de Delphi. La siguiente tabla muestra algunas estadísticas sobre las capacidades de estos manejadores de base de datos.

Tabla de comparación de tablas Paradox, dBase y Access

ATRIBUTO	PARADOX	dBASE	ACCESS
Máximos de registros	2,000 millones	1,000 millones	2,000 millones
Campos por tabla	255	1,024	1,024
Caracteres por campo	No aplicable	256	No aplicable
Bytes por registro	32,750	32,767	32,767

Como se puede observar las tres bases de datos son extremadamente poderosas y capaces de contener más datos de lo que la mayoría de las aplicaciones pueden llegar a necesitar. Paradox y Access ofrecen bastante más flexibilidad en la propia base de datos. Esto significa que las tablas pueden contener más información que sólo los datos sin procesar. Por ejemplo Access permite que la validación de datos ocurra al nivel de la base de datos, Paradox también tiene integrada la seguridad de contraseñas. Estas características son muy útiles sobre todo si la aplicación Delphi no será la única que acceda a los datos.

Seleccionar que modelo de base de datos es adecuado para una aplicación es a menudo una tarea difícil. Por suerte el motor de base de datos de Borlan es lo suficientemente flexible para cambiar de una base de datos a otra con poco esfuerzo. Si esto no proporciona suficiente flexibilidad, el motor de base de datos puede comunicarse con ODBC, que a su vez puede comunicarse con casi cualquier base de datos que existe en el mercado hoy en día.

Ventajas

- * Presenta programación estructurada y programación orientada a objetos.
- * Se podrá estar satisfecho al ver la primera aplicación desarrollada para Windows con un trabajo fácil y tan rápido como sea posible.
- * Requiere de una sintaxis precisa.
- * Su compilador realiza el programa ejecutable automáticamente.
- * Tiene edición multiarchivos.
- * Acceso a Bases de Datos.
- * Manejo de Multimedia.
- * Su curva de aprendizaje es corta.
- * Su editor de texto es amigable.
- * Delphi es un lenguaje muy poderoso.

- * Un usuario inexperto puede ser productivo con algunas semanas de utilización de Delphi.
- * Permite manipular otras aplicaciones para utilizarlas como componentes en aplicaciones propias (Ej. Word, Excell, Project, etc..), siempre y cuando dichas aplicaciones soporten OLE automation.
- * Una de las características más importantes incluidas son los controladores nativos, para bases de datos. Delphi soporta Microsoft Access y Foxpro, así como el acceso nativo a dBase. Esto significa que para el desarrollador tiene un acceso más rápido a estas fuentes de datos que si se utilizara controladores de *conectividad abierta de bases de datos* (ODBC, open database connectivity).
- * Por lo tanto Delphi es viable como herramienta para la migración de aplicaciones en Visual Basic a Delphi.
- * Interface al usuario amigable.
- * Su interprete es rápido en la creación de ejecutables.
- * Interprete poderoso que permite la detección de errores al momento de edición de programa.
- * Maneja aplicaciones Cliente-Servidor.
- * Puede ser utilizado como un servidor de OLE Automation.

Desventajas

- * El lenguaje esta poco difundido
- * Su limite de cadenas es de 255 caracteres, existen cadenas terminación cero que pueden ser de 65535 caracteres de extensión.
- * 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++

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. Borland es la pionera en ambientes bajo DOS y trabajar con 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.

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.

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 ha 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.

Ventajas.

- * 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.
- * Rapidez en ejecución de programas.
- * Manejo de variables locales y globales.
- * Total libertad para manejo de memoria y accesos a Sistema Operativo.

Desventajas.

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

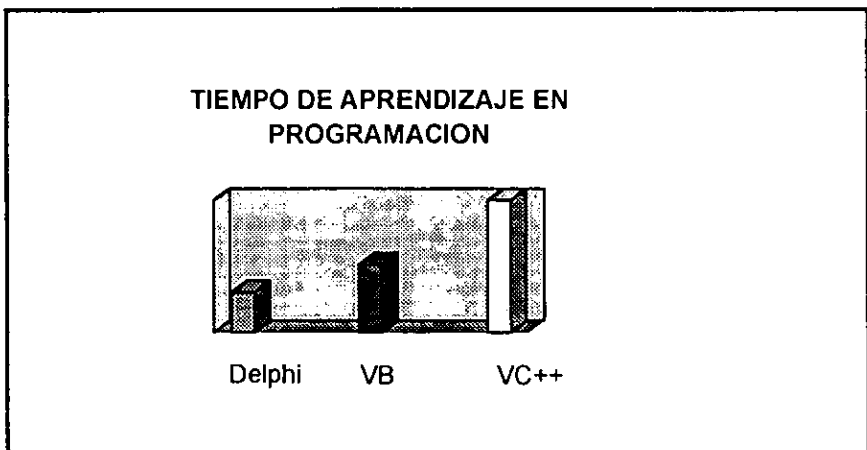


Figura 5.22 Gráfica comparativa de la curva de aprendizaje del software analizado.

FLEXIBILIDAD DEL SOFTWARE

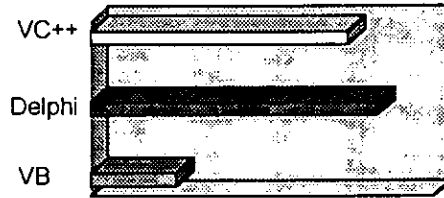


Figura 5.23 Gráfica que muestra comparativamente la flexibilidad del software analizado.

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 preconstruido <input checked="" type="checkbox"/>	<input checked="" 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 checked="" 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 type="checkbox"/>		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Si	<input type="checkbox"/>	No

Figura 5.24 Sumario de Características del Software.

Tomando como referencia los conceptos antes mencionados, se puede ya seleccionar el software que se utilizará para el desarrollo del Sistema de Control y Automatización de Evaluaciones para el CETIS 152. Este se elaborará en Delphi; debido a todas las características que observamos en la Figura 5.24 y las que se mencionarán a continuación:

- * Su curva de aprendizaje es baja (Figura 5.22.).
- * Su poder, flexibilidad y velocidad son adecuados (Figura 5.23).
- * 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*.
- * Se puede trabajar con la Base de Datos Access.

No hay duda que Delphi 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.

Seleccionado el lenguaje y el manejador de la base de datos, en páginas posteriores se describirá el desarrollo del sistema.

5.4 MANEJO DE DATOS

Antes de continuar, es necesario dar antes un panorama general de los usuarios que forman parte de la institución y los que podrán acceder al sistema.

Podrán tener acceso solo aquellos que pertenezcan al plantel (personal interno), como son los directivos, profesores, y secretarias. Se les asignará un número de clave con la cual podrán realizar ciertas actividades. Esto es con el fin de garantizar los atributos intrínsecos de la información, es decir, que se tenga integridad, confidencialidad y disponibilidad de los datos permitiendo:

- * Reducción de riesgos de pérdida, alteración o divulgación de la información.
- * Asegurar el uso de la información por personas autorizadas para ello.
- * Proveer información veraz y completa.
- * Asegurar la disponibilidad de la información permitiendo la continuidad en la operación.

Es por ello que no se debe de descartar la idea de crear niveles de seguridad en un sistema, en la siguiente tabla se especifican todas las claves ó password con los que cuenta el proyecto.(ver Figura5.25).

Teniendo como opciones las siguientes:

- * A altas
- * B bajas
- * C cambios
- * M modificaciones

ESTRUCTURA DE CONTROL Y AUTOMATIZACIÓN DE EVALUACIONES
NIVELES DE SEGURIDAD

CLAVE NIVEL	NOMBRE NIVEL	PRIVILEGIOS TABLAS																					
		CLAVE ACCESO		PROFESOR				ASIGNATURAS				TEMARIOS				REACTIVOS				EXAMENES			
		A	B	A	B	C	M	A	B	C	M	A	B	C	M	A	B	C	M				
0	ADMINISTRADOR	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			
1	COORDINADOR			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			
2	ESCRIBANO					X				X				X	X	X	X	X	X	X			
3	ESTUDIANTE							X				X			X					X			

Figura 5.25 Niveles de seguridad del Sistema

Como se puede observar en la tabla solo el administrador de la base de datos tiene acceso a todas las actividades y a todas las pantallas, él será el responsable de proteger y cuidar la información, de aplicar las medidas de seguridad especificadas por el propietario y garantizar su disponibilidad y acceso a las entidades facultadas de acuerdo con lo establecido por los propietarios.

También se encargarán de guardar la información que es propiedad de otros; en su mayoría los administradores son las entidades que proporcionan servicios de cómputo a los propietarios o a los usuarios, ya sean de la Institución o externos.

Funciones y responsabilidades:

- 1.-Deberán garantizar el envío y recepción de la información a usuarios autorizados.
- 2.-Será responsable del procesamiento y almacenamiento de la información, cuando aplique.
- 3.-En el caso de aplicaciones de mainframes, los Centros de Servicio de Información son los custodios.
- 4.-Para aplicaciones en micros, minis y redes locales o extendidas las responsabilidades las tendrá el propietario o el usuario, o en su caso, el administrador de la red.
- 5.-Deberá asesorar al propietario de la información respecto a las vulnerabilidad que puedan representar una amenaza para la información y sobre medidas específicas para proteger dicha información.

Usuario:

Se consideran usuarios de la información a todos los integrantes que forman parte del CETIS 152, que para el desempeño de sus actividades manejan información de la Institución. Son responsables de mantener su integridad, disponibilidad y confidencialidad conforme a lo dispuesto por el propietario y el administrador. Tendrán autoridad específica sobre la información, limitada por el

propietario, para verla, modificarla, añadirle o borrarla. Los usuarios son los que usarán la información para el desempeño de sus actividades.

Funciones y responsabilidades:

- 1.-Ellos deberán seguir las reglas especificadas por el propietario, y en su caso, por el administrador.
- 2.-El usuario de la información será cualquier funcionario autorizado por el propietario de la misma, para leerla, capturarla o actualizarla.
- 3.-El usuario de la información será responsable de:
 - * Utilizar la información únicamente para el propósito autorizado por el propietario.
 - * Cumplir con todos los controles establecidos por el propietario y el administrador así como con las normas de clasificación de la información.
 - * Asegurar que la información confidencial no será divulgada a persona alguna sin previa autorización del propietario.
 - * Acatar los lineamientos establecidos en el esquema de acceso y facultades.
 - * Notificar al propietario de la información de cualquier violación o intento de violación de las políticas, prácticas y procedimientos de seguridad informática.

Para obtener la información proporcionada por los usuarios, nos auxiliamos de pantallas que son la parte frontal del sistema, en ellas podremos capturar los datos necesarios, para que posteriormente se realicen una serie de validaciones para su almacenamiento en la base de datos.

VALIDACIÓN POR MÁSCARAS.

Este tipo de validación se realiza en Delphi, con el fin de 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, este tipo de validación requiere programación, que en nuestro caso se realizó en Delphi. Los procedimientos de validación que se realizaron son utilizados en los módulos donde se lleva a cabo la entrada de captura de la información la cual se obtiene en caracteres de presentación (caja ó en combobox) que especifican donde deben introducirse los datos, algunos datos deberán cumplir con ciertas características como son, longitud, fechas, teléfonos, aceptara en algunos casos cierto tipo de dato que se debe de escoger de una lista que se de previamente, etc., es decir, sólo considerara como información válida aquella que cumpla con las especificaciones de cada campo.

La siguiente tabla especifica la máscara de cada campo de la pantalla de PROFESOR³ (ver Figura 5.26).

³ Las pantallas restantes no se detallan por lo extenso del sistema, pero se realizaron de forma similar

CAMPO EN LA PANTALLA	ENTIDAD EN LA BASE DE DATOS	MASCARA
Clave	Claveprofe	00;1;
Apellido Pat.	Apepprofe	*
Apellido Mat	Apemprofe	*
Nombre	Nombprofe	*
Cargo	Puestprofe	*
Profesión	Perfiprofe	*
Academia	Areaprofe	*
Status	Statusprofe	*
Nivel Usuario	Nivelprofe	*
Calle	Calleprofe	*
Colonia	Colonprofe	*
Cod. Postal	Cpostprofe	000000;1;
Ciudad	Ciudadprofe	*
Estado Civil	Edociprofe	*4
Teléfono	Teleprofe	(999)000-0000;1;

FIGURA 5.26 Validación de Datos

5.5 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 (DBMS⁵), los tres componentes principales de un sistema de Base de Datos son: el *hardware*, el *software* DBMS y los datos por manejar.

⁴ *. No existe máscara para introducir datos en estos campos ya que son de tipo texto y no existe un formato especial para guardar la información.

⁵ *Database Management System*. Es un sistema de software que controla y manipula los datos, para eliminar redundancia y asegurarse de la integridad, consistencia y disponibilidad de los datos, entre otras características.

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 5.27 muestra la Base de Datos como un recipiente de datos a ser compartidos por varios programas.

El proceso de diseño involucra los tres procesos siguientes:

- * Agrupamiento de los datos globales de una Base de Datos en uno o más registros de tipo conceptual (Entidades).
- * Definición de las relaciones conectando entidades relacionadas en un modelo conceptual de Base de Datos.

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.

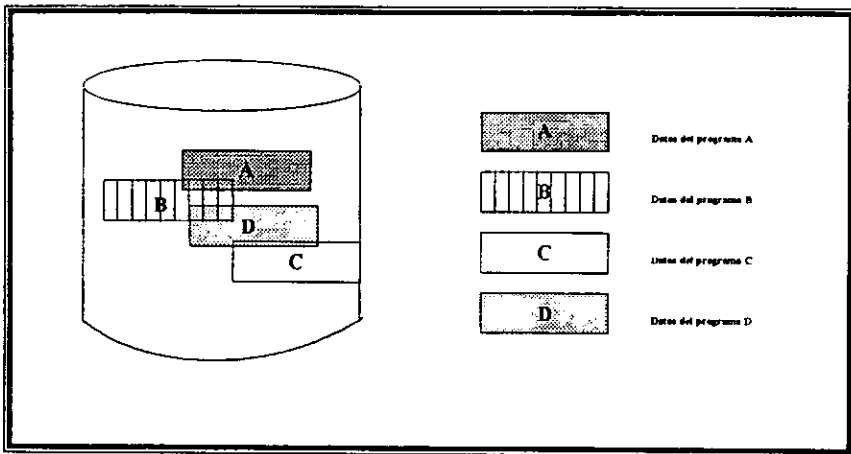


Figura 5.27 Base de Datos

DIAGRAMA DE BLOQUES PARA DESARROLLO DE LOS MÓDULOS PRINCIPALES

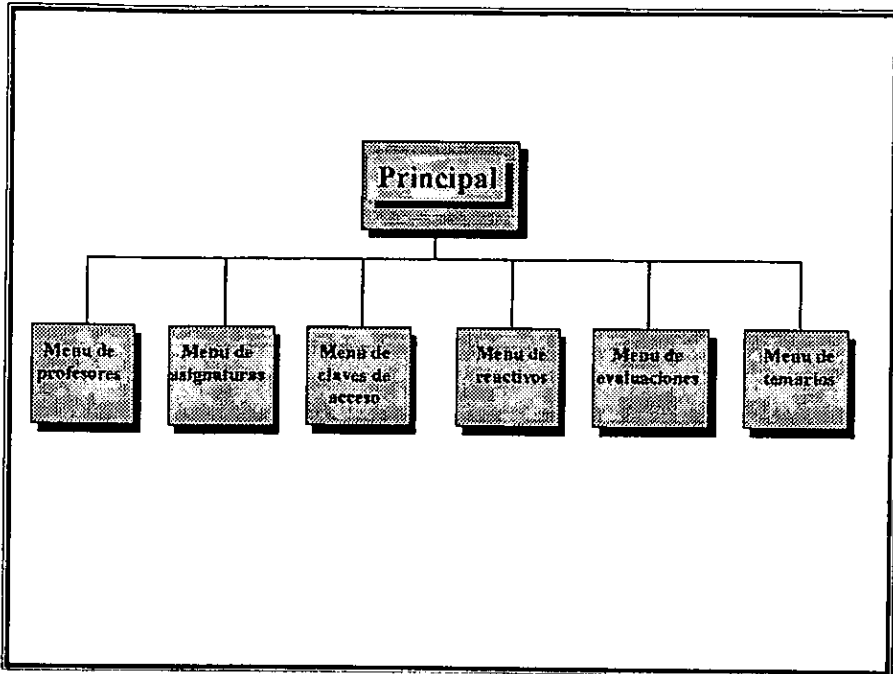


Figura 5.28 Módulo Principal

El diagrama que se muestra en la Figura 5.28 contiene todos los módulos disponibles del sistema. los cuáles son:

Menu de profesores

- Altas
- Bajas
- Cambios
- Consultas.

Menu de Asignaturas

- Altas
- Bajas
- Cambios
- Consultas.

Menu de Claves de Acceso

- Altas
- Bajas
- Cambios
- Consultas.

Menu de Consultas

- Altas
- Bajas
- Cambios
- Consultas.

Menu de Reactivos

- Altas
- Bajas
- Cambios
- Consultas.

Menu de Evaluaciones

- Altas
- Bajas
- Cambios
- Consultas.

Menu de Temarios

- Altas
- Bajas
- Cambios
- Consultas.

Menu de Reportes

- Altas
- Bajas
- Cambios
- Consultas.

La parte de captura es una de las partes importantes del sistema porque es donde se registran todos los datos que requiere el usuario dependiendo de la información que necesite en el momento, en base a campos que se definieron previamente en el sistema⁶. Por ejemplo en la pantalla de altas de profesores los datos que se requieren son los siguientes:

⁶ sólo se describirá una pantalla por lo extenso del sistema, las restantes son similares pero si desea información más detallada de cada pantalla ver el manual del usuario

Datos Personales:

Clave
Apellido paterno
Apellido materno
Nombre
Cargo
Profesión
Academia
Status
Nivel usuario

Domicilio particular:

Calle
Colonia
Codigo postal
Estado civil
Ciudad
Teléfono

Estos campos son utilizados en la pantalla de profesores que comprenden la parte 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.

Bajas.

En este submódulo se realiza un proceso de depuración manual en la base de datos. El proceso de depuración se lleva a cabo cuando el usuario lo requiera en el momento.

Cambios.

Existe un submódulo de cambios dentro de cada módulo como una herramienta al usuario 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.

Consulta

Las consultas permiten al encargado del sistema visualizar agrupando e identificando a un conjunto de datos con características similares, es decir, es una relación de uno a muchos. 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., esto se realiza por medio de rutinas para la selección de registros de características particulares o por selección de conjuntos.

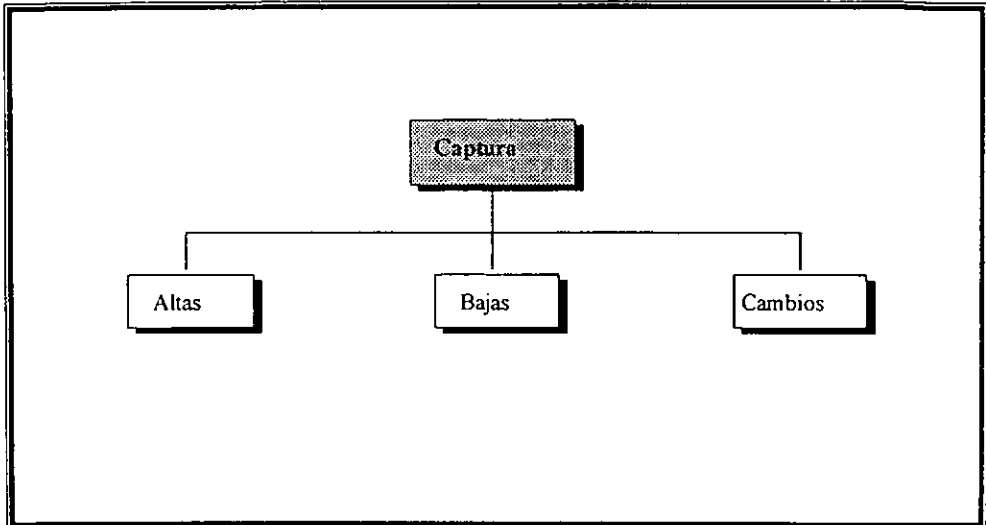


Figura 5.29 Módulo de Captura

INSERTAR ARCHIVOS

Dentro del proceso de captura existe una parte del módulo de alta reactivo que pueden insertarse gráficas o imágenes creadas bajo cualquier aplicación que trabaje en ambiente Windows como son; Power Point, Word, Excel, PaintBrush, Corel Draw, etc., el sistema podrá insertar solo aquellas imágenes que estén con formatos: mapa de bits (bmp), Metarchivo de windows (wmf), iconos(ico)⁷.

El usuario puede realizar sus diseños de la forma que más le convenga; utilizando diferentes tipos de gráficas, imágenes, colores, etc. pueden ir desde una gráfica sencilla, hasta poder pegar una imagen escaneada, podrán ser editadas al gusto del cliente, dependiendo del poder de la aplicación con la que este trabajando. Una vez creada o modificada la imagen esta podrá ser incorporada en el sistema, solo bastará seleccionar el botón de insertar una imagen, y automáticamente se realizará el pegado de la gráfica

Esto se puede llevar acabo por medio del componente Image, se pueden insertar gráficos en los formularios que se encuentren en los formatos que antes se describieron. Para ello se inserta un componente de este tipo en el formulario e indique el archivo de imagen a cargar a través del editor de imágenes que puede abrirse mediante la propiedad Picture. Además, puede indicarse también si el tamaño del componente Image debe adaptarse automáticamente al tamaño del archivo de imagen

⁷El pegado de una imagen se realiza en la alta de reactivos de Localización/Identificación de problemas y en Solución de problemas

(establecer propiedad Autosize a True) o si la imagen debe adaptarse al tamaño del componente Image (propiedad Stretched)

AYUDA.

El sistema proporciona información sobre el manejo y uso adecuado del sistema, en cada una de las pantallas de cada modulo, apoyándose en barras de estado la cual desplegará información del contenido de la ayuda con que cuenta el sistema de todos los campos que contiene la pantalla, solo bastará con posicionar el mouse en cada uno de los elementos de dichas pantallas, con el fin de facilitar y asesorar a los usuarios que lo requieran.

5.6 DIAGRAMA DE FLUJO DE DATOS

En las Figuras 5.30 y 5.31 se muestra el esquema general del funcionamiento del sistema.

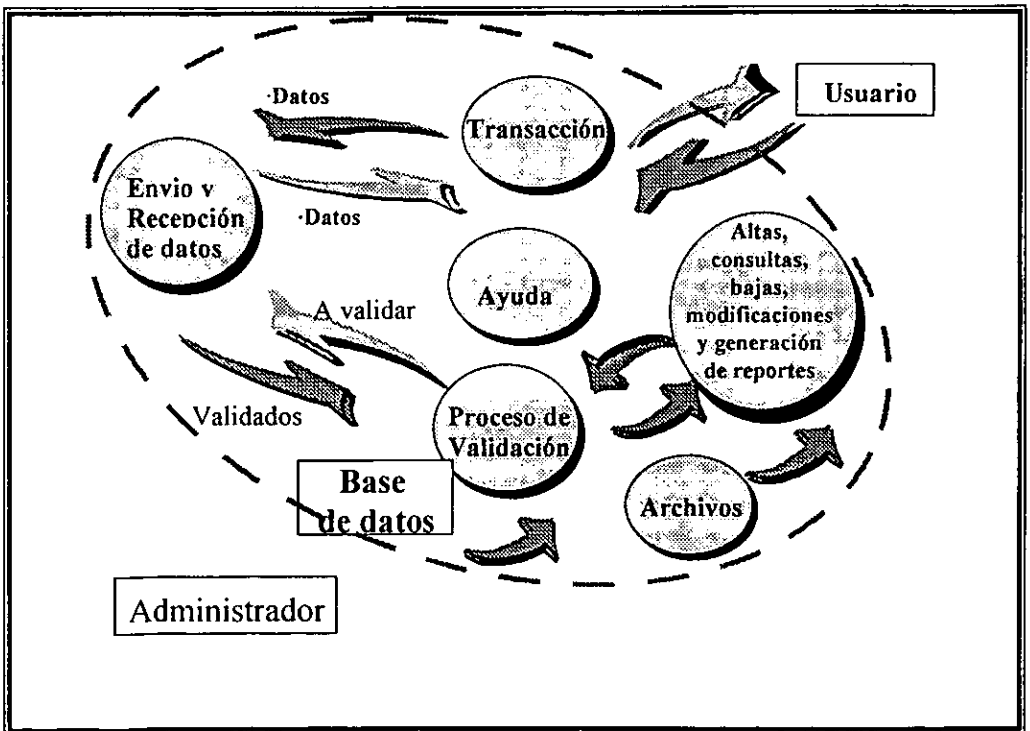


Figura 5.30 Diagrama de Flujo del Sistema

- * *Usuario*: Es el encargado de dar la información solicitada en el momento que se requiera en el sistema.
- * *Transacción*. Es el intercambio de información necesaria para que exista un diálogo⁸ entre el usuario y el sistema (hombre-máquina).
- * *Envío y recepción de datos*. Las interacciones que se llevan a cabo en aquellas máscaras que tiene la característica de tener un sin fin de opciones tiene la finalidad de enviar la información al módulo que le corresponda la ayuda de máscaras se puede definir que módulo debe de actuar, es decir, de máscaras las cuales tiene la característica de tener una serie de opciones, dependiendo de la selección que se haga se estable que módulo debe de atenderlo, es decir, si se selecciona , profesores, temarios, etc, o las altas, bajas, etc. de cada módulo antes descrito.
- * *Proceso de validación*. La validación de los datos se lleva a cabo por procesos, estos son una serie de subrutinas que tiene ciertas restricciones establecidas, para que la información cumpla con determinadas características y pueda almacenarse en la base de datos para su utilización posterior.
- * *Altas, consultas, bajas, modificaciones y generación de reportes*. Son todas la opciones con las que se pueden trabajar en el sistema, dependiendo del password con el que entren a la herramienta.
- * *Archivos*. . En las pantallas de alta reactivo de localización/identificación y de Solución de problemas se pueden insertar archivos de tipo bmp, ico, wmf, solo bastará seleccionar el botón de insertar imágenes, los archivos es la parte importante del reactivo ya que es la forma representativa del problema que se planteo.
- * *Ayuda*. En este módulo nos proporciona información sobre el manejo y uso adecuado del sistema.
- * *Base de datos*. Es la encargada de almacenar la información no redundante que se encuentra agrupada o estructurada y que es compartida por varios usuarios y/o aplicaciones.
- * *Administrador* . Es el responsable de proteger y cuidar la información, de aplicar las medidas de seguridad especificadas por el propietario y garantizar su disponibilidad y acceso a las entidades facultadas y de dar mantenimiento a la base de datos.

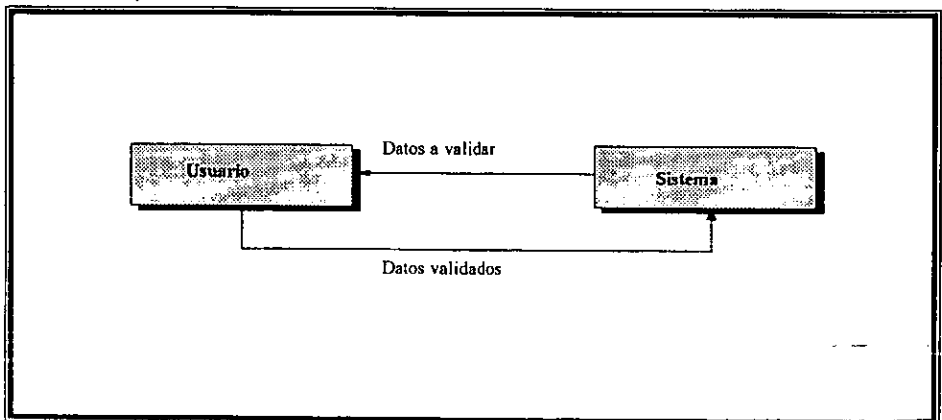


Figura 5.31 Diagrama de Flujo de los Datos

⁸ diálogo no verbal

5.7 DESARROLLO DE LA BASE DE DATOS

Como se determinó previamente la base de datos se desarrollo en Microsoft Access ver 4.0, con el fin de crear un sistema que permita dar Altas, Bajas, Cambios y Consultas de datos en forma amigable y dándole mayor flexibilidad al sistema.

Con la información obtenida en el análisis, se obtuvo que el sistema esta formado por once 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 los datos requeridos, sobre las cuales se va a trabajar.

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

A) Creación de la Base de Datos en Microsoft Access

Primeramente al entrar a Access aparece un menú como el que se muestra en la Figura 5.32 seleccionamos la opción de Archivos y elegimos crear una nueva base de datos.

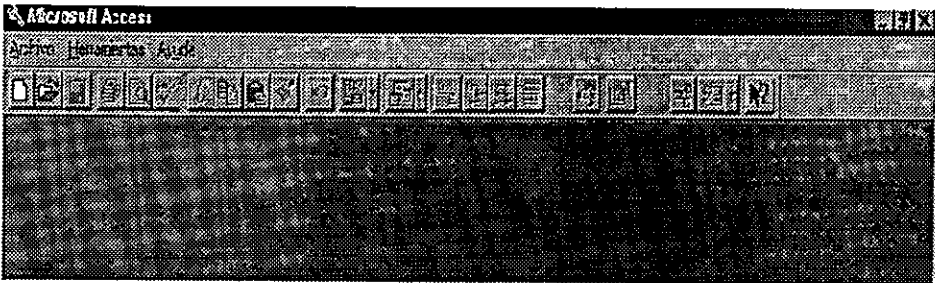


FIG 5.32 Menú inicial de Microsoft Access

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 5.33)

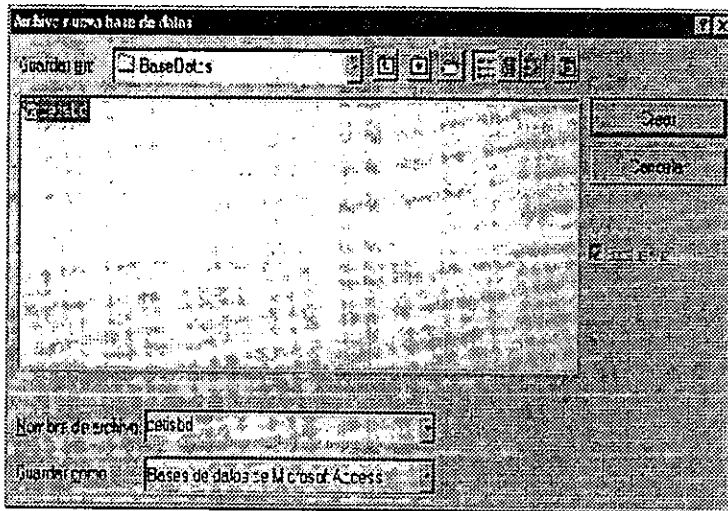


Figura 5.33 Generación de una nueva Base de Datos

En la Figura 5.34 se muestra la ventana de la Base de Datos, con los respectivos elementos que la forman.

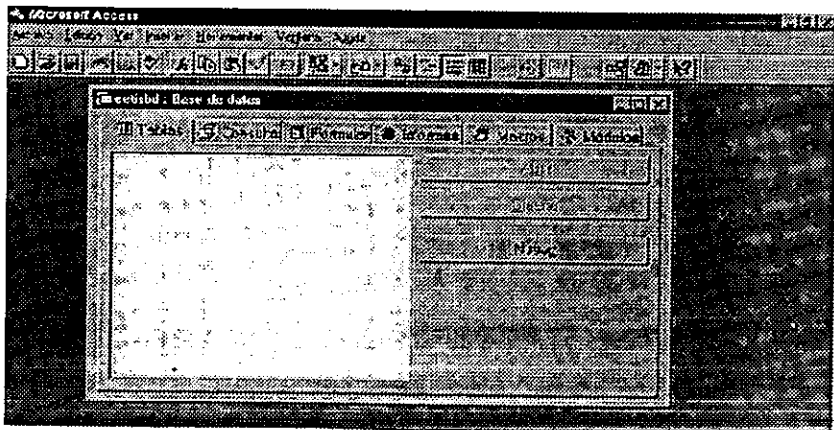


Figura 5.34 Ventana de Base de Datos

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:



para generar un nuevo objeto de la opción seleccionada

para ver la información de un objeto


para editar el objeto

B) Creación de la Tabla

Para crear la tabla, seleccionamos la opción de tablas y damos clic en la opción de nueva, esta opción, como las demás, nos permite elegir si deseamos que el asistente nos auxilie, para crear el objeto con cierta estructura o no.

En la parte superior se tiene la sección de declaración de campos que esta compuesta por tres columnas, la primera para darle un nombre al campo, la segunda para darle el tipo de datos que deseamos y la tercera para dar el descriptivo de lo que se va a introducir en el campo.

En la parte inferior tenemos la sección de propiedades de los campos que nos permite adecuar las propiedades a nuestras necesidades.

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 5.35 muestra como creamos la tabla Profesores que forma parte de nuestro sistema. En esta tabla se muestran los campos que la forman con sus respectivas características.

Nombre del campo	Tipo de datos	Descripción
CANPROFE	Número	ES LA CLAVE CON LA QUE SE IDENTIFICA AL PROFESOR, A CUAL SE LE ATRIBUYE EL NOMBRE DE PROFESOR.
NOMPROFE	Texto	ES EL NOMBRE PATERNO DEL PROFESOR.
AETNPROFE	Texto	ES EL APELLIDO PATERNO DEL PROFESOR.
CALEPROFE	Texto	ES LA CALLE Y NUMERO DE DOMICILIO DEL PROFESOR.
DOMPROFE	Texto	ES LA CIUDAD DEL DOMICILIO DEL PROFESOR.
CIENPROFE	Texto	ES LA CATEGORIA DEL DOMICILIO DEL PROFESOR.
COSTPROFE	Número	ES EL COSTO HOSITAL DEL PROFESOR.
TELEPROFE	Texto	ES EL NUMERO TELEFONICO DEL PROFESOR.
ESTPROFE	Texto	ES EL ESTADO CIVIL DEL PROFESOR.
PERPROFE	Texto	ES LA FORMA EN QUE HA REALIZADO EL PROFESOR.
STATPROFE	Texto	ES EL STATUS (ACTIVO / INACTIVO) QUE TIENE EL PROFESOR.
PUESTPROFE	Texto	ES EL PUESTO QUE OCUPA EL PROFESOR.

Propiedades del campo

Propiedad	Valor
Nombre del campo	TELEPROFE
Nombre de tabla	PROFESOR
Máscara de entrada	
Texto	
Validación de entrada	
Nombre de validación	
Validación	Si
Permitir la edición	No
Editable	No

El tipo de datos de este campo es texto y no numérico, aquí se le puede especificar una máscara de entrada y realizar la validación conveniente, pero este tipo de validación se realiza desde Delphi para que no acepte otros tipos de datos que no sean los requeridos.

FIGURA 5.35 Tabla de Profesor 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 nombre profesor, el cual, para fines prácticos en nuestra base de datos se llama TELEPROFE, tiene en la columna de tipo de dato texto y no numérico, aquí se le puede especificar una máscara de entrada y realizar la validación conveniente, pero este tipo de validación se realiza desde Delphi para que no acepte otros tipos de datos que no sean los requeridos.

Una vez introducidos los campos de la tabla y validados todos los diferentes tipos de datos, se procede a cerrar la tabla con el nombre de PROFESOR.

Una vez explicado la manera en como se creó la tabla anterior se procedió a realizar de manera similar las otras tablas que conforman a la base de datos, en la Figura 5.36 se muestra el nombre de todas las tablas creadas.

Nombre	Propiedad	Modificado	Creado
ASIGNACIONES		26/03/98 6:23:57 AM	27/03/98 11:53:59 AM
ASIGNATURAS		26/03/98 6:23:57 AM	11/04/98 11:44:54 PM
CLAYES		23/05/98 6:23:58 PM	26/04/98 9:23:55 PM
EXAMENES		6/02/98 12:27:03 PM	11/04/98 11:02:14 PM
REACTIVOS		29/03/98 6:23:56 AM	11/04/98 10:10:26
REACTIVOS1		26/03/98 6:23:56 AM	26/03/98 2:52:45 PM
REACTIVOS2		6/06/98 9:27:26 PM	6/06/98 2:35:06 PM
REACTIVOS3		6/06/98 9:27:26 PM	6/06/98 2:35:06 PM
REACTIVOS4		6/06/98 9:27:26 PM	6/06/98 9:30:49 PM
REACTIVOS5		6/06/98 9:27:26 PM	6/06/98 11:27:07 PM
TARPOS1		6/06/98 2:47:21 PM	6/06/98 2:46:19 PM
TEMPORIOS		03/06/98 11:02:14 PM	26/04/98 9:23:55 AM

Figura 5.36 Pantalla que muestra todas las tablas de la Base de Datos

5.8 DISEÑO Y CONSTRUCCIÓN DEL SOFTWARE.

Como se menciona anteriormente el software del Sistema de control y automatización de evaluaciones, se desarrolló en Delphi v. 3.0. y es precisamente en este tema donde se realizará y explicará el procedimiento del desarrollo del mismo.

Para crear una aplicación en Delphi 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 la **Biblioteca de Componentes Visuales (VCL)**. Cada objeto posee su propio conjunto de sucesos, propiedades y métodos. Para diseñar un objeto seleccione la herramienta deseada del **VCL**. 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.

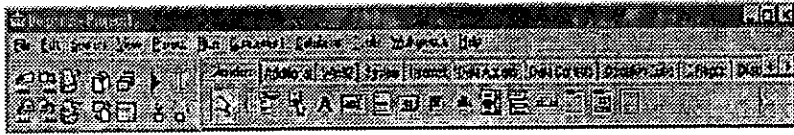


Figura 5.37 Biblioteca de Componentes visuales VCL

Las propiedades de cada objeto visual se pueden observar en la ventana del Inspector de Objetos, con su ayuda se realiza la conexión entre el diseño gráfico de aplicaciones Delphi en la pantalla y la funcionalidad con que se dota a los componentes y que se tiene que programar en el código fuente .

En la primera ficha del inspector de objetos, *Properties*, pueden determinarse todas las características esenciales de cada uno de los componentes (por ejemplo, tamaño, color, fuente, y comportamiento).

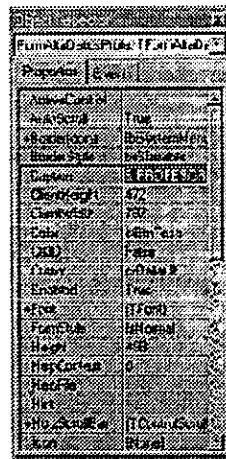


Figura 5.38 Inspector de Objetos

En la segunda ficha, *Events*, a la que se accede pulsando con el ratón la ficha de la parte superior del inspector de objetos , se determina como debe de reaccionar el componente que se está editando en ese momento a determinadas acciones del usuario del programa.

- 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.

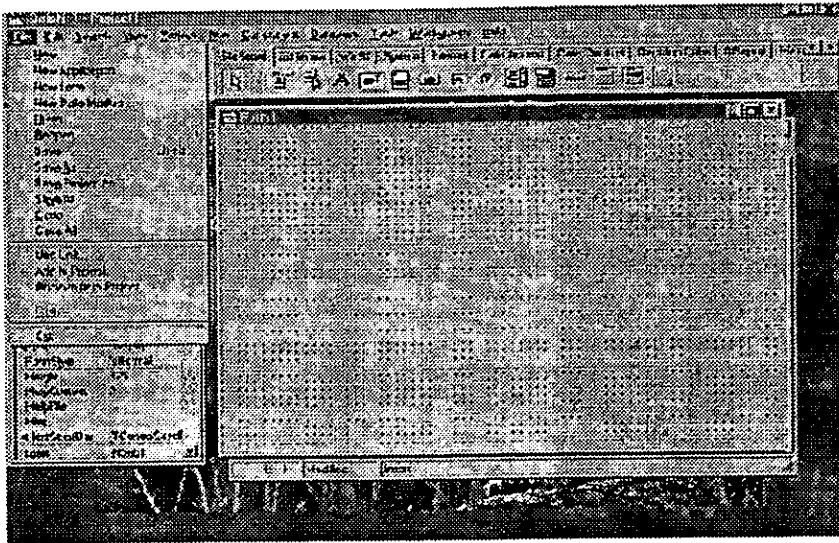


Figura 5.39 Entorno de Delphi

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 todos los colores posibles que soporte cada aplicación para el diseño de las imágenes.

De lo antes mencionado, se procede a la implementación de las pantallas:

Pantalla de Presentación

Para el desarrollo de esta pantalla, se realizaron los siguientes pasos:

1. Abrir Delphi 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 F11 para desplegar las propiedades de la forma que se muestran en una ventana llamada Object Inspector donde se definieron las siguientes propiedades:

Cursor=CrDefault
 Position= PoDesigned
 Color =CITeal
 Caption = "SISTEMA DE CONTROL Y AUTOMATIZACION DE EVALUACIONES"
 Name = FormCaratula
 Height = 508
 Width = 808

4. Para crear el escudo de la Universidad Nacional Autónoma de México se escoje del VCL la opción Additional y el icono de Image, se debe especificar en que ruta esta la imagen para que la pueda insertar en la pantalla, ya una vez definido esto dar Ok.

Picture =LoadPicture("C:\Delphi3\Images\iconns\escudo.ico")

5. Para crear la Barra de Menú de la pantalla principal del Sistema seleccione del VCL la opción Standard después MainMenu, en este momento se despliega en pantalla una ventana flotante, en la cual especificamos las opciones siguientes:

Caption = &Categorías
 Name = Categorías1

Para poder formar hijos en este Menú es necesario hacer un clic en la descripción que se acaba de realizar, automáticamente se crea un cuadro de bajo de este, se selecciona F11 y se coloca el cursor para definir las propiedades de *Caption*.y *Name* para cada Menú y cada opción deseada quedando de la siguiente manera:

CAPTION	NAME	NIVEL
Categorías	Categorías1	1
Profesor	Profesor1	2
Asignaturas	Asignaturas1	2
Reactivos	Reactivos1	2
Evaluaciones	Evaluaciones1	2
Salir	Salir1	2
Altas	Altas1	3
Bajas	Bajas1	3
Cambios	Cambios1	3
Consultas	Consultas1	3
Edit	Edit1	1
Corta	Corta1	2
Copia	Copia1	2
Borra	Borra1	2
Pega	Pega1	2
Catálogos	Catalogos1	1
Temarios	Temarios1	2

Tabla 5.40

6. Para incrustar los botones de la Pantalla principal se selecciona un panel, y se colocan encima de este los botones ajustando sus propiedades como:

Propiedades para el panel

Caption =
Name = Panel3
Color = CISilver
Height = 41
Width=737

Propiedades para los botones

Caption = &Comenzar
Name = BitComenzar
Color = CISilver
Height = 25
Width=105

Caption = &Salir
Name = BitSalir
Color = CISilver
Height = 25
Width=105

Este proceso se repite para cada botón de cada una de las pantallas del sistema.

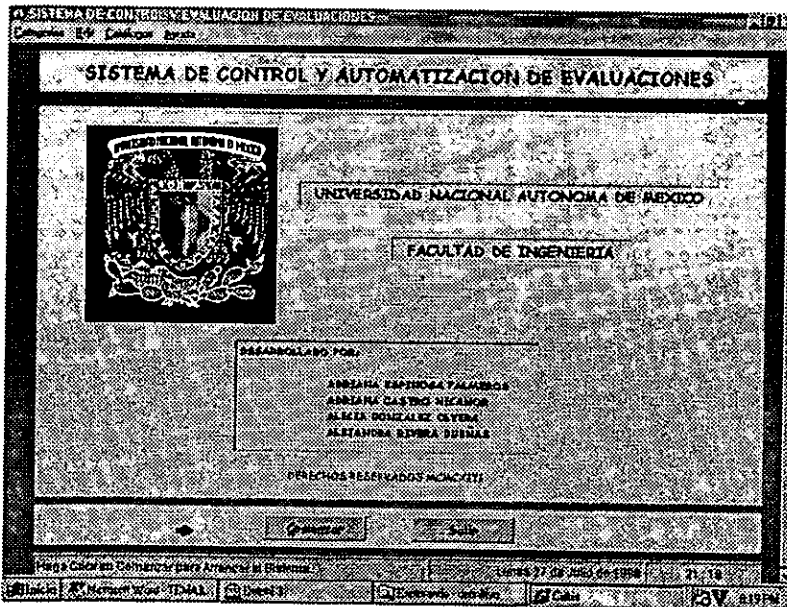


Figura 5.41 Presentación del proyecto

Pantalla de claves de acceso

Al seleccionar el botón de comenzar en la pantalla de presentación aparece otra ventana llamada claves de acceso en la cual debemos de teclear la clave de usuario y el password, para poder ingresar al sistema.

El desarrollo de esta pantalla se realizo de la siguiente manera:

1. Seleccionar una nueva forma, describiendo las propiedades similares a la que se explico anteriormente.
2. Seleccionar un panel del VCL en la pestaña que indica Standard, dimensionarlo automáticamente con el arrastre del mouse y definir las siguientes propiedades:

Caption = CLAVES DE ACCESO DEL SISTEMA
Name = Panel2
Color = ClMenu
Cursor = CrDefault
Visible = True

3. Para colocar títulos se selecciona y arrastra una Caja de Texto y se define sus propiedades:

Font = Arial
Color = Azul marino
Tipo = Negrita cursiva
Tamaño = 12
Cursor = CrDefault
Color = ClMenu
Visible = True

Para realizar los botones de los módulos con los que cuenta el sistema, se selecciona del VCL: en la pestaña de Additional dos botones definiéndolos con las siguientes propiedades:

Visible = True
Caption = Conectar
Caption = Cancelar
Height = 61
Width : 73

Cursor = CrDefault
 Color = ClMenu
 Visible = True

4. Para realizar los botones de los módulos con los que cuenta el sistema, se selecciona del VCL: en la pestaña de Additional un boton definiendolo con las siguientes propiedades:

Visible = True
 Glyph = LoadPicture("C:\Delphi3\Images\iconns*nombre del icono deseado*.ico")

Height = 61
 Width = 73

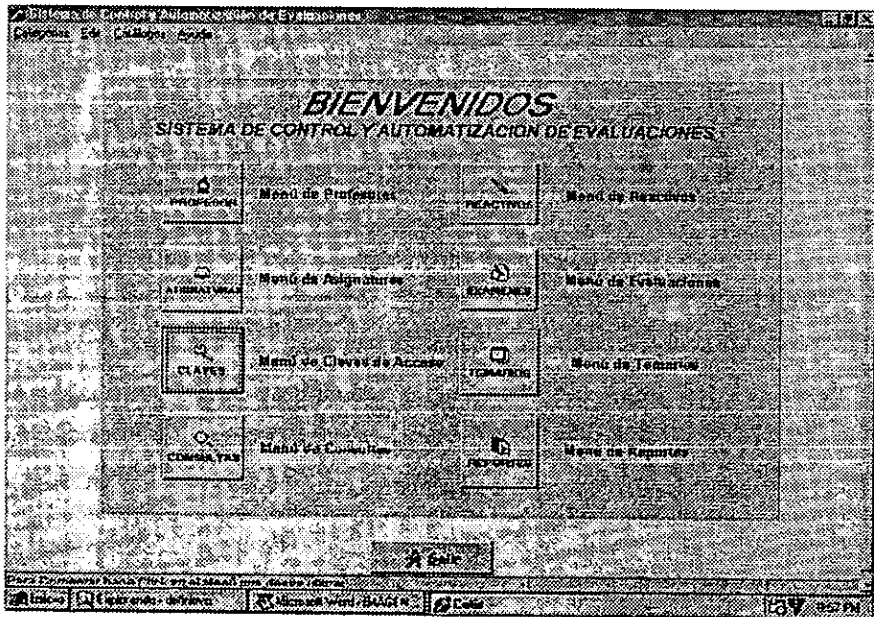


Figura 5.43. Pantalla del menú principal

Menú de Opciones

Esta pantalla aparecerá cuando se selecciono unos de los módulos del sistema.

1. Seleccione una nueva forma cuyas propiedades serán:

Caption = Menu de opciones
 Name = FormMenuOpciones
 Height = 229
 Width = 278

- Para definir las opciones que contiene esta forma seleccione y arrastre un RadioGroup dimensionelo con el arrastre del mouse según sus necesidades definiendo como sus propiedades las siguientes:

```
Name = GropuOpcionForm
Items = Tstrings(Altas
                Bajas
                Cambios
                Consultas)
Visible = True
```

- Para realizar los botones de la pantalla seleccione del VCL:en la pestaña de Additional dos botones describiendo las propiedades correspondientes.

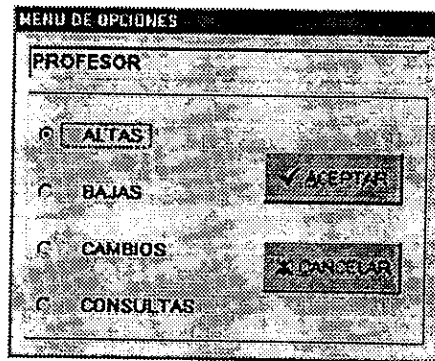


Figura 5.44 Menú de opciones

Pantalla de Altas

- Seleccionar una forma nueva.
- Para crear la barra del menu seleccione del VCL el icono de MainMenu y defina las propiedades similares a la anterior.
- Seleccionar un panel del VCL en la pestaña que indica Standard, dimensionarlo automáticamente con el arrastre del mouse definiendo sus propiedades similares a las descritas anteriormente.
- Para colocar titulos se selecciona y arrastre Cajas de Texto en el panel definiendo sus propiedades similares a las descritas anteriormente.
- Para crear las cajas de entrada de información se selecciona y se arrastra del VCL una caja de texto (EditMask) definiendo como propiedades las siguientes:

Para campos que tiene como entrada máscaras

EditMask = 00;1; para la clave del profesor

EditMask = 000000;1; para el código postal

EditMask = (900)000-0000; para el teléfono

Visible = True

Cursor = CrDefault

Para definir campos que no tienen como entrada una máscara se deja en blanco la opción de EditMask

6. Para realizar los botones de la pantalla seleccione del VCL:en la pestaña de Additional tres botones describiendo las propiedades correspondientes

The screenshot shows a Windows-style application window titled "ALTA PROFESOR". The window contains a form with the following fields and controls:

- CLAVE:** A text field with a mask of "00;1".
- APELLIDO PATERNO:** A text field.
- APELLIDO MATERNO:** A text field.
- NOMBRE:** A text field.
- CARGO:** A text field.
- PROFESION:** A text field.
- ACADEMIA:** A text field.
- STATUS:** A radio button labeled "ACTIVO".
- NIVEL USUARIO:** A text field.
- DISTRITO:** A text field.
- CALLE:** A text field.
- CIUDAD:** A text field.
- CODIGO POSTAL:** A text field with a mask of "0000".
- TELEFONO:** A text field with a mask of "(900)000-0000".
- Buttons:** Three buttons at the bottom: "Aceptar" (with a checkmark icon), "Cancelar" (with an 'X' icon), and "Salir" (with a door icon).

At the bottom of the window, there is a status bar with the text: "Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad. Copyright © 2000 por la Universidad de la Salle. Todos los derechos reservados." and the date "14 de mayo de 2000".

Figura 5.45 Pantalla de alta profesor

5. Para crear las cajas de entrada de información se selecciona y se arrastra del VCL una caja de texto (EditMask) definiendo como propiedades las siguientes:

Para campos que tiene como entrada máscaras

EditMask = 000,1; para la clave de la asignatura

EditMask = /-/-/ para la fecha de aplicación

Visible = True

Cursor = CrDefault

Para definir campos que no tienen como entrada una máscara se deja en blanco la opción de EditMask

6. Para realizar los botones de la pantalla seleccione del VCL: en la pestaña de Additional 6 botones describiendo las propiedades correspondientes.

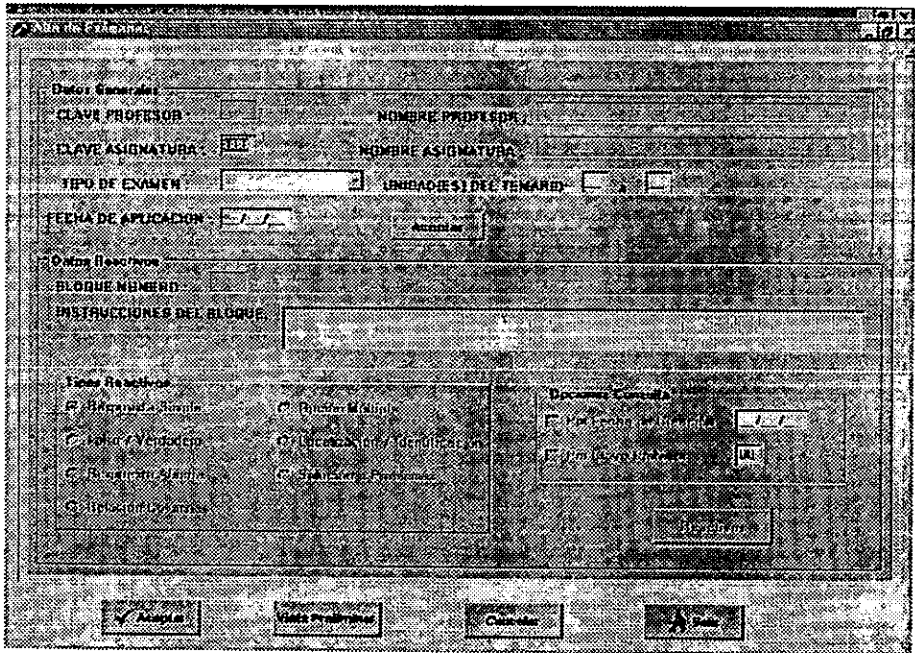


Figura 5.47 Pantalla de captura para la elaboración de exámenes

Pantalla de Reportes

1. Crear una forma nueva.
2. Seleccionar y arrastrar del VCL en la pestaña de Qreport el icono de QuikRep, es aqui donde vamos a crear nuestro reporte de los reactivos definiendo como propiedades las siguientes:

Name = QuikRep
DataSet = Tpaso → Es la tabla de paso, aqui se guardan temporalmente los reactivos para formar el examen.

Las propiedades restantes que da por default se dejan

3. Para indicarle lo que queremos visualizar en el reporte, se selecciona y se arrastra del VCL en el QuikRep una cajas de texto (QRLabel), las propiedades que da por default se dejan.

Este proceso se aplica para todas y cada una de las pantallas que conforman nuestro sistema.



6. RESULTADOS Y PRUEBAS

6.1 INTRODUCCION

Aunque se haya realizado una labor perfecta de análisis, diseño y programación, se debe de hacer un esfuerzo para verificar que no existan errores en el sistema. Generalmente se hace un trabajo "imperfecto" y es entonces cuando hay que realizar pruebas, las cuales se vuelven iterativas: la primera fase de ellas muestra la presencia de errores y las posteriores verifican si las fallas corregidas funcionan de forma correcta.

El proceso de desarrollar casos de pruebas de aceptación puede llevarse a cabo en paralelo con las actividades de implantación del diseño y de la programación, para que, en el momento en que los programadores terminen de escribir sus programas y de realizar sus propias pruebas unitarias, el equipo del analista - usuario esté listo con sus propios casos a probar.

6.2 ETAPA DE PRUEBAS

El proceso de pruebas de Software es, la parte fundamental para asegurar el correcto funcionamiento de los sistemas, ya que permite la eliminación y depuración de una gran parte de las posibles fallas dentro del mismo.

Un sistema probado de manera meticulosa, con base en una metodología perfectamente definida reduce en un gran porcentaje la probabilidad de tener problemas durante su utilización. Aún en caso de llegar a presentar algún error, se tiene la ventaja de poder identificar antecedentes previos completamente documentados, lo que lleva a dar respuestas correctivas mucho más oportunas.

Las etapas de pruebas debe ser desglosada en pequeñas fases, las cuales pueden ser identificadas como niveles de pruebas. Cada uno de estos niveles debe constar de actividades y tareas específicas orientadas a lo que se quiere probar, especificando cuales serán las entradas y las salidas ó resultados esperados.

La etapa de pruebas del Sistema se subdividió en las siguientes fases:

- ❖ **Pruebas Unitarias.** Se realizaron pruebas de la funcionalidad de cada uno de los módulos de manera independiente.
- ❖ **Pruebas Integrales.** Se realizaron pruebas del funcionamiento del sistema como un todo, tal y como lo vería el usuario final. Por medio de estas pruebas se aseguro que las funciones realizadas por el sistema fueran las definidas durante la etapa de análisis.
- ❖ **Pruebas Aceptación.** Esta prueba la realizó el usuario final, en donde se validó el Funcionamiento total del Sistema y se dio su aceptación.

Pruebas Unitarias

Son las pruebas técnicas y funcionales que se realizan a los programas conforme se terminan de codificar. Estas finalizan cuando se ha probado la totalidad de las funciones de las aplicaciones y no se detectan más defectos, estas pruebas se realizan por parte de Desarrollo de Sistemas.

Los objetivos a cubrir con estas pruebas son:

- ❖ Confirmar que el módulo o programa fue codificado correctamente.
- ❖ Determinar la funcionalidad de los programas o módulos desarrollados, es decir, que cada uno de ellos haga lo que tiene que hacer.
- ❖ Encontrar y corregir defectos de cada módulo o programa.

Existe una serie de actividades a realizar dentro del proceso de pruebas unitarias, las cuales se presentan en Figura 6.1

4. Se realizó el análisis de resultados de cada módulo en donde se determina si la información es correcta y si se procesó de acuerdo a los requerimientos del usuario.
5. De acuerdo al análisis anterior pudimos comprobar el funcionamiento de cada uno de los módulos, y además detectar y corregir errores de diseño y de programación no antes vistos en la etapa de análisis.
6. Además, también pudimos darnos cuenta si nuestro Guión de Pruebas cubre todas nuestras necesidades y en caso de no ser así podemos enriquecerlo agregando más información que no estuvo contemplada al principio de esta prueba.
7. Una vez depurado y probado de forma independiente el módulo podemos pasar a la parte de Pruebas de Integración.

Pruebas Integrales.

Las pruebas Integrales son aquellas que ensamblan a cada uno de los módulos de un sistema, es la etapa donde se integran todos los componentes que interactúan en un medio ambiente controlado.

La prueba Integral está diseñada para descubrir las fallas e incompatibilidades entre componentes, además de detectar cualquier implementación incorrecta de las especificaciones o incompatibilidad entre los módulos.

Los objetivos principales de una prueba integral son:

- ❖ Probar la integración de los módulos o programas.
- ❖ Verificar las interfaces entre componentes y equipos.
- ❖ Evaluar que cada componente y su complemento cumplan los requerimientos y especificaciones de diseño.

Existe una serie de actividades a realizar dentro del proceso de pruebas unitarias, las cuales se presentan en Figura 6.2

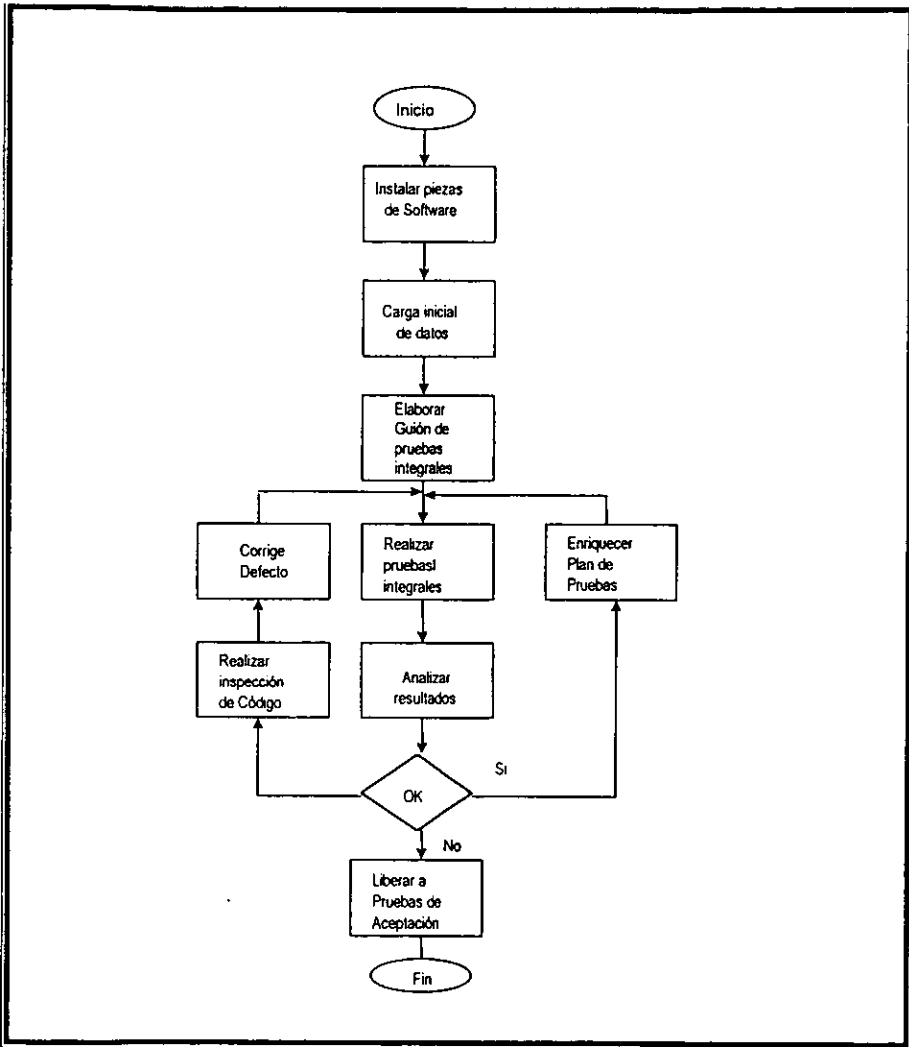


Figura 6.2 Flujo de actividades del Proceso de Pruebas Integrales

Pruebas de Aceptación

Estas pruebas son realizadas por el usuario final para verificar que la aplicación fue desarrollada conforme a sus requerimientos y que realiza todas las funciones correctamente.

Generalmente, tras el buen éxito de estas pruebas, el usuario emite una carta de aceptación, en donde señala que el sistema cumple con todos sus requerimientos.

En estas pruebas intervienen los siguientes responsables:

- ❖ Líder de Proyecto. Coordina las pruebas, valida los guiones de pruebas del usuario y asegura la corrección de cualquier defecto que se encuentre.
- ❖ Usuario. Responsable de desarrollar los guiones de pruebas, orientados a la validación del sistema y del cumplimiento de sus requerimientos.

Se realizó una prueba adicional al sistema para verificar su correcto funcionamiento en situaciones extremas:

Pruebas de Esfuerzo

Las transacciones de esfuerzo se identifican como situaciones imprevistas, que podría enfrentar un módulo o programa.

El objetivo de las Pruebas de Esfuerzo es el provocar situaciones de carga de trabajo y/o anormales para observar el comportamiento de la aplicación.

Como parte de una prueba de esfuerzo se pueden considerar las siguientes tareas:

- ❖ Copia de archivos.
- ❖ Generación de respaldos.
- ❖ Interpretación de mensajes.
- ❖ Generación de productos (archivos y reportes)
- ❖ Flujo de procesos de cada sistema.
- ❖ Reinicios y recuperaciones.
- ❖ Baja de servidores normal y anormalmente.
- ❖ Diálogos entre aplicaciones y/o computadoras.
- ❖ Validar casos técnicos de control.
- ❖ Validar casos técnicos de esfuerzo.
- ❖ Validar casos técnicos de comunicaciones.

Bienvenida



Bienvenido al “Sistema de Control y Automatización de Evaluaciones para el CETIS N°152. Este manual le permitirá poner manos a la obra rápidamente. Aquí aprenderá paso a paso a realizar una Evaluación con una interfaz intuitiva y agradable.

MANUAL DEL USUARIO

INTRODUCCIÓN

Propósito

Este manual tiene la intención de explicarle la manera correcta de utilizar este sistema.

Contenido

Este manual contiene ocho capítulos los cuales son descritos a continuación:



Inicializando el sistema

Capítulo 1



Comenzar a trabajar con Profesores

Capítulo 2



Comenzar a trabajar con Reactivos

Capítulo 3



Comenzar a trabajar con Asignaturas

Capítulo 4



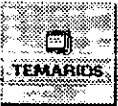
Comenzar a trabajar con Exámenes

Capítulo 5



Comenzar a trabajar con Claves

Capítulo 6




Comenzar a trabajar con Temarios



CONVENCIONES (Normas de estilo)

El presente manual del usuario emplea las siguientes normas estilísticas:

- ♦ Una flecha  implica un procedimiento a través de una o más acciones que deben seguirse paso a paso.
- ♦ Las listas enumeradas indican procedimientos con pasos que usted tiene que realizar en determinada secuencia.
- ♦ Cada capítulo en este manual de usuario comienza con una especificación de las partes de la Ventana del capítulo a tratar.

Normas sobre el uso del ratón:

Este término	Significa
Haga clic	Señale el botón que desee seleccionar y luego pulse e inmediatamente suelte el botón del Mouse pero sin mover el propio Mouse.
Haga doble clic	Señale lo que desee seleccionar y luego haga clic en el botón izquierdo del Mouse dos veces en rápida sucesión pero sin mover el Mouse.
Seleccione	Haga clic en la pestaña o el elemento que desee.
Arrastre y suelte	Señale lo que desee, pulse y mantenga pulsando el botón izquierdo del Mouse mientras mueve el propio Mouse. Una vez que haya movido el Mouse, coloque el puntero en la posición que desee, suelte el botón del Mouse

Al inicio de cada capítulo se da una descripción general de las partes que conforman las ventanas de cada modulo.

Partes Comunes de las Ventanas:

❖ Cuadros de texto

La información que usted escribe se realiza en un cuadro de texto. Al desplazarse a un cuadro de texto vacío, aparecerá un punto de inserción (barra vertical intermitente). El texto que escriba aparecerá en el punto de inserción.

Para seleccionar un cuadro de texto haga clic con el Mouse o pase de un cuadro de texto a otro con la tecla tabulador.

❖ Casillas de verificación

Una casilla de verificación presenta opciones no excluyentes de las cuales puede seleccionar tantas como necesite. Cuando se selecciona una casilla de verificación Examen, ésta contiene una marca de verificación. Para activar o desactivar la casilla de verificación examen haga clic o bien presiona la barra espaciadora.

❖ Ayuda

La ayuda en pantalla en el Sistema de Evaluación es una forma conveniente y rápida de obtener información a cerca de cualquier tarea que es esté realizando o cualquier función o cuadro de diálogo sobre el cual se desea saber más. Existe una Barra de Ayuda en cada pantalla del Sistema de Evaluación en el cual moverse con el Mouse de uno a otro elemento de la pantalla, va mostrando información sobre el elemento en selección.



Inicializando el sistema

En este capítulo se explica la forma de entrar en el sistema, lo cuál es muy fácil de hacer, solo siga las instrucciones tal como se le indican.

⇒ En el Menú de Inicio del Sistema



⇒ Con el mouse, haga click en

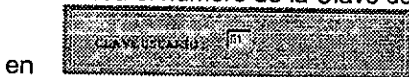


Aparecerá en la pantalla CLAVE DE ACCESO AL SISTEMA

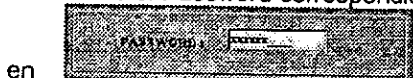


⇒ Para tener acceso al Sistema es necesario disponer de una de Clave que nos permita identificarnos como usuarios.

1.- Escriba el número de la Clave de Usuario

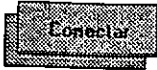


2.- Teclee el Password correspondiente

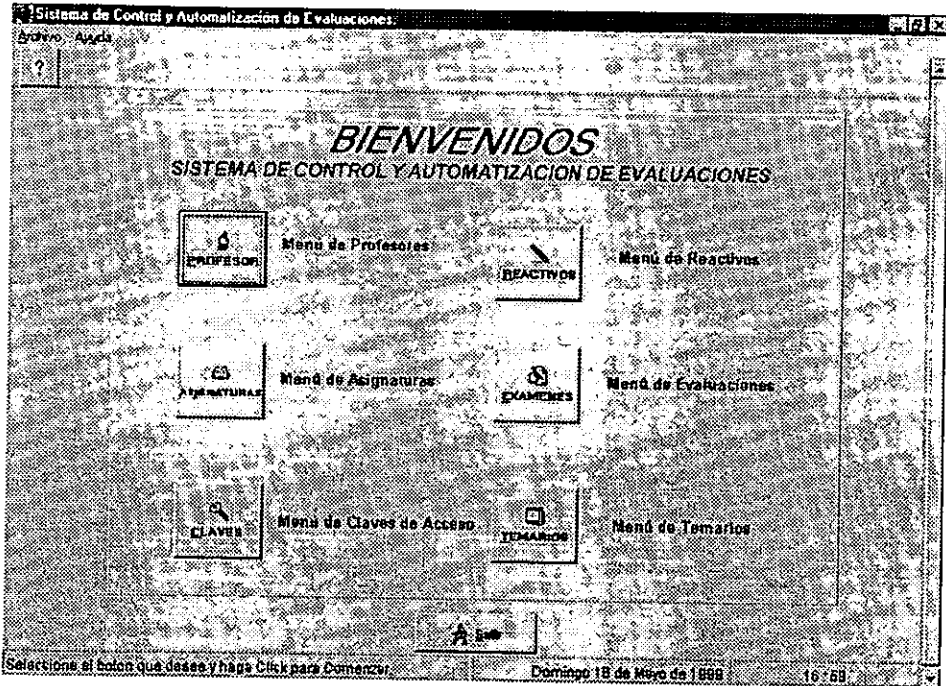


, verifique el formato de la palabra como la escribo al darla de Alta, Ejemplo: Azucena, la debe de teclear tal como esta aqui escrita.

➡ Con el mouse, haga click en



Aparecera la Pantalla el Menú Principal y aqui eligira las actividades que desee realizar.





7 Profesores

En este capítulo

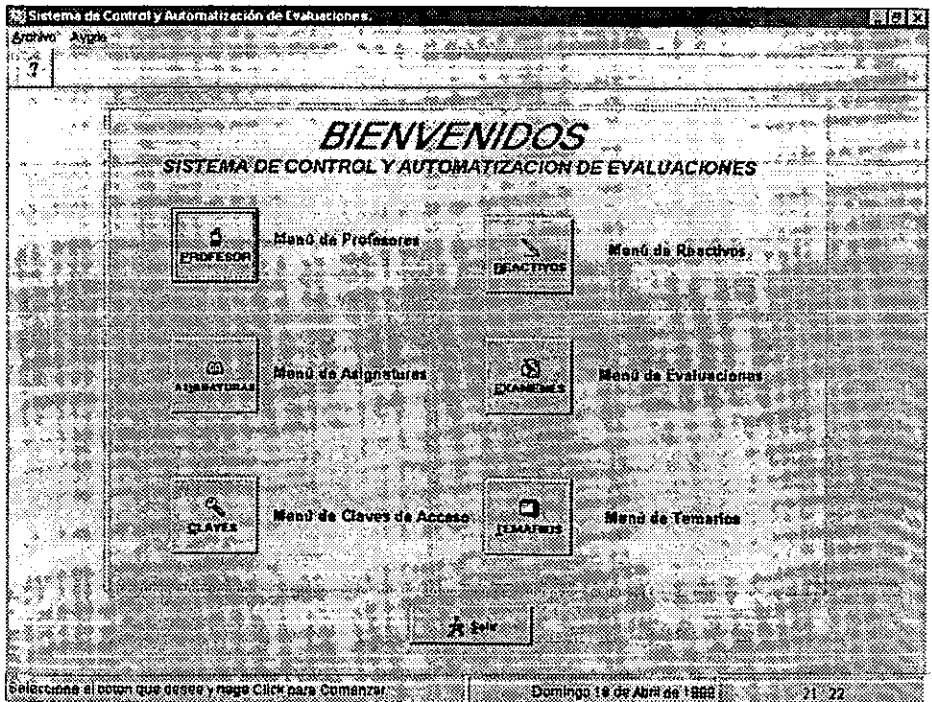
Los temas en este capítulo son:

- ❖ Como Agregar los datos de un Profesor
- ❖ Como Eliminar los datos de un Profesor
- ❖ Como Modificar los datos de un Profesor
- ❖ Como Consultar los datos de un Profesor

❖ Acceso a Profesores:



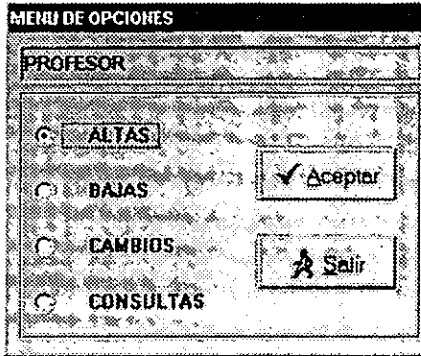
En el Menú Principal.
Aparecerá la pantalla MENÚ DE PROFESORES.



Para obtener acceso a los Profesores,
Haga clic en:



Usted podrá seleccionar un botón de opción de diferentes movimientos que se pueden hacer con los Profesores. MENÚ DE OPCIONES



❖ Como **Agregar** los datos de un Profesor:



Seleccione el botón de opción de Altas con el mouse, haga clic en



Aparecerá en la pantalla ALTA DATOS PROFESOR

ALTA DATOS PROFESOR

Activo: Activo

DATOS PERSONALES

CLAVE: 14

APPELLIDO PAPA: _____

APPELLIDO MAM: _____

NOMBRE: _____

CARGO: _____

PROFESION: _____

ACADEMIA: _____

STATUS: ACTIVO

NIVEL USUARIO: _____

DOMICILIO

CALLE: _____

SOLONGA: _____


COD. POSTAL: 00000

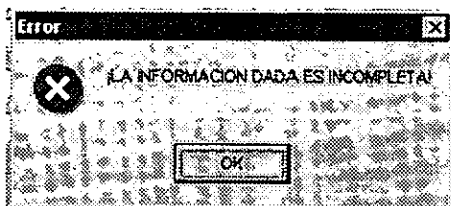
CUIDAD: _____

ESTADO CIVIL: _____

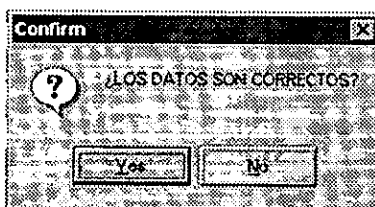
TELÉFONO: (00) 5000-0000

Teclea los datos del Profesor a dar de Alta y haga Click en Aceptar. Domingo 18 de Abril de 1999 71 / 51

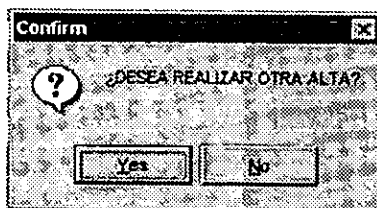
1. Recordar la clave que asigna el sistema para la nueva alta del profesor.
2. Escribir o llenar los cuadros de texto con los datos personales y domiciliarios , del profesor y oprima la tecla enter u haga clic en .
3. Es necesario que se den los datos completos para evitar el siguiente mensaje, en el cual haga click en OK.



4. Después de Grabar. Aparecerá el siguiente mensaje de confirmación:




5. Y también un mensaje para continuar con otra alta de datos de un Profesor.



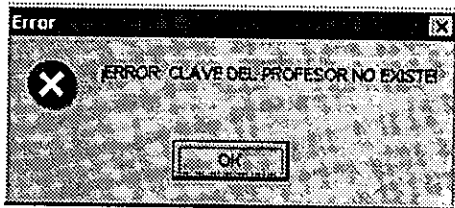
6. Finalmente, si la respuesta es NO. Entonces se regresará al menú principal.

❖ **Como Eliminar los datos de un Profesor.**

- En el Menú de Opciones, seleccione el botón de opción de **Bajas** con el mouse, y haga clic en .

Aparecerá en la pantalla BAJAS PROFESOR.

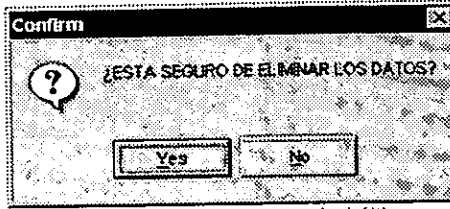
1. Recordando la clave que asigna el sistema cuando se dio la alta del profesor. Es necesario dar la clave correctamente, para evitar el siguiente mensaje, el cual es necesario contestarlo con O.K.



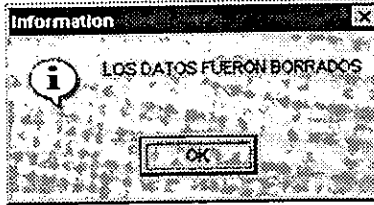
2. Cuando se escriba la clave correcta, oprima enter, y se aparecerán los datos del profesor que se desean eliminar. Haga clic en el botón




3. Para completar la Eliminación de los datos del Profesor, aparecerá el siguiente mensaje de confirmación:



- Finalmente, si la respuesta fue **Yes**, aparecerá el último mensaje. Haga click en **OK**. Y se regresará al menú principal.



❖ **Como Modificar los datos de un Profesor.**

- Seleccione en el MENÚ DE OPCIONES el botón de opción de Cambios haga clic . Aparecerá en la pantalla CAMBIOS DATOS PROFESOR.

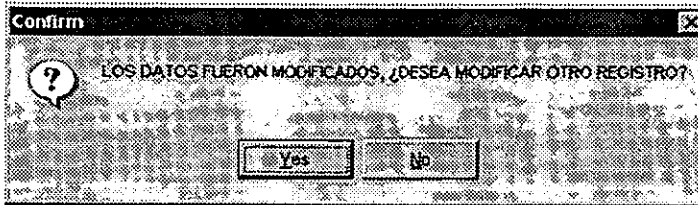
1. Escribir la clave del profesor a modificar y haga click en



2. Modifique los datos y haga click en



3. Después preguntará si los datos son correctos y confirmará con el siguiente mensaje.



4. Si se deseará modificar otro registro, se regresará a la pantalla de cambios profesor y pedirá nuevamente una clave, etc.

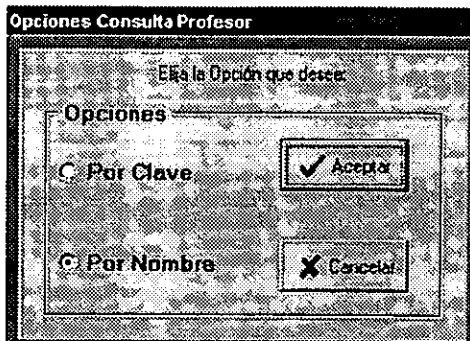
❖ **Como Consultar los datos de un Profesor**



Seleccione en el MENÚ DE OPCIONES el botón de opción de Consultas haga clic



.Aparecerá en la pantalla OPCIONES CONSULTA PROFESOR



Seleccionado la consulta de profesor por clave. Aparecerá la pantalla de

CONSULTA PROFESOR POR CLAVE y haga click en



CONSULTA DATOS PROFESOR

Archivo Ayuda

CONSULTA PROFESOR POR CLAVE

DATOS PERSONALES

CLAVE:

NOMBRE:

CARGO:

PROFESION:

ACADEMIA:

ESTADO:

NIVEL DE USUARIO: INVITADO

DOMICILIO

CALLE:

CIUDAD:

TELÉFONO:

ESTADO CIVIL:

COLONIA:

CÓDIGO POSTAL:

Para comenzar escriba la Clave del Profesor a Consultar y haga Click en Aceptar

Domingo 18 de Abril de 1999 23:1 Hrs

1. Escribir la clave del profesor que se desea consultar y haga click en



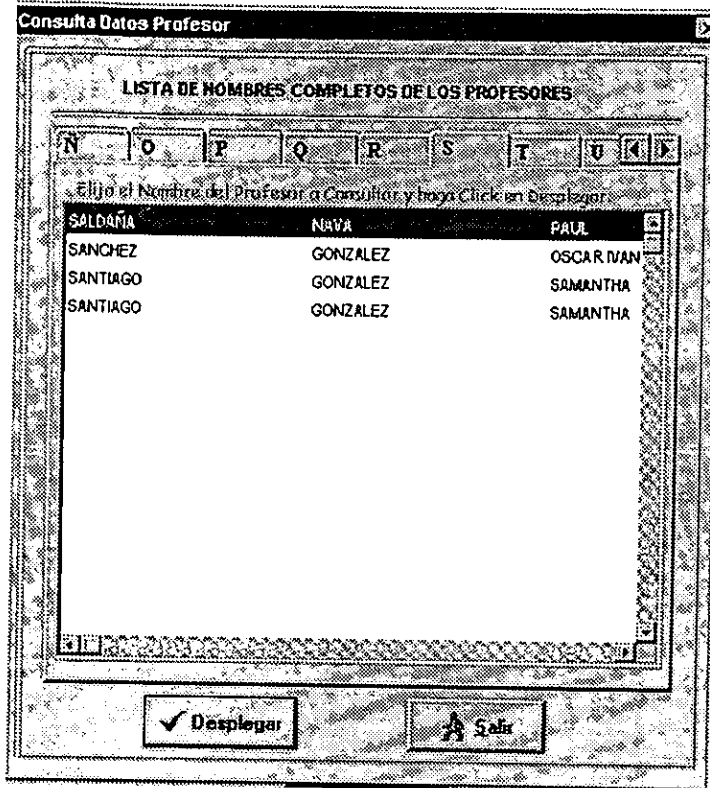
2. Puede usar también las flechas de la esquina superior derecha, para moverse entre otros datos de profesor.




Seleccionado la consulta de profesor por nombre. Aparecerá la pantalla de

CONSULTA PROFESOR POR NOMBRE y haga click en





1. Seleccionar la lengüeta correspondiente al apellido paterno del profesor, elija el nombre del profesor a consultar y haga click en 
2. Aparecerá el profesor elegido por nombre en la pantalla Consulta profesor por nombre.

CONSULTA DATOS PROFESOR

CONSULTA DATOS POR NOMBRE

PERSONAL

04

NOMBRE: SALDAÑA NAVA PAUL

CARGO: PROFESOR

PROFESION: ING. EN COMPUTACION

ACADEMIA: MATEMATICAS

STATUS: ACTIVO NIVEL: ADMINISTRADOR

DOMICILIO

CALLE: SIETE # 233 COLONIA: PORVENIR

CUIDAD: MEXICO D.F. CODIGO POSTAL: 2940

TEL.FONO: (15) 355-0096

ESTADO CIVIL: SOLTERO

Para comenzar escriba la Clave del Profesor en el cuadro correspondiente. Dominio: 16/07/2006 10:27:56 Hr.

3. Finalmente, haga click en





2 Reactivos

En este capítulo

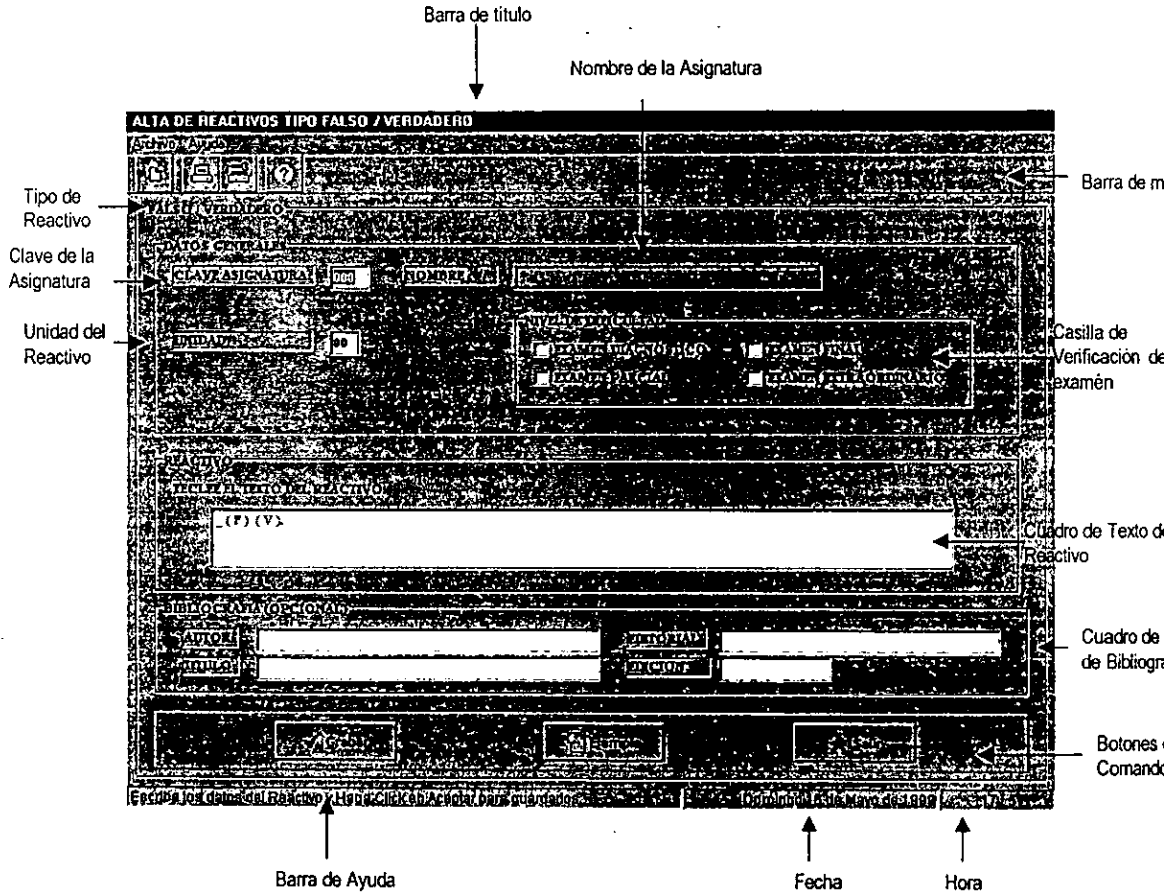
Los temas en este capítulo son:

- ❖ ¿Qué es un Reactivo?
- ❖ Partes generales de la ventana de Reactivos
- ❖ Tipos de Reactivos
- ❖ Acceso a Reactivos
- ❖ Como Agregar un Reactivo
- ❖ Como Eliminar un Reactivo
- ❖ Como Modificar un Reactivo
- ❖ Como Consultar un Reactivo

❖ ¿Qué es un Reactivo?

Es el planteamiento de una situación que requiere solución, que propone acciones o suscita reacciones que se traducen en respuestas, de cuyo grado de acierto sea posible hacer un diagnóstico sobre los alcances del aprendizaje.

❖ Partes generales de la ventana de Reactivos



❖ Tipos de Reactivos

El examinador tiene todavía un campo bastante amplio en el que puede desenvolverse para elaborar los reactivos, utilizando formas básicas de construcción sustancialmente diferentes tanto en su diseño como su aptitud para explorar aprendizajes de distinta indole y nivel.

Entre estos tipos básicos de reactivo tenemos:

1. Respuesta Simple o Breve
2. Respuesta Falso / Verdadero
3. Respuesta Abierta
4. Respuesta de Opción Múltiple
5. Respuesta de Relación de Columnas
6. Respuesta a Solución de problemas
7. Respuesta Localización / Identificación

➤ **REACTIVO TIPO RESPUESTA SIMPLE O BREVE**

Estos reactivos se caracterizan por el establecimiento de una proposición incompleta o una interrogante en que, a partir de la información reportada se espera el "completamiento" o respuesta correspondiente que consiste en la aportación de un término, frase específica, símbolo, dato, etc, cuyo conocimiento se busca comprobar.

➤ **REACTIVO TIPO RESPUESTA FALSO / VERDADERO**

Estos reactivos se caracterizan por limitar la respuesta a una de dos opciones o alternativas para calificar una aseveración o enunciado.

➤ **REACTIVO TIPO RESPUESTA ABIERTA**

Estos reactivos se caracterizan porque conceden al alumno una elaboración de dicha respuesta en un contenido muy definido, fijando una extensión máxima o una organización que al ser atendida, impide la pluralidad de las versiones, por lo menos desde el punto de vista estructural.

➤ **REACTIVO TIPO RESPUESTA OPCIÓN MÚLTIPLE**

Estos reactivos están constituidos en su forma clásica por un enunciado incompleto o una pregunta (cuerpo del reactivo) y varias posibles respuestas (opciones o alternativas), entre las cuales una completa responde correctamente al enunciado o pregunta inicial.

➤ **REACTIVO TIPO RESPUESTA RELACIÓN DE COLUMNAS**

Estos reactivos se caracterizan por pedir el establecimiento de relaciones entre elementos de dos grupos o series, y se requiere de instrucciones muy claras para orientar sobre el criterio que se ha de utilizar para establecer dichas relaciones.

➤ **REACTIVO TIPO RESPUESTA SOLUCIÓN A PROBLEMAS**

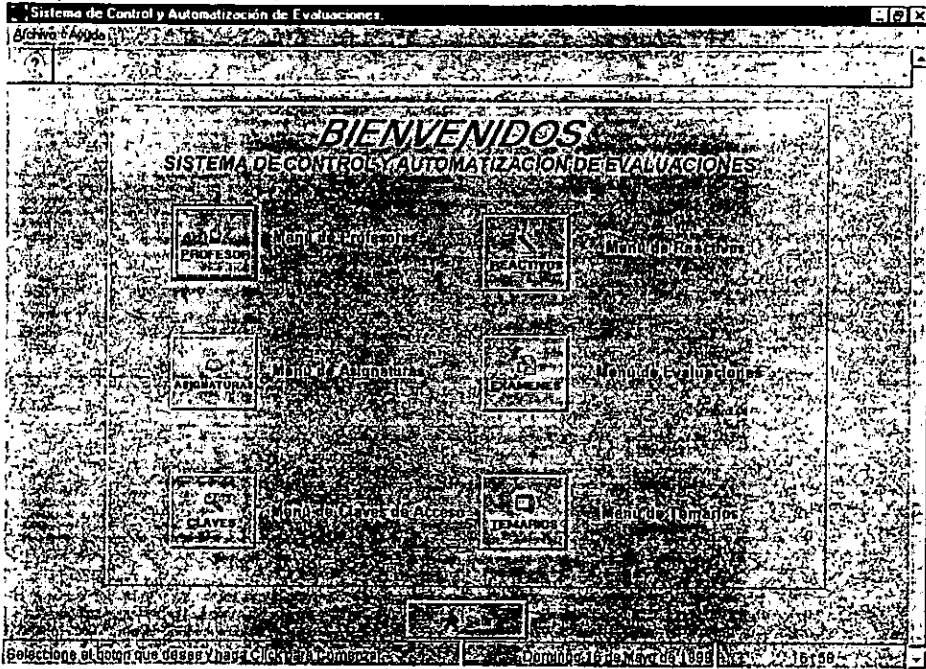
Un problema es una proposición compleja de elementos a la que corresponden una o varias soluciones o una incógnita a dejar, atendiendo al manejo que se haga de tales elementos. Es una cuestión que se trata de resolver mediante el estudio adecuado de los elementos que abarca; por su parte la educación en este estudio se halla íntimamente ligada con la aplicación de principio, leyes, fórmulas, normas o criterios. Se debe plantear un problema concreto que obligue al examinado a efectuar los pasos y realizar las operaciones que le llevan a la solución global.

➤ **REACTIVO TIPO RESPUESTA LOCALIZACIÓN / IDENTIFICACIÓN**

Estos reactivos se caracterizan por contener una serie de indicaciones, referidas a conductas que han de verificar sobre un material esquemático o gráfico dado en el mismo reactivo (mapas, diagramas, representaciones, topográficas, cuadros sinópticos, ilustraciones, modelos, etc.). Las conductas solicitadas consisten esencialmente en ubicar elementos o puntos sobre el material esquemático, lo cual implica aprendizajes que pueden llegar hasta el nivel de aplicación.

❖ Acceso a Reactivos

➡ En el MENÚ PRINCIPAL

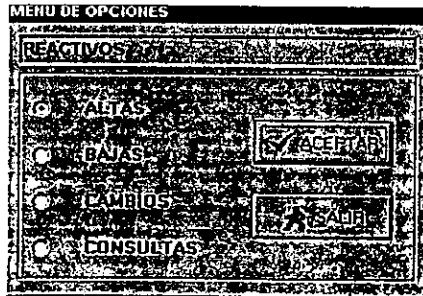


➡ Para obtener acceso a los Reactivos, haga clic en



En el Menú Principal.

Aparecerá la pantalla MENÚ OPCIONES REACTIVOS.



Usted podrá seleccionar un botón de opción de diferentes movimientos que se pueden hacer con los reactivos.

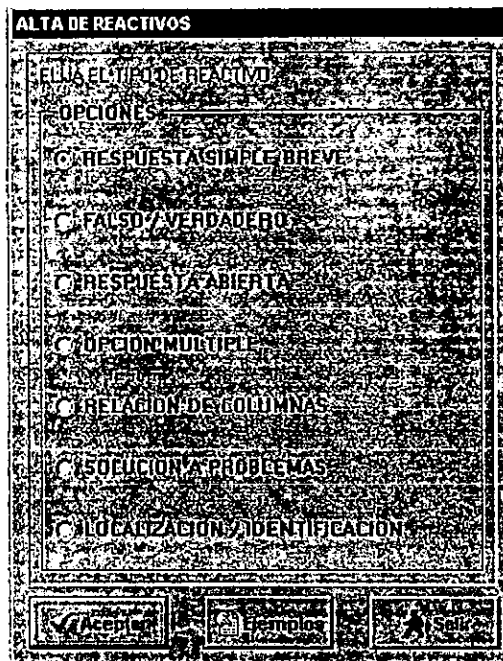
❖ **Como Agregar un Reactivo**



Seleccione el botón de opción de Altas con el mouse, haga clic en



. Aparecerá en la pantalla ALTA DE REACTIVOS



Usted podrá seleccionar un botón de opción de siete tipos de reactivos que contiene el sistema.

⇒ **EJEMPLO:** Si su selección es Respuesta Simple o Breve

Aparecerá la pantalla de ALTA REACTIVO RESPUESTA SIMPLE O BREVE.

ALTAS DE REACTIVOS TIPO RESPUESTA SIMPLE BREVE

RESPUESTA SIMPLE BREVE

DATOS GENERALES

CLAVE ASIGNATURA: NOMBRE:

UNIDAD: NIVEL DE DIFICULTAD:

EXAMEN DIAGNÓSTICO EXAMEN FINAL

EXAMEN PARCIAL EXAMEN TIPO ORDINARIO

REACTIVO

TECLA EL TEXTO DEL REACTIVO:

BIBLIOGRAFÍA (OPCIONAL)

AUTORE: BIBLIOGRAFIA:

EDITOR: EDICION:

Escriba los datos del Reactivo y Haga Click en Aceptar para guardarlo. 11:57

⇒ Para agregar un reactivo de ALTA RESPUESTA SIMPLE O BREVE

1. Escriba la clave de la asignatura a la que pertenecerá su reactivo

en y haga clic en

u oprima la tecla Enter.

2. El nombre de la Asignatura aparecerá en

3. Escriba la unidad a la pertenecerá el reactivo en

y oprima la tecla tabulador.

4. Seleccione el tipo de examen al que pertenecerá el reactivo, haciendo clic con el mouse o con la barra espaciadora en

NIVEL DE DIFICULTAD	
<input type="checkbox"/> EXAMEN DIAGNOSTICO	<input checked="" type="checkbox"/> EXAMEN FINAL
<input type="checkbox"/> EXAMEN PARCIAL	<input type="checkbox"/> EXAMEN EXTRAORDINARIO

Y continúe con la tecla tabulador.

5. Escriba la pregunta o texto de su reactivo en el cuadro de texto nombrado REACTIVO y oprima tabulador. Si se equivoca utilice la tecla retroceso. El tipo del reactivo es distinto, emplee el tabulador para moverse también entre los distintos cuadros de texto del reactivo.

6. Escriba en el cuadro de texto BIBLIOGRAFÍA, los datos requeridos los cuales

BIBLIOGRAFIA	
<input type="text" value="AUTOR"/>	<input type="text" value="AUTORIA"/>
<input type="text" value="TITULO"/>	<input type="text" value="EDICION"/>

pueden dejarse en blanco. Moverse entre los datos con la tecla tabulador o haga clic en el cuadro donde desee escribir.

7. Haga clic en



8. Aparecerá un mensaje ¿LOS DATOS SON CORRECTOS?. Haga clic en Yes. Su Reactivo será agregado o dado de Alta.
9. Aparecerá un mensaje ¿LOS DATOS FUERON GRABADOS DESEADAR OTRA ALTA?. Si hace clic en Yes aparecerá nuevamente la pantalla de Alta de Reactivo, Si hace clic en NO regresará al Menú Principal.

➡ Para dar de Alta un Reactivo TIPO FALSO / VERDADERO deberá escoger la opción correspondiente.

Aparecerá la pantalla de ALTA DE REACTIVOS TIPO FALSO / VERDADERO

ALTA DE REACTIVOS TIPO FALSO / VERDADERO

Archivo | Ayuda

FALSO / VERDADERO

DATOS GENERALES

CLAVE ASIGNATURA: NOMBRE:

ENTIDAD: NIVEL DE ESTUDIOS:

EXAMEN DIAGNOSTICO EXAMEN FINAL

EXAMEN PARCIAL EXAMEN EXTRAORDINARIO

REACTIVO

SELECCIONA EL TIPO DEL REACTIVO

(F) (V)

BIBLIOGRAFIA (OPCIONAL)

AUTOR: AUTORIAL:

TITULO: EDICION:

ESCRIBA LOS DATOS DEL REACTIVO Y Haga Click en Aceptar para Validar

12 de Mayo de 1998

➡ Los pasos siguientes para agregar o dar una alta de este reactivo fueron descritos en la parte de **Como Agregar un Reactivo**.

- ☞ Para dar de Alta un Reactivo TIPO RESPUESTA ABIERTA deberá escoger la opción correspondiente.

Aparecerá la pantalla de ALTA DE REACTIVOS TIPO RESPUESTA ABIERTA

ALTA DE REACTIVOS TIPO RESPUESTA ABIERTA

Archivo Ayuda

RESPUESTA ABIERTA

DATOS GENERALES

CLAVE ASIGNATURA:
 NOMBRE ID:
 NOMBRE DEL ASIGNATURA:

UNIDAD I. E. C. A.:
 NIVEL DE DIFICULTAD:
 EXAMEN TIPO TEST
 EXAMEN FINAL
 EXAMEN PARCIAL
 EXAMEN EXTRAORDINARIO

REACTIVO

TECLEE EL TEXTO DEL REACTIVO:

BIBLIOGRAFIA (OPCIONAL)

AUTOR:
 EDITORIAL:

TITULO:
 EDICION:

Estos son los datos del Reactivo y Haga Click en Aceptar para guardarlo. Jueves 8 de Mayo de 1999 12:31

- ☞ Los pasos siguientes para agregar o dar una alta de este reactivo fueron descritos en la parte de **Como Agregar un Reactivo**.

➤ Para dar de Alta un Reactivo TIPO OPCIÓN MULTIPLE deberá escoger la opción correspondiente.

Aparecerá la pantalla de ALTA DE REACTIVOS TIPO OPCIÓN MULTIPLE

➤ Los pasos siguientes para agregar o dar una alta de este reactivo fueron descritos en la parte de **Como Agregar un Reactivo.**

- Para dar de Alta un Reactivo TIPO RELACIÓN DE COLUMNAS deberá escoger la opción correspondiente.

Aparecerá la pantalla de ALTA DE REACTIVOS TIPO RELACIÓN DE COLUMNAS.

ALTA DE REACTIVOS TIPO RELACION DE COLUMNAS

REACTIVO TIPO RELACION DE COLUMNAS

DATOS GENERALES

CLAVE ASIGNATURA: [000] NOMBRE: []

UNIDAD: [00] NIVEL DE DIFICULTAD:

EXAMEN DIAGNOSTICO EXAMEN FINAL

EXAMEN PARCIAL EXAMEN EXTRAORDINARIO

REACTIVO

[] [] []

BIBLIOGRAFIA

AUTOR: [] EDITORIAL: []

TITULO: [] EDICION: []

Aceptar Cancelar Borrado Salir

Clickear la Clave del Profesor a Modificar y haga Click en Aceptar para desplegar 31 de Mayo de 1999 12:34

- Los pasos siguientes para agregar o dar una alta de este reactivo fueron descritos en la parte de **Como Agregar un Reactivo**.

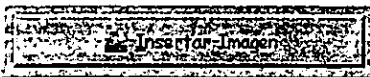
Para dar de Alta un Reactivo TIPO SOLUCIÓN A PROBLEMAS deberá escoger la opción correspondiente.

Aparecerá la pantalla de **ALTA DE REACTIVOS TIPO SOLUCIÓN A PROBLEMAS**

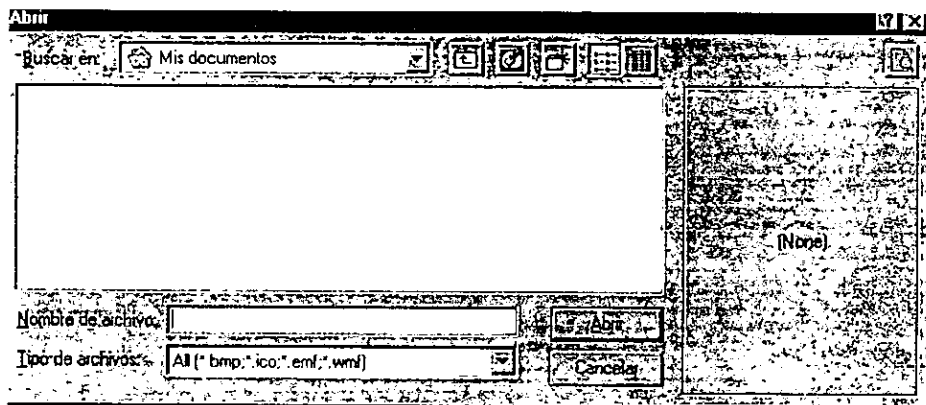
Los pasos siguientes para agregar o dar una alta de este reactivo fueron descritos en la parte de **Como Agregar un Reactivo**.

Los reactivos TIPO SOLUCIÓN A PROBLEMAS Y TIPO LOCALIZACIÓN / IDENTIFICACIÓN manejan algunas particularidades, puesto que son Reactivos tipo esquemáticos o gráficos.

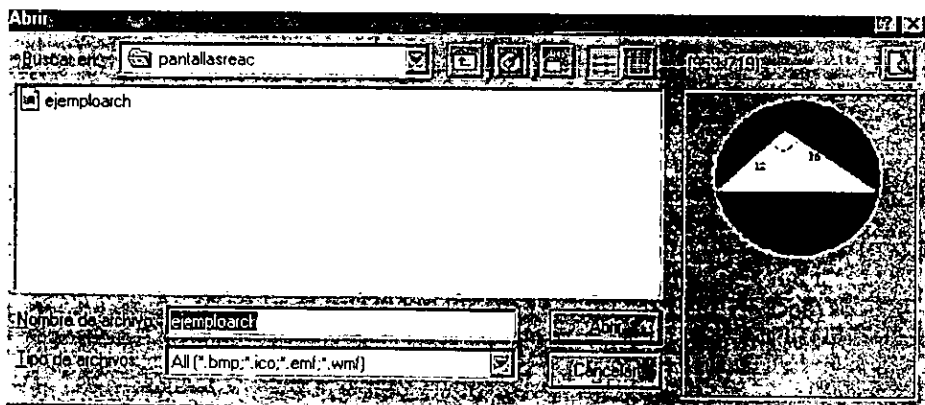
1. Para agregar una imagen o gráfico a este tipo de reactivos haga clic en



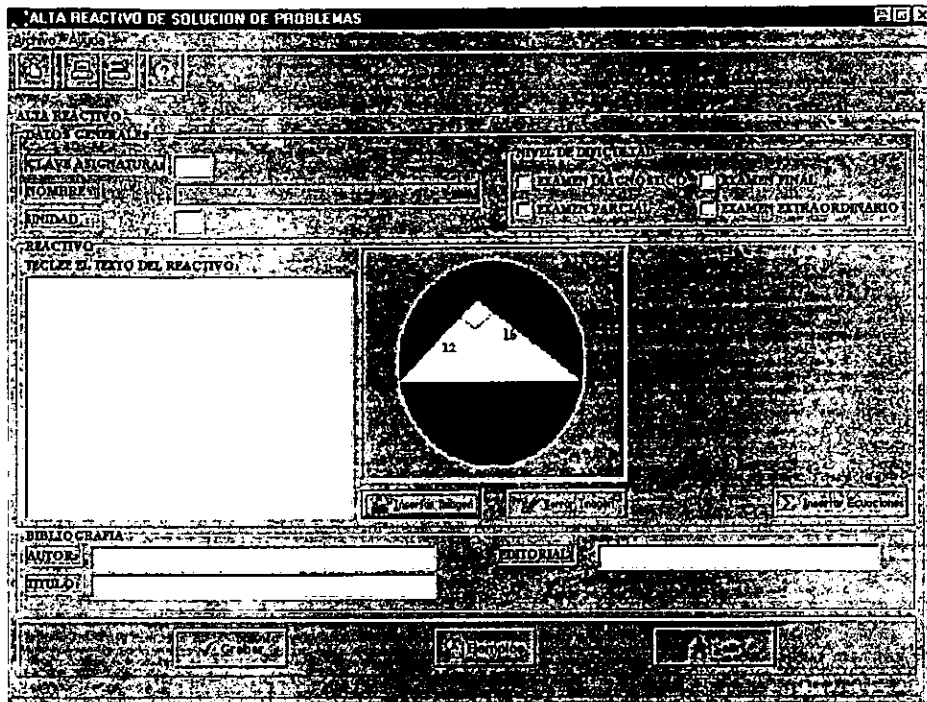
2. Aparecerá una pantalla desde la cuál podrá dar la ruta para importar la imagen (La imagen que se desee importar previamente deberá estar en formato *.bmp, *.ico, *.emf, *.wmf)



3. Una vez seleccionada la imagen presionar el botón de Abrir



4. La imagen aparecerá en el recuadro de imagen



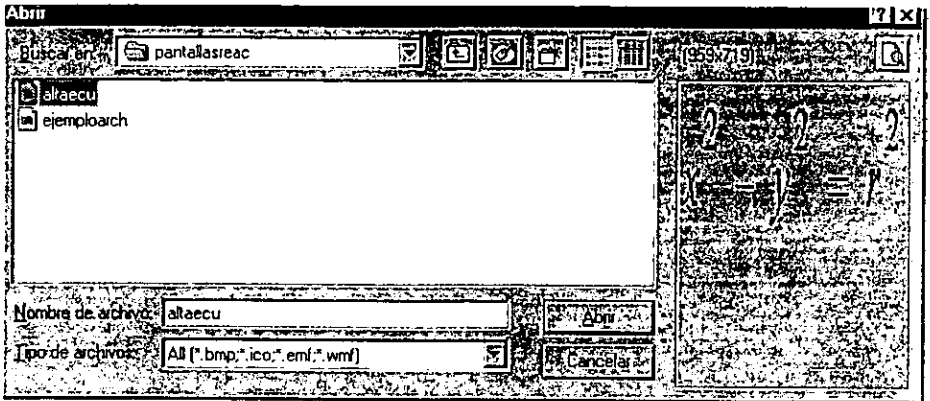
5. Si no desea cambiar esta imagen por otra solo presione el botón.



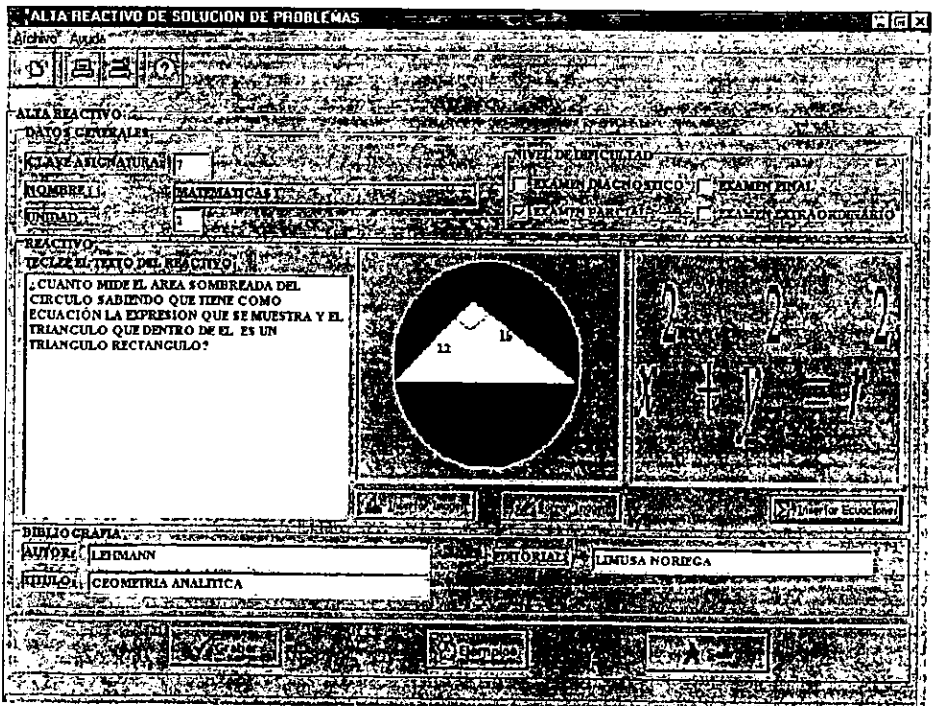
6. Para agregar OPERACIONES U ECUACIONES. Haga clic en el botón



7. Aparecerá una pantalla desde la cuál podrá dar la ruta para importar la ecuación (La imagen que se desee importar previamente deberá estar en formato *.bmp, *.ico, *.emf, *.wmf)



8. La ecuación aparecerá en el recuadro de ecuación



⇒ Para dar de Alta un Reactivo TIPO LOCALIZACIÓN / IDENTIFICACIÓN deberá escoger la opción correspondiente.

Aparecerá la pantalla de ALTA DE REACTIVOS TIPO LOCALIZACIÓN E IDENTIFICACIÓN.

ALTA REACTIVO DE LOCALIZACION O IDENTIFICACION

ALTA REACTIVO

DATOS GENERALES

CLAVE ASIGNATURA:

NOMBRE:

CIUDAD:

NIVEL DE DIFICULTAD

EXAMEN TECNICO:

EXAMEN EXTRAORDINARIO:

EXAMEN EXTRAORDINARIO:

REACTIVO

REACTIVO

BIBLIOGRAFIA


AUTOR:

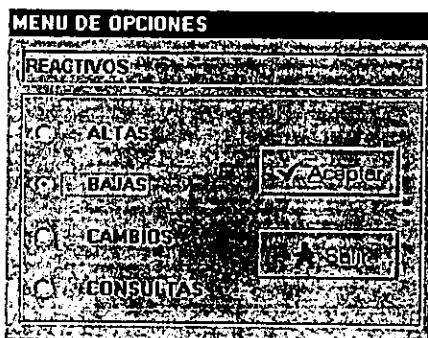
TITULO:

EDITORIAL:

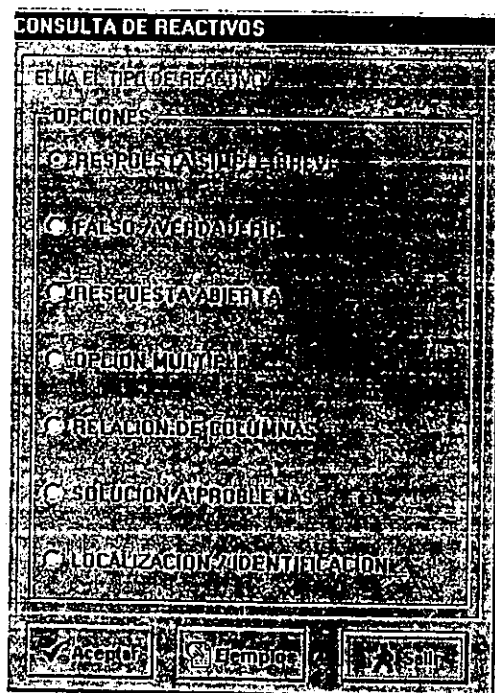
⇒ Los pasos siguientes para agregar o dar una alta de este reactivo fueron descritos en la parte de **Como Agregar un Reactivo y en Reactivo de Solución a Problemas.**

❖ Como Eliminar un Reactivo

- Seleccione el botón de opción de Bajas con el mouse, haga clic en . Aparecerá en la pantalla ALTA DE REACTIVOS




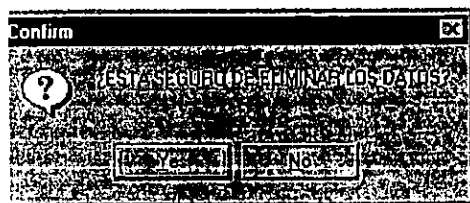
- Aparecerá la pantalla de MENÚ DE REACTIVOS, elija tipo de reactivo que desea eliminar y haga clic en 



⇒ **EJEMPLO:** Si su selección es Respuesta Simple o Breve

Aparecerá la pantalla de CONSULTA DE REACTIVOS TIPO RESPUESTA SIMPLE O BREVE.

⇒ Haga clic en  y aparecerá una ventana de confirmación.



Haga clic en el botón de YES.
Finalmente una confirmación de que los datos fueron eliminados.
Haga clic en OK.

- ☞ Para dar de Baja un Reactivo TIPO FALSO / VERDADERO deberá seleccionar el reactivo que desea eliminar.

Aparecerá la pantalla de CONSULTA DE REACTIVOS TIPO FALSO/VERDADERO

- ☞ Haga clic en

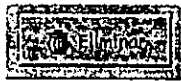


Los pasos siguientes para eliminar o dar de baja este reactivo fueron descritos en la parte de **Como Eliminar un Reactivo**.

- ➡ Para dar de Baja un Reactivo TIPO RESPUESTA ABIERTA deberá seleccionar el reactivo que desea eliminar.

Aparecerá la pantalla de CONSULTA DE REACTIVOS TIPO RESPUESTA ABIERTA

- ➡ Haga click en



- ➡ Los pasos siguientes para eliminar o dar de baja este reactivo fueron descritos en la parte de **Como Eliminar un Reactivo.**

➤ Para dar de Baja un Reactivo OPCIÓN MULTIPLE deberá seleccionar el reactivo que desea eliminar.

Aparecerá la pantalla de CONSULTA DE REACTIVOS TIPO OPCIÓN MULTIPLE

➤ Haga click en



➤ Los pasos siguientes para eliminar o dar de baja este reactivo fueron descritos en la parte de **Como Eliminar un Reactivo.**

- Para dar de Baja un Reactivo TIPO RELACIÓN DE COLUMNAS deberá seleccionar el reactivo que desea eliminar.

Aparecerá la pantalla de CONSULTA DE REACTIVOS TIPO RELACIÓN DE COLUMNAS

- Haga clic en



- Los pasos siguientes para eliminar o dar de baja este reactivo fueron descritos en la parte de **Como Eliminar un Reactivo**.

- ☞ Para dar de Baja un Reactivo TIPO SOLUCIÓN A PROBLEMAS deberá seleccionar el reactivo que desea eliminar.

Aparecerá la pantalla de CONSULTA DE REACTIVOS TIPO SOLUCIÓN A PROBLEMAS

- ☞ Haga clic en



- ☞ Los pasos siguientes para eliminar o dar de baja este reactivo fueron descritos en la parte de **Como Eliminar un Reactivo**.

- Para dar de Baja un Reactivo TIPO LOCALIZACIÓN / IDENTIFICACIÓN deberá seleccionar el reactivo que desea eliminar.

Aparecerá la pantalla de CONSULTA DE REACTIVOS TIPO LOCALIZACIÓN / IDENTIFICACIÓN

- Haga clic en



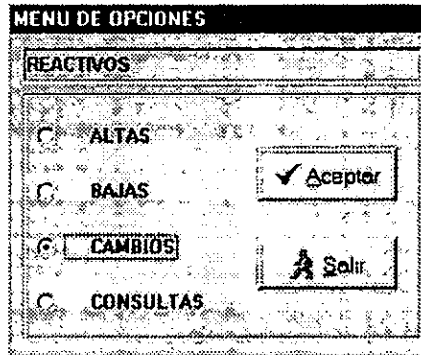
- Los pasos siguientes para eliminar o dar de baja este reactivo fueron descritos en la parte de **Como Eliminar un Reactivo**.

❖ Como Modificar un Reactivo

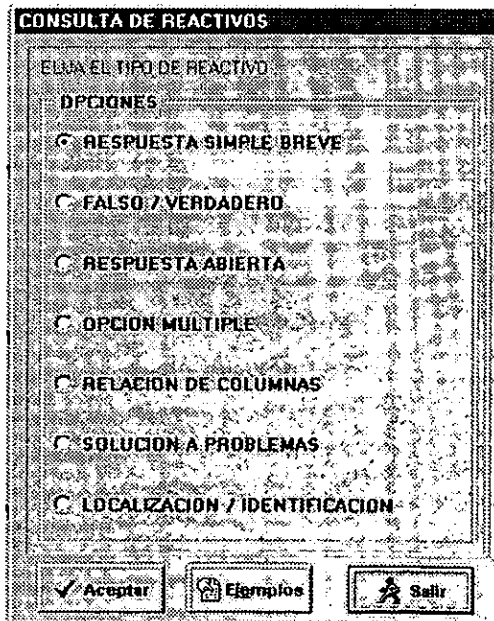
➤ Seleccione en el MENÚ DE OPCIONES el botón de opción de Cambios haga clic



. Aparecerá en la pantalla MENÚ REACTIVOS



➤ Aparecerá la pantalla de MENÚ DE REACTIVOS, elija tipo de reactivo que desea modificar y haga clic en





Seleccionado el TIPO DE REACTIVO. Aparecerá la pantalla de CONSULTA DE REACTIVOS

PARA REALIZAR UNA CONSULTA, BAJAR Y MODIFICACION DE REACTIVOS ES NECESARIO BUSCAR PRIMERO LOS CUERPOS DE DATOS ESPECIFICANDO LAS CARACTERISTICAS PRINCIPALES DE ESTOS

Asignatura

CLAVE:

NOMBRE:

Opciones:

Elegir las Opciones para Consulta

Por Unidad de Disciplina Por Fecha de Examen

Por Unidad del Examen Por Tipo de Examen

Consultar

Para comenzar Teclée la Clave de la Asignatura a la que pertenece el Reactivo y haga Click en Aceptar

Seleccione el botón que desea y haga Click para Comenzar Sabado 22 de Mayo de 1999 23:21




Escriba la clave de la Asignatura del reactivo que desea Modificar o cambiar y con la tecla tabulador muevase entre las opciones para consultar y seleccione una con la barra espaciadora o haga click con el mouse en





Aparecerá en orden de registros los reactivos que existen en la Base de Datos sobre esa Asignatura.

⇒ **EJEMPLO:** Si su selección es Respuesta Simple o Breve

Aparecerá la pantalla de CONSULTA DE REACTIVOS TIPO RESPUESTA SIMPLE O BREVE.

Haga clic en  y el sistema le permitirá modificar los datos que usted desee.

Realizados los cambios verá que esta  habilitado el boton de .

Haga click  en este boton y aparecera el mensaje

Si responde Yes aparecera el sig. Mensaje

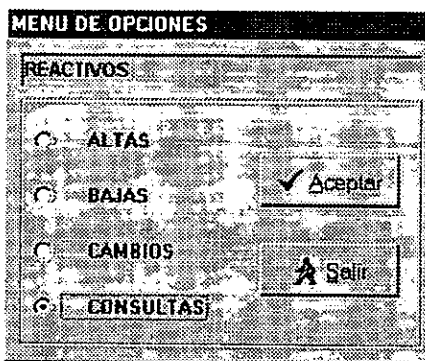


❖ Como Consultar un Reactivo

➤ Seleccione en el MENÚ DE OPCIONES el botón de opción de Consultas haga clic



.Aparecerá en la pantalla MENÚ REACTIVOS

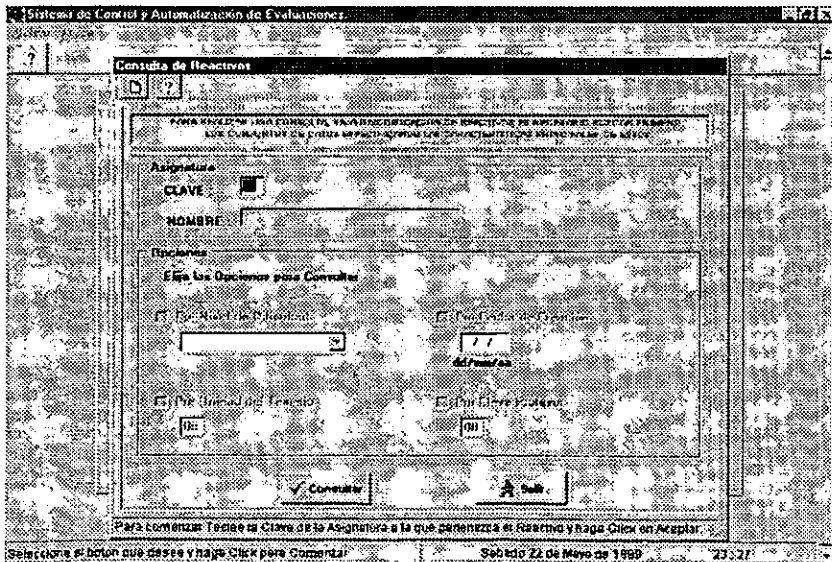


➤ Aparecerá la pantalla de MENÚ DE REACTIVOS, elija tipo de reactivo que desea consultar y haga clic en





Seleccionado el TIPO DE REACTIVO. Aparecerá la pantalla de CONSULTA DE REACTIVOS



- ⇒ Escriba la clave de la Asignatura del reactivo que desea Modificar o cambiar y con la tecla tabulador muevase entre las opciones para consultar y seleccione una con la barra espaciadora o haga click con el mouse en



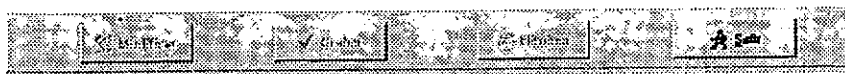
- ⇒ Aparecerá en orden de registros los reactivos que existen en la Base de Datos sobre esa Asignatura.

⇒ **EJEMPLO:** Si su selección es Respuesta Localización / Identificación

Aparecerá la pantalla de CONSULTA DE REACTIVOS TIPO RESPUESTA LOCALIZACIÓN / IDENTIFICACIÓN



Los botones de opcion estarán inhabilitados



Por lo tanto solo puede consultar los diferentes reactivos existentes.



3 Asignaturas

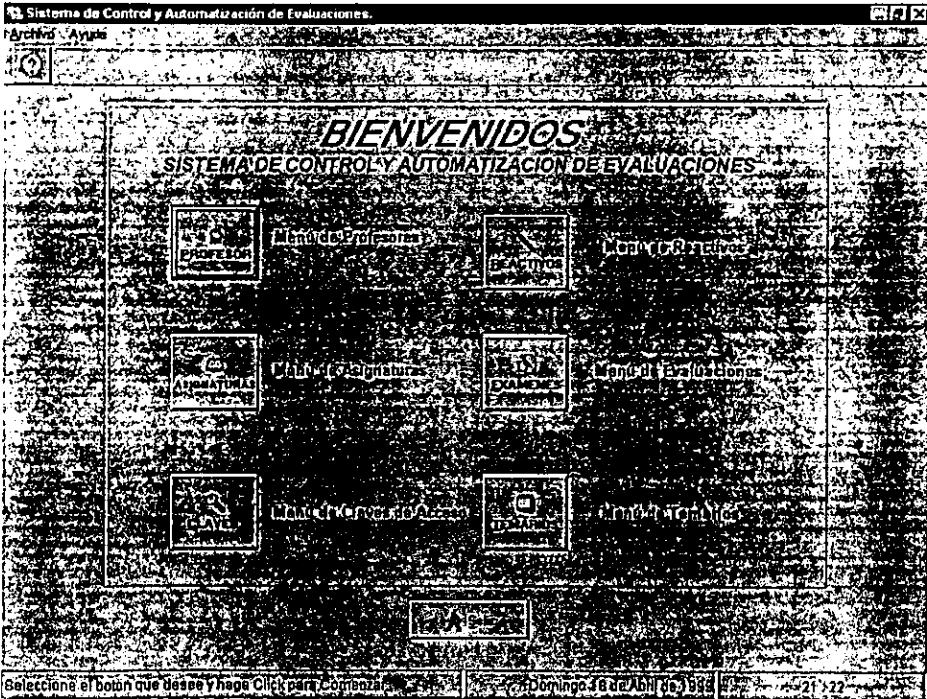
En este capítulo

Los temas en este capítulo son:

- ❖ Como Agregar una Asignatura
- ❖ Como Eliminar una Asignatura
- ❖ Como Modificar los datos de una Asignatura
- ❖ Como Consultar los datos de una Asignatura

❖ Acceso a Asignaturas:

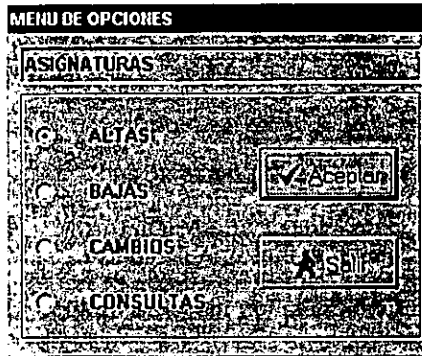
⇒ En el Menú Principal.
Aparecerá la pantalla MENÚ DE ASIGNATURAS.



Para obtener acceso a las Asignaturas,
Haga clic en:



Usted podrá seleccionar un botón de opción de diferentes movimientos que se pueden hacer con los Profesores. MENÚ DE OPCIONES

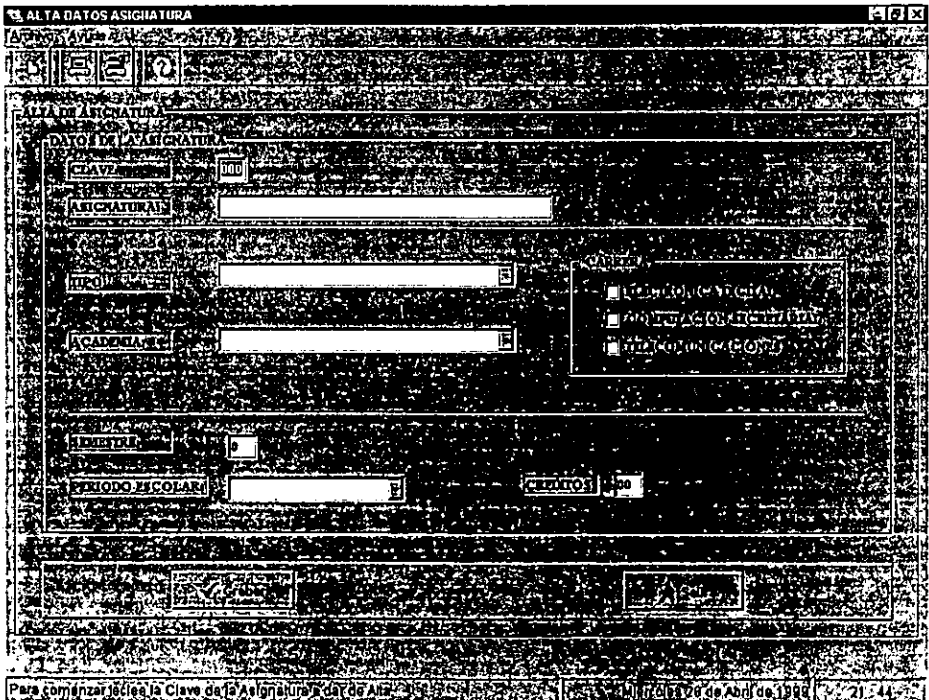



❖ **Como Agregar una Asignatura:**

➤ Seleccione el botón de opción de Altas con el mouse, haga clic en



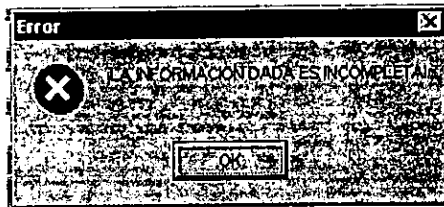
Aparecerá en la pantalla ALTA DATOS ASIGNATURAS



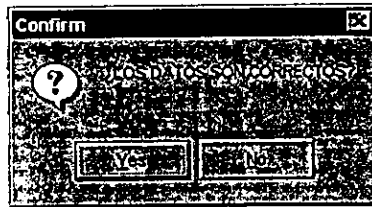
1. Recordar la clave de la nueva alta de la Asignatura, para posteriores consultas.
2. Escribir o llenar los cuadros de texto con los datos completos de la asignatura, y oprima la tecla enter u haga clic en 
3. Es necesario asegurarse anteriormente (en consultas) de la clave de la asignatura, para evitar el siguiente mensaje:



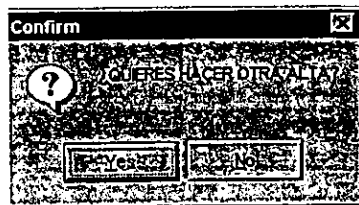
4. Es necesario que se den los datos completos para evitar el siguiente mensaje, en el cual haga click en OK.



4. Después de Grabar. Aparecerá el siguiente mensaje de confirmación:




5. Y también un mensaje para continuar con otra alta de datos de un Profesor.




6. Finalmente, si la respuesta es NO. Entonces se regresará al menú principal.

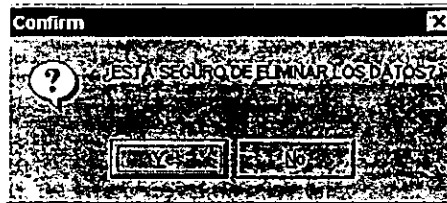
❖ Como Eliminar una Asignatura.

➤ En el Menú de Opciones, seleccione el botón de opción de **Bajas** con el mouse, y haga clic en 

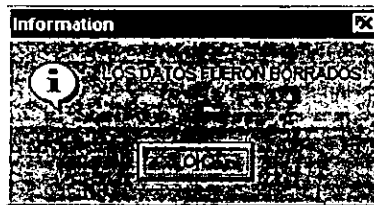
Aparecerá en la pantalla BAJAS DATOS ASIGNATURA.

1. Recordando la clave que se dio la alta de la Asignatura. Es necesario dar la clave correctamente, para evitar un mensaje de error.
2. Cuando se escriba la clave correcta, oprima enter, y se aparecerán los datos de la Asignatura que se desean eliminar. Hagá clic en el botón 

3. Para completar la Eliminación de los datos del Profesor, aparecerá el siguiente mensaje de confirmación:



4. Finalmente, si la respuesta fue Yes, aparecerá el último mensaje. Haga click en OK. Y se regresará al menú principal.

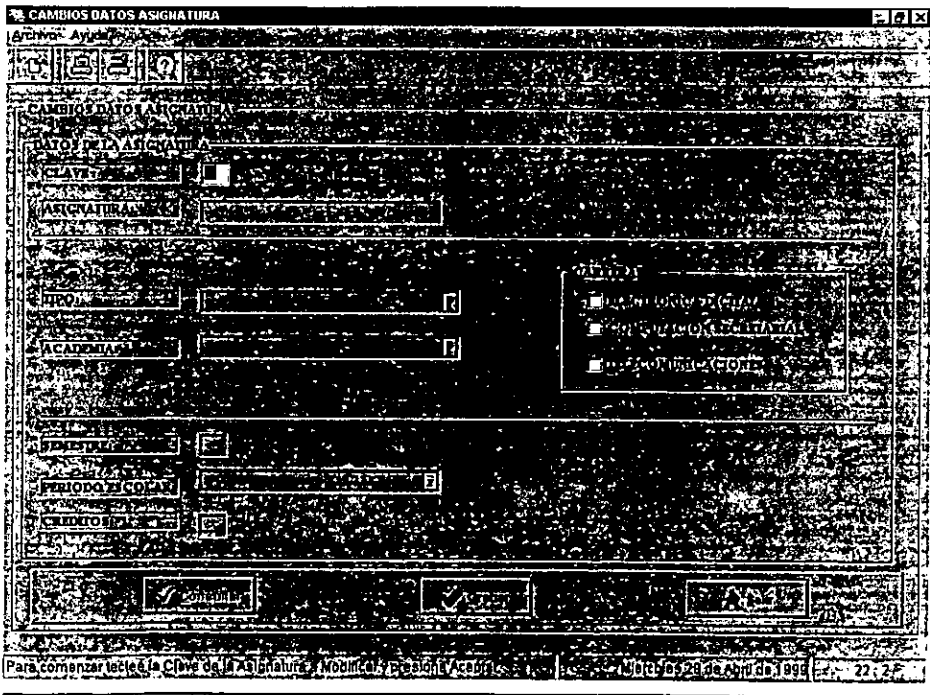


❖ **Como Modificar los datos de una Asignatura.**

Seleccione en el MENÚ DE OPCIONES el botón de opción de Cambios haga clic



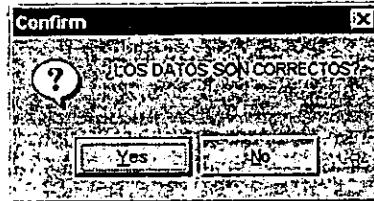
. Aparecerá en la pantalla CAMBIOS DATOS ASIGNATURA.



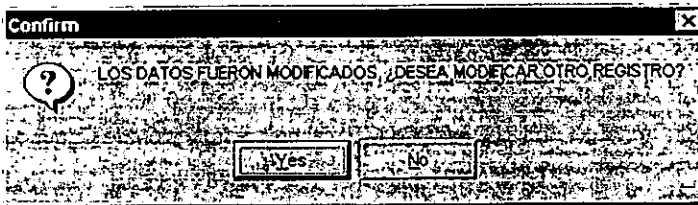
1. Escribir la clave de la Asignatura a modificar y haga click en
2. Modifique los datos y haga click en



3. Después preguntará si los datos son correctos y confirmará con el siguiente mensaje



4. Si se deseará modificar otro registro, se regresará a la pantalla de cambios datos asignatura y pedirá nuevamente una clave, etc.



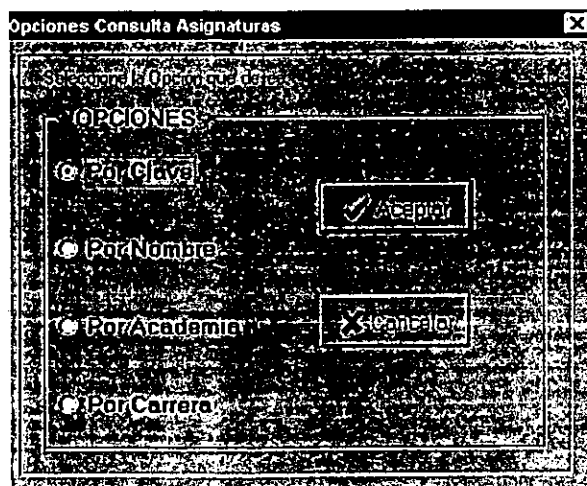
❖ Como Consultar los datos de una Asignatura.



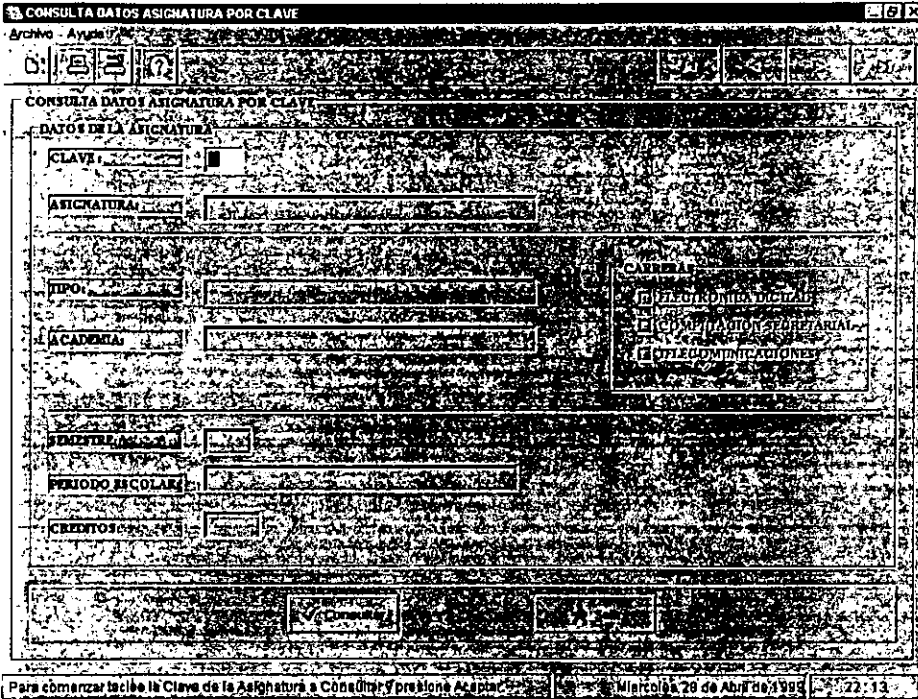
Seleccione en el MENÚ DE OPCIONES el botón de opción de Consultas haga clic



Aparecerá en la pantalla OPCIONES CONSULTA ASIGNATURAS



➡ 1) Seleccionado la consulta de asignatura por clave. Aparecerá la pantalla de CONSULTA DATOS ASIGNATURA POR CLAVE y haga click en



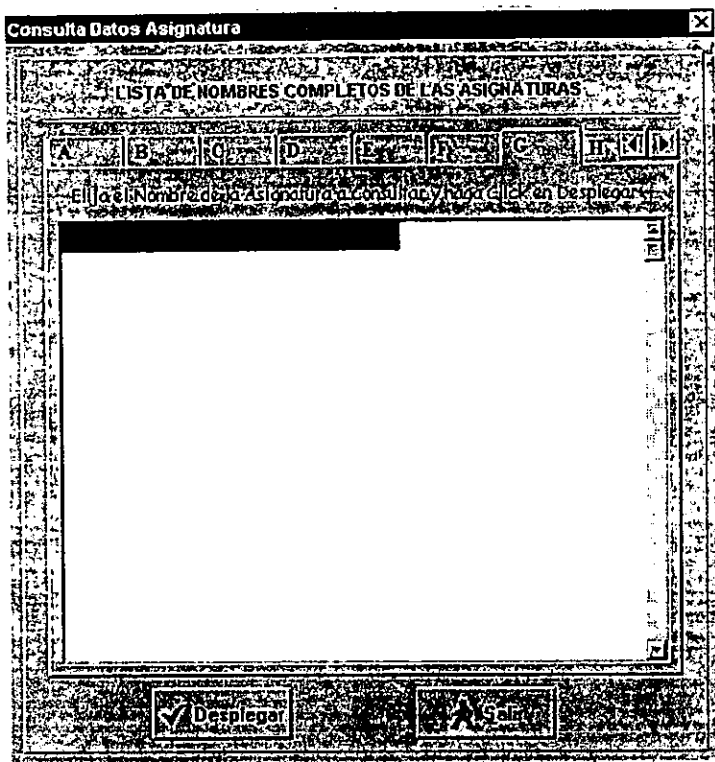
2. Puede usar también las flechas de la esquina superior derecha, para moverse entre otros datos de las asignaturas.




2) Seleccionado la consulta de profesor por nombre. Aparecerá la pantalla de

➡ CONSULTA ASIGNATURA POR NOMBRE y haga click en





1. Seleccionar la lengüeta correspondiente a la primera letra de la asignatura, elija el Nombre de la asignatura a consultar y haga click en 

2. Aparecerá la asignatura elegida por nombre en la pantalla Consulta asignatura por nombre.

CONSULTA DATOS ASIGNATURA POR NOMBRE

Archivos

CONSULTA DATOS ASIGNATURA POR NOMBRE

DATOS DE LA ASIGNATURA

CLAVE DE LA ASIGNATURA:

ASIGNATURA:

TIPO DE ASIGNATURA:

ACADEMIA:

CATEGORIA DE ASIGNATURA:

- ELECTRONICA INICIAL
- INGENIERIA EN SISTEMAS
- COMPUTACION SECRETARIAL
- TELECOMUNICACIONES

SEMESTRE:

PERIODO ESCOLAR:

CREDITOS:

Para comenzar teclee la Clave de la Asignatura y Presione Aceptar

Miércoles 28 de Abril de 1999 12:22:22

3. Puede usar también las flechas de la esquina superior derecha, para moverse entre otros datos de las asignaturas.

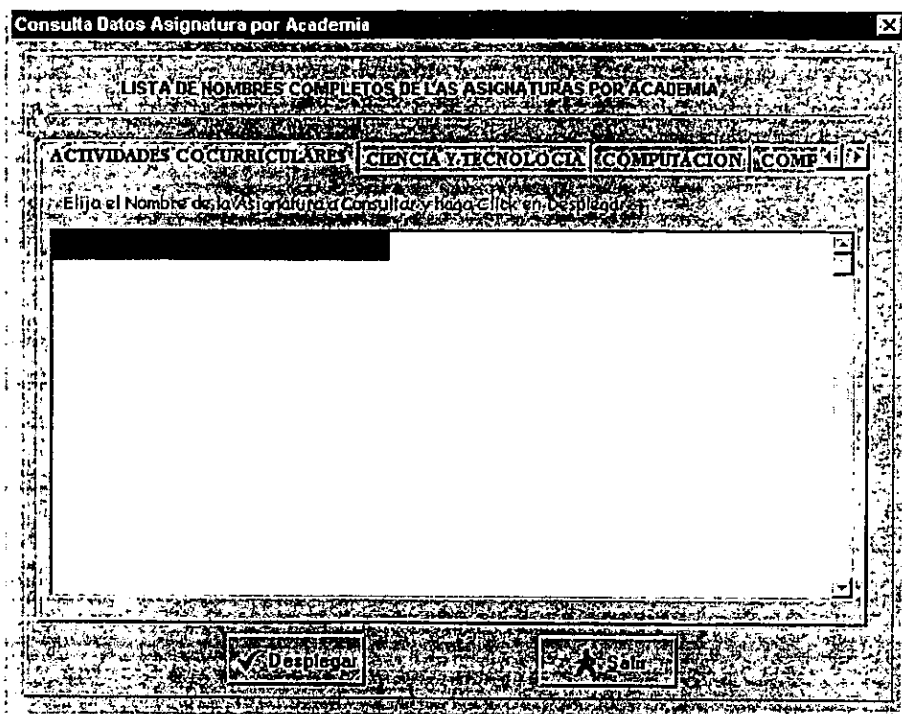


4. Finalmente, haga click en



- 3) Seleccionado la consulta de Asignatura por academia. Aparecerá la pantalla de CONSULTA ASIGNATURA POR ACADEMIA y haga click en



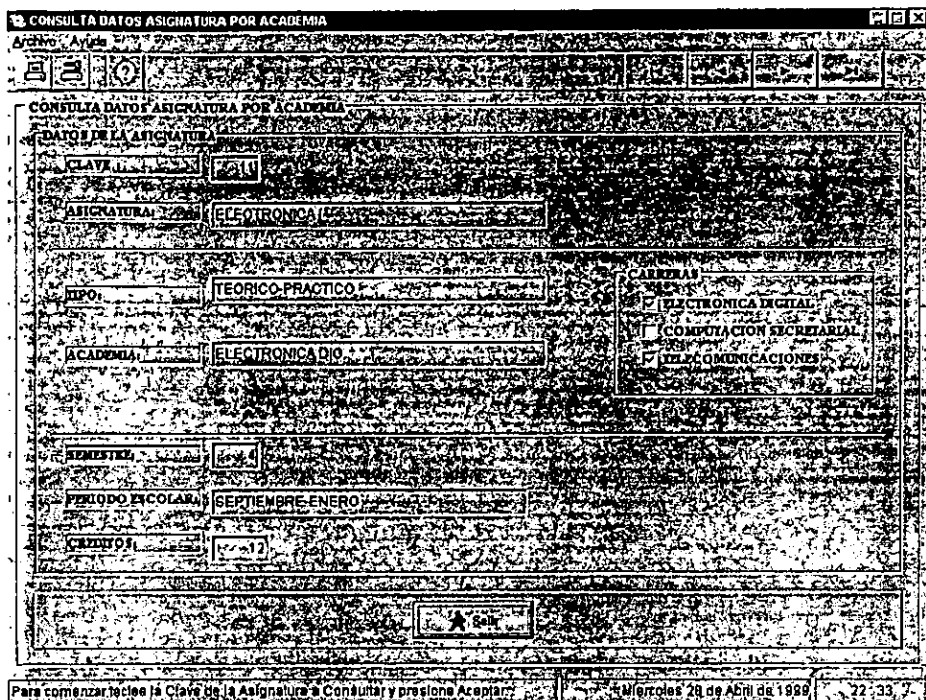


1. Seleccionar la lengüeta correspondiente a la academia de la asignatura, elija el

Nombre de la asignatura a consultar y haga click en



2. Aparecerá la asignatura elegida por nombre en la pantalla Consulta asignatura por academia.



3. Puede usar también las flechas de la esquina superior derecha, para moverse entre otros datos de las asignaturas.

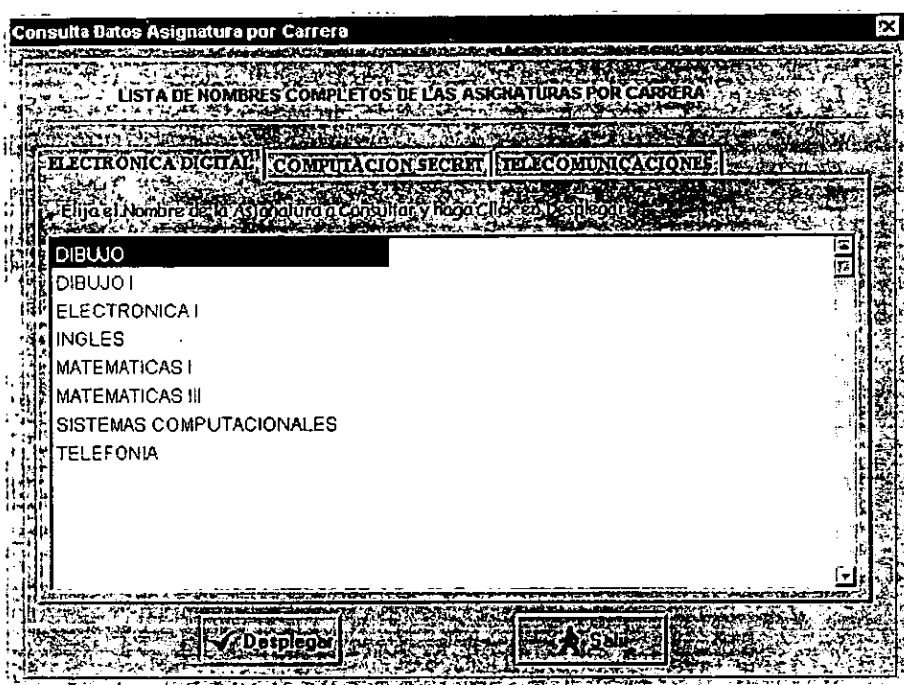


4. Finalmente, haga click en



3) Seleccionado la consulta de Asignatura por carrera. Aparecerá la pantalla de CONSULTA ASIGNATURA POR CARRERA y haga click en



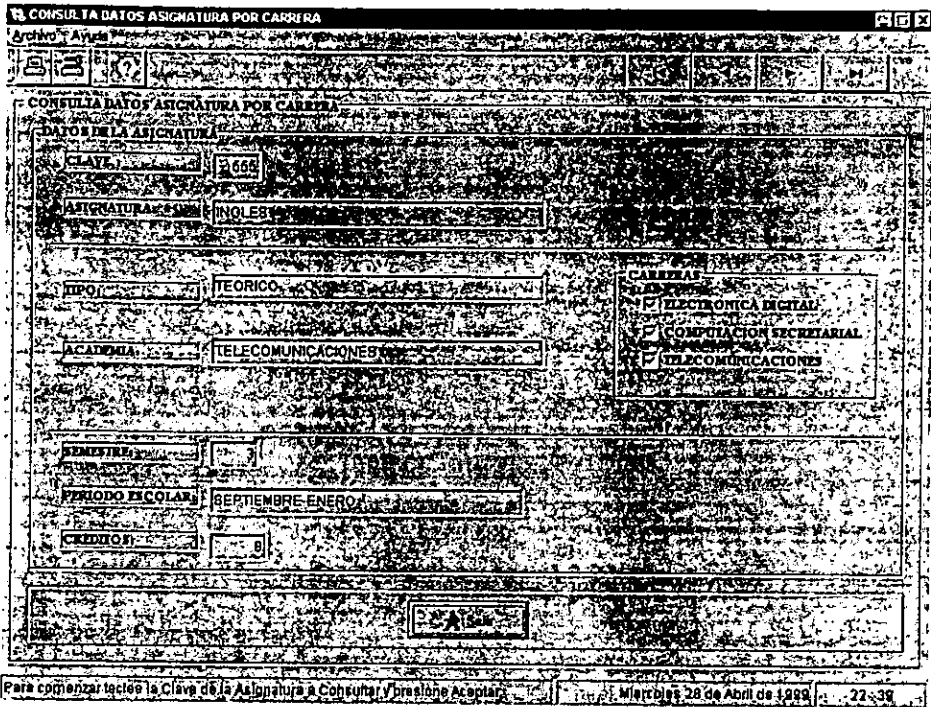


1. Seleccionar la lengüeta correspondiente a la academia de la carrera, elija el

Nombre de la asignatura a consultar y haga click en



2. Aparecerá la asignatura elegida por nombre en la pantalla Consulta asignatura por carrera.



- 3. Puede usar también las flechas de la esquina superior derecha, para moverse entre otros datos de las asignaturas.



- 4. Finalmente, haga click en





4 Exámenes

En este capítulo

Los temas en este capítulo son:

- ❖ Exámenes
- ❖ Partes de la ventana de Exámenes
- ❖ Como Agregar un Examen

❖ **Exámenes**

Esta parte del Sistema, es la más importante ya que aquí daremos forma y cuerpo a la elaboración del examen.

Partes de la Ventana de Exámenes.

Pantalla con solo Texto (Sin imagen).

Barra de Titulo

Tipo de movimiento

Clave del Profesor

Clave de la Asignatura

Fecha de aplicación del Exámen

Datos para aplicación del exámen

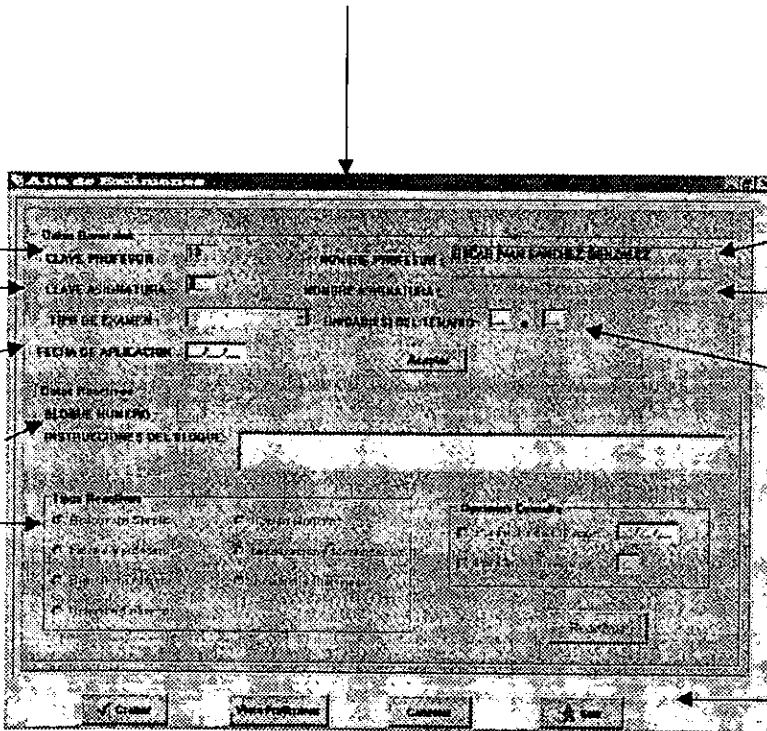
Tipos de Exámenes

Nombre del Profesor

Nombre de la Asignatura

Unidades del Temario a examinar

Botones d comando



Acceso a Exámenes



En el MENÚ PRINCIPAL

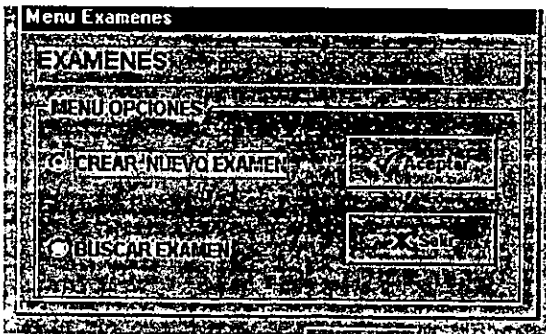


Para obtener acceso a las asignaturas, haga click en



En el Menú Principal.

Aparecerá la pantalla MENÚ DE EXAMENES



Usted podrá seleccionar un botón de opción de diferentes movimientos que se pueden hacer con los Exámenes.

Como crear un examen

⇒ Seleccione el botón de Crear Exámen con el mouse, haga click en



Aparecerá en la pantalla ALTAS DE EXAMENES

La imagen muestra una ventana de software titulada "Alta de Exámenes". La interfaz está organizada en secciones:

- Datos Generales:** Incluye campos para "CLAVE PROFESOR" (con el valor "19"), "NOMBRE PROFESOR" (con el valor "OSCAR WAR SANCHEZ BONZALEZ"), "CLAVE ASIGNATURA" (con el valor "2"), "NOMBRE ASIGNATURA", "TIPO DE EXAMEN" (con un menú desplegable), "UNIDADES DEL TEMARIO" (con un menú desplegable), y "FECHA DE APLICACION" (con un campo de fecha).
- Datos Reactivos:** Incluye "BLOQUE NUMERO" y "INSTRUCCIONES DEL BLOQUE" (con un área de texto).
- Tipos Reactivos:** Una lista de opciones con radio buttons: "O Respuesta Simple", "O Opción Múltiple", "O Falso y Verdadero", "O Llenado de espacios", "O Selección de imágenes", "O Selección de palabras", "O Selección de palabras", "O Selección de palabras", "O Selección de palabras".
- Opciones Consulta:** Incluye "Opción Fecha de Creación" (con un campo de fecha) y "Opción Clave Examen" (con un campo de texto).

En la parte inferior de la ventana, hay una barra de botones con: "Aceptar", "Volver Profesor", "Cancelar", y "Salir".

Dependiendo de la clave del Profesor, aparecerá en Datos Generales su clave de acceso y nombre.

1.- Escriba el número de la Clave de Asignatura en

CLAVE ASIGNATURA:

Y oprima la tecla Tabulación

- 2.- Con el mouse haga clic en la flecha para especificar el tipo de examen
Elija la opción deseada y Oprima la tecla Tabulador

Un campo de texto con el título "TIPO DE EXAMEN" y una flecha de selección a la derecha.

- 3.- Escriba el rango de Unidades dentro de las cuales se aplicará el examen

Un campo de texto con el título "UNIDAD(ES) DEL TEMARIO:" y un icono de selección a la derecha.

Y Oprima la tecla Tabulador

- 4.- Escriba la fecha con la que desea que aparezca en el examen

Un campo de texto con el título "FECHA DE APLICACION" y un icono de calendario a la derecha.

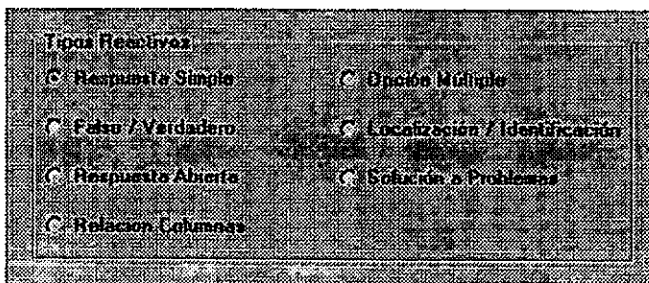
Y Oprima la tecla Aceptar

Un botón rectangular con el texto "Aceptar" en el centro.

- 5.- Se desactivara el Bloque de Datos Generales y se activara el de Datos Reactivos, aparecerá automáticamente el número de bloque, escriba las instrucciones que aparecerán por Bloque:

Un campo de texto con el título "INSTRUCCIONES DEL BLOQUE" y un área de entrada de texto a la derecha.

- 6.- Seleccione el tipo de pregunta

Un formulario con el título "Tipos Reactivos:" y una lista de opciones con botones de radio: Respuesta Simple, Respuesta Múltiple, Falso / Verdadero, Localización / Identificación, Respuesta Abierta, Solución a Problemas, y Relacion Columnas.

Y oprima la tecla Tabulador

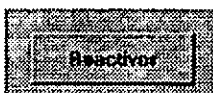
7.- Seleccione con el mouse el tipo de búsqueda que hará para encontrar los reactivos que contiene la base de datos y anotará el dato que se le pide dependiendo de la opción escogida.

Opciones Consulta

Por Fecha de Creación

Por Clave Profesor

Y oprima la tecla



8.- Aparecerá en la pantalla los reactivos existentes con los datos de la búsqueda, seleccionaremos el reactivo con el botón de Seleccionar.

CONSULTA DE REACTIVOS TAO HEMISFERIO SUR

RESPUESTA ABIERTA
EL RESPUESTA NIM: []

DATOS GENERALES

CLAVE ALICATORIA: []

FECHA DE CREACION / MODIFICACION: 10/19

UNIDAD: []

NIVEL DE DIFICULTAD

EXAMEN DIAGNOSTICO EXAMEN EXTRAORDINARIO

EXAMEN PARCIAL

MARCA PROFESOR: []

REACTIVO

TEXTO DEL REACTIVO:

DEFINA QUE ES UN SISTEMA OPERATIVO

BIBLIOGRAFIA (OPCIONAL)

AUTOR: ALICIA GONZALEZ

EDITORIAL: MCGRAW HILL

TITULO: SISTEMAS OPERATIVOS

Seleccionar Cancelar Aceptar

Para obtener más información consulte el manual de usuario que viene con el sistema

Fecha: 30 de Noviembre de 1998 72 / 51 Ptas

- 9.- Aparecerá el mensaje de Reactivo Seleccionado
- 10.- Nuestro exámen quedara listo para agregarle un nuevo bloque de preguntas.



- 11.- Si deseamos ver como va quedando nuestro examen solo presionamos el botón



- 12.- Aparecerá la imagen de cómo va quedando nuestro exámen

Centro de Estudios Tecnológicos Industrial y de Servicios No.152

EXAMEN FINAL DE SISTEMAS COMPUTACIONALES

EXAMEN FINAL DE SISTEMAS COMPUTACIONALES

FECHA APLICACION 12/12/98

NOMBRE DEL ALUMNO: _____ NUMERO ACIERTOS _____

NOMBRE PROFESOR: ALICIA GONZALEZ OLVERA CALIFICACION _____

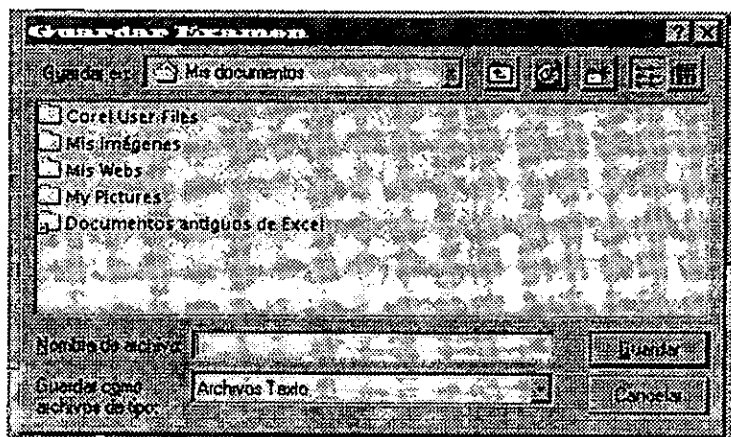
1 INSTRUCCIONES: Conteste correctamente las siguientes preguntas. (VALOR 30 PTOS)

1 DEFINA QUE ES UN SISTEMA OPERATIVO.

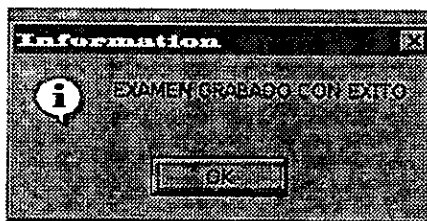
3

- 13.- Cuando deseemos salvar nuestro exámen seleccionaremos el botón Grabar

14.- Nos pedirá la ruta en donde deseamos salvar



15.- Una vez seleccionado la carpeta donde se salvara el examen como archivo , damos el nombre y extensión con el que deseamos guardar el examen, hacemos clic con el mouse en la opción de Guardar y aparecerá el botón.





5 Claves de Acceso

En este capítulo

Los temas en este capítulo son:

- ❖ Claves de Acceso
- ❖ Partes de la Ventana de Clave
- ❖ Como Agregar una Clave
- ❖ Como Eliminar una Clave
- ❖ Como Modificar una Clave
- ❖ Como Consultar una Clave

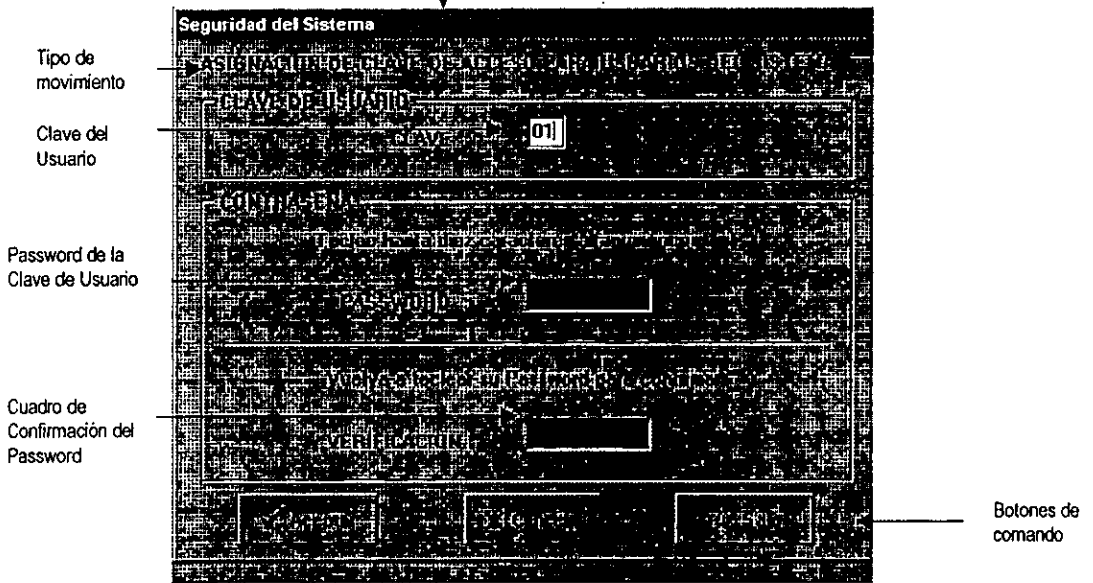
❖ **Claves de Acceso**

Para tener Acceso al sistema es necesario contar con una Clave, esta clave nos identifica como usuarios y nos permite los accesos a los menús de acuerdo a nuestros privilegios.

❖ Partes de la ventana de Claves

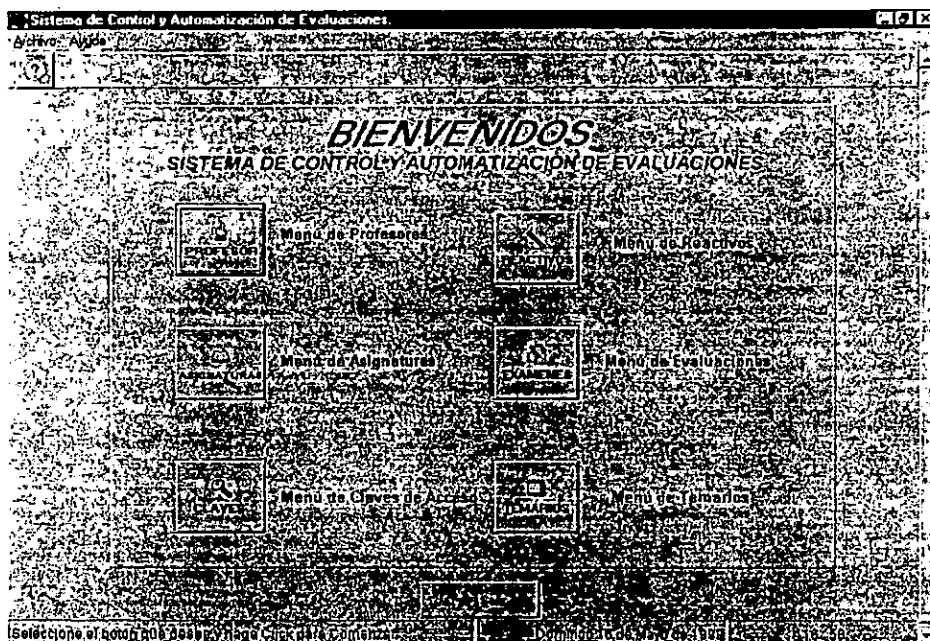
Pantalla con solo Texto (Sin imagen).

Barra de Titulo



Acceso a claves

➡ En el MENÚ PRINCIPAL

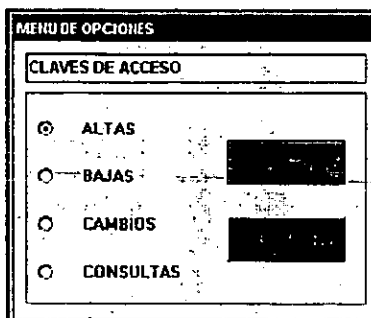


Para obtener acceso a las claves, haga clic en



En el Menú Principal.

Aparecerá en la pantalla MENÚ DE OPCIONES DE CLAVES

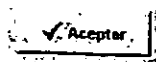


Usted podrá seleccionar un botón de opción de diferentes movimientos que se pueden hacer con las Claves de Acceso.

❖ Como Agregar una clave



Seleccione el botón de opción de Altas con el Mouse y haga clic en



Aparecerá en la pantalla ASIGNACIÓN DE CLAVES

1.- Escriba el número de clave por asignar en

Y oprima la tecla
Tabulador o Enter

2.- Escriba su Password deseado y vuelva escribiirlo para verificarlo en

Y oprima la tecla
Tabulador o enter

3.- Si cree que ha cometido un error al llenarlo oprima en



Y podrá comenzar de nuevo a llenar los datos

4.- Si ya esta seguro de haber llenado correctamente oprima en



Ahora ya tiene su clave asignada

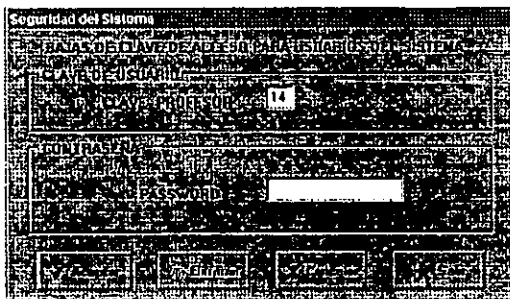
❖ Como Eliminar una clave



Seleccione el botón de opción de Bajas con el Mouse y haga clic en



Aparecerá en la pantalla BAJAS DE CLAVE DE ACCESO CLAVES

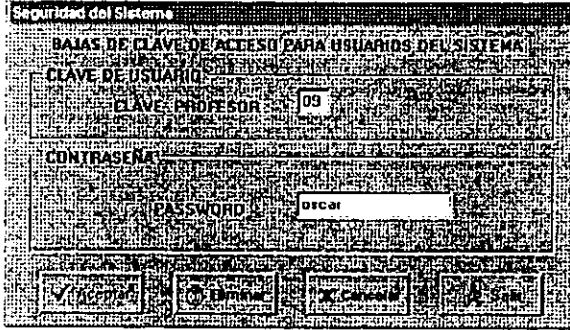


1.- Escriba el número de clave a dar de baja en

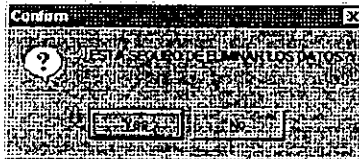


Y oprima la tecla Tabulador o Enter

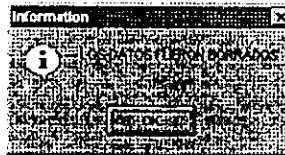
2.- Cuando aparezcan los datos en la pantalla solo debemos hacer click en el botón de Eliminar.



3.- Aparecerá un mensaje de confirmación



4.- Mandará un mensaje de confirmación.



❖ Como Modificar una clave



Seleccione el botón de opción de Cambios con el Mouse y haga clic en



Aparecerá en la pantalla CAMBIO DE CLAVE DE ACCESO

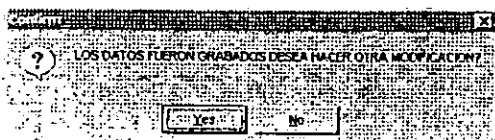
1.-Para poder hacer cambios de Claves de Acceso debemos teclear la Clave del Usuario y el Password del Usuario y hacemos click en



2.- Modificamos la clave de Acceso y presionamos en

3.- Pide confirmación de los datos

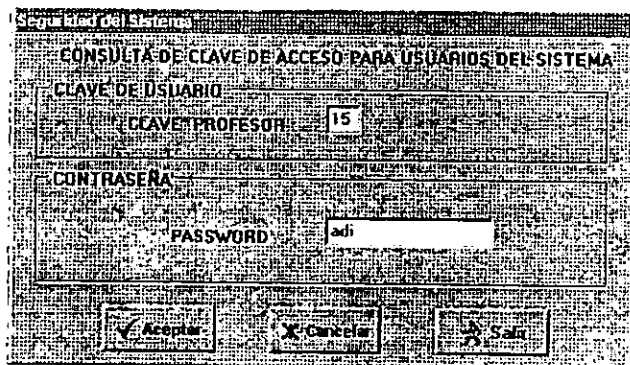
4.- Si respondemos Yes hará los cambios que se le indicaron y preguntará si deseamos hacer más cambios.



❖ Como Consultar una clave



Seleccione el botón de opción de Consultas con el Mouse y haga clic en
Aparecerá en la pantalla CONSULTA DE CLAVES
ACCESO



1.-Escribimos el Número de Profesor y presionamos en Aceptar y se despliega la clave de Password correspondiente.



6 Temarios

En este capítulo

Los temas en este capítulo son:

- ❖ Temario
- ❖ Partes de la ventana de Temarios
- ❖ Como Agregar un Temario
- ❖ Como Eliminar un Temario
- ❖ Como Modificar un Temario
- ❖ Como Consultar un Temario

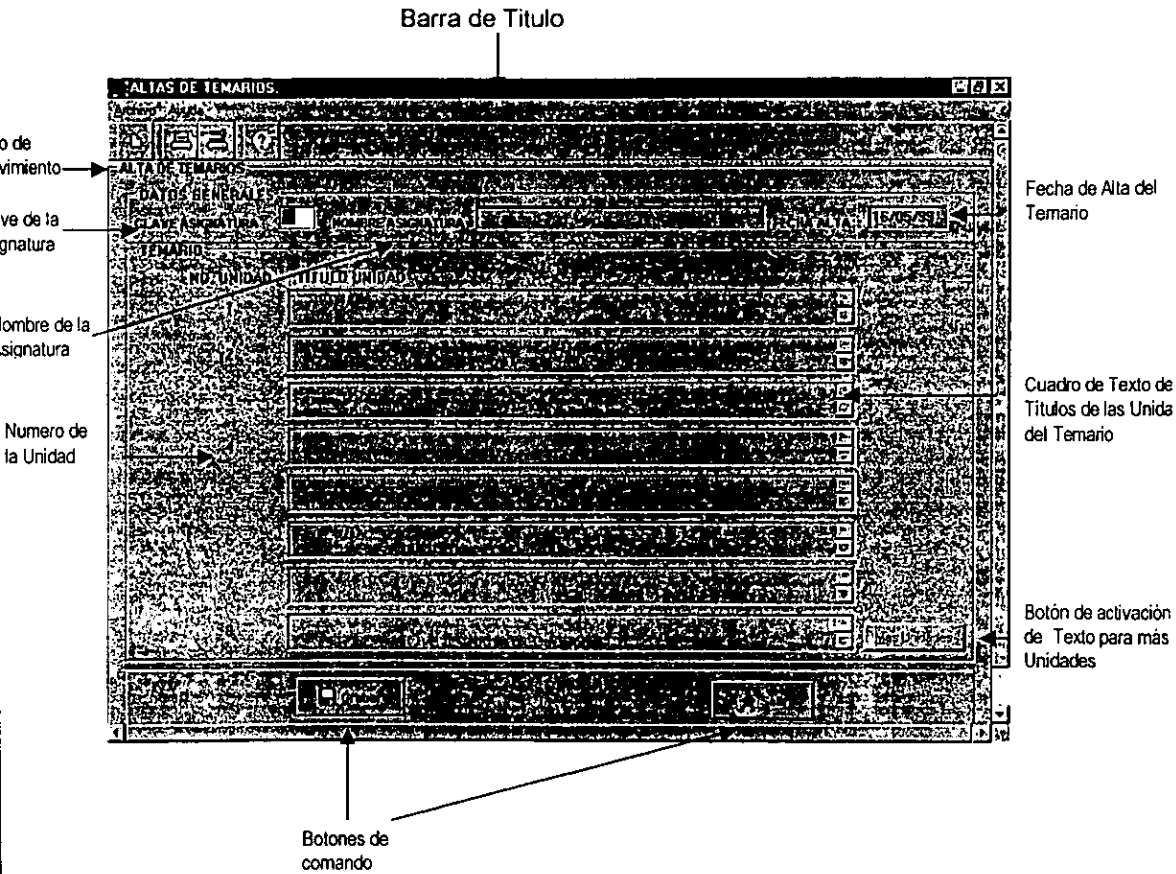
❖ Temarios

La parte de temarios sirve de ayuda a los profesores, ya que aquí se incluyen todos los temas a tratar en el curso que irán a impartir.

Este módulo contiene como los demás altas, bajas, cambios y consultas de Temarios.

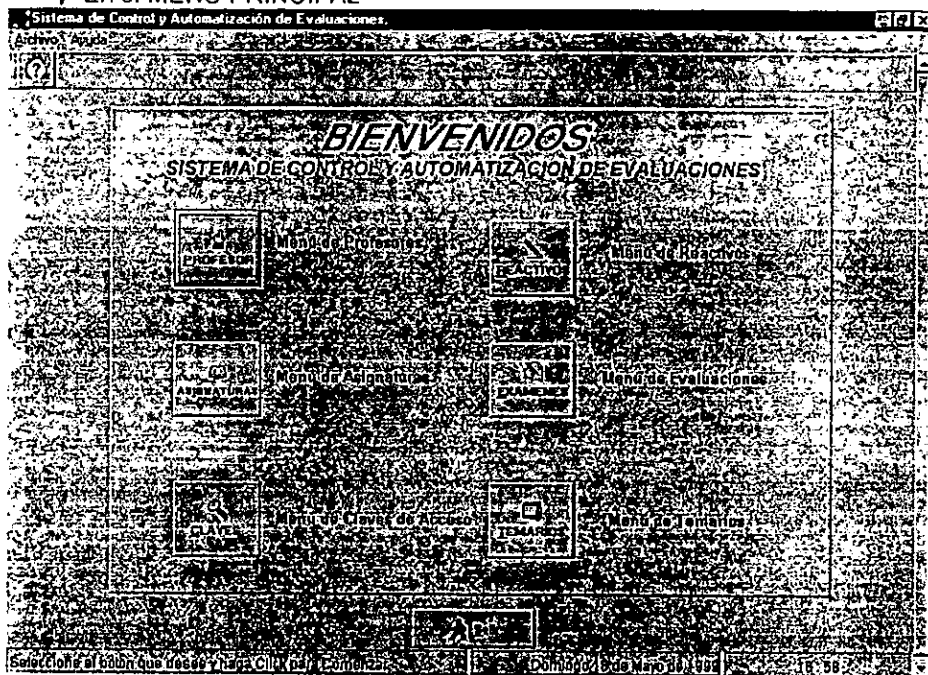
❖ Partes de la ventana de Temarios

La mayoría de las pantallas o ventanas tienen ciertos elementos en común. Tales como una barra de título y una barra de menú. No obstante, no todas las ventanas contienen todos los elementos, y por esto veremos las partes de la Ventana de Profesores.



❖ Acceso a Temarios

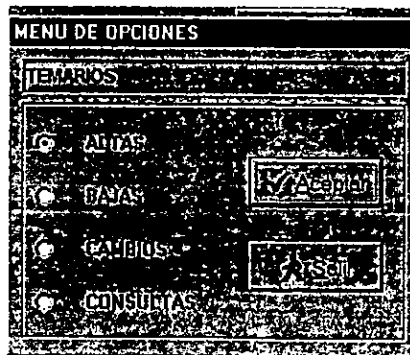
➡ En el MENÚ PRINCIPAL



➡ Para obtener acceso a los Temarios, haga clic en



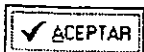
En el Menú Principal.
Aparecerá la pantalla MENÚ OPCIONES TEMARIOS.



Usted podrá seleccionar un botón de opción de diferentes movimientos que se pueden hacer con los Temarios.

❖ Como Agregar un Temario

☞ Seleccione el botón de opción de Altas con el Mouse, haga clic en



Aparecerá en la pantalla ALTA DE TEMARIO

1.- Escriba la clave de la asignatura a la que pertenecerá su reactivo

en

ó presione la tecla Enter.

y haga clic en



2.- El nombre de la Asignatura aparecerá en


HOMBRE ASIGNATURA:

3.- Aparecerá automáticamente la fecha del sistema

en FECHA ALTA :

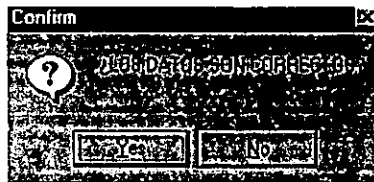
4.- Escriba el Temario de la Asignatura, solamente el Nombre de la Unidad

NO. UNIDAD	TITULO UNIDAD
1	UNIDAD PERC. INTERCULTURALES
2	LINEAS
3	VOLUMEN
4	FIGURAS
5	TRIPULANA
6	
7	
8	

5.- En caso de tener más asignaturas que las mostradas haga click en 

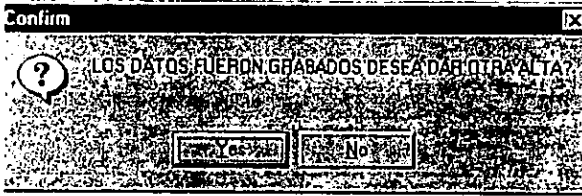
6.- Haga clic en 

7.- Aparecerá el sig. Mensaje



Si responde Yes, aparecerá lo siguiente:

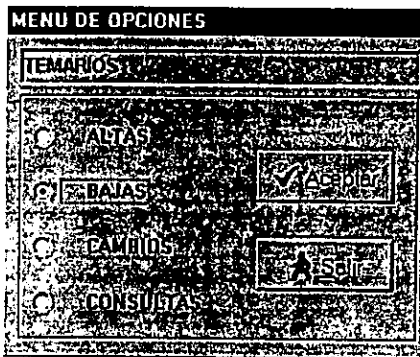
8.- Aparecerá el sig. mensaje



Si responde Yes, aparecerá una nueva pantalla de Alta de Reactivo, Si responde No volverá al Menú Principal.

❖ Como Eliminar un Temario

☞ En la pantalla MENU DE TEMARIO



Seleccione el botón de opción de Bajas con el Mouse, haga clic en



☞ Aparecerá la pantalla de BAJAS DE TEMARIOS

BAJAS DE TEMARIOS

Archivo Ayuda

BAJAS DE TEMARIOS

DATOS GENERALES

CLAVE ASIGNATURA: []

NOMBRE ASIGNATURA: [DIBUJO DE LAS UNIDADES]

FECHA DE ACTUALIZACION: [16/05/99]

TEMARIO

NO. UNIDAD	TITULO UNIDAD
1.	CONCEPTOS INTRODUCTORIOS
2.	LINEAS
3.	VOLUMEN
4.	FORMAS
5.	PERSPECTIVA

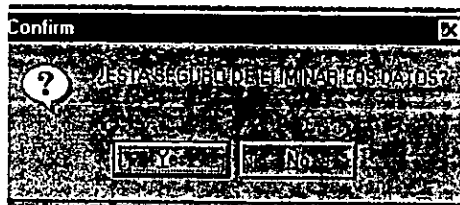
Eliminar Actualizar

1.- Repetimos el 1er Paso de Altas de Temarios

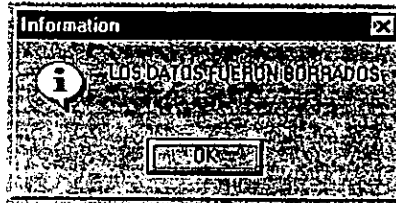
2.- Cuando aparezcan los datos en la pantalla solo debemos hacer click en el botón de Eliminar el cual ya estará habilitado.



3.- Aparecerá un mensaje de confirmación



4.-Si respondió con Yes, Mandará un mensaje de confirmación.

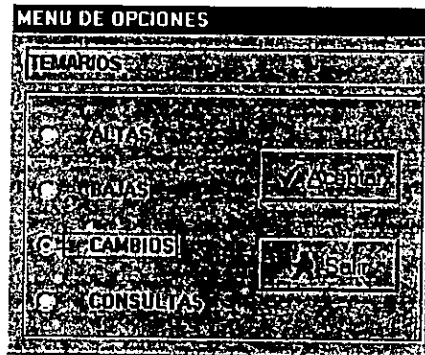


❖ Como Modificar un Temario

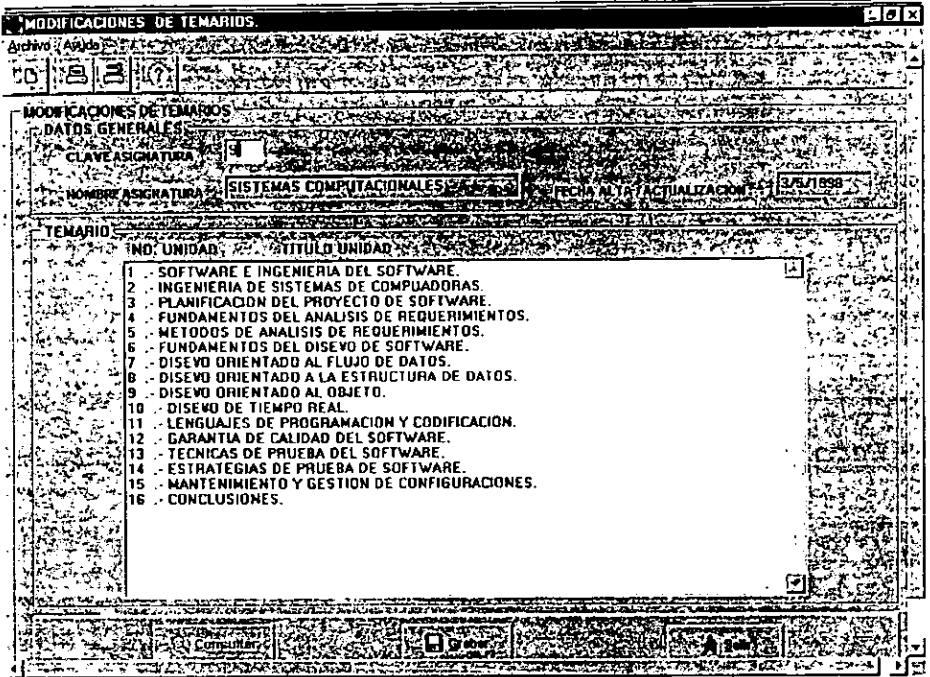
⇒ Seleccione en el MENÚ DE OPCIONES el botón de opción de Cambios



.Aparecerá en la pantalla MENÚ TEMARIOS



⇒ Aparecerá la pantalla de CAMBIOS DE TEMARIOS

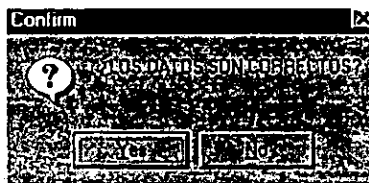


1.- Repetimos el 1er Paso de Altas de Temarios

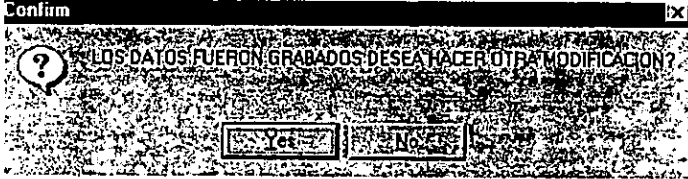
2.- hacemos los cambios correspondientes y presionamos en



3.- Pide confirmación de los datos



4.-Si respondemos Yes hará los cambios que se le indicaron y preguntará si deseamos hacer más cambios.



Si responde Yes, aparecerá una nueva pantalla de Modificación, Si responde No volverá al Menú Principal.

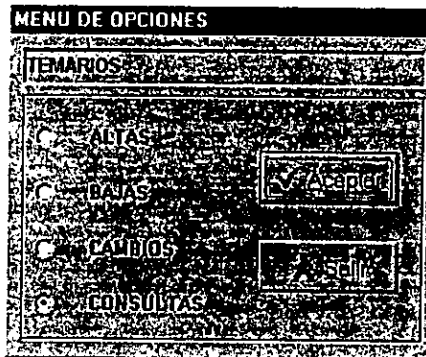
❖ Como Consultar un Temario



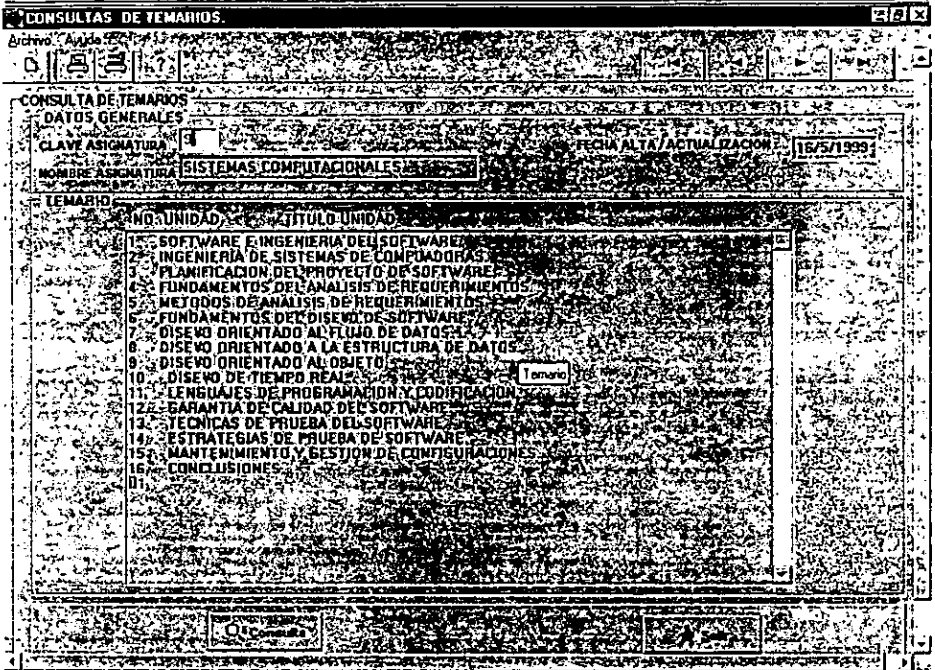
Seleccione en el MENÚ DE OPCIONES el botón de opción de Consultas



.Aparecerá en la pantalla MENÚ DE TEMARIOS



Aparecerá la pantalla de CONSULTAS DE TEMARIOS



1.- Repetimos el 1er Paso de Altas de Temarios, se despliega la información correspondiente

2.- En esta opción solo podemos visualizar la información. Si deseamos otra consulta presionamos en



y aparecerá la pantalla en limpio, lista para hacer otra consulta.



8. 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 más representativos es *Delphi*.

Delphi 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 las instituciones que deben ser debidamente organizadas con el fin de maximizar su valia.

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 *Delphi* 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.

Durante la elaboración de este proyecto, se ratificó la gran importancia que ha adquirido el uso de las redes, ya que nos permite que pueda ser utilizado en un futuro garantizando que tanto los equipos de computo como sus usuarios, compartan y manejen grandes cantidades de información de manera óptima así como evitar la duplicidad en la misma.

De este trabajo se puede concluir que sí se cumplió el objetivo del mismo, que era obtener un sistema de control y automatización de evaluaciones para el CETIS N°152.

La elaboración de la tesis nos dio 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.

CONCLUSIONES

La aplicación resultó ampliamente eficiente y amigable.

Al concluir este trabajo se resuelve la problemática para controlar y automatizar las evaluaciones que actualmente realiza el CETIS N°152 en forma manual, facilitando la tarea a la Administración escolar y al personal docente, así como la reducción de costos por pérdida de tiempo y simplicidad en procesos académicos y administrativos.

La experiencia obtenida en este trabajo es invaluable porque se asumió la responsabilidad del sistema en forma total. Además se pusieron en práctica los conocimientos obtenidos en forma académica y verlos cristalizados en una aplicación que es útil y que está en producción, esto reviste un logro importantísimo.

Así mismo, consideramos que al realizar este sistema cumplimos con nuestros propios objetivos, ya que hemos obtenido reconocimiento y se nos ha ubicado como profesionistas serios, capaces y comprometidos con la sociedad y nuestro presente. Así como confirmar que la Facultad de Ingeniería y la Universidad Nacional Autónoma de México siguen siendo instituciones que le dan a nuestro país personas altamente calificadas.

“ Por mi raza hablará el espíritu ”

Castro Nicanor María Adriana
Espinosa Palmeros Adriana
González Olvera Alicia
Rivera Dueñas Alejandra

unit AltaTemarios;

```

interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs,
  DB, DBTables, ComCtrls, Buttons, ExtCtrls, StdCtrls,
  DBCtrls, Mask;

Type
TFormAltaTemarios = class(TForm)
  StatusBar: TStatusBar;
  GroupBox3: TGroupBox;
  GroupBox4: TGroupBox;
  Label14: TLabel;
  Label15: TLabel;
  Label16: TLabel;
  SpeedAceptar: TSpeedButton;
  EditClaveAsign: TMaskEdit;
  DBEdit1: TDBEdit;
  EditFechaAct: TEdit;
  DataSource2: TDataSource;
  TAsign: TTable;
  CInsertTemario: TQuery;
  Timer1: TTimer;
  TTemarios: TTable;
  GroupBox5: TGroupBox;
  LabelUnidad1: TLabel;
  LabelUnidad2: TLabel;
  LabelUnidad3: TLabel;
  LabelUnidad4: TLabel;
  Label13: TLabel;
  Label14: TLabel;
  LabelUnidad5: TLabel;
  LabelUnidad6: TLabel;
  LabelUnidad7: TLabel;
  LabelUnidad8: TLabel;
  SpeedAgrega: TSpeedButton;
  MemoUnidad1: TMemo;
  MemoUnidad2: TMemo;
  MemoUnidad3: TMemo;
  MemoUnidad4: TMemo;
  MemoUnidad5: TMemo;
  MemoUnidad6: TMemo;
  MemoUnidad7: TMemo;
  MemoUnidad8: TMemo;
  MemoTemario: TMemo;
  Panel2: TPanel;
  SpeedCancelar: TSpeedButton;
  SpeedGrabar: TSpeedButton;
  BitSalir: TBitBtn;
  procedure SpeedAceptarClick(Sender: TObject);
  procedure FormActivate(Sender: TObject);
  procedure DeshabilitaGrales;
  procedure HabilitaGrales;
  procedure BitSalirClick(Sender: TObject);
  procedure SpeedGrabarClick(Sender: TObject);
  procedure ArmaTextoTemario;
  procedure MuestraHint(Sender: TObject);
  procedure SpeedCancelarClick(Sender: TObject);
  procedure MemoUnidad1KeyPress(Sender: TObject); var
  Key: Char;
  procedure MemoUnidad2KeyPress(Sender: TObject); var
  Key: Char;
  procedure MemoUnidad3KeyPress(Sender: TObject); var
  Key: Char;
  procedure MemoUnidad4KeyPress(Sender: TObject); var
  Key: Char;
  procedure MemoUnidad5KeyPress(Sender: TObject); var
  Key: Char;
  procedure MemoUnidad6KeyPress(Sender: TObject); var
  Key: Char;
  procedure MemoUnidad7KeyPress(Sender: TObject); var
  Key: Char;

```

```

  procedure MemoUnidad8KeyPress(Sender: TObject); var
  Key: Char;
  procedure Timer1Timer(Sender: TObject);
  procedure LimpiarFormato;
  procedure MemoUnidad8Exit(Sender: TObject);
  procedure SpeedAgregaClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  FormAltaTemarios: TFormAltaTemarios;
  NumUnidad: integer;
  Clave: integer;
implementation
  uses UnidadGral;
  [SR *.DMX]

  procedure TFormAltaTemarios.SpeedAceptarClick(Sender:
  TObject);
  begin
    If (EditClaveAsign.Text <> ' ') then
    Begin
      Clave := QuitaEspacios(EditClaveAsign.Text);
      TAsign.Active := True;
      If TAsign.FindKey([Clave]) Then
      Begin
        TTemarios.Active := True;
        If TTemarios.FindKey([Clave]) Then
        Begin
          MessageDlg('ERROR EL TEMARIO
          CORRESPONDIENTE A ' +
          TAsignaturaYa
          EXISTE!'), mtError, [mbOK], 0);
          EditClaveAsign.Text := ' ';
          EditClaveAsign.SetFocus;
        End
        Else
        Begin
          DeshabilitaGrales;
          MemoUnidad1.SetFocus;
        End;
      End
      Else
      Begin
        MessageDlg('ERROR LA CLAVE DE LA
        ASIGNATURA NO EXISTE! ',
        mtError, [mbOK], 0);
        EditClaveAsign.Text := ' ';
        EditClaveAsign.SetFocus;
      End;
    End
    Else
    Begin
      MessageDlg('POR FAVOR TECLEE LA CLAVE DE LA
      ASIGNATURA! ',
      mtError, [mbOK], 0);
      EditClaveAsign.Text := ' ';
      EditClaveAsign.SetFocus;
    End;
  end;

  procedure TFormAltaTemarios.DeshabilitaGrales;
  begin
    MemoUnidad1.Enabled := True;
    MemoUnidad2.Enabled := True;
    MemoUnidad3.Enabled := True;
    MemoUnidad4.Enabled := True;
    MemoUnidad5.Enabled := True;
    MemoUnidad6.Enabled := True;
    MemoUnidad7.Enabled := True;
    MemoUnidad8.Enabled := True;

```

APENDICE A

```

MemoUnidad1.Color := clWindow;
MemoUnidad2.Color := clWindow;
MemoUnidad3.Color := clWindow;
MemoUnidad4.Color := clWindow;
MemoUnidad5.Color := clWindow;
MemoUnidad6.Color := clWindow;
MemoUnidad7.Color := clWindow;
MemoUnidad8.Color := clWindow;

end;

procedure TFormAltaTemarios.HabilitaGrades;
begin
    EditClaveAssign.SetFocus;

    MemoUnidad1.Enabled := False;
    MemoUnidad2.Enabled := False;
    MemoUnidad3.Enabled := False;
    MemoUnidad4.Enabled := False;
    MemoUnidad5.Enabled := False;
    MemoUnidad6.Enabled := False;
    MemoUnidad7.Enabled := False;
    MemoUnidad8.Enabled := False;
    SpeedAgrega.Enabled := False;

    MemoUnidad1.Color := clMenu;
    MemoUnidad2.Color := clMenu;
    MemoUnidad3.Color := clMenu;
    MemoUnidad4.Color := clMenu;
    MemoUnidad5.Color := clMenu;
    MemoUnidad6.Color := clMenu;
    MemoUnidad7.Color := clMenu;
    MemoUnidad8.Color := clMenu;
end;

procedure TFormAltaTemarios.FormActivate(Sender:
TObject);
var
    Horas,Minutos,Sec, MSec : word;
    Hora : string;
begin
    NumUnidad := 1;
    EditFechaAct.Text := DateToStr(Date);
    *** BLOQUE PARA FECHA/HORA ***
    ArmaFecha;
    StatusBar.Panels.Items[1].Text := Fecha;

    DecodeTime(Time, Horas, Minutos, Sec, MSec);
    Timer.Interval := 60000 - (Sec * 1000);
    Hora := IntToStr(Horas) + ':' +
IntToStr(Minutos) + ':' + HrS;
    StatusBar.Panels.Items[2].Text := Hora;
    .....
    Application.OnHint := BuestraHint;
    HabilitaGrades;
    LimpiarFormato;
end;

procedure TFormAltaTemarios.BitSallirClick(Sender:
TObject);
begin
    Close;
end;

procedure TFormAltaTemarios.SpeedGrabarClick(Sender:
TObject);
var
    BandError : integer;
begin
    ArmaTextoTemario;
    BandError := 0;
    Try
    If MessageDlg('LOS DATOS SON
    MUY ALTOS', mtConfirmation,
    [mbYes,mbNo], 0) = mYes then
end;

```

```

QInsertTemario.Prepare;
QInsertTemario.Params[0].AsInteger := Clave;
QInsertTemario.Params[1].AsString :=
EditFechaAct.Text;
QInsertTemario.Params[2].AsMemo :=
MemoTemario.Text;
QInsertTemario.ExecSQL;
end
else
begin
    BandError := 1;
    EditClaveAssign.SetFocus;
end;
except
    MessageDlg('ERROR AL TEMARIO, POR FAVOR
    VERIFIQUE LOS DATOS',
    mtError,[mbOK], 0);
    BandError := 1;
    Raise;
end;
If BandError = 0 then
begin
    If MessageDlg('LOS DATOS FUERON GRABADOS DESE-
    PAR OTRA ALTA?',
    mtConfirmation,[mbYes,mbNo], 0) = mYes then
    begin
        HabilitaGrades;
        LimpiarFormato;
    end
    else
    Close;
end;
end;

procedure TFormAltaTemarios.ArmaTextoTemario;
var
    Unidad : string;
begin
    If (MemoUnidad1.Text <> '') then
    begin
        MemoTemario.Lines.Add(IntToStr(NumUnidad)+'
        '+MemoUnidad1.Text);
    end;
    Inc(NumUnidad);
    If (MemoUnidad2.Text <> '') then
        MemoTemario.Lines.Add(IntToStr(NumUnidad)+'
        '+MemoUnidad2.Text);
    Inc(NumUnidad);
    If (MemoUnidad3.Text <> '') then
        MemoTemario.Lines.Add(IntToStr(NumUnidad)+'
        '+MemoUnidad3.Text);
    Inc(NumUnidad);
    If (MemoUnidad4.Text <> '') then
        MemoTemario.Lines.Add(IntToStr(NumUnidad)+'
        '+MemoUnidad4.Text);
    Inc(NumUnidad);
    If (MemoUnidad5.Text <> '') then
        MemoTemario.Lines.Add(IntToStr(NumUnidad)+'
        '+MemoUnidad5.Text);
    Inc(NumUnidad);
    If (MemoUnidad6.Text <> '') then
        MemoTemario.Lines.Add(IntToStr(NumUnidad)+'
        '+MemoUnidad6.Text);
    Inc(NumUnidad);
    If (MemoUnidad7.Text <> '') then
        MemoTemario.Lines.Add(IntToStr(NumUnidad)+'
        '+MemoUnidad7.Text);
    Inc(NumUnidad);
    If (MemoUnidad8.Text <> '') then
        MemoTemario.Lines.Add(IntToStr(NumUnidad)+'
        '+MemoUnidad8.Text);
    Inc(NumUnidad);
end;

```

APENDICE A

```

Procedure TFormAltaTemarios.MuestraHint(Sender:
TObject);
Begin
  If Length(Application.Hint) > 0 then
  Begin
    StatusBar.SimplePanel := True;
    StatusBar.SimpleText := Application.Hint;
  End
  Else
    StatusBar.SimplePanel := False;
end;

procedure TFormAltaTemarios.SpeedCancelarClick(Sender:
TObject);
begin
  HabilitaGrales;
  LimpiarFormato;
end;

procedure TFormAltaTemarios.MemoUnidad1KeyPress(Sender:
TObject);
  var Key: Char;
begin
  Key:=AnsiUpperCase(Key)[1];
end;

procedure TFormAltaTemarios.MemoUnidad2KeyPress(Sender:
TObject);
  var Key: Char;
begin
  Key:=AnsiUpperCase(Key)[1];
end;

procedure TFormAltaTemarios.MemoUnidad3KeyPress(Sender:
TObject);
  var Key: Char;
begin
  Key:=AnsiUpperCase(Key)[1];
end;

procedure TFormAltaTemarios.MemoUnidad4KeyPress(Sender:
TObject);
  var Key: Char;
begin
  Key:=AnsiUpperCase(Key)[1];
end;

procedure TFormAltaTemarios.MemoUnidad5KeyPress(Sender:
TObject);
  var Key: Char;
begin
  Key:=AnsiUpperCase(Key)[1];
end;

procedure TFormAltaTemarios.MemoUnidad6KeyPress(Sender:
TObject);
  var Key: Char;
begin
  Key:=AnsiUpperCase(Key)[1];
end;

procedure TFormAltaTemarios.MemoUnidad7KeyPress(Sender:
TObject);
  var Key: Char;
begin
  Key:=AnsiUpperCase(Key)[1];
end;

procedure TFormAltaTemarios.MemoUnidad8KeyPress(Sender:
TObject);
  var Key: Char;
begin
  Key:=AnsiUpperCase(Key)[1];
end;

```

```

procedure TFormAltaTemarios.LimpiarFormato;
Begin
  MemoTemario.Lines.Clear;
  MemoUnidad1.Lines.Clear;
  MemoUnidad2.Lines.Clear;
  MemoUnidad3.Lines.Clear;
  MemoUnidad4.Lines.Clear;
  MemoUnidad5.Lines.Clear;
  MemoUnidad6.Lines.Clear;
  MemoUnidad7.Lines.Clear;
  MemoUnidad8.Lines.Clear;

  LabelUnidad1.Caption := '1 -';
  LabelUnidad2.Caption := '2 -';
  LabelUnidad3.Caption := '3 -';
  LabelUnidad4.Caption := '4 -';
  LabelUnidad5.Caption := '5 -';
  LabelUnidad6.Caption := '6 -';
  LabelUnidad7.Caption := '7 -';
  LabelUnidad8.Caption := '8 -';

  EditClaveAssign.Text := '';
End;

procedure TFormAltaTemarios.Timer1Timer(Sender:
TObject);
var
  Horas,Minutos,Sec, HSec : word;
  Hora : string;
begin
  DecodeTime(Time, Horas, Minutos, Sec, HSec);
  Hora := IntToStr(Horas) + ':' + IntToStr(Minutos);
  StatusBar.Panels.Items[2].Text := Hora;
  Timer1.Interval := 60000;
end;

procedure TFormAltaTemarios.MemoUnidad8Exit(Sender:
TObject);
begin
  If (MemoUnidad8.Text <> '') then
  begin
    SpeedAgrega.Enabled := True;
  end;
end;

procedure TFormAltaTemarios.SpeedAgregaClick(Sender:
TObject);
begin
  ArmaTextoTemario;
  MemoUnidad1.Lines.Clear;
  MemoUnidad2.Lines.Clear;
  MemoUnidad3.Lines.Clear;
  MemoUnidad4.Lines.Clear;
  MemoUnidad5.Lines.Clear;
  MemoUnidad6.Lines.Clear;
  MemoUnidad7.Lines.Clear;
  MemoUnidad8.Lines.Clear;

  LabelUnidad1.Caption := IntToStr(NumUnidad);
  LabelUnidad2.Caption := IntToStr(NumUnidad+1);
  LabelUnidad3.Caption := IntToStr(NumUnidad+2);
  LabelUnidad4.Caption := IntToStr(NumUnidad+3);
  LabelUnidad5.Caption := IntToStr(NumUnidad+4);
  LabelUnidad6.Caption := IntToStr(NumUnidad+5);
  LabelUnidad7.Caption := IntToStr(NumUnidad+6);
  LabelUnidad8.Caption := IntToStr(NumUnidad+7);
  MemoUnidad1.SetFocus;
end;
end;

```

APENDICE A

```

unit BajasTemarios;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  Db, DBTables, ComCtrls, Buttons, ExtCtrls,
  StdCtrls, DBCtrls, Mask;

type
  TFormBajasTemarios = class(TForm)
    StatusBar: TStatusBar;
    GroupBox3: TGroupBox;
    GroupBox4: TGroupBox;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    EditClaveAsign: TMaskEdit;
    DBEditNomAsign: TDBEdit;
    DataSource2: TDataSource;
    TTemarios: TTable;
    Timer1: TTimer;
    GroupBox5: TGroupBox;
    Label13: TLabel;
    Label14: TLabel;
    Panel2: TPanel;
    BitSalir: TBitBtn;
    DBMemoTemario: TDBMemo;
    DBEditFecha: TDBEdit;
    SpeedAceptar: TSpeedButton;
    SpeedCancelar: TSpeedButton;
    TAsign: TTable;
    DataSource1: TDataSource;
    BitEliminar: TSpeedButton;
    Instruccion2: TLabel;
    Instruccion1: TLabel;
    QDeleteTemario: TQuery;
    procedure SpeedAceptarClick(Sender:
    TObject);
    procedure FormActivate(Sender: TObject);
    procedure BitSalirClick(Sender: TObject);
    procedure MuestraHint(Sender: TObject);
    procedure SpeedCancelarClick(Sender:
    TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure LimpiarFormato;
    procedure BitEliminarClick(Sender:
    TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FormBajasTemarios: TFormBajasTemarios;
  NumUnidad: integer;
  Clave: integer;
implementation

uses UnidadGral;
($R *.DFM)

procedure
TFormBajasTemarios.SpeedAceptarClick(Sender:
TObject);
begin
  If (EditClaveAsign.Text <> ' ') then
  Begin
    Clave :=
    QuitaEspacios(EditClaveAsign.Text);
    TTemarios.Active := True;
    If TTemarios.FindKey([Clave]) Then
    Begin
      TAsign.Active := True;
      If TAsign.FindKey([Clave])<>
      true Then
        Begin
          MessageDlg('LA ASIGNATURA
          NO EXISTE! ', mtError, [mbOK], 0);
          EditClaveAsign.Text := '
          ';
          EditClaveAsign.SetFocus;
        End
      Else
        Begin
          BitEliminar.Enabled :=
          True;
          SpeedAceptar.Enabled :=
          False;
          Instruccion1.Visible :=
          False;
          Instruccion2.Visible :=
          True;
        End
      End
    End
  Else
    Begin
      MessageDlg('ERROR EL TEMARIO NO
      EXISTE! ',
      mtError, [mbOK], 0);
      EditClaveAsign.Text := '
      ';
      EditClaveAsign.SetFocus;
    End;
  End
End
Else
Begin
  MessageDlg('POR FAVOR TECLEE LA
  CLAVE DE LA ASIGNATURA! ',
  mtError, [mbOK], 0);
  EditClaveAsign.Text := '
  ';
  EditClaveAsign.SetFocus;
End;
end;

procedure
TFormBajasTemarios.FormActivate(Sender:
TObject);
var
  Horas, Minutos, Sec, MSec : word;

```

APENDICE A

```

Hora : string;
begin
  NumUnidad := 1;
  (** BLOQUE PARA FECHA/HORA **)
  ArmaFecha;
  StatusBar.Panels.Items[1].Text := Fecha;

  DecodeTime(Time, Horas, Minutos, Sec,
MSec);
  Timer1.Interval := 60000 - (Sec * 1000);
  Hora := IntToStr(Horas) + ' : ' +
IntToStr(Minutos)+' Hrs';
  StatusBar.Panels.Items[2].Text := Hora;
  (**.....**)
  Application.OnHint := MuestraHint;
end;

procedure
TFormBajasTemarios.BitSalirClick(Sender:
TObject);
begin
  Close;
end;

Procedure
TFormBajasTemarios.MuestraHint(Sender:
TObject);
Begin
  If Length (Application.Hint) > 0 then
  Begin
    StatusBar.SimplePanel := True;
    StatusBar.SimpleText :=
Application.Hint;
  End
  Else
    StatusBar.SimplePanel := False;
end;

procedure
TFormBajasTemarios.SpeedCancelarClick(Sender:
TObject);
begin
  LimpiarFormato;
end;

procedure TFormBajasTemarios.LimpiarFormato;
Begin
  BitEliminar.Enabled := False;
  SpeedAceptar.Enabled := True;
  Instruccion1.Visible := True;
  Instruccion2.Visible := False;

  DBMemoTemario.Lines.Clear;
  DBEditFecha.Clear;
  DBEditNomAssign.Clear;
  EditClaveAssign.Text := '';
  EditClaveAssign.SetFocus;
End;

```

```

procedure
TFormBajasTemarios.Timer1Timer(Sender:
TObject);
var
  Horas,Minutos,Sec, MSec : word;
  Hora : string;
begin
  DecodeTime(Time, Horas, Minutos, Sec,
MSec);
  Hora := IntToStr(Horas) + ' : ' +
IntToStr(Minutos);
  StatusBar.Panels.Items[2].Text := Hora;
  Timer1.Interval := 60000;
end;

procedure
TFormBajasTemarios.BitEliminarClick(Sender:
TObject);
begin
  if MessageDlg('¿ESTA SEGURO DE ELIMINAR LOS
DATOS?',mtConfirmation,
               {mbYes,mbNo},0) = mrYes
then
  begin
    QDeleteTemario.Prepare;
    QDeleteTemario.Params[0].AsInteger :=
QuitaEspacios(EditclaveAssign.Text);
    QDeleteTemario.ExecSQL;
    LimpiarFormato;
    MessageDlg('LOS DATOS FUERON
BORRADOS',mtInformation, {mbOK},0);
  end;
end;

end.

```

APENDICE A

```

unit CambiosTemarios;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  Db, DBTables, ComCtrls, Buttons, ExtCtrls,
  StdCtrls, DBCtrls, Mask;

type
  TFormCambiosTemarios = class(TForm)
    StatusBar: TStatusBar;
    GroupBox3: TGroupBox;
    GroupBox4: TGroupBox;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    EditClaveAsign: TMaskEdit;
    DBEditNomAsign: TDBEdit;
    DataSource2: TDataSource;
    Temarios: TTable;
    Timer1: TTimer;
    GroupBox5: TGroupBox;
    Label13: TLabel;
    Label14: TLabel;
    Panel2: TPanel;
    BitSalir: TBitBtn;
    DBMemoTemario: TDBMemo;
    TAsign: TTable;
    DataSource1: TDataSource;
    DBEditFecha: TDBEdit;
    SpeedAceptar: TSpeedButton;
    SpeedCancelar: TSpeedButton;
    QUpdateTemario: TQuery;
    SpeedGrabar: TSpeedButton;
    procedure SpeedAceptarClick(Sender:
    TObject);
    procedure FormActivate(Sender: TObject);
    procedure BitSalirClick(Sender: TObject);
    procedure MuestraHint(Sender: TObject);
    procedure SpeedCancelarClick(Sender:
    TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure LimpiarFormato;
    procedure BitGrabarClick(Sender:
    TObject);
    procedure DBMemoTemarioKeyPress(Sender:
    TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FormCambiosTemarios: TFormCambiosTemarios;
  NumUnidad :integer;
  Clave : integer;

```

```

implementation
uses UnidadGral;
{$R *.DFM}

procedure
TFormCambiosTemarios.SpeedAceptarClick(Sender
: TObject);
begin
  If (EditClaveAsign.Text <> ' ') then
  Begin
    Clave :=
    QuitaEspacios(EditClaveAsign.Text);
    TTemarios.Active := True;
    If TTemarios.FindKey([Clave]) Then
    Begin
      TAsign.Active := True;
      If TAsign.FindKey([Clave])<>
      true Then
        Begin
          MessageDlg('LA ASIGNATURA
          NO EXISTE!', mtError, [mbOK], 0);
          LimpiarFormato;
        End
        Else
        Begin
          DBMemoTemario.Color :=
          clWindow;
          SpeedGrabar.Enabled :=
          True;
          SpeedAceptar.Enabled :=
          False;
        End
        End
        Else
        Begin
          MessageDlg('ERROR EL TEMARIO NO
          EXISTE!',
          mtError, [mbOK], 0);
          LimpiarFormato;
        End;
        End
        Else
        Begin
          MessageDlg('POR FAVOR TECLEE LA
          CLAVE DE LA ASIGNATURA!',
          mtError, [mbOK], 0);
          LimpiarFormato;
        End;
      end;

    procedure
    TFormCambiosTemarios.FormActivate(Sender:
    TObject);
    var
      Horas,Minutos,Sec, MSec : word;
      Hora : string;
    begin
      NumUnidad := 1;
      (** BLOQUE PARA FECHA/HORA **)
      ArmaFecha;

```


APENDICE A

```

StatusBar.Panels.Items[1].Text := Fecha;

DecodeTime(Time, Horas, Minutos, Sec,
MSec);
Timer1.Interval := 60000 - (Sec * 1000);
Hora := IntToStr(Horas) + ' : '+
IntToStr(Minutos)+' Hrs';
StatusBar.Panels.Items[2].Text := Hora;
(*****)
Application.OnHint := MuestraHint;
SpeedGrabar.Enabled := False;
SpeedAceptar.Enabled := true;
end;

procedure
TFormCambiosTemarios.BitSalirClick(Sender:
TObject);
begin
    Ttemarios.Active := False;
    TAsign.Active := False;
    Close;
end;

Procedure
TFormCambiosTemarios.MuestraHint(Sender:
TObject);
Begin
    If Length (Application.Hint) > 0 then
        Begin
            StatusBar.SimplePanel := True;
            StatusBar.SimpleText :=
Application.Hint;
        End
    Else
        StatusBar.SimplePanel := False;
end;

procedure
TFormCambiosTemarios.SpeedCancelarClick(Sende
r: TObject);
begin
    LimpiarFormato;
end;

procedure
TFormCambiosTemarios.LimpiarFormato;
Begin
    DBMemoTemario.Lines.Clear;
    DBMemoTemario.Color := clInactiveBorder;
    DBEditFecha.Clear;
    DBEditNomAsign.Clear;
    EditClaveAsign.Text := '';
    EditClaveAsign.SetFocus;
    SpeedGrabar.Enabled := False;
    SpeedAceptar.Enabled := true;
End;

procedure
TFormCambiosTemarios.Timer1Timer(Sender:
TObject);

```

```

var
    Horas,Minutos,Sec, MSec : word;
    Hora : string;
begin
    DecodeTime(Time, Horas, Minutos, Sec,
MSec);
    Hora := IntToStr(Horas) + ' : '+
IntToStr(Minutos);
    StatusBar.Panels.Items[2].Text := Hora;
    Timer1.Interval := 60000;
end;

procedure
TFormCambiosTemarios.BitGrabarClick(Sender:
TObject);
var BandError : integer;
begin
    BandError := 0;
    Try
        If MessageDlg('¿LOS DATOS SON
CORRECTOS?',mtConfirmation,
[mbYes,mbNo],0) = mrYes
then
        begin
            ArmaFecha;
            DBEditFecha.Text := Fecha2;
            QUpdateTemario.Prepare;
            QUpdateTemario.Params[0].AsString :=
Fecha2;
            QUpdateTemario.Params[1].AsMemo :=
DBMemoTemario.Text;
            QUpdateTemario.Params[2].AsInteger :=
QuitaEspacios(EditClaveAsign.Text);
            QUpdateTemario.ExecSQL;
        end
    Else
        begin
            BandError := 1;
            EditClaveAsign.SetFocus;
        end;
    Except
        MessageDlg('ERROR AL MODIFICAR
TEMARIO, POR FAVOR VERIFIQUE LOS DATOS',
,mtError,[mbOK],0);
        BandError := 1;
        Raise;
    End;

    If BandError = 0 then
        begin
            If MessageDlg('LOS DATOS FUERON
GRABADOS DESEA HACER OTRA MODIFICACION?',
,mtConfirmation,[mbYes,mbNo],0) =
mrYes then
                begin
                    LimpiarFormato;
                end
            else
                Close;
        end;
    end;
end;

```

APENDICE A

```

end;

procedure
TFormCambiosTemarios.DBMemoTemarioKeyPress(Sender: TObject;
var Key: Char);
begin
    Key:=AnsiUpperCase(Key)[1];
end;

end.

unit ConsultaTemarios;

interface

uses
Windows, Messages, SysUtils, Classes,
Graphics, Controls, Forms, Dialogs,
Db, DBTables, ComCtrls, Buttons, ExtCtrls,
StdCtrls, DBCtrls, Mask;

type
TFormConsultaTemarios = class(TForm)
    StatusBar: TStatusBar;
    GroupBox3: TGroupBox;
    GroupBox4: TGroupBox;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    EditClaveAsign: TMaskEdit;
    DBEditNomAsign: TDBEdit;
    DataSource2: TDataSource;
    Ttemarios: TTable;
    Timer1: TTimer;
    GroupBox5: TGroupBox;
    Label13: TLabel;
    Label14: TLabel;
    Panel2: TPanel;
    BitSalir: TBitBtn;
    DBMemoTemario: TDBMemo;
    TAsign: TTable;
    DataSource1: TDataSource;
    DBEditFecha: TDBEdit;
    SpeedAceptar: TSpeedButton;
    SpeedCancelar: TSpeedButton;
    procedure SpeedAceptarClick(Sender:
TObject);
    procedure FormActivate(Sender: TObject);
    procedure BitSalirClick(Sender: TObject);
    Procedure MuestraHint(Sender: TObject);
    procedure SpeedCancelarClick(Sender:
TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure LimpiarFormato;
private
    { Private declarations }
public

```

```

    { Public declarations }
end;

var
    FormConsultaTemarios:
TFormConsultaTemarios;
    NumUnidad : integer;
    Clave : integer;
implementation
uses UnidadGral;
{$R *.DFM}

procedure
TFormConsultaTemarios.SpeedAceptarClick(Sender:
TObject);
begin
    If {EditClaveAsign.Text <> ' '} then
        Begin
            Clave :=
QuitaEspacios(EditClaveAsign.Text);
            Ttemarios.Active := True;
            If Ttemarios.FindKey({Clave}) Then
                Begin
                    TAsign.Active := True;
                    If TAsign.FindKey({Clave})<>
true Then
                        Begin
                            MessageDlg(';LA ASIGNATURA
NO EXISTE!',mtError,[mbOK],0);
                            EditClaveAsign.Text := '
';
                            EditClaveAsign.SetFocus;
                        End;
                    End
                Else
                    Begin
                        MessageDlg(';ERROR EL TEMARIO NO
EXISTE!',
                            mtError,[mbOK],0);
                            EditClaveAsign.Text := '
';
                            EditClaveAsign.SetFocus;
                        End;
                    End
                Else
                    Begin
                        MessageDlg(';POR FAVOR TECLEE LA
CLAVE DE LA ASIGNATURA!',
                            mtError,[mbOK],0);
                            EditClaveAsign.Text := '
';
                            EditClaveAsign.SetFocus;
                        End;
                    End;
                end;
            end;

procedure
TFormConsultaTemarios.FormActivate(Sender:
TObject);
var
    Horas,Minutos,Sec, MSec : word;
    Hora : string;
begin

```

APENDICE A

```

    NumUnidad := 1;
    (** BLOQUE PARA FECHA/HORA **)
    ArmaFecha;
    StatusBar.Panels.Items[1].Text := Fecha;

    DecodeTime(Time, Horas, Minutos, Sec,
MSec);
    Timer1.Interval := 60000 - (Sec * 1000);
    Hora := IntToStr(Horas) + ' : ' +
IntToStr(Minutos) + ' Hrs';
    StatusBar.Panels.Items[2].Text := Hora;
    (**.....**)
    Application.OnHint := MuestraHint;
end;

procedure
TFormConsultaTemarios.BitSalirClick(Sender:
TObject);
begin
    Close;
end;

Procedure
TFormConsultaTemarios.MuestraHint(Sender:
TObject);
Begin
    If Length (Application.Hint) > 0 then
        Begin
            StatusBar.SimplePanel := True;
            StatusBar.SimpleText :=
Application.Hint;
        End
    Else
        StatusBar.SimplePanel := False;
end;

procedure
TFormConsultaTemarios.SpeedCancelarClick(Send
er: TObject);
begin
    LimpiarFormato;
end;

procedure
TFormConsultaTemarios.LimpiarFormato;
Begin
    DBMemoTemario.Lines.Clear;
    DBEditFecha.Clear;
    DBEditNomAsign.Clear;
    EditClaveAsign.Text := '';
    EditClaveAsign.SetFocus;
End;

procedure
TFormConsultaTemarios.Timer1Timer(Sender:
TObject);
var
    Horas,Minutos,Sec, MSec : word;
    Hora : string;
begin

```

```

    DecodeTime(Time, Horas, Minutos, Sec,
MSec);
    Hora := IntToStr(Horas) + ' : ' +
IntToStr(Minutos);
    StatusBar.Panels.Items[2].Text := Hora;
    Timer1.Interval := 60000;
end;

end.

```

Listado de Figuras

FIGURA	NOMBRE DE LA FIGURA	PÁGINA
2.1	Representación de un proceso	6
2.2	Flujo de entrada y flujo de salidas en un proceso	7
2.3	Representación gráfica de archivo	7
2.4	Representación gráfica de la entidad externa	9
2.5	Ejemplo de representación de una entidad	12
2.6	Formato para escribir la relación entre dos entidades	12
2.7	Opcionalidad	13
2.8	El grado representado por una "pata de gallo" y una línea simple	13
2.9	Representación gráfica de una relación entre la entidad	13
2.10	Matriz de relaciones	14
2.11	La barra UID indica que la relación con la asignatura es parte del UID de reactivo	16
2.12	La evaluación no tiene UID	
2.13	Candidatos a UID: "clave profesor", "clave alumno" y "nombre"	17
2.14	En la entidad Asignatura (a) y (b)	18
2.15	Reglas de las formas normales	21
2.16	DTE incompleto	24
2.17	Notación del diccionario de datos	26
2.18	Mapa de instancias	30
3.1	Componentes de red	39
3.2	Principales medios físicos de conexión	41
3.3	Topologías físicas básicas	43
3.4	Modelo OSI	50
3.5	Arquitectura de TCP/ IP	52
3.6	Dispositivos de interconectividad y el modelo OSI	53
3.7	Trama de Ethernet	58
3.8	Red LAN Ethernet 10Base-T	70
4.1	Elementos del diseño de sistemas	75
4.2	Seis objetivos en el diseño de salidas	76
4.3	Seis objetivos en el diseño de entradas	77
4.4	Tres secciones de una pantalla	78
4.5	Cinco objetivos en el diseño del almacenamiento de datos	80
4.6	Diseño de procedimientos	82
4.7	Diagrama de contexto SISE	84

4.8	DFD Módulo de claves de acceso	85
4.9	DFD Módulo de profesores	86
4.10	DFD Consultas de profesores	87
4.11	DFD Módulo de asignaturas	88
4.12	DFD Consultas de asignaturas	89
4.13	DFD Módulo de reactivos	90
4.14	DFD Consultas de reactivos	91
4.15	DFD Módulo de exámenes	92
4.16	DFD Módulo de temarios	93
5.14	Muestra la ventana de Una Base de Datos de Access	112
5.15	Una Tabla de Access	114
5.16	Proceso para realizar una aplicación con imagen Windows	117
5.17	Paquetes y lenguajes que manejan programación "Visual"	118
5.18	Desarrollo de una aplicación para una interface gráfica	119
5.19	Acceso a la base de datos nativas con Delphi CS	122
5.20	Acceso a servidores nativos SQL con la edición SQL de Report Smith	123
5.21	Ficha Técnica de Visual Basic	123
5.22	Gráfica comparativa de la curva de aprendizaje del software analizado	127
5.23	Gráfica que muestra comparativamente la flexibilidad del software Analizado	128
5.24	Sumario de Características del Software	128
5.25	Niveles de seguridad del Sistema	130
5.26	Validación de Datos	132
5.27	Base de Datos	133
5.28	Módulo Principal	134
5.29	Módulo de Captura	137
5.30	Diagrama de Flujo del Sistema	138
5.31	Diagrama de Flujo de los Datos	139
5.32	Menú inicial de Microsoft Access	140
5.33	Generación de una nueva Base de Datos	141
5.34	Ventana de Base de Datos	141
5.35	Tabla de Profesor de la Base de Datos	143
5.36	Pantalla que muestra todas las tablas de la Base de Datos	144
5.37	Biblioteca de Componentes visuales VCL	145
5.38	Inspector de Objetos	145
5.39	Entorno de Delphi	146
5.41	Presentación del proyecto	148
5.42	Ventana de Clave de Acceso	150

5.43	Pantalla del menú principal	151
5.44	Menú de Opciones	152
5.45	Pantalla de alta profesor	153
5.46	Pantalla de consulta de profesores	154
5.47	Pantalla de captura para la elaboración de exámenes	155

Listado de Tablas

TABLA	NOMBRE DE LA TABLA	PÁGINA
1.1	Ventajas de las Bases de Datos Vs. Archivos Tradicionales	2
3.1	Tabla de comparativo de cables	41
3.2	Relación entre el modelo OSI y el conjunto de protocolos TCP/ IP	51
3.3	Análisis de la red LAN	66
3.4	Relación entre las topologías y los medios de transmisión	67
3.5	Características de los medios de transmisión	67
3.6	Especificaciones Ethernet	68
4.1	Tabla de profesores	95
4.2	Tabla de asignaturas	95
4.3	Tabla de temarios	95
4.4	Tabla de reactivos tipos del 1 al 3	96
4.5	Tabla de reactivos tipo 4	96
4.6	Tabla de reactivos tipo 5	97
4.7	Tabla de reactivos tipo 6	97
4.8	Tabla de reactivos tipo 7	98
4.9	Tabla de exámenes	98
4.10	Tabla de claves	98
4.11	Tabla de paso	99
4.12	Diagrama entidad-relación. Base de datos del SISE	99
5.1	Evaluación de Tiempo de Procesamiento en Batch	106
5.2	Evaluación de Tiempo Importación y generación de índices	107
5.3	Evaluación de Soporte de Núm Máximo de archivos abiertos	107
5.4	Evaluación de Facilidad de Uso	107
5.5	Evaluación que Proporciona Seguridad	108
5.6	Evaluación de la Integridad de los Datos	108
5.7	Evaluación de la Presentación de Calidad en los Reportes y Formas	108
5.8	Evaluación en la Facilidad de Programación	109
5.9	Evaluación en la Compartición de los datos con otras bases de datos	109
5.10	Evaluación en la Diversidad en Tipos de Datos	109
5.11	Evaluación en el Manejo de Graficos para formulario	110
5.12	Evaluación para la Puntuación final	110
5.13	Puntuación de la Comparación	110

Glosario

Almacenamiento	Pertenciente a un dispositivo dentro del cuál pueden introducirse datos para retenerlos y poder ser recuperados después.
Ancho de banda	Un intervalo de frecuencia entre dos límites definidos y expresados en ciclos por segundo o hertz.
ANSI	(American National Standards Institute, Instituto nacional de estándares americanos). Es una organización que define las normas de código y los esquemas de señalización en los Estados Unidos y los representa en la organización de estándares internacionales (ISO) y dentro del comité consultivo internacional para telegrafía y telefonía (CCITT).
Aplicación	Programa que es creado para ayudar al usuario en una tarea específica.
Automatización	Implementación de procesos por medios automáticos.
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
Backbone	Es el canal de comunicación principal, al cual se interconectan segmentos individuales de red, para formar una red Lan o Wan más grande.
Broadcast	La transmisión de un mensaje hacia dos o más estaciones al mismo tiempo.
BUFFER	Área reservada de memoria que en conjunto se comporta como una sola unidad.
Carriers	Compañías de telecomunicaciones que prestan el servicio de transporte de información sobre un canal de comunicaciones propietario.
CCITT	(Consultative Committee for International Telephony and Telegraphy) Comité consultivo internacional para telefonía y telegrafía. Una organización internacional de normas de comunicaciones. Es uno de los cuatro órganos de la Unión Internacional de Telecomunicaciones, fundada en 1865, con sede central en Ginebra y compuesta por más de 150 países miembros.
Circuito Virtual	Es un trayecto a través de una red de conmutación de paquetes de tipo mailla que adopta el aspecto de un circuito conectado físicamente y dedicado. Los sistemas finales y nodos a lo largo de la red predefinen y mantienen un circuito virtual, pero el trayecto actual a través de la red de conmutación de paquetes puede cambiar.
Cliente	Programa de computadora que utiliza los recursos dados por otra máquina o programa a través de la red.

Control de flujo	Técnica de ajuste de velocidad que se utiliza en las comunicaciones de datos para impedir el bloqueo de dispositivos receptores y la pérdida de datos.
Dato	Cualquier representación como caracteres o cantidades analógicas a las cuales se les asigna un significado.
DCE	(Equipo de Comunicación de Datos). Es un dispositivo que mantiene y termina una conexión entre el equipo terminal de datos y un medio de comunicación (por ejemplo un módem).
Diagrama de bloques	Un diagrama de un Sistema, instrumento, computadora o programa el cuál se divide en partes seleccionadas y se representan por cuadros o líneas conectoras.
Diagrama de flujo	Representación gráfica para la definición, análisis o solución de un problema en el cuál se utilizan símbolos para representar operaciones, datos, flujo o equipo en la descripción de un problema
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.
DTE	(Equipo Terminal de Datos). Es un dispositivo que transmite o recibe datos.
Encapsulación	Consiste en la inserción de un encabezado correspondiente a un protocolo dentro de otro paquete, asociado a un protocolo distinto.
Encriptar	Alterar o codificar datos para prevenir accesos no autorizados.
Firewall	Un ruteador lógico que crea una barrera que aísla segmentos de red y protege al resto de la red de otros incidentes, tal como, accesos no autorizados.
Frame	Es una unidad de transmisión lógica manejada en los niveles de enlace de datos, está incluye, su propio control de información para direccionamiento y chequeo de error, además de la dirección de transmisión.
FTP	Protocolo de transferencia de archivos. Es un programa de transferencia de archivos en entornos TCP / IP. Se utiliza para conectarse con otro sistema y ejecutar varias órdenes de generación de listas y transferencia de archivos entre ambos sistemas.

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.
Host	Cualquier computadora capaz de ejecutar programas de aplicación que se encuentran conectadas a una red.
IEEE	(Institute of Electrical and Electronic Engineers, Instituto de Ingenieros Eléctricos y Electrónicos). Es una sociedad con base en Estados Unidos que desarrolla, entre otras cosas, normas para la comunicación de datos.
ISO	(International Standards Organization) Organización de estándares internacionales. Una organización que establece estándares (normas) internacionales de comunicaciones e intercambio de información, fundada en 1946 con sede en Ginebra.
Internet	Es una malla mundial de computadoras y redes de computadoras interconectadas en una única gran red de comunicaciones extendida por todo el mundo. La malla se refiere al hecho de que Internet es una red de redes que utiliza el protocolo TCP / IP para la transmisión de datos.
Intranet	Red de uso privado que emplea los mismos estándares y herramientas que Internet.
LAN	Local Area Network. Cualquier tecnología física de redes que opera a altas velocidades sobre distancias cortas (por ejemplo a pocos cientos de metros).
MAN	Red de área Metropolitana. Este tipo de red se constituye cuando se conecta un conjunto de redes locales en una misma área geográfica pequeña.
Multicast	Un mensaje enviado simultáneamente a un grupo específico de nodos dentro de una red.
Multimedia	Conjunción de vídeo, imágenes, sonidos, texto, gráficas, etc. en un programa que permite la más 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	Interconexión de sistemas abiertos. Las normas OSI fomentan los entornos abiertos de conexión de red que permite a los sistemas de computadoras de múltiples fabricantes, comunicarse unos con otros mediante el uso de protocolos que han sido aceptados internacionalmente por los miembros de ISO.
Paquete	Un conjunto de datos manipulados por una red en un formato bien definido que incluye un encabezado y un campo de datos.
PPP	Es un protocolo de comunicación serie que opera sobre líneas de enlace telefónico o alquiladas(dedicadas) para proporcionar conexiones dentro de redes.
Protocolo	Conjunto de convenciones y normas para establecer un diálogo a distintos niveles de implementación de una arquitectura de redes.
Red	Una red de computadoras es un sistema de comunicaciones de datos que enlaza dos o más computadoras y dispositivos periféricos.
Red Virtual	Una asociación lógica que permite a los usuarios comunicarse entre si como si estuvieran conectados al mismo segmento de red independientemente del segmento en el que estén conectados físicamente.
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.
Software	Conjunto de programas que hacen uso a su vez de los respectivos conjuntos de datos para su funcionamiento.
SONET	(Synchronous Optical Network). Red sincrónica óptica. Es la especificación ANSI para las transmisiones de fibra óptica y define la infraestructura global genérica para la transmisión de información sincrónica y asincrónica (datos sensibles al tiempo), puede transmitir en los rangos multimegabit y multigigabit.

SQL	Structure Query Language. Lenguaje de consulta estructurado que se utiliza para consultar, actualizar y gestionar bases de datos relacionales.
TCP / IP	(Transmission Control Protocol / Internet Protocol). Es un software desarrollado por el departamento de defensa de Estados Unidos. El IP trabaja con el nivel OSI y TCP trabaja en los niveles cuatro y cinco de OSI.
Throughput	Es la cantidad de bits transmitidos en un determinado periodo de tiempo. Por lo general, este término se relaciona con el aprovechamiento del ancho de banda.
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.
WAN	Red de área Global. Se constituye cuando se comunican un conjunto de Lan's o de Man's.
WWW	(World Wide Web). Malla extensa mundial, proporciona servicios de localización de información mediante la utilización de enlaces de hipertexto que conectan un documento con otro.

Bibliografía

- ❖ Dan Osier, Steve Grobman, Steve Batson. Aprendiendo Delphi en 14 días. Editorial: Prentice Hall, 1998.
- ❖ Dan Osier, Steve Grobman, Steve Batson. Aprendiendo Delphi en 21 días. Editorial: Prentice Hall, 1998.
- ❖ Manning Michelle M. Delphi 2 Guía Oficial de Borland. Editorial: Prentice Hall.
- ❖ Pressman Roger S. Ingeniería de Software, Un enfoque práctico. Editorial: MacGrawHill, 1992.
- ❖ Kendall Kenneth E., Kendall Julie E. Analisis y Diseño de Sistemas. Editorial: Prentice Hall, 1991.
- ❖ Ahuja, Vijay. Design and Analysis of computer Communication Networks. Editorial: Mc Graw Hill Book Company. 1986.
- ❖ Black, Uyles. Reds de Computadoras, Protocolos, Normas e Interfaces. Coedición Macrobit rama. 1990.
- ❖ Reglamento DGETI. Informe 1989-1996.
- ❖ Davidson, Robert P. And Muller, Nathan J. Internetworking LANs: Operation, Design and Management. Artech House. Boston London 1992.
- ❖ FitzGerald, Jerry. Comunicación de Datos en los Negocios (conceptos básicos, seguridad y diseño). Editorial: Megabyte Noriega editores. México 1993.
- ❖ Rodriguez G.Jorge E. Introducción a las Redes de Área Local. Editorial: Mc Graw Hill. 1ª Edición. 1996.
- ❖ Stantifaller, Michael. TCP / IP and NFS Internetworking in a UNIX Environment. Editorial: Addison Wesley publishing company. 1992
- ❖ Raya Cabrera Jose Luis . Redes Locales y TCP / IP. Editorial: Ra-ma. 1995.
- ❖ Tanenbaum, Andrew S. Redes de Ordenadores. Segunda edición en español. Editorial: Prentice Hall. 1989.

Hemerografía

- ❖ Rello Corzo, A : Diseño de red. RED 49: 2-6, 1994.
 - ❖ El ABC de las redes locales. Especial RED. 1991.
 - ❖ Luczak, M: Interconexión de redes. RED 7: 35-38, 1991.
 - ❖ Carreño, J: Tendencias en redes locales. RED 54: 24-28, 1995.
 - ❖ Cortés; P.: Tendencias en cableado. RED 73: 20-26, 1996.
 - ❖ Guerrero, G.: Tendencias en administración de redes. RED 72: 16-26, 1996.
 - ❖ Peral, G.: ¿Cómo entender la administración de redes y sistemas?. RED 60: 12-14, 1995.
-