

45



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

"SISTEMA DE APOYO INFORMATIVO PARA
LA LINEA DE INFORMACION DEL CONADIC"

T E S I S

PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION

P R E S E N T A N
GUTIERREZ JUAREZ VICTOR
RAMIREZ MARTINEZ ROSA



ASESOR DE TESIS: ING. LUCILA PATRICIA ARELLANO MENDOZA

MEXICO, D.F.

AGOSTO DEL 2000



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

† Mamá:

Te agradezco el haberme dado la vida, el haberme dado siempre el ejemplo de honestidad, de trabajo y perseverancia para llegar a una meta. Hoy le doy gracias por haberme dado la oportunidad de darme una madre como tu. Este triunfo es para ti. Que Dios te tenga junto a él, por tu memoria no te fallaré. Muchas Gracias Mamá.

Papá:

Has sido un gran maestro para mí, ya que me enseñaste a luchar siempre por lo justo. Te agradezco el que hayas siempre arriesgado tu vida en el trabajo por todos, gracias por todos estos años de felicidad, este triunfo es para ti también. Dios te conserve siempre conmigo y que sigas siendo mi soporte tanto en los momentos difíciles y compartir los momentos de felicidad. Muchas Gracias Papá.

A mis hermanos:

Gracias por su apoyo a Rosalía, a María Eugenia y Erik y que nos sigamos llevando muy bien como hasta ahora y apoyándonos entre todos siguiendo el ejemplo que nos dejó mamá. Gracias a todos.

A Paty:

Eres la persona que me ha devuelto las ganas de vivir y luchar por mis ideas. Gracias por acompañarme en los momentos complicados y alentarme siempre por cada reto que tengo en la vida. Te amo y estoy seguro que Dios nos ayudará para cumplir nuestra meta que tu y yo anhelamos tanto. Gracias mi vida.

A la Fam. Bernal Monzón:

Gracias por darme la mano cuando más la necesitaba. Agradezco sus atenciones hacia mí. Dios los mantenga siempre unidos.

Al Lic. Juan Carlos Cabrera:

Por la ayuda incondicional que siempre tuvo hacia nosotros, por compartir sus conocimientos y sobre todo por la gran calidad de persona que es. Espero nunca pierdas esa forma de ser y de ayudar a las personas que quieren sobresalir en la vida. Este trabajo es para ti. Muchas Gracias.

A la UNAM:

Gracias por albergarme en sus instalaciones e inculcarme a través de sus profesores siempre el sentido de superación y brindarme los conocimientos para la vida laboral. Siempre tendré la camiseta bien puesta.

VICTOR GUTIÉRREZ JUÁREZ

A MIS PADRES:

Sabiendo que jamás existirá una forma
de agradecer una vida de lucha,
sacrificio, y esfuerzo constantes,
sólo deseo que comprendan que él
logro mío es suyo, que mi esfuerzo
es inspirado en ustedes y que son
mi único ideal.

Por apoyarme en cada meta de mi vida
con respeto y admiración
mil gracias.

A MI HIJO Y ESPOSO:

por el apoyo y comprensión
que me brindaron para concluir esta etapa de mi vida.
mil gracias.

A MIS HERMANOS Y FAMILIA:

Por su ejemplo y comprensión
Mil gracias.

AL LIC. JUAN CARLOS CABRERA:

por habernos apoyado
con su experiencia y
conocimientos en la realización
del presente trabajo
mil gracias.

A MIS PROFESORES:

por su gran vocación para enseñar
y por su interés al guiarme
por el camino del conocimiento
para pasar por alto la ignorancia
mil gracias.

ROSA RAMÍREZ MARTÍNEZ



Páginas

| | |
|--|-----------|
| Introducción | 1 |
| 1. Antecedentes de las drogas | 3 |
| 1.1 Antecedentes del Consejo Nacional Contra las adicciones (CONADIC) | 6 |
| 1.1.1 Composición del CONADIC | 8 |
| 1.1.2 Objetivo general del CONADIC | 9 |
| 1.1.3 Atribuciones del CONADIC | 11 |
| 1.2 La subdirección de orientación e información | 13 |
| 1.2.1 Aspectos generales del centro de atención telefónica salud-hable | 13 |
| 1.2.1.1 Antecedentes de la línea salud-hable | 13 |
| 1.2.1.2 Misión de la línea salud-hable | 14 |
| 1.2.1.3 Objetivo genérico de la línea salud-hable | 15 |
| 1.2.1.4 Objetivos específicos | 15 |
| 2. Estudio de Factibilidad y Análisis del Sistema | 16 |
| 2.1 Objetivo del sistema | 17 |
| 2.2 Definición del problema | 19 |
| 2.3 Especificación de requerimientos | 20 |
| 2.4 Estrategias de solución | 21 |
| 2.5 Justificación del sistema | 24 |
| 2.6 Análisis de requerimientos | 29 |
| 2.6.1 Características del personal del departamento | 29 |
| 2.6.2 Infraestructura del equipo | 31 |
| 2.6.2.1 Establecimiento de criterios de selección | 32 |
| 2.6.3 Consideraciones de software | 33 |
| 2.6.3.1 Establecimiento de criterios de selección | 38 |
| 2.7 Restricciones del sistema | 49 |
| 3. Estrategia del sistema | 50 |
| 3.1 Teoría de las bases de datos | 51 |
| 3.1.1 Objetivo del sistema administrador de bases de datos | 52 |

| | |
|--|------------|
| 3.2 Arquitectura de un sistema de administración de bases de datos (DBMS)..... | 53 |
| 3.3 Estructura de la base de datos relacional del sistema..... | 55 |
| 3.4 Modelado de atributos..... | 57 |
| 3.5 Normalización..... | 58 |
| 3.6 Álgebra relacional..... | 82 |
| 4. Diseño del sistema..... | 83 |
| 4.1 Modelado de entidades..... | 85 |
| 4.2 Diseño de las relaciones..... | 87 |
| 4.2.1 Relación orgcaracterísticas..... | 90 |
| 4.2.2 Relación orgcontacto..... | 92 |
| 4.2.3 Relación orggenerales..... | 92 |
| 4.2.4 Relación orgobjetivo..... | 93 |
| 4.2.5 Relación orgservadicional..... | 94 |
| 4.2.6 Relación orgservicio..... | 94 |
| 4.2.7 Relación orgtratamientos..... | 95 |
| 4.2.8 Relación Detalle..... | 96 |
| 4.2.9 Relación subtítulo..... | 96 |
| 4.2.10 Relación tema..... | 97 |
| 4.2.11 Relación título..... | 97 |
| 4.2.12 Relación organexo..... | 98 |
| 4.2.13 Relación acceso..... | 98 |
| 4.2.14 Relación concepto..... | 99 |
| 4.2.15 Relación folio..... | 99 |
| 4.2.16 Relación usuario..... | 99 |
| 5. Desarrollo del sistema..... | 102 |
| 5.1 Creación de relaciones..... | 112 |
| 5.2 Creación de consultas..... | 115 |
| 5.3 Creación de formularios..... | 118 |
| 6. Pruebas y ajuste..... | 126 |
| 6.1 Tipos de pruebas..... | 127 |
| 6.2 Problemas que se presentaron en el desarrollo del sistema..... | 130 |
| 6.3 Ventajas y desventajas del sistema..... | 130 |
| Conclusiones..... | 132 |
| Diccionario de datos..... | 136 |
| Bibliografía..... | 144 |
| Manual técnico..... | 147 |
| Manual del usuario..... | 202 |

INTRODUCCIÓN

El incremento del abuso de drogas, tabaquismo, alcoholismo y sus repercusiones en el desarrollo de muchas sociedades en el mundo, entre ellas la nuestra, es un hecho que preocupa a la opinión pública, a la comunidad científica y a los responsables de la formulación de políticas y toma de decisiones. Una muestra de esta inquietud son las acciones y programas que amplios sectores en nuestro país llevan a cabo para prevenir el consumo de estas sustancias y brindar atención y rehabilitación a los adictos.

El objetivo de esta tesis es presentar una herramienta informática al Centro de Atención Telefónica Salud – Hable perteneciente a la Subdirección de Orientación e Información del CONADIC para la automatización en el manejo de los datos con referencia a los temas de drogas, alcoholismo y tabaquismo que ahí se manejan, contemplando también el brindar información necesaria y en su caso canalizar a las personas que así lo requieran a los diferentes organismos públicos o privados encargados de esa tarea.

El presente trabajo consta de seis Capítulos, conclusión, diccionario de datos, bibliografía y dos Anexos en los cuales se mencionará en forma breve el contenido de cada uno de ellos con el cual se pretende dar una idea del contexto de esta tesis.

En el primer capítulo se presentarán los antecedentes del Consejo Nacional contra las Adicciones (CONADIC). En este capítulo tratamos de dar un pequeño antecedente histórico de lo que actualmente es este organismo.

El capítulo segundo contiene el estudio de factibilidad y análisis del Sistema. En él exponemos los alcances de este trabajo, el problema a resolver y sus posibles soluciones, así como una metodología que cubre los pasos necesarios para seguir el ciclo de desarrollo del sistema.

También haremos una evaluación de los recursos de software, hardware, tipo de personal y usuarios para el cual esta hecho el sistema, así como las restricciones que conlleva el desarrollo del mismo.

En el capítulo tercero se menciona la teoría de las bases de datos, así como un análisis de los manejadores de bases de datos.

En el capítulo cuarto realiza el diseño del Sistema. En esta parte del trabajo se incluyen las etapas correspondientes al diseño del modelo conceptual y la especificación de los datos.

En el capítulo quinto se menciona el desarrollo del Sistema. En esta fase se realiza, escribe y depura el programa fuente, siguiendo los estándares de programación que se adecuen al desarrollo del sistema.

En el capítulo sexto contiene la puesta en marcha y pruebas. En esta parte se prueban los diversos módulos y se integran para probar el sistema completo a fin de ser aprobado.

Finalmente contiene dos anexos

Anexo A: Manual Técnico. En este manual se mencionan los temas referentes al manejo técnico del sistema.

Anexo B: Manual de Operación. En este anexo se mencionan los temas de funcionamiento operativo de la línea y las situaciones específicas dentro de la línea.

CAPÍTULO I
ANTECEDENTES DE LAS DROGAS.

CAPÍTULO I

ANTECEDENTES DE LAS DROGAS.

El problema de la drogadicción es universal, prácticamente no existe nación que no se vea afectada en alguna medida por el abuso de sustancias que causan dependencia, lo mismo las de consumo lícito como el alcohol, la nicotina y los medicamentos psicotrópicos, o las de uso ilegal que afectan el psiquismo y la conducta, principalmente en los jóvenes. Estas últimas, por las implicaciones que el narcotráfico conlleva, significan riesgos adicionales para los gobiernos, incluso pueden poner en peligro la estabilidad interna de los Estados, o provocar fricciones entre países.

Por lo pronto, todo el esfuerzo que muchos países, incluido México, realizan para combatirlos, no está dando los frutos deseados; el consumo de drogas continúa y en algunos casos se extiende, lo mismo que los delitos asociados a su producción y distribución.

El estudio de las adicciones en México se ha realizado primero desde una perspectiva clínica, y recientemente como un problema de salud, por trascender al nivel social, rebasar instituciones y naciones. Los niveles de intensidad con que se presentan en cada país y sus regiones son diferentes, por lo cual las acciones para hacerles frente deben adecuarse a sus características.

La mayor parte de la información disponible para integrar un diagnóstico de la situación que guardan las adicciones en México, proviene de estudios epidemiológicos efectuados en hogares, escuelas y grupos de alto riesgo; también de estudios clínicos de gran relevancia e investigaciones en el campo de las neurociencias, sobre todo en el caso de los solventes inhalables y las anfetaminas. Pero la mejor secuencia histórica es la que se obtiene con las encuestas realizadas en escuelas de educación media, media superior y superior por el Instituto Mexicano de Psiquiatría (IMP), la realizada por la Encuesta Nacional de Adicciones (ENA), de la Dirección General de Epidemiología (DGE) y de la Secretaría de Salud (SSA).

Estos estudios indican un incremento relativo en la prevalencia del uso de sustancias adictivas por parte de la población estudiantil en los últimos 20 años, resultando interesante destacar el hecho de que los problemas más importantes en nuestro país en materia de adicciones, por sus repercusiones sobre la salud de los consumidores, sobre la economía y la sociedad, son el alcoholismo y el tabaquismo.

Si observamos las estadísticas de mortalidad, se verá que entre las diez primeras causas de muerte están todas aquellas vinculadas directamente con el alcoholismo, en fin, podemos preguntarnos ¿Cuántos deficientes mentales tienen como antecedente el llamado síndrome del feto alcohólico? ¿Cuántos de los miles de niños que sufren maltrato son de hecho víctimas indirectas del alcoholismo de sus padres?

En suma, por la importancia y trascendencia del tema para el desarrollo económico y social de nuestra sociedad, hemos desarrollado el presente trabajo, que esperamos contribuya de alguna manera a la extinción de estos problemas.

1.1 ANTECEDENTES DEL CONSEJO NACIONAL CONTRA LAS ADICCIONES (CONADIC)

El Plan Nacional de Desarrollo 1995-2000 destaca que la salud en un sentido amplio no significa únicamente la ausencia de enfermedad, sino un estado de bienestar físico y mental enmarcado en un contexto propicio para su sustento. El logro pleno de las capacidades para el trabajo, la educación y la cultura, sólo es posible cuando existen condiciones de salud adecuadas.

De acuerdo con esta premisa, la reforma del Sistema Nacional de Salud adquiere un doble compromiso: mejorar y ampliar la calidad de los servicios mediante la reestructuración de las instituciones y fortaleciendo su coordinación.

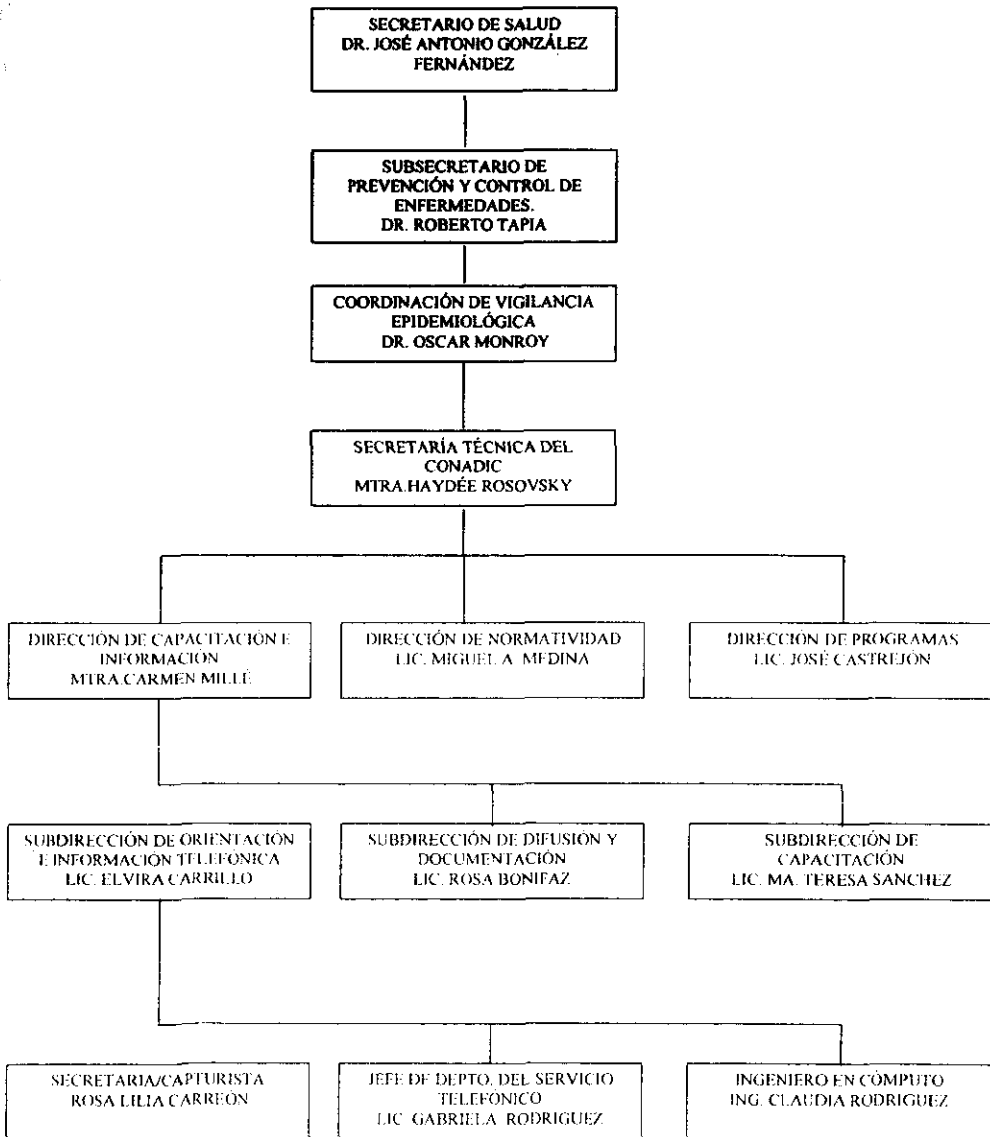
La salud incide prácticamente en todas las actividades económicas, sociales y culturales, y es un elemento imprescindible para el desarrollo del país. La salud de la población es entonces, tanto el motor como la consecuencia del progreso económico y social, pero requiere de múltiples acciones para preservarla o restituirla entre los individuos, y prevenir las enfermedades que les afectan.

Así, al considerar que las adicciones son un importante problema de salud pública se crea el Consejo Nacional contra las Adicciones (CONADIC). Esta es una instancia creada por Decreto Presidencial en julio de 1986, pero tiene antecedentes en consejos anteriores que separaban las acciones contra el alcoholismo de las demás farmacodependencias. El artículo primero del citado decreto señala lo siguiente:

"Se crea el Consejo Nacional contra las Adicciones con el objeto de promover y apoyar las acciones de los sectores público, social y privado, tendientes a la prevención y combate de los problemas de salud pública causados por las adicciones, así como para proponer y evaluar los Programas Nacionales contra el alcoholismo y abuso de bebidas alcohólicas, el Tabaquismo y la Farmacodependencia".

Los fundamentos legales para su creación se encuentran en la propia Constitución Política de los Estados Unidos Mexicanos, la Ley de Salud, la Ley de Planeación, el Plan Nacional de Desarrollo, el Programa Nacional de Salud y el Programa Nacional para el Control de Drogas.

1.1.1 COMPOSICIÓN DEL CONADIC



1.1.2 OBJETIVO GENERAL DEL CONADIC

El Consejo Nacional contra las Adicciones (CONADIC) se creó con el objeto de promover, coordinar y apoyar las acciones de los sectores público, social y privado, tendientes a prevenir y combatir los problemas de salud pública causados por las adicciones.

Para cumplir su cometido, las acciones del CONADIC se sustentan en la concertación y el fortalecimiento de nuevos esquemas de cooperación interinstitucional, cuyo marco de acción son los Programas contra el Alcoholismo y el abuso de bebidas alcohólicas, el Tabaquismo y la Farmacodependencia.

Estos programas se integran en dos apartados: el primero enfocado al diagnóstico del problema de las adicciones a nivel nacional, actualizando el marco jurídico vigente y los recursos existentes para enfrentar el fenómeno. El segundo comprende las acciones a realizar, estableciendo previamente los objetivos general y específicos, así como las estrategias y líneas de acción.

En el programa contra el Alcoholismo y el abuso de bebidas alcohólicas se presentaron datos relevantes que permiten tener una visión integral del problema, entre los que destacan:

- La disponibilidad, producción, distribución y comercialización de bebidas alcohólicas.
- Los problemas médicos y psicosociales, incluyendo la morbilidad relacionada con la ingestión de bebidas embriagantes, los actos delictivos y los problemas familiares, sociales y laborales originados por el consumo de alcohol.

Respecto al programa contra el Tabaquismo, se abordan los siguientes temas:

- **Problemas de salud asociados con el consumo de tabaco.**
- **Producción y consumo.**
- **Análisis cuantitativos del tabaquismo.**
- **Datos de salud relacionados con el tabaquismo en México**

En cuanto al programa contra la farmacodependencia, se presentan datos y conceptos relevantes con el objeto de tener una visión clara del problema, entre los que se consideran:

- Los antecedentes del problema en nuestro país, y de las acciones que se han llevado a cabo para combatirlo.
- La conceptualización de la farmacodependencia, así como la clasificación de las drogas y los daños que causan a la salud.
- El análisis de la etiología del problema.
- Los problemas psicosociales relacionados con la farmacodependencia.

El CONADIC propone estos programas a todas las instituciones de los distintos sectores que desarrollan acciones en los campos de atención y educación para la salud, investigación y legislación en materia de adicciones.

1.1.3 ATRIBUCIONES DEL CONADIC

De acuerdo con el Reglamento de Organización y Funcionamiento del Consejo Nacional contra las Adicciones (CONADIC), el consejo tiene las siguientes atribuciones:

1. Proponer a las dependencias y entidades vinculadas a los programas contra el Alcoholismo y el abuso de bebidas alcohólicas, el Tabaquismo y la Farmacodependencia, las acciones que contribuyan al adecuado cumplimiento de los mismos.
2. Sugerir acciones de seguimiento derivadas de la ejecución de los programas mencionados, evaluar sus resultados y, en su caso, proponer las adecuaciones y modificaciones pertinentes.
3. Recomendar medidas sobre el control de la publicidad relativa a bebidas alcohólicas, tabaco y otras farmacodependencias.
4. Promover, en forma permanente, actividades de análisis e investigación para apoyar las acciones contra el alcohol, el tabaco y las otras drogas.
5. Recomendar acciones preventivas de los problemas de salud pública provocados por el alcoholismo, el abuso en el consumo de alcohol, el tabaquismo y las otras farmacodependencias, así como difundirlas, promoverlas y apoyarlas.
6. Proponer la integración de grupos de trabajo tendientes a la planeación y ejecución de accidentes en materia preventiva, de tratamiento y rehabilitación.
7. Proponer las reformas que se estimen convenientes a las disposiciones legales aplicables a la producción, comercialización y consumo de bebidas alcohólicas, tabaco y otras sustancias psicoactivas.

-
8. Sugerir mecanismos de coordinación entre las autoridades federales y estatales para la eficaz ejecución de los programas referidos.
 9. Servir de foro para exponer los criterios de las autoridades y organismos representados en el consejo, acerca de las campañas al público, actividades de prevención, presentación de servicios asistenciales, investigación científica, legislación y formación de recursos humanos.
 10. Fomentar, dentro de los programas de educación para la salud, la orientación a la familia y a la comunidad sobre la disminución en el consumo de alcohol, tabaco y otras drogas, así como de otras sustancias susceptibles de producir dependencia, y proponer acciones que tiendan a disminuir su consumo.
 11. Promover el cambio de elementos condicionantes de las adicciones en el ambiente social y familiar por medio de normas, valores y actitudes sociales que legitiman su uso.
 12. Proponer políticas con relación a la oferta de tabaco y bebidas alcohólicas, con criterios de salud pública.
 13. Promover acciones que reduzcan los riesgos asociados con el consumo de tabaco y bebidas alcohólicas.
 14. Sugerir la ejecución de programas de identificación temprana del fumador y consumidor de bebidas alcohólicas, con énfasis en la mujer embarazada y personas con problemas de salud, a fin de concientizarlas sobre los efectos que, en su estado, pueden producir esas adicciones.

1.2 LA SUBDIRECCIÓN DE ORIENTACIÓN E INFORMACIÓN.

Dicha subdirección inició sus funciones el 15 de Julio de 1998, contando solo con cuatro líneas de atención al público conocidas como ADICTEL dicha línea contaba con personal no dedicado, conformada hasta el momento por 8 personas de las cuales 4 están contestando la línea a pesar de ser personal no dedicado, es decir con trabajadores que no estaban totalmente capacitados ni eran de planta en esta actividad. Siendo este programa un antecedente del Servicio de Orientación Telefónica del CONADIC, llamada actualmente salud- hable.

1.2.1 ASPECTOS GENERALES DEL CENTRO DE ATENCIÓN TELEFÓNICA SALUD-HABLE.

1.2.1.1 Antecedentes de la Línea Salud-Hable.

El servicio de Orientación Telefónica en el CONADIC inició sus funciones en febrero de 1997 como apoyo al Programa de Prevención y Control de Adicciones, generado por la necesidad de ofrecer a la población un medio directo, accesible y confidencial en el que se encontraran respuestas a dudas e inquietudes en materia de adicciones.

Tal es el caso del convenio que se da entre CONADIC y Fundación Azteca, en donde pretenden unir esfuerzos para disminuir el consumo de las drogas en nuestro país. Ambas experiencias han detectado la necesidad actual de contar con un servicio telefónico de orientación y canalización, constituyendo éste como un sensor de las necesidades de la población y de las tendencias de consumo, a través del registro de las llamadas.

Fundación Azteca es una organización estructurada para brindar apoyo a los sectores más desprotegidos, con una estructura amplia en cuanto a recursos técnicos, humanos, tecnológicos y financieros se refiere.

Dentro de los múltiples objetivos de FUNDACIÓN AZTECA se encuentran los de: fomentar la conciencia de responsabilidad social y brindar apoyo a instituciones no lucrativas, dándose así diversas funciones y alianzas con otros organismos para trabajar conjuntamente en el logro de estos.

Para coadyuvar a este efecto CONADIC apoyará con los recursos humanos capacitados en esta materia y Fundación Azteca proporcionará los recursos técnicos, tecnológicos y materiales para desarrollar el Centro Telefónico que permita brindar una mayor atención a la población y contar con las bases de datos confiables y actualizadas que propicien la toma de decisiones objetivas y oportunas.

1.2.1.2 Misión de la Línea Salud-Hable

Asegurar un servicio de atención telefónica integral en materia de prevención, canalización y orientación de adicciones, las 24 horas del día, los 365 días del año, apoyándose en la colaboración que para este efecto hace Fundación Azteca y la Secretaría de Salud a través del Consejo Nacional Contra las Adicciones, uniendo sus esfuerzos, recursos y tecnología; buscando como propósito fundamental apoyar a la sociedad en la disminución del consumo de drogas.

1.2.1.3 Objetivo Genérico de la Línea Salud-Hable

Ofrecer a la población un medio directo, accesible y confidencial en el que el público encuentre respuestas y soluciones a sus dudas e inquietudes en relación con las drogas, proporcionando los servicios de orientación psicológica y de trabajo social o mediante canalizaciones a organismos públicos o privados y centros especializados que cuenten con servicios de apoyo o rehabilitación y generando información que muestre las tendencias de casos reportados y de organismos de todo el país, de tal manera que permitan tomar decisiones para reorientar nuevas acciones, estrategias y campañas.

1.2.1.4 Objetivos Específicos

- Brindar apoyo vía telefónica a los solicitantes y fomentar la búsqueda de tratamiento y/o rehabilitación
- Canalizar a los solicitantes a los organismos públicos y/o privados que presten servicios de apoyo y rehabilitación.
- Establecer un proceso de seguimiento de casos.
- Realizar un registro de información que permita detectar nuevas tendencias o necesidades de la población.
- Evaluar el impacto de las campañas en medios masivos de información

CAPÍTULO II
ESTUDIO DE FACTIBILIDAD Y ANÁLISIS DEL SISTEMA

CAPÍTULO II

ESTUDIO DE FACTIBILIDAD Y ANÁLISIS DEL SISTEMA

Dentro de este capítulo se exponen los alcances del presente trabajo, el problema a resolver y las posibles soluciones, así como una metodología que cubre los pasos necesarios para seguir el ciclo de desarrollo del sistema.

Se hará una evaluación de los recursos del equipo, ya que debemos seleccionar aquellos que convengan al desarrollo del sistema y se especificará para que tipo de personal y usuario estará hecho el sistema, así como las restricciones que tendrá este de acuerdo a los recursos que el Consejo posee para llevar a cabo el desarrollo del mismo. Cabe hacer mención que se realizará la selección del software requerido para satisfacer las necesidades tanto del Consejo como del propio sistema.

2.1 OBJETIVO DEL SISTEMA.

Diseño, desarrollo, e implantación de un sistema informático de apoyo para facilitar el procesamiento y manipulación de datos que en materia de atención del alcoholismo, tabaquismo y farmacodependencia se refiere, así como también de aquellos que integran la cédula de cada organismo al cual se canalizaran los casos para la línea **SALUD-HABLE**.

La integración de los datos de ésta fuente permitirá la divulgación a todas las áreas del Consejo, ya sea en pantalla o mediante la elaboración de informes que apoyen tanto al personal como a usuarios externos que en algún momento requieran consultar la información contenida en el llamado árbol de información.

Los datos compilados se integrarán al directorio de organismos del servicio telefónico de orientación del CONADIC, conformando un banco de información para la canalización adecuada de los solicitantes.

Este sistema permitirá contar con un catálogo actualizado de los servicios disponible para la atención de las adicciones en sus diferentes niveles, lo que facilitará la coordinación y el acceso a los servicios por parte de los usuarios.

La información teórica referente a los temas de drogas, alcoholismo, tabaquismo así como sus respectivos efectos y repercusiones están disponibles en una estructura denominada árbol de información, al cual podrá tener acceso las personas que lo requieran.

2.2 DEFINICIÓN DEL PROBLEMA.

El problema principal consiste en el manejo de la información que se tiene en cuanto a organismos y conceptos básicos de todo lo referente a las adicciones, ya que está hecho mediante fichas de trabajo clasificadas, por lo cual es necesario validar la información recurriendo a procesos manuales que en términos de tiempo afectan notablemente la actividad de análisis.

Se observa que por tratarse de un método manual, obsoleto y poco seguro el control de la información es difícil de ser llevado a la práctica; y los objetivos del CONADIC se ven retrasados por:

- La falta de coordinación entre el distinto personal de las áreas del Consejo.
- La lentitud y dificultad de obtener resultados precisos.

El proceso de mantenimiento que se tiene actualmente para los bancos de información no permite actualizar todos los datos afectados al ritmo de los nuevos cambios que van surgiendo al transcurso del tiempo.

Por otro lado, en cuanto a la obtención de los textos de información, son hechos en computadora, pero toda información se tiene que obtener manualmente por medio de un largo y tedioso proceso.

En resumen:

- No se cuenta con un sistema flexible que permita de manera veraz y oportuna obtener los conceptos básicos en cuanto a adicciones se refiere o información de organismos para una adecuada canalización.
- Existe retraso en la entrega de información requerida por personal directivo del Consejo.
- Al existir información redundante existe inconsistencia de datos.

2.3 ESPECIFICACIÓN DE REQUERIMIENTOS.

Este departamento cuenta con una gran cantidad de información la cual es posible explotar en beneficio de los usuarios y empleados; misma que hasta la fecha no ha sido posible sistematizar tal vez por el poco tiempo que lleva. Con la escasa experiencia con la que se cuenta ha sido posible definir las necesidades reales de procesar dicha información para capturar y ordenar los datos que se necesiten dando la presentación que se requiera, para dar un servicio adecuado a las necesidades de los usuarios.

La solución de este problema es de gran importancia para este proyecto aunque sin duda con el mejoramiento de los procesos de control y el manejo de información se tendrá una mejor visión sobre los problemas que se presentan, es decir, de acuerdo a las características de los casos reportados, realizando estas funciones confiables y rápidamente. Esto permitirá una atención al público de mejor calidad satisfaciendo sus necesidades de información y canalización.

Así, el futuro de esta línea, aprovechando los recursos tecnológicos que la computación ofrece debe, según este trabajo, encaminarse a consolidar el establecimiento de un sistema de cómputo para el manejo de la información, cumpliendo sus objetivos y propósitos de apoyar vía telefónica a dichos solicitantes y fomentar la búsqueda de tratamiento, rehabilitación y canalización a los organismos públicos y privados que presten dichos servicios y den este tipo de apoyo.

Esta tesis tiene como principal idea la de controlar en la computadora toda la información necesaria de tal manera que permita lo siguiente:

- Altas.
- Bajas.
- Modificaciones.
- Consultas en formato RTF.
- Búsqueda de información y de organismos en la base de datos.

La creación de este sistema de apoyo deberá ofrecer mejores posibilidades de atención al solicitante, ya que el operador deberá contar con una capacitación que le permitirá un mayor conocimiento del sistema para así proporcionar un mejor servicio.

2.4 ESTRATEGIA DE SOLUCIÓN.

Enseguida se presentan las estrategias de solución que de acuerdo a las consideraciones de requerimientos de la problemática, serán analizadas para comparar ventajas y desventajas. La selección de una de ellas será considerada como propuesta de solución.

Las posibles alternativas son:

1. Completar el método actual para dar la información vía telefónica.
2. Realizar un sistema de información que se apegue a las necesidades identificadas hasta la fecha debido a la experiencia que se tiene en atención a los solicitantes.
3. Adquirir un sistema que pudiera apegarse a algunas de las necesidades inmediatas de los solicitantes.

Análisis de alternativas

- Analizando la primera alternativa:

VENTAJAS:

1. Los gastos en capacitación al personal serían mínimos
2. Se requeriría de poco tiempo en implantar el nuevo sistema para dar la información manualmente.
3. Los gastos en recursos materiales serían mínimos

DESVENTAJAS:

1. El acceso manual a la información seguiría siendo un obstáculo para una atención eficiente al solicitante. Esto se podría traducir a una mala canalización.
2. Existiría poca eficiencia en el control de la información proporcionada al solicitante.
3. Lentitud en el proceso.

- Analizando la segunda alternativa

VENTAJAS:

1. El sistema se realizará apegándose a las necesidades propias de los usuarios identificados por el personal.
2. El sistema será amigable para el operador, es decir, fácil de manejar.
3. El sistema se hará de forma modular para facilitar cualquier tipo de mantenimiento.
4. El sistema dará un rápido acceso a la información, uno de los puntos principales del desarrollo de este trabajo.

DESVENTAJAS.

1. El tiempo estimado para su realización es mayor al de las otras alternativas.
 2. Los gastos de capacitación al personal serán mayores, ya que se les dará un curso para el manejo del sistema.
 4. Los gastos en cuanto a equipo tecnológico serán mayores.
- Analizando la tercera alternativa.

VENTAJAS:

1. Los beneficios serían inmediatos en cuanto a tiempo.

DESVENTAJAS:

1. El costo del sistema sería elevado.
2. Tal vez se tendría adquirir equipo tecnológico adecuado al sistema.
3. Se dependería de la compañía donde se adquirió dicho sistema para su mantenimiento y la capacitación de los operadores.

Selección de alternativas:

A partir del análisis de las alternativas mencionadas hemos considerado que la mejor opción es la de realizar el sistema que se apegue a las necesidades, ya que trataremos de cubrir todos los requerimientos planteados considerándolo como un beneficio debido a su potencial hacia los solicitantes futuros.

2.5 JUSTIFICACIÓN DEL SISTEMA

Con la realización de este sistema se podrán eliminar los procesos manuales de cotejo de datos, se optimizará el almacenamiento de la información disminuyendo considerablemente el tiempo para obtener información requeridos al Consejo.

Se tendrá mayor seguridad y veracidad de la información, además de que podrá ser fácilmente actualizada.

Se aprovecharán los recursos con los que cuenta el consejo.

En suma, el sistema que se desarrollará deberá incorporar los aspectos básicos para que el propio usuario cree y de mantenimiento a sus archivos, en forma de pantallas interactivas. Adicionalmente permitirá la generación de informes descriptivos que el usuario desee.

La creación de este sistema de información deberá ofrecer mejores posibilidades para la toma de decisiones que contribuyen al mejoramiento general del Consejo, teniendo como propósito principal que el usuario disponga de una herramienta sencilla de operar que cuente con los conceptos básicos de adicciones y con un pequeño manual operativo de cómo reaccionar o que hacer en ciertas situaciones ante los usuarios de la línea telefónica de información.

Las empresas necesitan obtener ventajas competitivas en el mercado altamente cambiante. Uno de los principales problemas al que se enfrentan los tomadores de decisiones es cómo obtener esta ventaja sin necesidad de invertir una gran cantidad de tiempo y trabajo en procesos costosos para lograr integrar el ambiente tecnológico ya instalado.

Las empresas necesitan reducir los costos operacionales aprovechando los sistemas operacionales existentes – que generalmente incluyen las aplicaciones y los sistemas de bases de datos relacionales que éstas utilizan. Independientemente del motor de base de datos que conduce los sistemas operacionales, la necesidad del negocio de integrar y replicar datos es la misma.

Dentro de los manejadores de base de datos más robustos se encuentran Oracle y Sybase con los cuales mencionamos ventajas y sus desventajas:

Oracle es un gestor de bases de datos, es decir, es un sistema software que permite consultar, manipular y controlar datos mediante SQL.

Oracle: Integra soluciones para aplicaciones con los siguientes componentes:

1. Métodos Oracle.
2. Oracle Designer y Developer.
3. En la integración de plataforma de empresa contiene tres módulos que son: transporte, transformación y adaptadores.
4. El módulo de productos se encuentran los módulos de consultas avanzadas, integración rota, directorios de internet y productos asociados.

Las virtudes principales de Oracle son:

- Infraestructura robusta.
- Integración flexible.
- Aplicación autónoma.

Comparado con Visual Basic que es un software para fronts es netamente programado para hacer conexión, consulta y modificación de la base de datos que se encuentra detrás. Oracle cuenta con demasiados módulos los cuales tienen que ser muy bien asimilados por los programadores y esto se traduce un mayor costo de capacitación y de inversión en horas-hombre. Los cuales para el Consejo Nacional contra las

Adiciones son complicados la absorción de costos y para la puesta en producción del sistema.

En cuanto a los programadores de este sistema también significa una capacitación extra para Oracle del cual no existen muchos centros de capacitación y el costo de una capacitación esta fuera del alcance de ellos.

Otras de las ventajas de Visual Basic en aplicaciones cliente servidor sobre oracle son:

- Bajo costo (aunque no se conoce el costo real del procesamiento y la conversión a esta aplicación).
- Mejor distribución de la carga de procesamiento.
- Capacidad para adaptarse a cambios tecnológicos, tanto de hardware como de software.

Una mas de las desventajas de Oracle frente a Visual Basic es que contiene un largo y tedioso camino en las fases de diseño de desarrollo del sistema a diferencia de los tradicionales son:

- Estrategia.
- Análisis.
- Diseño.
- Construcción Pruebas.
- Implementación.
- Mantenimiento.

Sybase: Es un manejador de base de datos relacional contiene su propio motor de base de datos.

Una de las principales desventajas de Sybase ante Visual Basic, es que el primero cuenta con tres productos mas de los cuales depende para su buen funcionamiento.

Otra desventaja de Sybase son sus diferentes soluciones que hacen mas elevado el costo horas-hombres y enfocado al sistema de CONADIC la prontitud en que el sistema tiene que estar en producción y estas:

- Sybase Adaptive Server Enterprise – Una solución de administración de datos escalable y de alta performance.
- Sybase Adaptive Server Enterprise Replication – Una solución integrada para administración y replicación de datos para ambientes de bases de datos Sybase.
- Sybase Enterprise Studio – Una solución completa para administrar, replicar e integrar datos entre distintas bases de datos de sistemas operacionales.
- Sybase Mainframe Connect – Una solución optimizada para integrar los datos de mainframe a la empresa.

Para empresas que no tienen bases de datos Sybase como base de sus sistemas operacionales, pero que aún así requieren integración y replicación de datos, Sybase tiene dos soluciones:

Sybase Enterprise Connect – Una solución completa para la integración y replicación de sistemas de bases de datos No-Sybase.

Sybase Mainframe Connect – Una solución optimizada para integrar los datos de mainframe a la empresa.

La principal causa de la preferencia de Visual Basic y Access como fronts y base de datos, es que en principio CONADIC arrancó el proyecto con un bajo presupuesto en cuanto a software se refiere. El sistema ya en producción será migrado a SQL para tener una mejor robustez en almacenamiento de datos Y se conservará los front-end con Visual Basic.

2.6.1 ANÁLISIS DE REQUERIMIENTOS.

2.6.2 CARACTERÍSTICAS DEL PERSONAL DEL DEPARTAMENTO.

El personal que operará el sistema será capacitado para obtener y dar información en forma eficiente, para cubrir estas necesidades se debe cumplir con los perfiles siguientes:

OPERADOR DE LÍNEA DE INFORMACIÓN

Escolaridad: Pasante de licenciatura en psicología o licenciatura en Trabajo Social o técnico en Trabajo Social.

Conocimientos: Conocimientos básicos para la intervención en áreas de salud, para orientación y prevención de enfermedades, de técnicas de entrevista y manejo de PC.

SUPERVISOR DE LÍNEA DE INFORMACIÓN

Escolaridad: Licenciatura en psicología (titulado)

Conocimientos: Conocimientos de psicología clínica, intervención en crisis, básicos para la intervención en áreas de salud, básicos para la atención y prevención de enfermedades, de técnicas de entrevista, manejo de PC y de grupos de trabajo.

SUBDIRECTOR

Escolaridad: Licenciatura en psicología Clínica titulado (indispensable).

Conocimientos: Conocimientos de psicología clínica, de técnicas de intervención en crisis y terapia breve, manejo de PC y de grupos de trabajo, conocimientos en atención telefónica, instrumentación de programas de capacitación, conocimientos de

Intervención en áreas de salud, básicos para orientación y prevención de enfermedades y de técnicas de entrevista.

JEFE DE DEPARTAMENTO

Escolaridad: Licenciatura en Psicología (titulado).

Conocimientos: Conocimientos de técnicas de intervención en crisis y terapia breve, manejo de PC y de grupos de trabajo, conocimientos en atención telefónica, implementación de programas de capacitación, conocimientos de intervención en áreas de salud, básicos para orientación y prevención de enfermedades de técnicas de entrevista.

INGENIERO DE CÓMPUTO

Escolaridad: Ingeniero en sistemas computacionales (titulado).

Conocimientos: Administración de base de datos, SQL Server 6.5, comunicaciones y redes, manejo de herramientas visuales, MS-DOS, WINDOWS 95 y OFFICE 97, equipos PC y servidores.

CAPTURISTA

Escolaridad: Sexto semestre de licenciatura en informática / ingeniería en sistemas computacionales o carrera a fin.

Conocimientos: Manejo de MS-DOS, WINDOWS 95 Y OFFICE 97

SECRETARIA-CAPTURISTA

Escolaridad: Carrera técnica o comercial.

Conocimientos: Manejo de PC, WINDOWS 95 y OFFICE 97, impresoras, fax, módem, mecanografía y básicos de taquigrafía

2.6.2 INFRAESTRUCTURA DEL EQUIPO

En el departamento se cuenta con el equipo necesario para concentrar y manipular la información facilitando su acceso y mantenimiento, ya que es un equipo de gran capacidad de procesamiento y almacenamiento, las características técnicas del equipo son:

Servidor: COMPAQ PROLIANT 3000, 128 RAM, 3 HD de 5 Gigantes en arreglo Raid 5, Tarjeta Smart array, Sistema operativo, WINDOWS NT 4.0

Switch: 3 COM linkswitch 1100

Impresora: HP Láser jet 5

Estación de trabajo: COMPAQ 4000N, 32 RAM, 2 GB disco Windows NT y tarjeta de red integrada.

Conmutador Harris: MAP 20-20, 128 tos, reléase 23, alimentación CD 50 agentes de ACD y reportes ACD, Sistema de estadísticas EAS.

No brake: Sistema de fuerza ininterrumpible de 12 KVA con 30 minutos de respaldo a 100 % de carga.

Grabadora Harris: Grabadora multicanal, con una unidad de grabación de un canal, expandible a más anuncios por un canal.

Sistema de telefonía informatizada (correo de voz): Sistema de telefonía Vox de 4 puertos con computadora Pentium a 166 Mhz. con software Vox para aplicación Biper y licencia de uso de programa.

2.6.2.1 Establecimiento de Criterios de Selección

Los criterios de selección del equipo que consideramos son los siguientes, dado que se necesita un sistema de información automatizado que brinde un servicio oportuno y eficiente dando solución al problema, esto permitirá una atención al público de mejor calidad satisfaciendo sus necesidades como podrá ser información y canalización.

Económico: Enfocado especialmente en minimizar los gastos en la compra de equipo debido a que se empleará el equipo ya existente en el departamento.

Facilidad de uso: El manejo del equipo sea sencillo, aún más cuando en este caso el sistema será manejado por personal capacitado para esta tarea.

Flexibilidad para el crecimiento: Se requiere que el sistema permita el crecimiento de los recursos de cómputo de acuerdo a las distintas necesidades que se vayan presentando.

Factor de entrenamiento: Que sea fácil la capacitación en el manejo y administración de los recursos, ya que puede salir en ocasiones más caro la capacitación del personal que el sistema mismo, sino se tiene cuidado.

Factor de dependencia: Que el equipo que sea seleccionado no dependa de terceras personas para su buen funcionamiento, debido a cualquier falla que se presente la pueda resolver la persona encargada en ese momento.

2.6.3 CONSIDERACIONES DE SOFTWARE

La selección de las características y la arquitectura de una herramienta de diseño de sistemas que cumpla con los requisitos, ventajas y desventajas, no es un problema fácil de resolver. En los últimos años han ido apareciendo en el mercado múltiples productos de bases de datos, tanto para microcomputadoras como para grandes sistemas, haciendo de estos un vehículo común ampliamente utilizado en entornos muy diversos de máquinas y sistemas operativos. El objetivo principal de la producción de software es el de resolver los problemas de tipo empresariales, técnicos, etc. a los que se enfrentan frecuentemente las empresas y/o instituciones.

Para la realización de este trabajo se tomaron en cuenta diferentes alternativas con características similares, los cuales tienen ciertas diferencias que afectan el modo de desarrollar las aplicaciones en cada uno de los sistemas. En este capítulo se resumen las principales diferencias que existen entre ellos, con el fin de elegir el más adecuado a los requerimientos, necesidades y restricciones que se han venido planteando a lo largo de este trabajo.

VISUAL BASIC 6.0: Diseñado para crear aplicaciones generales de Microsoft Windows, a diferencia de Access que fue diseñado para crear y manipular bases de datos.

Visual Basic permite crear una aplicación. EXE autónoma, se necesita la biblioteca de tiempo de ejecución de Visual Basic para ejecutarlas, a diferencia de Access que permite crear aplicación de tiempo de ejecución, que es un archivo de base de datos .MDB que se ejecuta utilizando el archivo ejecutable runtime de Access, disponible en el Access Developer's Toolkit.

Visual Basic 6.0 presenta ADO o DAO en español como el nuevo estándar para acceso a datos con la desventaja frente a Access de que no permite mejoras a objetos de acceso a datos. Entre los controladores OLE DB que se incluyen están SQL Server 6.5, Oracle 7.3.3, Microsoft Access, ODBC y SNA Server.

Visual Basic 6.0 Proporciona también un conjunto completo de herramientas para integrar bases de datos con cualquier aplicación. Entre las características de bases de datos se incluyen herramientas de diseño para crear y modificar bases de datos de SQL Server 6.5, Oracle 7.3.3 o superior y AS/400.

VISUAL FOXPRO 6.0: Proporciona las herramientas necesarias para crear y administrar componentes y aplicaciones de bases de datos de 32 bits de alto rendimiento. Sus sólidas herramientas y el lenguaje orientado a objetos de datos céntricos lo hace ideal para generar aplicaciones multicapa modernas y escalables que integran la computación cliente - servidor e Internet.

Las mejoras en el entorno de desarrollo y en el grupo de herramientas proporcionan al programador una gran productividad y capacidad de personalización. Proporciona bibliotecas de clases reutilizables previamente generadas, lo que permite a los programadores agregar funcionalidades comunes a las aplicaciones, como manipulación de datos, detección actualizada de conflictos, búsqueda y recuperación de datos.

Al igual que Access y Visual Basic la versión reciente son más fáciles de usar y aprender facilitando el trabajo a los usuarios, lo que reduce el tiempo de aprendizaje de los programadores de versiones anteriores. Utilizando su galería de componentes para mejorar las aplicaciones, solo hay que arrastrar y colocar objetos en el proyecto desde esta Galería de la misma manera que en Access.

Microsoft Access 97 para los sistemas operativos Windows 95 y Windows NT proporciona la eficacia de las bases de datos relacionales siendo este último el sistema operativo que se va a utilizar. Facilita la información que necesita para tomar mejores decisiones. Integra datos de hojas de cálculo y otras bases de datos, y constituye una forma fácil de buscar respuestas, compartir información en redes internas (INTRANET) y en Internet, ya que este es un factor importante para el buen funcionamiento del sistema desarrollado.

Diseñado para crear y manipular bases de datos Access 97 Admite informes personalizados para una base de datos a diferencia de Visual Basic que sólo admite informes personalizados en la Professional Edition.

Access admite el lenguaje de macros y el lenguaje Access Basic a diferencia de Visual Basic que no permite el lenguaje de macro. La implantación del lenguaje Basic para Access y Visual Basic es muy similar, pero no idéntica.

Prácticamente no se necesita nada de código para enlazar controles a los orígenes de datos. Para conectar el control a cualquier origen de datos sólo es necesario configurar dos propiedades en la ventana Propiedades, siendo esta una gran ventaja para facilitar su uso a usuarios nuevos.

Permite arrastrar y colocar rápidamente controles personalizados enlazados a datos para crear formularios o informes sin demasiadas horas de esfuerzo. Características como el Asistente para Ayuda que facilitan la búsqueda de respuestas a preguntas acerca de la utilización de Access 97, ayudando a los usuarios a

aprovechar al máximo sus herramientas de software a diferencia de PARADOX que no cuenta con una ayuda que nos permita una búsqueda de respuestas a nuestras dudas.

Visual Basic está disponible en tres versiones, cada uno para un juego específico de requisitos para su desarrollo.

La edición de Aprendizaje de Visual Basic les permite a programadores crear aplicaciones poderosas fácilmente para Microsoft Windows 95 y Windows NT. Incluye todos los mandos intrínsecos, más la reja, etiqueta, y mandos del datos-límite. La documentación proporciona con el una edición que incluye Aprenda Visual Basic ahora, la Guía del Programador y la ayuda en línea.

La edición Profesional les proporciona un juego de herramientas, Incluye todos los rasgos de la edición de Aprendizaje, más los mandos de ActiveX adicionales, incluso los mandos de Internet, y el Escritor del Informe de crystal report. La documentación proporcionada con la edición profesional incluye la guía del programador, la ayuda en línea, la guía de herramientas de componentes y los Informes de crystal report para el manual de usuario de Visual Basic.

La edición de la Empresa permite crear aplicaciones distribuidas robustas. Incluye todos los rasgos de la edición Profesional, más el gerente de automatización, gerente del componente, herramientas de dirección de banco de datos, el Microsoft SourceSafe. La documentación impresa proporcionada con la edición de la Empresa incluye todos los pedazos de la edición profesional, más las Aplicaciones de Cliente/Servidor con el libro Visual Basic y la guía del usuario de sourcesafe.

Dentro de Visual Basic existe una gran herramienta llamada DAO para el acceso a bases de datos. Proporciona información detallada sobre el motor banco de datos, los datos controlan y acceden objetos (DAO) programando interfaces.

Visual Básico le permite crear y manipula datos que acceden a objetos en banco de datos, Campos y objetos.

DAO le permite escribir programas que acceden muchos tipos diferentes de bancos de datos, incluso bancos de datos nativos que usan el mismo formato como Access 97, los bancos de datos externos como FoxPro o Paradox y ODBC cliente/servidor bancos de datos como el Microsoft SQL Servidor.

Dentro del DAO existe una herramienta muy poderosa llamada Recordset que referencia a un objeto diseñado por Visual Basic para acceder a los datos de una base.

Recordset de tipo Dynaset (hoja de respuestas dinámica) permite filtrar, ordenar, extraer y actualizar datos de más de una relación; es un conjunto dinámico de registros.

Recordset de tipo Snapshot (hoja de respuestas instantánea) es un objeto estático de registros; Los datos obtenidos la primera vez no se actualizan dinámicamente, razón por la cual son más rápidos.

A diferencia de los objetos Recordset de tipo Dynaset o Snapshot, el objeto Recordset de tipo Table no puede hacer referencia a más de una relación base y no puede ser accedido utilizando una sentencia SQL que filtre u ordene los datos; cualquier modificación de los datos se refleja automáticamente.

2.6.3.1 Establecimientos de Criterios de Selección

Algunas de las características más importantes que requerimos del manejador de bases de datos a seleccionar son las siguientes:

- Que el paquete sea confiable y eficiente en el manejo de grandes cantidades de información.
- Que presente facilidad para el manejo en usuarios casuales del sistema final debido a la parte referente al árbol de información, así como versatilidad de herramientas para las pantallas de captura e informes referentes a la parte de organismos.
- Que no utilice muchos recursos.
- Que su costo no sea muy elevado.

Para la comparación y evaluación de estos manejadores de bases de datos se consideraron los siguientes criterios generales:

La integración de los datos de ésta fuente permitirá la divulgación a todas las áreas del consejo, ya sea en pantalla o mediante la elaboración de informes que apoyen al personal.

- Compatibilidad con versiones anteriores de los programas.
- Calidad de las lecciones guiadas y de la ayuda en línea (F1).
- Claridad y buena organización de la documentación que se imprima, debido a su importancia para la divulgación a todas las áreas del Consejo.

- Las necesidades de diseño, desarrollo, e implantación de programas especializados de bases de datos y el grado de complejidad que se necesita para facilitar el procesamiento y manipulación de datos en materia de alcoholismo, tabaquismo y farmacodependencia.
- La necesidad de automatización del servicio que se presta para lograr un servicio de mejor calidad en la institución.
- La facilidad en el manejo y corrección de errores para los casos de fallas por parte del usuario.
- Bajo costo.

Deducimos que todos los paquetes son flexibles en su manejo debido a que cuentan con herramientas prácticamente completas conteniendo la documentación adecuada y explícitamente documentada que permite un mejor y fácil manejo de cada uno de ellos.

Microsoft Access 97 ofrece varias mejoras significativas en el rendimiento.

| | |
|---|--|
| Retardo en la carga. | Microsoft Access solo carga los componentes de software que son necesarios para las bases de datos, como Visual Basic para aplicaciones y DAO, cuando son necesarios. Esto reduce el tiempo que tarda en cargar las bases de datos y mejora el rendimiento global. |
| Formularios e informes simplificados. | Puede abrir muchos formularios e informes de manera más rápida, porque los formularios e informes que no tienen procedimientos de evento ya no incluyen un módulo de formulario o informe. |
| Propiedad Finalizar al producirse un error (FailOnError). | Puede optimizar las consultas de actualización en gran volumen para orígenes de datos ODBC enviando la consulta al servidor, donde todos los registros apropiados se procesan a la vez en lugar de registro por registro. |

| | |
|--|--|
| Carga sin llamadas mejorada. | Microsoft Access no carga los módulos, incluidos los módulos de formulario, hasta que se ejecuta el código de Visual Basic del módulo. Esto mejora el rendimiento global. |
| Administración de estado compilado mejorada. | Puede mantener el estado compilado de la base de datos incluso si la modifica. Sólo se descompilará el código modificado y aquel código que dependa del código modificado. |
| Velocidad de los controles Activex. | Microsoft Access ha mejorado el rendimiento de los controles ActiveX incrustados. |
| Cuadros combinados más rápidos. | Microsoft Access ha mejorado el rendimiento de los cuadros combinados en formularios. |
| Control de imagen. | Puede mostrar imágenes independientes en los formularios e informes mucho más rápido utilizando un control de imagen en lugar de un marco de objeto independiente. |
| Analizador de rendimiento | Este asistente analiza los objetos de base de datos y proporciona sugerencias para acelerar y mejorar su funcionamiento. |
| Vista previa de informes más rápida. | Puede ver los informes más rápidamente, porque los eventos sólo se activan la primera vez que pasa las páginas de un informe. |

Microsoft Access proporciona muchas características nuevas diseñadas para ayudar a utilizar fácilmente Internet y programar aplicaciones para el World Wide Web.

Necesita un explorador de Web, tal como Microsoft Internet Explorer y un módem, una conexión de Intranet u otra conexión de red para tener acceso a Internet y aprovechar algunas de estas características.

| | |
|------------------------------------|--|
| Importar o vincular archivos HTML. | Importar o vincular relaciones (de sólo lectura) o listas desde un archivo HTML utilizando el comando Obtener datos externos del menú Archivo. |
| Exportar objetos a formato HTML. | Exportar informes a formato HTML estático, y hojas de datos y formularios a formato HTML estático o dinámico utilizando el comando Guardar como o exportar del menú Archivo. Mejore la apariencia, la coherencia y el desplazamiento por sus páginas Web mediante un archivo de plantilla de HTML. |

| | |
|------------------------------------|--|
| Asistente para publicar en el Web. | Crear una aplicación de Web con el Asistente para publicar en el Web mediante el comando Guardar como HTML del menú Archivo. Puede enviar hojas de datos, formularios o informes a formato HTML estático o dinámico utilizando uno o varios archivos de plantilla de HTML; crear una página principal; almacenar todos los archivos en una carpeta especificada como una publicación de Web, copiar los archivos a un servidor Web utilizando el Asistente para publicación en el Web y guarde un perfil de publicación de Web para utilizar posteriormente. |
| Acceso a servidores FTP Y HTTP. | Importar o vincular datos (de sólo lectura) en servidores FTP o HTTP, exportar un objeto de base de datos a un servidor FTP y agregar o modificar una lista de ubicaciones de servidores FTP mediante el cuadro Buscar en el cuadro de diálogo Abrir. |
| Hipervínculos. | Almacenar un hipervínculos en una tabla. Agregar la dirección del hipervínculo en una hoja de datos o formulario utilizando el botón Insertar hipervínculo. Un hipervínculo puede ser una ruta de acceso a un archivo en la unidad de disco duro, una ruta de acceso UNC o una dirección URL. Cuando hace clic en un campo de hipervínculo, Microsoft Access salta a un objeto, un documento, una página Web u otro destino. |

| | |
|--------------------------------|--|
| Destinos. | <p>Saltar a una base de datos u objeto de Microsoft Access desde un hipervínculo en una hoja de datos, formulario o informe, o desde dentro de otros documentos de Microsoft Office, tales como un documento de Microsoft Word o una hoja de cálculo de Microsoft Excel. Saltar a un documento de Office desde un hipervínculo dentro de una hoja de datos.</p> <p>Formulario o informe. Hipervínculo en formularios e informes. Crear una etiqueta, control de imagen o botón de comando en un formulario o informe que salta a un destino de hipervínculo.</p> |
| Controles ActiveX. | Explorar World Wide Web, ver páginas Web, tener acceso a otros documentos en su PC, en la red o en Internet y transferir datos desde internet a través de un formulario de la aplicación. |
| Controles ActiveX adicionales. | Programar aplicaciones Web brillantes y dinámicas. Disponible en Microsoft Office 97, edición Developer. |
| Herramientas de Office. | Examinar fácilmente bases de datos y objetos de Microsoft Access o documentos de Office utilizando una funcionalidad similar a la de la barra de herramientas de Microsoft Internet Explorer. |

| | |
|--|---|
| Búsqueda rápida en el Web. | Ubicar bases de datos de Microsoft Access y otros documentos de Microsoft Office que están indexados mediante las propiedades de base de datos y archivo. |
| Réplica. | Sincronizar la réplica de una base de datos con una réplica o Diseño principal en un servidor de Internet. |
| Ayudante de Office. | El ayudante de Office le ofrece Ayuda para su programa de Microsoft Office. Cuando necesite Ayuda, haga clic en el botón Ayudante de Office. El ayudante proporciona de forma automática temas de Ayuda relacionados con aquello en lo que se encuentra trabajando, incluso antes de que haga alguna pregunta. También puede hacer una pregunta determinada escribiéndola en el Ayudante. Puede elegir un Ayudante que se adapte a su personalidad y a su forma de trabajar. El ayudante es compartido por todos los programas de Office, por lo que cualquier opción que cambie afectará al Ayudante de todos los programas de Office. |
| Seguimiento de la actividad de la base de datos con Outlook Journal. | Outlook Journal se utiliza para realizar el seguimiento de cuando se abrió o cerró una base de datos, o cuando se imprimió un objeto. Para obtener detalles, consulte la Ayuda que se proporciona con Microsoft Outlook. |

| | |
|--|--|
| Obtener los contactos de Outlook desde una base de datos de Microsoft Access. | Se puede importa datos desde una base de datos de Microsoft Access a la carpeta de contactos de Outlook. También puede volver a exportar datos de Outlook a Microsoft Access. Para Obtener detalles, consulte la Ayuda que se proporciona con Microsoft Outlook. |
| Retención del formato del texto. | Cuando se copian o cortan datos desde formularios y hojas de datos de Microsoft Access, se retiene los valores de la fuente, alineación y formato de los números para los encabezados de las columnas y los datos cuando se pegan en hojas de cálculo de Microsoft Excel. |
| Envío de subinformes a otros formatos de archivo. | Cuando se envía un informe con un subinforme a un formato de archivo .txt, .rtf, o .xls, el subinforme se envía junto con el informe principal. |
| Formato e impresión de datos de Microsoft Excel en Informes de Microsoft Access. | Cuando se necesita la funcionalidad de agrupación y formato de los informes de Microsoft Access para los datos que tenga en hojas de cálculo de Microsoft Excel, puede hacer clic en informe de Access en el menú Datos de Microsoft Excel para iniciar el Asistente para informes de Microsoft Access. Para obtener los detalles, consulta la Ayuda que se proporciona con Microsoft Excel. |

| | |
|---|---|
| <p>Mover de forma permanente datos de Microsoft Excel a Microsoft Access.</p> | <p>Cuando una hoja de cálculo de Microsoft Excel deja de ser la herramienta apropiada para los datos, se puede crear una base de datos de Microsoft Access desde la hoja de cálculo haciendo clic en Convertir a Access en el menú Datos de Microsoft Excel. Para obtener los detalles, consulte la Ayuda que se proporciona con Microsoft Excel.</p> |
| <p>Creación de un control de tabla dinámica en un formulario de Microsoft Access.</p> | <p>Cree un control de tabla dinámica en un formulario de Microsoft Access. Mediante el asistente para tablas dinámicas, se puede crear un control en un formulario que permite resumir grandes cantidades de datos usando un formato y el método de cálculo que elija. Una tabla dinámica es como una consulta de tabla de referencias cruzadas, pero en la que puede cambiar dinámicamente los encabezados de las filas y las columnas para obtener una vista distinta de los datos. (Para obtener ventaja de este asistente, necesita tener instalado Microsoft Excel).</p> |
| <p>Vincular una hoja de cálculo.</p> | <p>Se pueden vincular datos de una hoja de cálculo Microsoft Excel o Lotus 1-2-3, siempre que los datos estén organizados en el formato de tabla apropiado.</p> |
| <p>Uso de Microsoft Access como un componente ActiveX.</p> | <p>Cuando se utiliza Microsoft Access como un componente ActiveX, se pueden manipular objetos de Microsoft Access desde otra aplicación, tal como Microsoft Excel.</p> |
| <p>Arrastrar y colocar objetos de base de datos.</p> | <p>La siguiente funcionalidad de arrastrar y colocar está ahora disponible en Microsoft Access:</p> <ul style="list-style-type: none"> • Puede arrastrar y colocar objetos de base de datos entre bases de datos abiertas de Microsoft Access. (Debe tener en ejecución dos instancias de Microsoft Access). • Puede arrastrar, colocar tablas y consultas de Microsoft Access en otras aplicaciones, tal como Microsoft Word y Microsoft Excel. • Puede crear una tabla arrastrando y colocando un rango de celdas desde una hoja de cálculo de Microsoft Excel a la ventana de base de datos. • Puede arrastrar y colocar objetos OLE en un campo de un formulario en la vista. |

Microsoft Access 97 incorpora las siguientes características nuevas para hacer aún más fácil el trabajo con los datos.

| | |
|---|---|
| <p>Actualización de los datos en un cuadro de lista o cuadro combinado de un campo de búsqueda.</p> | <p>Para mejorar el rendimiento, los datos del cuadro de lista o cuadro combinado de un campo de búsqueda que provienen de otra tabla ya no se actualizan de forma automática cuando se actualiza la otra tabla. Puede actualizar los datos del campo búsqueda presionando la tecla F9.</p> |
| <p>Nueva interpretación de las fechas introducidas en un formato de año reducido.</p> | <p>En los controles de hojas de datos y de modificación, Microsoft Access interpreta ahora de la siguiente forma las fechas introducidas con un formato de año reducido, para significar el siglo veintiuno.</p> |
| <p>Intervalo de fechas para el formato de año reducido interpretación.</p> | <p>1/1/00 hasta 12/31/29 1/1/2000 hasta 12/31/2029</p> <p>1/1/30 hasta 12/31/99 1/1/1930 hasta 12/31/1999</p> |
| <p>Utilice réplica de base de datos para sincronizar múltiples copias de la base de datos.</p> | <p>Mediante la réplica, múltiples usuarios en distintas ubicaciones pueden trabajar en sus propias copias de una base de datos y luego sincronizar los datos comunes de toda las réplicas.</p> <p>Puede crear una réplica parcial que contenga sólo un subconjunto de los registros de una réplica completa usando un filtro para restringir los datos a sincronizar.</p> <p>Por ejemplo, puede mantener el diseño principal de la base de datos de ventas completa en las oficinas centrales y réplicas parciales de los datos regionales en las oficinas regionales. Los beneficios de este enfoque incluyen una mayor seguridad, réplicas menores de bases de datos, tráfico de red reducido, y menor costo de las telecomunicaciones.</p> |
| <p>Compatibilidad con IntelliMouse al desplazarse por una hoja de datos o formulario.</p> | <p>Si tiene IntelliMouse, puede utilizarlo para desplazarse en una hoja de datos o formulario.</p> |
| <p>Arrastre y coloque los datos o envíelos a otras aplicaciones.</p> | <p>Puede seleccionar los datos de un formulario u hoja de datos y arrastrarlos y colocarlos en Microsoft Word o Microsoft Excel. También puede seleccionar datos de un formulario u hoja de datos y guardarlos como texto, como texto enriquecido (que puede abrirse con Microsoft Word y con la mayoría de los procesadores de texto de Microsoft Windows y de los programas de diseño electrónico), o como una hoja de trabajo de Microsoft Excel.</p> |

| | |
|---|--|
| Compruebe la ortografía y corrija automáticamente los errores de escritura. | Puede comprobar la ortografía de todos los campos Texto o Memo o de entradas seleccionadas en hojas de datos y formularios. También puede hacer que Microsoft Access corrija de forma automática los errores comunes de ortografía y de escritura mientras escribe; y convertir a mayúsculas los nombres de los días de la semana. |
| Orden de los subformularios. | Ahora también se pueden ordenar los registros de una subformulario. |
| Guarde el orden de una tabla, consulta o formulario. | Cuando guarde una tabla, consulta o formulario, Microsoft Access guarde cualquier orden que haya establecido y lo vuelve a aplicar de forma automática cada vez que abra ese objeto. Si basa un nuevo formulario o informe en una tabla o consulta que tenga guardado un determinado orden con la misma, el formulario o informe o hereda el orden. Cada vez que abra el formulario o informe, Microsoft Access vuelve a aplicar ese orden de la forma automática. |
| Formato de hojas de datos. | Cuando guarde una tabla, consulta o formulario, Microsoft Access guarda cualquier orden que haya establecido y lo vuelve a aplicar de forma automática cada vez que abra ese objeto. Si basa un nuevo formulario o informe en una tabla o consulta que tenga guardado un determinado orden con la misma, el formulario o informe, Microsoft Access vuelve a aplicar ese orden de forma automática. |
| Formato de hojas de datos. | Las características de una hoja de datos se pueden cambiar de las siguientes formas: Cambiando la apariencia de las celdas para que aparezcan elevadas o hundidas, cambiando el color de las líneas de la cuadrícula, mostrando sólo las líneas de la cuadrícula horizontales o verticales o ambas. |
| Mostrando datos en un color y estilo de fuente distinta. | Animando las filas de tal forma que pueda verlas deslizarse hacia arriba y rellenar el hueco de un registro eliminado. Esta es la opción predeterminada, aunque puede desactivar la animación si lo prefiere. |

Microsoft Access 97 hace que filtrar sea más fácil incluso con las siguientes nuevas formas de filtrar los registros.

| | |
|----------------------|---|
| Filtrar por entrada. | Hacer clic con el botón secundario en un campo de formulario u hoja de datos y luego en el menú contextual, escribir en el cuadro Filtrar por el valor exacto que está buscando en ese campo, o una expresión cuyo resultado desea usar como criterios. |
|----------------------|---|

| | |
|---|--|
| Filtro por selección. | Directamente en el formulario u hoja de datos, seleccionar todo o parte de un valor, y luego hacer clic en el botón Filtro por selección en la barra de herramientas para buscar todos los registros que tengan el valor seleccionado. |
| Filtro excluyendo la selección. | Directamente en el formulario u hoja de datos, seleccionar todo o parte de un valor, y luego hacer clic con el botón secundario y luego hacer clic en Filtro excluyendo la selección en el menú contextual para buscar todos los registros que no contengan el valor seleccionado. |
| Filtro por formulario. | Especificar el valor o valores que está buscando en una vista del formulario u hoja de datos con campos en blanco (en lugar de datos). Especifiquen un valor que estén buscando escribiéndolo o eligiéndolo de una lista del campo. Nota todavía puede crear filtros avanzados en la ventana filtro u orden avanzado usando la cuadrícula de diseño, llamada anteriormente cuadrícula consulta por ejemplo (Query By Example , QBE). |
| Filtrar registros de una tabla o consulta. | Usando cualquiera de las técnicas de filtrado, ahora puede filtrar los registros mostrados en una vista Hoja de datos de tabla o consulta. |
| Guardar un filtro con una tabla, consulta o formulario. | El filtro se guarda cuando se guarda la tabla, consulta o formulario, sin embargo, no se aplica de forma automática cuando se vuelven a abrir cualquiera de estos objetos. Se aplican cuando sean necesarios. |
| Basar un nuevo formulario o informe en los datos filtrados. | Puede crear un formulario o informe directamente desde un conjunto filtrado de registros al mismo tiempo que se muestran los registros en una tabla abierta o consulta, o puede basar el formulario o informe en una tabla cerrada o consulta que se haya guardado con un filtro. En cualquier caso, el formulario o informe hereda el filtro. |
| Aplicar, eliminar o cambiar un filtro o un orden heredado. | Las siguientes propiedades le permiten manipular un filtro u orden que haya heredado un informe: Filtro (filter), ActivarFiltro (filterOn), OrdenarPor (OrderBY), y ActivarOrden (OrderByOn). Filtro (filter) y OrdenarPor (ordenBy) se aplican también a los formularios, tablas, y consultas; sin embargo, estas propiedades se pueden cambiar usando las características de filtro y orden en lugar de establecer las propiedades directamente. |
| Aplicar un filtro a un subformulario. | El filtro limita los registros del subformulario sin limitar los registros del formulario principal. |

Microsoft Access 97 ofrece las siguientes mejoras para la impresión y vista preliminar de los objetos de base de datos.

| | |
|--|---|
| <p>Características de la vista preliminar.</p> | <ul style="list-style-type: none"> • El botón vista de la barra de herramientas hace que sea fácil cambiar a las demás vistas del objeto de base de datos. • El botón una página muestra una página. • El botón dos páginas muestra dos páginas. • El botón varias páginas le permite seleccionar el número de páginas de la vista preliminar. • El cuadro Zoom enumera el porcentaje de zoom en el que puede ver el objeto. • Si tiene Microsoft IntelliMouse, puede usar el botón redondo para moverse por la página actual. |
| <p>Comandos y cuadros de diálogo.</p> | <p>Los siguientes comandos y cuadros de diálogo se han rediseñado o se les ha cambiado el nombre para que sean más fáciles de usar y para que coincidan con los utilizados en las demás aplicaciones de Microsoft Office:</p> <ul style="list-style-type: none"> • El cuadro de diálogo imprimir es el mismo que se utiliza en las demás aplicaciones de Microsoft Office. • El comando configurar impresión del menú archivo se llama ahora configurar página. • El botón imprimir de la barra de herramientas imprime el objeto seleccionado sin mostrar el cuadro de diálogo imprimir. • El comando muestra preliminar se llama ahora vista previa del diseño y está disponible en el menú ver y haciendo clic en la flecha junto al botón vista de la barra de herramientas. • El comando definición de impresión se llama ahora documentador y está disponible apuntando al comando analizar del menú herramientas. |

Tomando en cuenta los criterios de selección, el de menor costo es \$7,000. Aunque Access se puede concluir que es el manejador más básico ya que lleva a los usuarios de la mano y conjuntamente con visual basic que es una herramienta gráfica muy poderosa hacen a los sistemas poderosos en acceso de datos y vista.

Visual basic nos hace la programación más fácil ya que la mayoría de sus funciones y procedimientos están automatizados. Y sus controles traen una cierta cantidad de propiedades, fáciles de modificar. Además su entorno de desarrollo

totalmente visual, facilita la creación de aplicaciones, sumado a esto la compatibilidad que tiene con Windows.

Cabe mencionar que se cuenta ya en el CONADIC con el manejador Access y visual basic, además de contar con el equipo adecuado para su buen funcionamiento evitando así que exista problema alguno en cuanto al factor económico, por otro lado emplea una configuración mínima de software para su funcionamiento, además debido a su característica de llevar a los usuarios casi de la mano permite una mayor facilidad de uso increíble.

De acuerdo con lo anterior y tomando en cuenta las ventajas y desventajas que presenta cada uno de estos paquetes de software, y analizando los requerimientos que necesitamos para la elaboración del presente trabajo para dar solución al problema planteado durante su desarrollo y tomando en cuenta el ahorro en cuanto a costo se refiere, siendo este uno de los factores importantes para la toma de decisiones.

En este aspecto, el manejador de bases de datos que mejor se adapta a este proyecto, es Access y visual basic como soporte gráfico, y son los paquetes seleccionados finalmente para el desarrollo del sistema que se implantará físicamente en este departamento.

2.7 RESTRICCIONES DEL SISTEMA.

- Limitaciones en los proyectos: Un solo proyecto puede contener hasta 32,000 identificadores que incluyen formas, controles módulos, variables, constantes, procedimientos, funciones, y objetos. Las constantes en Visual Basic no pueden tener más de 255 caracteres y los nombres de formas, controles, módulos y clases no pueden ser más largos de 40 caracteres. Visual Basic no impone ningún límite en el número real de objetos distintos en un proyecto.
- El número de registros esta sujeto a la capacidad de Access.

CAPÍTULO III
ESTRATEGIA DEL SISTEMA.

CAPÍTULO III

ESTRATEGIA DEL SISTEMA

3.1 TEORÍA DE LAS BASES DE DATOS.

En el lenguaje coloquial, con la expresión base de datos se designa una colección de datos que es administrada por un sistema de administración de bases de datos, que se abrevia DBMS (Data Base Management System, sistema de administración de bases de datos o simplemente sistema de bases de datos).

Las bases de datos de uso frecuente siempre se mantienen en algún orden determinado: alfabético, cronológico, por población o quizá por código postal, etc. Este ordenamiento lo realiza el programador al crear la estructura de la base de datos para que resulte más fácil trabajar con ella.

Normalmente, en una base de datos de información, ésta se organiza y se mantiene en una relación compuesta por filas y columnas (esta es la base de todos los DBMS actuales). Las filas de datos conocidas registros se llaman tuplas y las columnas conocidas como campos se les llama atributos. Así, se podrá notar que cada tupla contiene varios atributos. Un atributo es un elemento componente de una tupla. La totalidad de los atributos integran una tupla.

Los trabajos relacionados con el manejo de una base de datos, cualquiera que sea su tipo y finalidad, si se realizan a través de una computadora, resultan más prácticos y cómodos. Esto es porque la computadora lleva a cabo las tareas tediosas y pesadas de administración de la base de datos y lo hace rápido.

En la actualidad existe una gran cantidad y variedad de software que se ajusta a cada una de las necesidades enfocadas a la automatización y sobre todo al guardado de grandes cantidades de información.

Existe una diferencia importante entre un administrador de bases de datos humano y un sistema de manejo de bases de datos computarizado. De ahí que el objetivo principal de este trabajo sea de sustituir esos laboriosos trabajos manuales combinados con demasiado papel, por lo tanto, se trata de automatizar la información y acabar con los archivos manuales que son tediosos y que consumen mucho tiempo.

3.1.1 OBJETIVO DEL SISTEMA ADMINISTRADOR DE BASES DE DATOS.

Los sistemas administradores de bases de datos permiten a los diseñadores estructurar la información y así manejar cantidades muy voluminosas de datos y efectuar muchas operaciones simultáneas en ellos.

Se espera que un sistema:

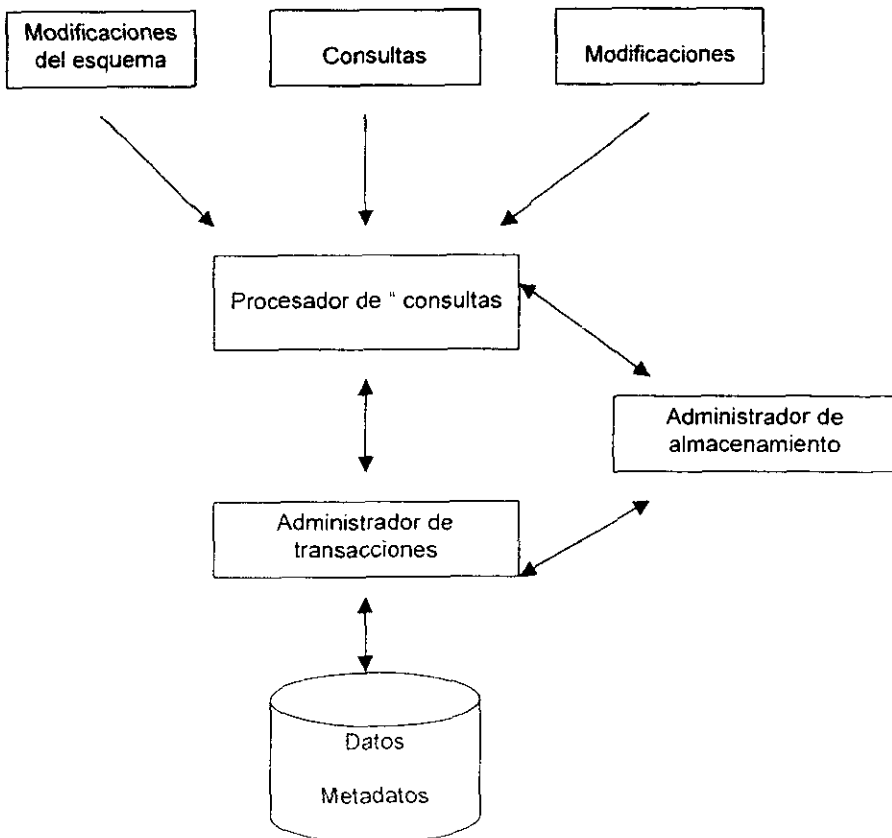
1. Permita a los usuarios crear otras bases de datos y especificar su esquema (estructura lógica de los datos) por medio de un lenguaje especializado denominado lenguaje de definición de datos.
2. Ofrezca a los usuarios la capacidad de consultar los datos (una consulta es un tecnicismo de base de datos que formula una pregunta sobre los datos) y modificarlos, usando para ello un lenguaje apropiado, llamado a menudo lenguaje de consulta o lenguaje de manipulación de datos.
3. Soporte el almacenamiento de cantidades muy voluminosas de datos durante un largo periodo, protegiéndolos contra accidentes o utilización no autorizada y permitiendo el acceso eficiente para hacer consultas y modificar la base de datos.

-
4. Controle el acceso simultáneo a los datos por parte de muchos usuarios, sin permitir que las acciones de uno de ellos afecten a los otros ni que los accesos simultáneos corrompan los datos accidentalmente.

3.2 ARQUITECTURA DE UN SISTEMA DE ADMINISTRACIÓN DE BASES DE DATOS (DBMS).

En la presente sección se describe brevemente la estructura de un sistema de administración de bases de datos (DBMS)

Principales componentes de un sistema de administración de bases de datos:



En la figura anterior se muestran las partes esenciales de un sistema de administración de bases de datos. Por convención, los componentes en forma de disco indican un lugar para almacenar datos. Nótese la palabra metadatos son aquellos datos que contendrán los nombres de las relaciones, los nombres de los atributos de éstas y los tipos de datos de los atributos.

El administrador del almacenamiento consta de dos componentes: el administrador de búfer y el administrador de archivos.

1. *El administrador de búfer.*- Se encarga de la memoria principal. Obtiene bloques de datos del disco a través del administrador de archivos y selecciona una página de la memoria.
2. *El administrador de archivos.*- Lleva un control de la localización de los archivos en el disco y a solicitud del administrador de búfer, obtiene el bloque (o bloques) que contiene uno de ellos.

El administrador de consultas se encarga de convertir una consulta o manipulación de la base de datos, que puede estar expresada en un nivel muy alto, en una serie de peticiones de datos almacenados, como determinadas tuplas de una relación o las partes de un índice en una relación.

El administrador de transacciones se encarga de se ejecuten varias transacciones de manera simultánea, existen tres requisitos para su ejecución:

1. **Atomicidad:** Deberá realizarse la transacción completa o nada de ella.
2. **Consistencia:** Una base de datos generalmente tiene un concepto de un "estado coherente", en el cual la información cumple las expectativas que podamos tener. Por ejemplo una situación coherente apropiada de la base de datos de una línea aérea es que un mismo asiento no sea asignado a dos pasajeros.

-
3. **Aislamiento:** Cuando dos o más transacciones son ejecutadas al mismo tiempo, es preciso aislar sus efectos.
 4. **Durabilidad:** Si la transacción ha cumplido su función su efecto no debería perderse en caso de una falla del sistema, incluso si la falla sobreviene apenas concluida la transacción.

3.3 ESTRUCTURA DE LA BASE DE DATOS RELACIONAL DEL SISTEMA.

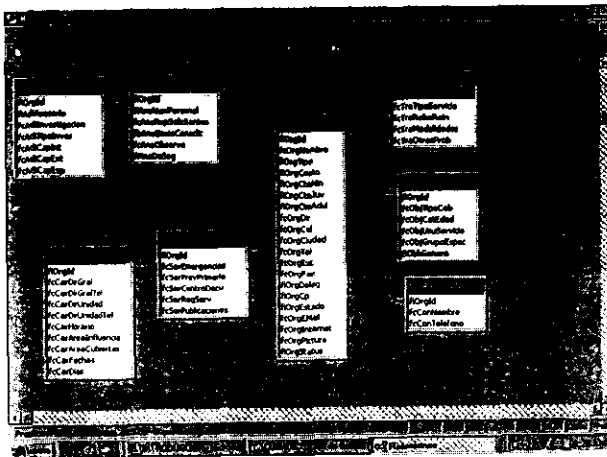
Una base de datos relacional es un sistema administrador de bases de datos que manipula información enlazándolas o relacionándolas entre sí por un campo común.

Las características del modelo relacional son tres:

1. **Simplicidad:** Las relaciones son de una forma familiar y explicables por sí mismas para representar. La mayoría de la gente ha utilizado datos en forma de relación, no se requiere un entrenamiento especial para entender y utilizar los datos que se representan en las relaciones. En pocas palabras son amigables a los usuarios.
2. **Precisión:** Las relaciones correctamente diseñadas mantienen un rigor matemático y dicen lo que significa, pueden ser implementadas y procesadas por una variedad de configuraciones de software y hardware. En pocas palabras son amigables a la computadora.
3. **Flexibilidad:** Las relaciones no solo muestran la estructura de los datos sino que pueden mostrar los datos también: Esto nos permite manejar el modelo antes de implementarlo. En otras palabras las relaciones son apropiadas no solo para modelar datos sino para procesarlos también.

El modelo relacional nos ofrece una manera única de representar los datos: como una relación bidimensional denominada relación.

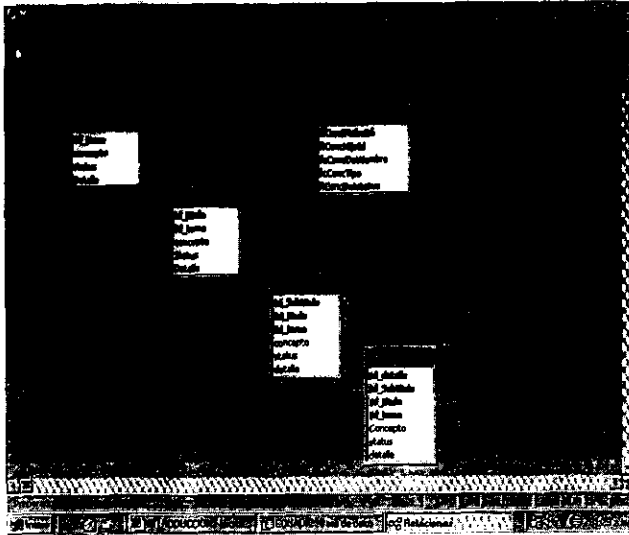
En este sistema que se desarrolló existen relaciones de muchos a uno a la relación principal de datos generales por medio de un campo (llave) que se llama fiorgid en la parte de organismos como se muestra a continuación:



El modelado de relaciones tiene tres importantes consideraciones como son:

1. Descubrir relaciones, es decir, concentrarse en Verbos.
2. Definir el alcance de la relación, definir si la relación es de interés para el sistema.
3. Definir el tipo de relación, 1 a 1, muchos a 1 y 1 a muchos, muchos a muchos.
4. Documentarlas en el diagrama Entidad-Relación.
5. Documentarlo en relaciones.

En la parte del árbol de información la relación es en cascada, ya que cada una de las relaciones necesita estar relacionada con la anterior para guardar datos y las llaves se relacionan de la siguiente forma:



3.4 MODELADO DE ATRIBUTOS.

Para determinar el modelado de atributos se toman en cuenta las siguientes consideraciones:

1. Describir atributos, concentrarse en adjetivos o adverbios.
2. Definir el alcance del atributo, decidir si el atributo es de interés al sistema.
3. Determinar una llave primaria para el atributo, Determinar cual es la mejor opción para colocar el atributo en una relación (eliminar duplicidad colocando los atributos de manera correcta).
4. Documentar el atributo en alguna relación.

Los métodos para colocar atributos son:

1. Método gramatical (por experiencia).
2. Normalización.

3.5 NORMALIZACIÓN.

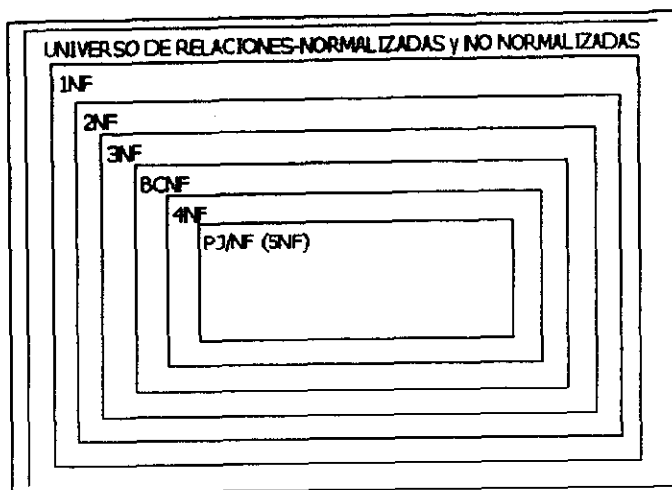
Normalización es el proceso de convertir una relación(TABLA) en otras de forma tal que se cumplan ciertas restricciones.

El proceso de normalización es básicamente obtener un conjunto de relaciones a partir de una de ellas utilizando proyecciones de la relación original.

Una característica fundamental de dichas proyecciones es que sean sin perdida ni ganancia, esto se refiere a que debe ser posible deducir de las relaciones proyectadas la información original, ni más ni menos.

La teoría de la normalización está basada en el concepto de formas normales, todas las relaciones normalizadas están en 1FN; algunas relaciones en 1FN también están en 2FN; y algunas relaciones en 2FN también están en 3FN. Por lo general, los diseñadores, debe escoger relaciones en 3FN para diseñar una base de datos.

La definición original de Codd acusaba ciertas insuficiencias y por lo tanto se dio una definición revisada mas fuerte en el sentido que cualquier relación que estuviera en 3FN según la nueva definición estaba ciertamente en 3FN según la definición antigua, y no según la definición nueva. La 3FN nueva a veces se llama Forma Normal de Boyce/Codd (FNBC) para distinguirla de la forma antigua. Mas tarde Fagin definió una cuarta forma normal (4FN) nueva y, más recientemente, otra forma normal que llamó forma normal de proyección-reunión (FN/PR, también conocida como 5FN).



Formas Normales.

Primera forma normal (1FN).

Una relación esta en PRIMERA FORMA NORMAL(1NF) si todos los dominios base de los campos tienen valores atómicos, es decir un campo solo tiene un valor y no un conjunto de valores.

Tomemos como ejemplo el caso de una relación para llevar la información de materias y calificaciones de materias de alumnos, un posible diseño se ilustra en las relaciones:

| MATRICULA | CALIFS |
|-----------|-----------------------------|
| 331678 | CB-001 10, MA-001 9 |
| 337890 | F-001 7, H-002 8 |
| 337777 | CB-002 7, CS-056 8 |
| 446789 | MA-031 7, CB-072-8, H-002 9 |

Ejemplo Materias y Calificaciones de Alumnos

Observando la relación nos damos cuenta que el dominio del campo MATRICULA si cumple la característica de tener valores atómicos, pero el campo CALIFS tiene el problema de que el dominio utilizado proporciona conjuntos de valores, es decir no esta en 1NF. Un posible mejor diseño sería cómo se ilustra en la relación.

| | |
|--------|-----------|
| 331678 | CB-001 10 |
| 331678 | MA-001 9 |
| 337890 | F-001 7 |
| 337890 | H-002 8 |
| 337777 | CB-002 7 |
| 337777 | CS-056 8 |
| 446789 | MA-031 7 |
| 446789 | CB-072 8 |
| 446789 | H-002 9 |

Mejor Diseño de Materias y Calificaciones de Alumnos

Aun así observamos que este diseño adolece del problema de que todavía el campo CALIF es un conjunto de valores(una clave de materia y una calificación) es decir todavía no esta en 1NF.

Un mejor diseño sería como se ilustra en la relación

| MATRICULA | CLAVE | CALIF |
|-----------|--------|-------|
| 331678 | CB-001 | 10 |
| 331678 | MA-001 | 9 |
| 337890 | F-001 | 7 |
| 337890 | H-002 | 8 |
| 337777 | CB-002 | 7 |
| 337777 | CS-056 | 8 |
| 446789 | MA-031 | 7 |
| 446789 | CB-072 | 8 |
| 446789 | H-002 | 9 |

Ejemplo de 1NF de Materias y Calificaciones de Alumnos

Revisando esta relación observamos que todos los dominios de los campos proporcionan valores atómicos, es decir ninguno de los campos tiene un conjunto de valores; por lo tanto esta relación ya está en 1NF.

Segunda forma normal (2FN).

Una relación está en 2NF si está en 1NF y además cumple la condición de que cada atributo no-llave depende de la llave primaria.

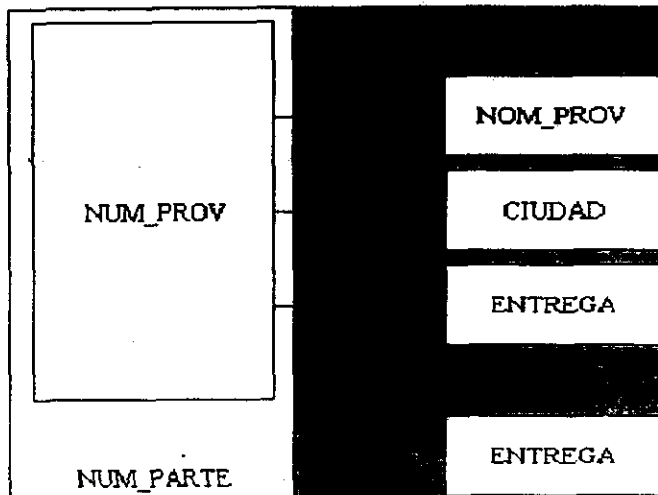
La llave primaria es el CONJUNTO MÍNIMO DE ATRIBUTOS que cumplen la característica de UNICIDAD y que se escogió para ser la LLAVE. Si existe más de un CONJUNTO MÍNIMO DE ATRIBUTOS QUE CUMPLEN LA UNICIDAD, al conjunto escogido como LLAVE se le llama LLAVE PRIMARIA y a los demás conjuntos se les llama LLAVES CANDIDATO.

Supongamos que tenemos una relación que permite llevar la información de los proveedores de una empresa junto con las partes y cantidad en las que son suministradas, se ilustra en la relación:

| NUM_PROV | NOM_PROV | CIUDAD | ENTREGA | NUM PARTE | CANTIDAD |
|----------|----------|----------|---------|-----------|----------|
| 1 | JUAN | ACAPULCO | 2 | P1 | 100 |
| 1 | JUAN | ACAPULCO | 2 | P2 | 100 |
| 2 | PEDRO | MERIDA | 4 | P1 | 200 |
| 2 | PEDRO | MERIDA | 4 | P3 | 100 |
| 2 | PEDRO | MERIDA | 4 | P5 | 300 |
| 3 | ANA | TIJUANA | 5 | P3 | 200 |
| 3 | ANA | TIJUANA | 5 | P4 | 100 |
| 3 | ANA | TIJUANA | 5 | P5 | 300 |
| 4 | RENE | DF | 1 | P1 | 200 |
| 5 | PATRICIA | REYNOSA | 2 | P1 | 100 |
| 5 | PATRICIA | REYNOSA | 2 | P4 | 200 |
| 6 | RENATA | OAXACA | 2 | P5 | 100 |

Proveedores, Partes y Cantidades Suministradas

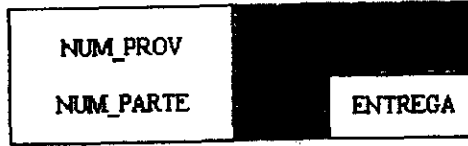
Considerando que la llave primaria es NUM_PROV, NUM_PARTE, tendríamos las siguientes relaciones de dependencia entre los campos no-llave y los campos de la llave candidato, como se ilustra en la figura:



Modelo de Num_prov y Num_parte

| NUM_PROV | NUM_PARTE | CANTIDAD |
|----------|-----------|----------|
| 1 | P1 | 100 |
| 1 | P2 | 100 |
| 2 | P1 | 200 |
| 2 | P3 | 100 |
| 2 | P5 | 300 |
| 3 | P3 | 200 |
| 3 | P4 | 100 |
| 3 | P5 | 300 |
| 4 | P1 | 200 |
| 5 | P1 | 100 |
| 5 | P4 | 200 |
| 6 | P5 | 100 |

Información Partes



Campos de Partes

Tercera forma normal (3FN).

Una relación está en 3NF si esta en 2NF y además se cumple que todos los atributos de la relación no dependen transitivamente de la llave primaria.

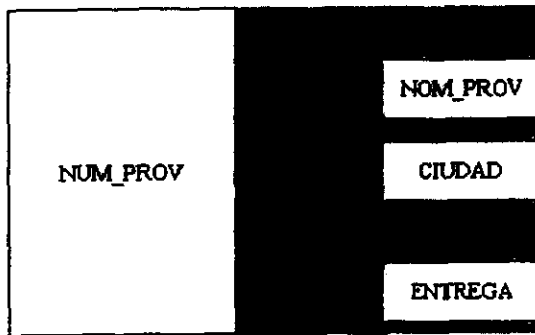
Es decir no se da la situación de que un campo dependa de la llave y otro dependa del primero.

Tomemos como ejemplo la relación de proveedor anterior:

| NUM_PROV | NOM_PROV | CIUDAD | ENTREGA |
|----------|----------|----------|---------|
| 1 | JUAN | ACAPULCO | 2 |
| 2 | PEDRO | MÉRIDA | 4 |
| 3 | ANA | TIJUANA | 5 |
| 4 | RENE | DF | 1 |
| 5 | PATRICIA | REYNOSA | 2 |
| 6 | RENATA | OAXACA | 2 |

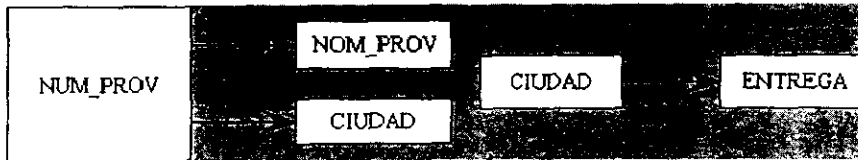
Información Proveedores

Pero analizando en detalle cada atributo de la relación nos dimos cuenta que el atributo ENTREGA depende realmente de CIUDAD y no directamente del NUM_PROV, es decir tenemos la siguiente relación de dependencia, como se ilustra en la figura



Campos de Proveedores (3NF)

O más claramente, como se ilustra en la figura:



Otra forma de mostrar los Campos de Proveedores

Para normalizar esta relación se realizan dos proyecciones teniéndose, como se ilustra en las tablas:

| NUM_PROV | NOM_PROV | CIUDAD |
|----------|----------|----------|
| 1 | JUAN | ACAPULCO |
| 2 | PEDRO | MÉRIDA |
| 3 | ANA | TIJUANA |
| 4 | RENE | DF |
| 5 | PATRICIA | REYNOSA |
| 6 | RENATA | OAXACA |

| Ciudad | Entrega |
|----------|---------|
| ACAPULCO | 2 |
| MERIDA | 4 |
| TIJUANA | 5 |
| DF | 1 |
| REYNOSA | 2 |
| OAXACA | 2 |

Forma Normal de Boyce-Codd (BCNF)

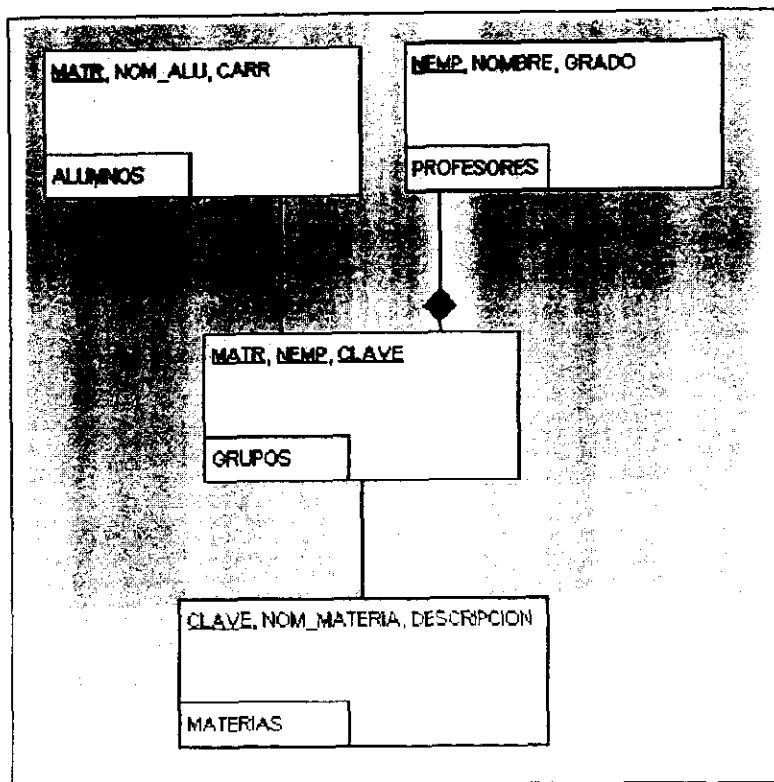
Para ilustrar la necesidad de una forma normal adicional daremos un ejemplo, como se ilustra en la figura: BASE DE DATOS DE ALUMNOS-PROFESORES-MATERIAS.

Supongamos que deseamos llevar la información de que materias lleva cada alumno y que profesor se las imparte.

Se sabe además que:

- Un alumno lleve una o más materias.
- Un profesor imparte sólo una materia.
- Una materia puede ser impartida por uno o más profesores.
- En una materia puede haber inscritos uno o más alumnos.

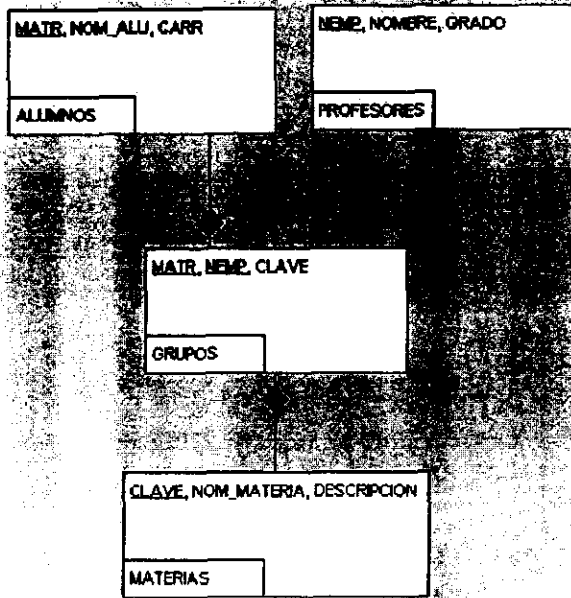
Un posible diseño sería(considerando que ya se definieron previamente los campos de cada entidad):



Bases de Datos de Alumnos-Profesores-Materias

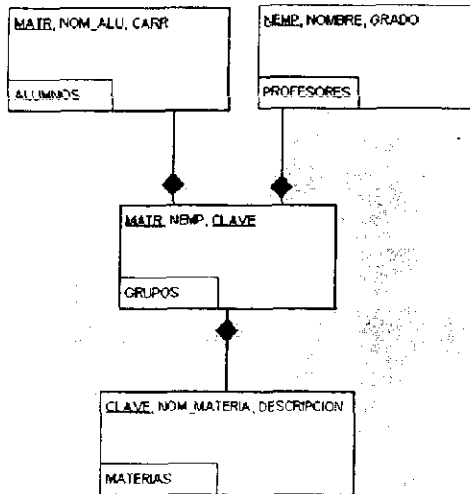
Pero como un profesor sólo imparte una materia no es posible que un alumno lleve dos ó más materias con el mismo profesor, de donde concluimos que no se requiere que la llave primaria de la relación GRUPOS sea **MATR+NEMP+CLAVE** sino solamente **MATR+CLAVE** ó **MATR+NEMP**, es decir que tenemos dos llaves candidato **MATR+CLAVE** y **MATR+NEMP**.

De forma que dos mejores diseños se ilustran en las figuras:



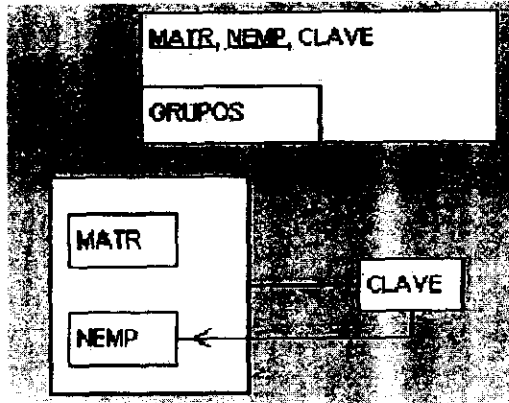
Bases de Datos Alumnos-Profesores-Materias

ó



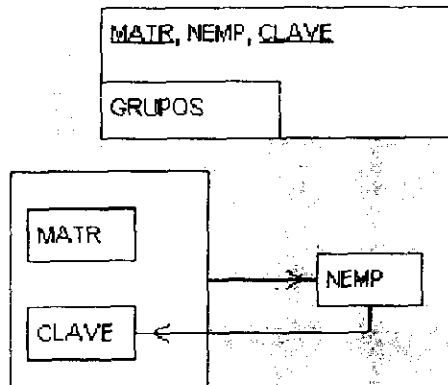
Base de Datos de Alumnos-Profesores-Materias

Construyendo los diagramas de dependencia funcional para la relación GRUPOS de los dos diseños tenemos la figura:



Dependencia Funcional para la Relación de Grupos

NOTA: La flecha de NEMP a CLAVE indica que el profesor determina la materia, puesto que solo imparte una.



Dependencia Funcional de la Relación de Grupos

En cualquiera de los dos casos podemos observar que la relación GRUPOS está en 3NF porque todos los campos no-llave no dependen transitivamente de la llave primaria.

Pero aún así la relación tiene ciertas anomalías, v.g. si solo se tiene a un alumno inscrito en una materia impartida por un profesor, al dar de baja a dicho alumno, se pierde la información de que materia imparte dicho profesor.

Para evitar este tipo de anomalías se ha definido la BCNF que dice:

- Una relación está en BCNF si cada determinante es una llave candidato.
- Un DETERMINANTE es cualquier atributo o conjunto de atributos del cuál otro atributo es dependiente funcionalmente en forma completa.

Analizando el diagrama de dependencias y las llaves candidato tenemos lo siguiente:

LAS LLAVES CANDIDATO SON:

MATR+CLAVE

MATR+NEMP

LOS DETERMINANTES SON:

MATR+CLAVE

MATR+NEMP

NEMP (Puesto que determina la CLAVE).

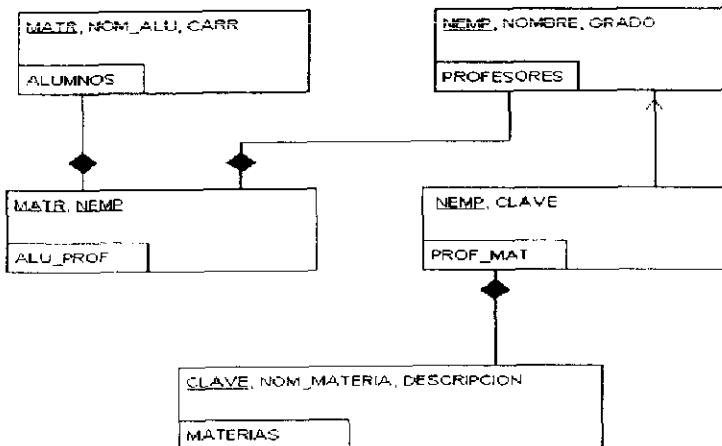
De donde concluimos que la relación GRUPOS no está en BCNF.

Para poner a GRUPOS en BCNF tenemos que analizar sus posibles proyecciones, como se ilustra en la relación:

| Proyección | Comentario |
|--------------------|--|
| GRUPOS[MATR,CLAVE] | Se pierde la información de que profesor le da clase a un alumno, puesto que hay varios profesores para una materia. |
| GRUPOS[MATR,NEMP] | OK, pero se requiere adicionalmente la proyección GRUPOS[NEMP,CLAVE] para tener la información original. |
| GRUPOS[NEMP,CLAVE] | OK, pero se requiere la proyección GRUPOS[MATR,NEMP] para recuperar la información original. |

Grupos en BCNF

De lo anterior concluimos que la relación grupos debe descomponerse en las proyecciones: GRUPOS[MATR,NEMP] y GRUPOS[NEMP,CLAVE], teniendo el siguiente diseño de nuestra base de datos, que se ilustra en la figura:



Base de Datos de Alumnos-Profesores-Materias

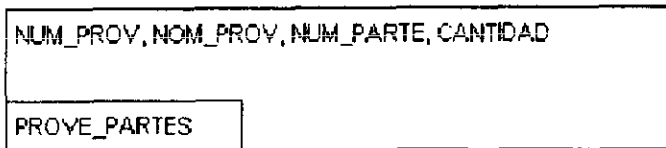
De donde podemos observar que ALU_PROF ya está en BCNF(es toda llave) y que PROF_MAT ya está en BCNF puesto que solo existe un determinante(que es NEMP).

Adicionalmente se puede verificar que la anomalía de perder la información de que materia da un profesor al borrar al único alumno que estaba inscrito se ha evitado.

NOTA: generalmente se cumple que una relación no está en BCNF si las llaves candidatas se traslapan.

BASE DE DATOS DE PROVEEDORES-PARTES.

Supongamos que tenemos la siguiente relación que se ilustra en la figura:



Ejemplo de la Relación Proveedor-Parte

Se sabe:

- Un proveedor suministra varias partes.
- Una parte puede ser suministrada por varios proveedores.
- NUM_PROV es único para cada proveedor.

- NUM_PARTE es único para cada parte.
- NOM_PROV es único para cada proveedor.

Los diagramas de dependencia funcional podrían ser los que se ilustran en las figuras:

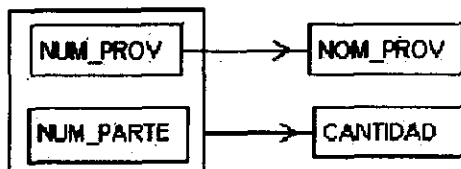


Diagrama de Dependencia Funcional

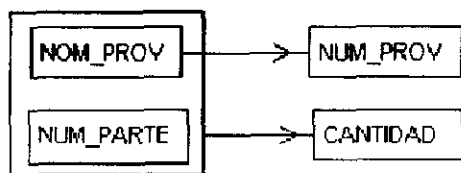


Diagrama de Dependencia Funcional

Por lo que al no haber dependencias transitivas se concluye que esta relación está en 3NF.

Tenemos entonces:

LAS LLAVES CANDIDATO SON:

NUM_PROV+NUM_PARTE

NOM_PROV+NUM_PARTE

LOS DETERMINANTES SON:

NOM_PROV+NUM_PARTE

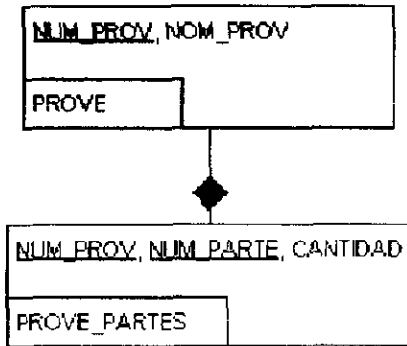
NUM_PROV+NUM_PARTE

NUM_PROV ó

NUM_PARTE

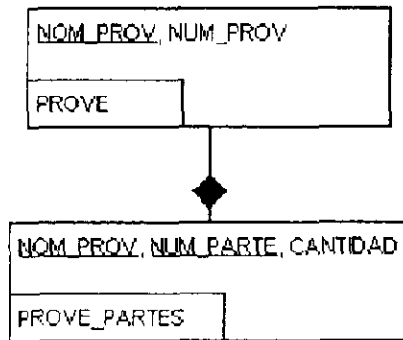
De lo que se concluye que no cumple con los requisitos de BCNF(se puede observar fácilmente que las llaves candidato se traslapan ó que no todos los determinantes son llaves candidato).

Para poner está relación en BCNF es necesario crear dos proyecciones que se ilustran en las figuras:



Proyecciones para BCNF

ó



Proyecciones para BCNF

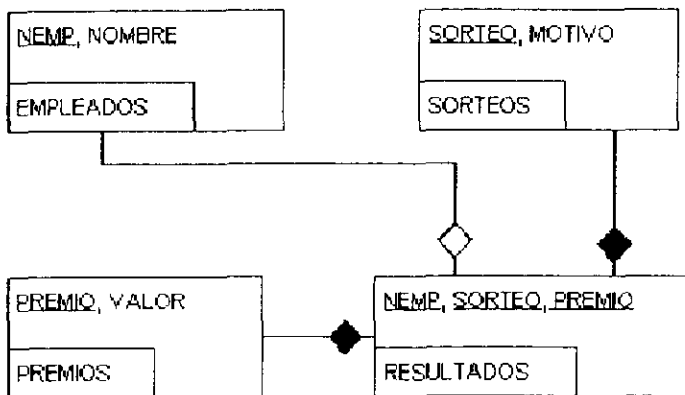
BASE DE DATOS DE EMPLEADOS-SORTEOS-PREMIOS.

En una empresa se tiene la costumbre de realizar **SORTEOS** entre los **EMPLEADOS** para otorgar **PREMIOS**.

Se sabe además que:

- En un **SORTEO** no necesariamente todos se sacan **PREMIO**.
- Cada sorteo tiene diversos **PREMIOS**.
- Los **PREMIOS** no son repetidos en el mismo **SORTEO**, pero sí en **DIFERENTES** **SORTEOS**.
- Un **PREMIO** solo se lo saca un **EMPLEADO**.
- Un **EMPLEADO** puede participar en diversos **SORTEOS**, pero a lo mucho se saca un **PREMIO** en **CADA** **SORTEO**.

Un posible diseño podría ser el que se ilustra en la figura:



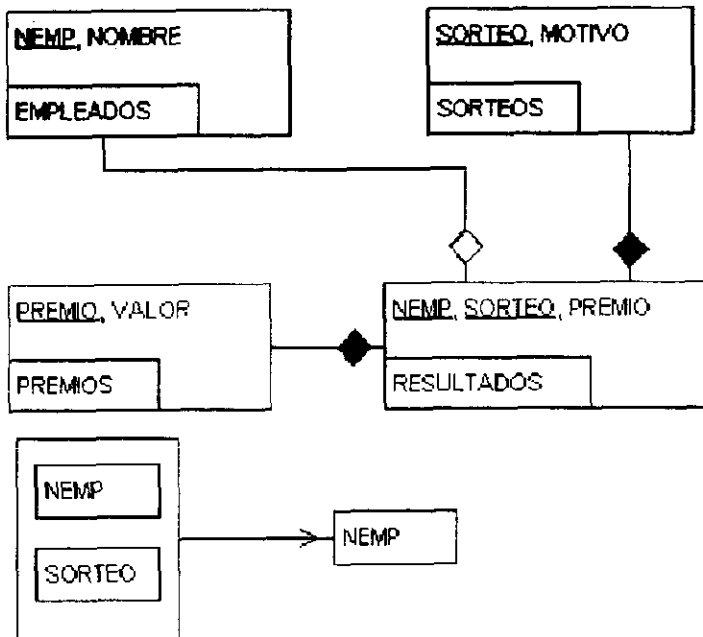
Bases de Datos de Empleados-Sorteos-Premios

Observando en detalle la relación resultados observamos que no es necesario que la llave sea NEMP+SORTEO+PREMIO debido a que:

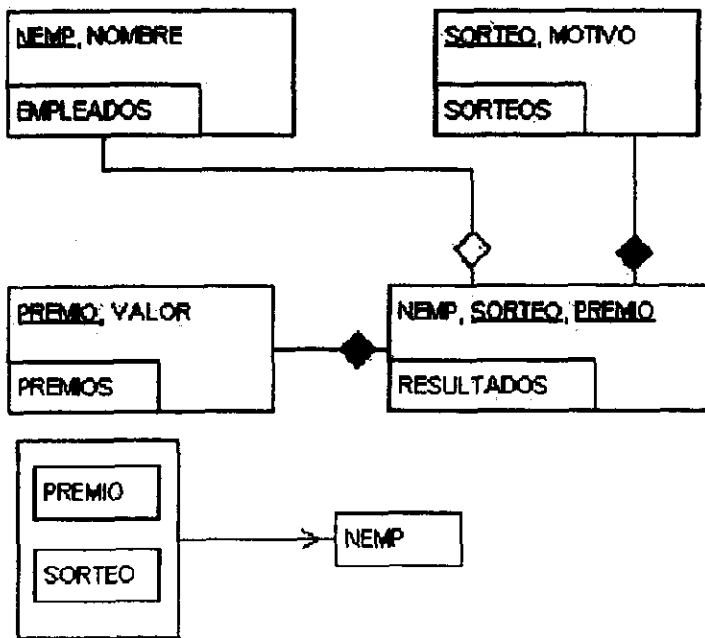
Un PREMIO no se repite en el MISMO SORTEO.

Un EMPLEADO solo se saca a lo mucho un PREMIO en cada SORTEO.

De acuerdo a esto el diseño podrían ser los que se ilustran en las figuras:



Bases de Datos de Empleados-Sorteos-Premios



Bases de Datos de Empleados-Sorteos-Premios

De donde:

LAS LLAVES CANDIDATOS SON:

NEMP+SORTEO

PREMIO+SORTEO

LOS DETERMINANTES SON:

NEMP+SORTEO

PREMIO+SORTEO

De lo cuál se concluye que la relación está en BCNF, puesto que todos los determinantes son llaves candidatas.

NOTA: Este es un caso en el que las llaves candidato se traslapan pero aún así la relación está en BCNF.

De la discusión de este último ejemplo, se puede ver que no siempre las llaves candidato traslapadas dan lugar a problemas.

Cuarta forma normal (4FN).

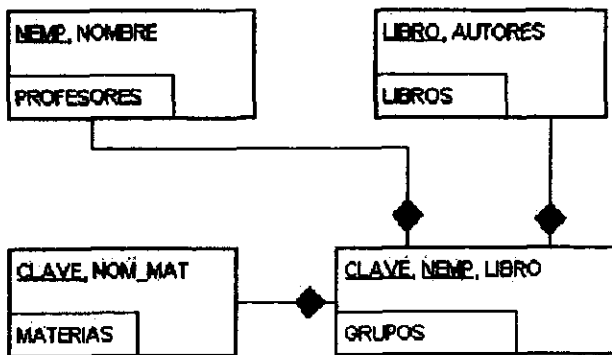
Existen algunas relaciones que a pesar de estar en BCNF presentan cierto grado de redundancia. Lo cuál a su vez tiene como consecuencia que tengamos anomalías al realizar altas o cambios.

Para ilustrar esto, consideremos que en una universidad se desea llevar el control de materias, profesores y libros de texto.

Se tiene la siguiente información:

- Una materia puede ser impartida por uno o más profesores.
- Una materia tiene asociados uno o más libros de texto.
- Un profesor puede impartir una o más materias.
- Un profesor al impartir una materia usa todos los libros de texto de esa materia.
- Un libro de texto puede ser utilizado en una o más materias.

De este modo un posible diseño sería el que se ilustra en la figura:



Base de Datos de Materias-Profesores-Libros de Texto

Analizando en detalle la relación GRUPOS nos podemos dar cuenta que se requiere que toda sea llave, puesto que:

Un mismo libro se puede repetir para diferentes combinaciones de profesor y materia.

Un mismo profesor se puede repetir para diferentes combinaciones de materias y libros.

Una misma materia se puede repetir para diferentes combinaciones de profesor y libro.

En este sentido al ser la relación GRUPOS toda LLAVE ya está en BCNF pero tiene algunos problemas:

Al dar de alta a un nuevo profesor para una materia tenemos el problema de dar de alta tantos registros como libros haya para la materia.

Al dar de alta un libro para una materia, se tienen que dar de alta tantos registros como profesores haya de la materia.

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

La BCNF no nos ayuda a corregir estos problemas, por lo que se requiere una forma normal adicional:

Una relación está en Cuarta Forma Normal(4NF) si al tener dependencias multivaluadas de la forma

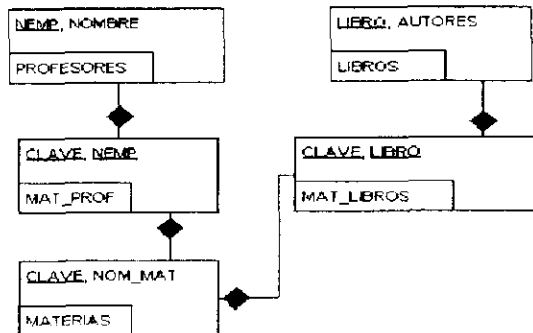
A-B todas las dependencias funcionales dependen de A. Es decir todas las dependencias multivaluadas son dependencias funcionales.

Para el caso de la relación analizada tenemos:

CLAVE - NEMP

CLAVE - LIBRO

Para este caso un mejor diseño separaría las dependencias multivaluadas dando la figura:



Base de Datos de Alumnos-Profesores-Libros de Texto

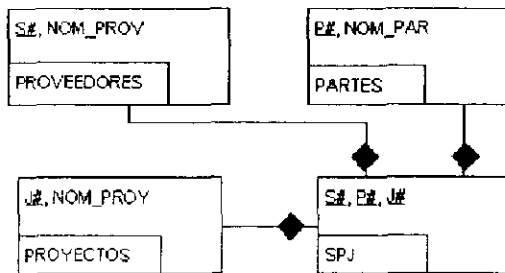
Quinta forma normal (5FN).

Dentro del proceso de normalización presentado, se ha visto que se realizan proyecciones sin pérdida. Normalmente se tiene que una relación es descompuesta en dos de sus proyecciones y al realizar el JOIN de dichas proyecciones se recupera la relación original. Existen algunas relaciones en las cuales el realizar dos proyecciones de la relación original no es suficiente para tener una descomposición sin pérdida (ni ganancia).

Tomemos el caso de PROVEEDORES, PARTES y PROYECTOS en donde se tiene que:

- Un proveedor puede suministrar diferentes partes a diferentes proyectos.
- Una parte puede ser usada por diferentes proyectos y suministrada por diferentes proveedores.
- Un proyecto puede usar diferentes partes y ser proveido por diferentes proveedores.

En este sentido un posible diseño sería el que se ilustra en la figura:



Modelo Entidad de Proveedores, Partes y Proyectos en 5FN

SI S_x, P_y aparece en SP

Y P_y, J_w aparece en PJ

Y S_x, J_w aparece en SJ

ENTONCES aparece SX, P_y, J_w en SPJ.

En este sentido se define la dependencia al join(JD) de la siguiente forma:

Una relación satisface una JD si la relación original se produce por la juntura de las proyecciones indicadas en JD. v.g.

Si la relación R consta de campos A, B, C satisface la dependencia JD(AB, AC) si R se produce de AB JOIN AC.

COROLARIO: Esto implica que:

A-B nddocument

SI S_x, P_y aparece en SP

Y P_y, J_w aparece en PJ

Y S_x, J_w aparece en SJ

ENTONCES aparece SX, P_y, J_w en SPJ.

3.6 **ÁLGEBRA RELACIONAL.**

El álgebra relacional ofrece algunas formas simples pero potentes de construir nuevas relaciones a partir de las anteriores. En ella las expresiones comienzan con las relaciones como operandos; las relaciones pueden representarse con su nombre o explícitamente como una lista de tuplas.

Las operaciones del álgebra relacional caen en cuatro grandes clases:

1. Las operaciones habituales de los conjuntos son: unión, intersección y diferencia aplicadas a relaciones.
2. Las operaciones que suprimen partes de una relación: la "selección" elimina algunos renglones (tuplas) y la "proyección", algunas columnas.
3. Las operaciones que combinan las tuplas de dos relaciones, entre ellas el "producto cartesiano", que combina en pares las tuplas de dos relaciones en todas las formas posibles y varias clases de operaciones de "reunión", que selectivamente forman parejas de tuplas a partir de dos relaciones.
4. Una operación denominada "cambio de nombre" (renombramiento) que no afecta a las tuplas de una relación, pero que sí modifica el esquema; por ejemplo, los nombres de consulta real como éste y que, sin embargo, no forman parte del álgebra relacional.

Las operaciones anteriores no bastan para efectuar cualquier cálculo acerca de las relaciones; de hecho son muy limitadas. Pero captan gran parte de lo que realmente se puede hacer con las bases de datos y constituyen una parte considerable del lenguaje estándar de consulta relacional como lo es SQL, que se utiliza para hacer una selección desde Visual Basic a las relaciones de Access y que es muy útil para los combobox y ListBox de Visual Basic de este sistema.



CAPÍTULO IV
DISEÑO DEL SISTEMA.

CAPÍTULO IV

DISEÑO DEL SISTEMA.

El proceso de diseñar una base de datos comienza con el análisis de la información que debe contener y de las relaciones entre los componentes de la información. A menudo la estructura de la base, denominada esquema de base de datos, se especifica en uno de los lenguajes o notaciones adecuados para expresar los diseños.

A menudo los diseñadores comienzan desarrollando un sistema que utilice el modelo entidad/relación o un modelo basado en objetos y luego lo traducen al modelo relacional para implementarlo. El motivo por el cual los sistemas de datos basados en el modelo relacional dominan el mercado es que su simplicidad permite utilizar lenguajes elegantes y potentes de consulta de bases de datos.

El modelo utilizado y desarrollado en este sistema es el modelo relacional, es un modelo extraordinariamente útil porque no tiene mas que un solo concepto para la modelación de datos: la "relación", una relación bidimensional donde se ordenan los datos.

En el modelo relacional, un diseño consta de uno o más esquemas de relación. En un diseño, al conjunto de esquemas de las relaciones se le llama esquema de base de datos relacional o simplemente esquema de base de datos. Una de las características más importante del modelo relacional es que se exige que los componentes de una tupla sean atómicos, o sea que pertenezcan a algún tipo elemental como enteros o cadenas de caracteres.

Y la característica más crucial de la estructura de datos relacional es que las asociaciones entre tuplas (renglones) se representan únicamente por valores de datos en columnas sacadas de un dominio común.

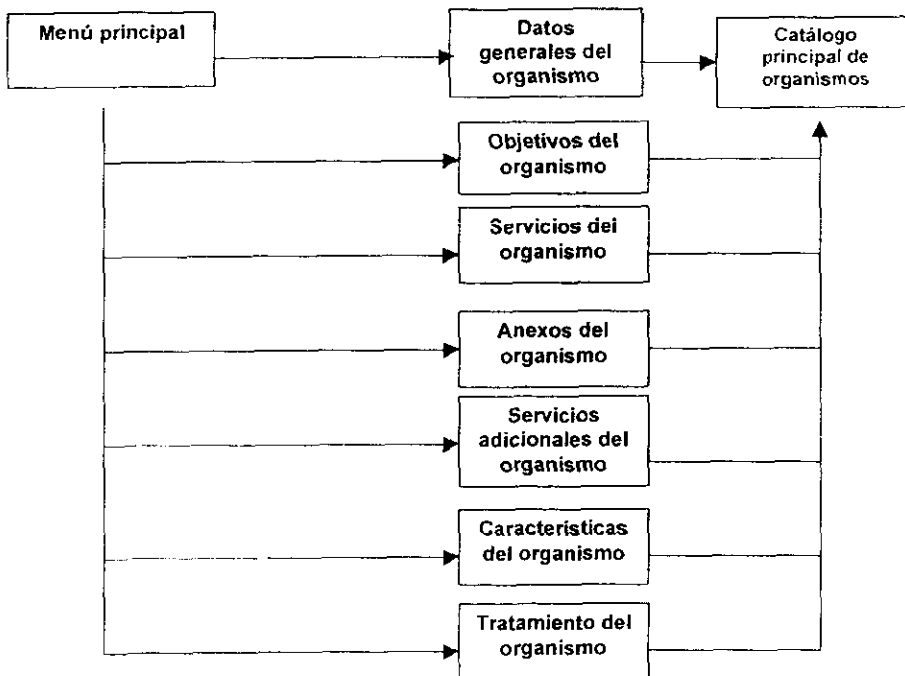
En el presente capítulo se determinará los atributos y objetivos del modelo de datos a utilizar; además se diseñará el modelo relacional del sistema.

Se describirán también los elementos de los que se constituye la base de datos.

4.1 MODELADO DE ENTIDADES.

Para el modelado de entidades se utiliza la metodología basada en entidad – relación, esta metodología tiene ventajas como son: flexibilidad, especifica las limitantes de los datos y sus relaciones.

El diagrama del modelado de entidades se muestra a continuación:



Entidad datos generales del organismo: Esta entidad es de gran interés, ya que, son los datos del organismos y sin estos datos no se podrá canalizar adecuadamente al solicitante, de acuerdo a su localización y condición.

Entidad objetivos del organismo: Esta entidad esta enfocada al tipo de población, grupos, usuarios y categorías, a los que se enfocan las diferentes instituciones para así cubrir todas las esferas de la sociedad y en particular del organismo del cual se trate.

Entidad servicios del organismo: En esta entidad se da la opción de obtener una respuesta a una emergencia, así como la prevención y orientación para evitar las diferentes causas de las adicciones. También en esta entidad se da la posibilidad de obtener información de publicaciones y fuentes bibliográficas, así como un centro de documentación del organismo de que se trate.

Entidad anexos del organismo: En esta entidad se capturan los requisitos del solicitante y las observaciones pertinentes, así como el seguimiento del Consejo Nacional contra las Adicciones.

Entidad servicios adicionales del organismo: Dentro de esta entidad se capturan los servicios adicionales que ofrece el organismo tales como capacitación especializada interna y externa, tipo de investigación y asesoría legal.

Entidad características del organismo: Dentro de esta entidad se captura la información general del organismo así como los directores generales y de unidades, horarios de servicio, áreas que abarca el organismo, días de servicio y los contactos con los que se cuenta para contactar dicho organismo.

Entidad tratamiento del organismo: Dentro de esta entidad se captura modalidades terapias, el tipo de servicio y la rehabilitación para el solicitante y su reinserción.

4.2 DISEÑO DE RELACIONES.

Después del modelo entidad-relación el paso siguiente es diseñar la estructura de cada una de las relaciones para conformar la base de datos, ya que, esta contendrá los atributos particulares de las entidades dentro de las cuales se indicarán las llaves foráneas y las llaves primarias, así como la descripción de la información que se almacena.

En Microsoft Access, las relaciones son el elemento fundamental para almacenar datos. Una definición formal señala que una relación es un objeto, o una entidad que se identifica a través de sus atributos o campos (columnas), y puede ser la abstracción de algo real o intangible. Cada una debe centrarse en un solo aspecto de negocio, por ejemplo, datos generales de clientes, clases de crédito de la empresa, etc.

Por otro lado, una relación es muy similar a una hoja de datos de Microsoft Excel, ya que está formada por filas y columnas. Sin embargo, en Microsoft Access, las columnas son conocidas como campos y almacenan datos con ciertas características o atributos. Por ejemplo, en una relación de clientes, puede existir un campo para teléfonos y otro para direcciones electrónicas. Por su parte, las filas son llamadas registros y ahí es donde se agregan las direcciones o teléfonos de cada cliente.

DIAGRAMA E/R (ORGANISMOS)

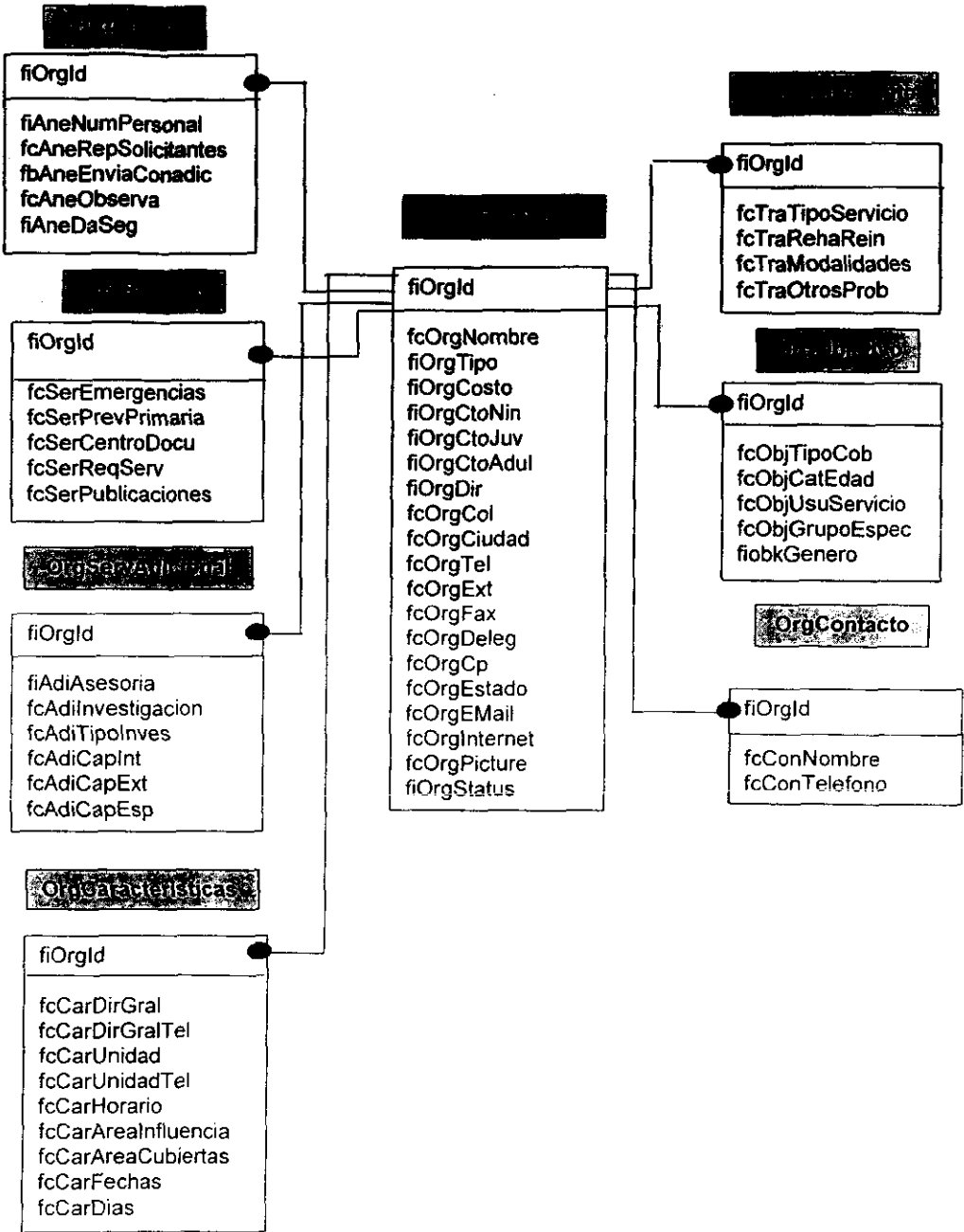
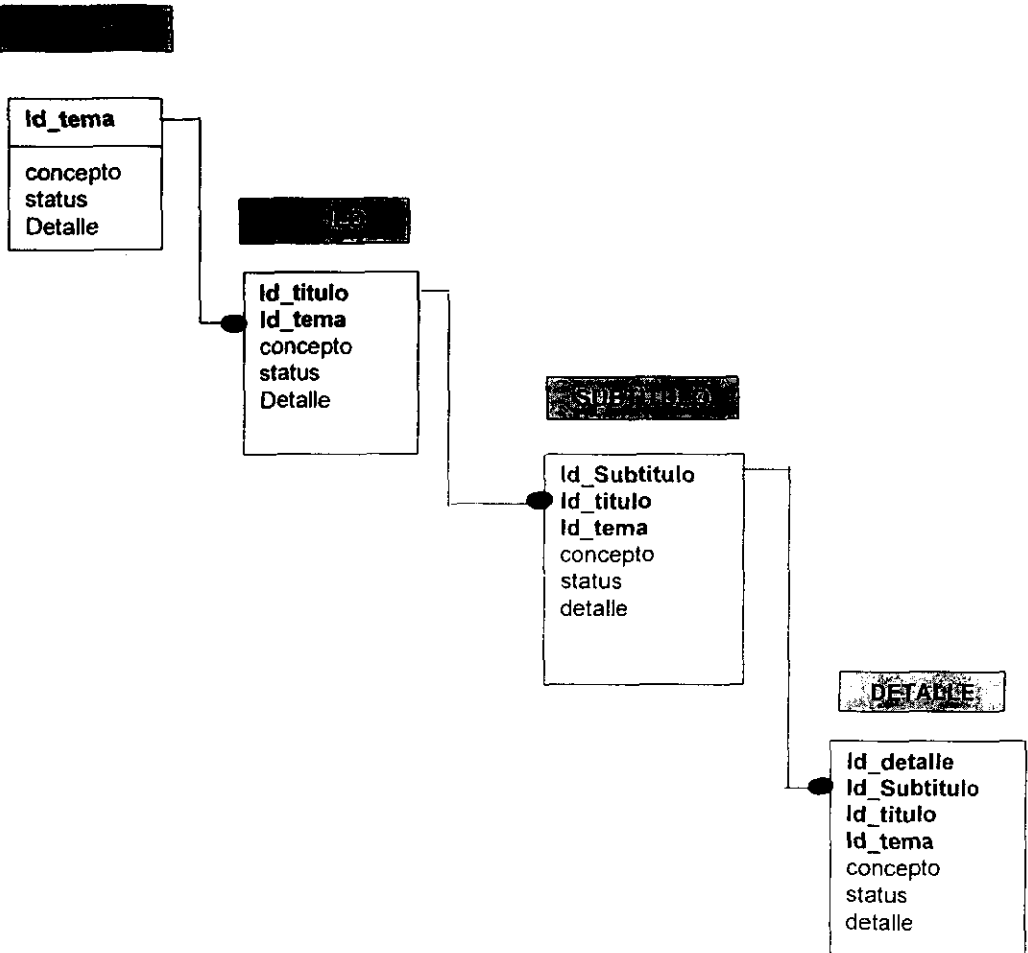
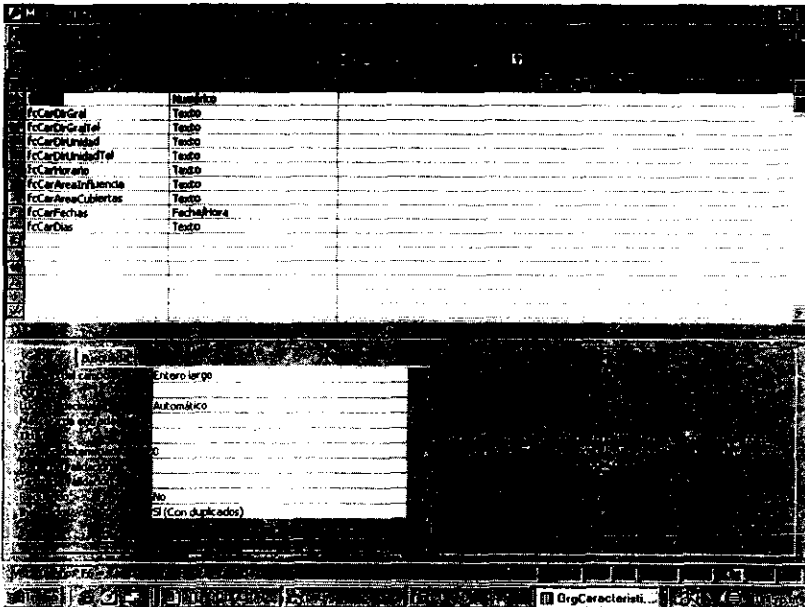


DIAGRAMA E/R (ARBOL DE INFORMACIÓN)



4.2.1 RELACIÓN ORGCARACTERÍSTICAS

La relación se forma de varios atributos que conforman la información referente a los datos del director general, director de unidad, el horario de atención del organismo, las áreas de influencia, las áreas cubiertas, fecha de inicio de actividades y los días de servicio.



| Nombre | Tipo |
|--------------------|------------|
| fCarDirGen | Texto |
| fCarDirUnid | Texto |
| fCarDirUnidad | Texto |
| fCarDirUnidadTel | Texto |
| fCarHorario | Texto |
| fCarAreaInfluencia | Texto |
| fCarAreaCubiertas | Texto |
| fCarFechas | Fecha/hora |
| fCarDias | Texto |

| Propiedad | Valor |
|-----------|---------------------|
| Formato | Entero largo |
| Formato | Automático |
| Formato | 0 |
| Formato | No |
| Formato | Si (Con duplicados) |

En Access, cada una de las tuplas tiene por default propiedades sobre ellas, como son: Tamaño del campo, Formato, Título, Valor predeterminado, Regla de validación, Texto de validación, Indexado, etc.

Dentro de cada una de las relaciones se siguió con una regla del lenguaje SQL para identificar los campos y ahorrar la consulta del diseño de la relación para ver el tipo de dato

Para saber que son campos de algún tipo de dato se representa con una letra f de field en inglés seguida de las letras i de entero, c de carácter y b de booleano.

La estructura de general de la relación es la siguiente:

| fiOrgId | Numérico | Entero largo |
|---------------------|------------|---------------|
| FcCarDirGral | Texto | 50 |
| fcCarDirGralTel | Texto | 30 |
| fcCarDirUnidad | Texto | 50 |
| fcCarDirUnidadTel | Texto | 30 |
| fcCarHorario | Texto | 15 |
| fcCarAreaInfluencia | Texto | 150 |
| fcCarAreaCubiertas | Texto | 150 |
| fcCarFechas | Fecha/Hora | Fecha general |
| fcCarDias | Texto | 10 |

Los atributos de esta relación tienen la información relacionada con los directores generales, directores de unidad, contactos y la información de los días de servicios que ofrece el organismo del que se trate.

4.2.2 RELACIÓN ORGCONTACTO

Los atributos que conforman esta relación contienen la información de los contactos que se tienen para obtener comunicación con el organismo, incluyendo nombre y teléfono; esta relación es muy importante dentro del formulario de características del organismo.

La estructura general de la relación es la siguiente:

| Nombre del atributo | Tipo de dato | Longitud |
|---------------------|--------------|----------|
| FiOrgId | Número | 50 |
| FcConNombre | Texto | 50 |
| FcConTelefono | Texto | 30 |

4.2.3 RELACIÓN ORGGENERALES

Los siguientes atributos que conforman esta relación contienen la información de los datos generales del organismo.

Esta relación es muy importante, ya que, con ella se obtienen los datos generales del organismo como nombre, dirección, costo, tipo de institución, colonia, así como su dirección de correo electrónico para tener contacto con él.

| | | |
|---------------|--------------|--------------|
| fiOrgId | Autonumérico | Entero largo |
| fcOrgNombre | Texto | 80 |
| fiOrgTipo | Numérico | Entero largo |
| fiOrgCosto | Numérico | Entero largo |
| fiOrgCtoNin | Numérico | Entero largo |
| fiOrgCtoJuv | Numérico | Entero largo |
| fiOrgCtoAdul | Numérico | Entero largo |
| fcOrgDir | Texto | 50 |
| fcOrgCol | Texto | 50 |
| fcOrgCiudad | Texto | 50 |
| fcOrgTel | Texto | 30 |
| fcOrgExt | Texto | 10 |
| fcOrgFax | Texto | 15 |
| fiOrgDeleg | Numérico | Entero largo |
| fiOrgCp | Numérico | Entero largo |
| fiOrgEstado | Numérico | Entero largo |
| fcOrgEMail | Texto | 50 |
| fcOrgInternet | Texto | 50 |
| fcOrgPicture | Texto | 50 |
| fiOrgStatus | Sí/No | Sí/No |

4.2.4 RELACIÓN ORGOBJETIVO

Los atributos que conforman esta relación contienen la información de la cobertura de la población a servicios médicos, las categorías por edad son las diferentes edades de atención, la población a la que se le puede dar el servicio, grupos específicos y el género.

| Nombre del atributo | Tipo | Longitud |
|---------------------|----------|--------------|
| fiOrgId | Numérico | Entero largo |
| fcObjTipoCob | Texto | 150 |
| fcObjCatEdad | Texto | 150 |
| fcObjUsuServicio | Texto | 150 |
| fcObjGrupoEspec | Texto | 150 |
| fiObkGenero | Texto | 50 |

4.2.5 RELACIÓN SERVADICIONAL

Los atributos de esta relación están relacionados con la capacitación especializada, interna, externa, así como el tipo de investigación que se desarrolla dentro del organismo.

| Nombre del atributo | Tipo | Longitud |
|---------------------|----------|--------------|
| fiOrgId | Numérico | Entero largo |
| fiAdiAsesoría | Numérico | Entero largo |
| fcAdiInvestigacion | Texto | 150 |
| fcAdiTipoInves | Texto | 150 |
| fcAdiCapInt | Texto | 150 |
| fcAdiCapExt | Texto | 150 |
| fcAdiCapEsp | Texto | 150 |

4.2.6 RELACIÓN ORGSERVICIO

Los atributos de esta relación están relacionados con los servicios de emergencia que ofrece el organismo y con la información para la prevención de alguna droga.

Dentro de los servicios que el organismo ofrece también existen centros de documentación, es decir, préstamo bibliotecario, publicaciones que fueron editadas en el último año y los requisitos para obtener este servicio.

| fiOrgId | Numérico | Entero largo |
|--------------------|----------|--------------|
| fcSerEmergencias | Texto | 150 |
| fcSerPrevPrimaria | Texto | 150 |
| fcSerCentroDocu | Texto | 150 |
| fcSerReqServ | Texto | 150 |
| fcSerPublicaciones | Texto | 150 |

4.2.7 RELACIÓN ORGTRATAMIENTO

Los atributos de esta relación comprenden el tipo de servicio en cuanto a consultas o internamiento se refiere, así como las modalidades de terapia y si ofrece rehabilitación; también ofrece la atención a otros problemas como son anorexia, bulimia, relaciones autodestructivas, comedores y juegos compulsivos.

| Nombre del atributo | Tipo | Longitud |
|---------------------|----------|--------------|
| fiOrgId | Numérico | Entero largo |
| fcTraTipoServicio | Texto | 150 |
| fcTraRehaRein | Texto | 150 |
| fcTraModalidades | Texto | 150 |
| fcTraOtrosProb | Texto | 150 |

4.2.8 RELACIÓN DETALLE

Los atributos de esta relación están relacionados en cascada con las relaciones de subtítulo, título y tema; ya que en el diseño de esta base de datos y del formulario en visual las ramas del árbol están ligadas por cada una de las llaves de estas relaciones. En esta relación se describen los datos históricos y teóricos útiles como información para el CONADIC y usuarios externos acerca de las diferentes drogas.

| Nombre del atributo | tipo | Longitud |
|---------------------|--------------|--------------|
| Id_detalle | Autonumérico | Entero largo |
| Id_Subtitulo | Numérico | Entero largo |
| Id_titulo | Numérico | Entero largo |
| Id_tema | Numérico | Entero largo |
| Concepto | Texto | 50 |
| status | Si/No | Si/No |
| detalle | Texto | 50 |

4.2.9 RELACIÓN SUBTÍTULO

Los atributos de esta relación están relacionados en cascada con las relaciones de título y tema; ésta relación describe exactamente la derivación de la droga a la que se quiere acceder y ver su detalle y así obtener información más detallada.

| Nombre del atributo | tipo | Longitud |
|---------------------|--------------|--------------|
| Id_Subtitulo | Autonumérico | Entero largo |
| Id_titulo | Numérico | Entero largo |
| Id_tema | Numérico | Entero largo |
| concepto | Texto | 100 |
| status | Si/No | Si/No |
| detalle | Texto | 50 |

4.2.10 RELACIÓN TEMA

Esta relación describe los diferentes temas que abarca el CONADIC como son alcoholismo, tabaco, drogadicción, etc. Y que son posibles escoger con la facilidad del árbol que presenta visual basic.

| Nombre del atributo | Tipo | Longitud |
|---------------------|--------------|--------------|
| Id_tema | Autonumérico | Entero largo |
| concepto | Texto | 100 |
| status | Sí/No | Sí/No |
| Detalle | Texto | 50 |

4.2.11 RELACIÓN TÍTULO

Esta relación esta relacionada con la relación de tema por medio de su llave foránea Id_tema; la información que nos describe esta relación es específicamente la droga y el árbol de visual basic nos permite seleccionarla y acceder al subtítulo deseado.

| Nombre del atributo | Tipo | Longitud |
|---------------------|--------------|--------------|
| Id_titulo | Autonumérico | Entero largo |
| Id_tema | Numérico | Entero largo |
| concepto | Texto | 100 |
| Status | Sí/No | Sí/No |
| Detalle | Texto | 50 |

4.2.12 RELACIÓN ORGANEXO

Los atributos de esta relación están relacionados con los requisitos que el solicitante debe tener para ser atendido en ese organismo así como las observaciones pertinentes. Dentro de esta relación existen atributos que se refieren al número de personal del organismo, enviar reporte al CONADIC así como el seguimiento que da el organismo al solicitante.

| Nombre del atributo | Tipo | Longitud |
|----------------------|------------|--------------|
| fiOrgId | N Numérico | Entero largo |
| fiAneNumPersonal | N Numérico | Entero |
| fcAneRepSolicitantes | T Texto | 150 |
| fbAneEnviaConadic | S Si/No | Si/No |
| fcAneObserva | T Texto | Texto |
| fiAneDaSeg | S Si/No | Si/No |

4.2.13 RELACIÓN ACCESO

Esta relación contiene la clave para la cual se puede tener acceso al sistema y así tener un control de usuarios.

| Nombre del atributo | Tipo | Longitud |
|---------------------|---------|----------|
| fcAccesold | T Texto | 8 |
| fcAccNombre | T Texto | 40 |
| fcAccPass | T Texto | 10 |

4.2.14 RELACIÓN CONCEPTO

Los atributos de esta relación es una de las más importantes ya que esta relación actúa como catálogo y de la cual se puede hacer el llamado por medio del snapshot o dynaset de visual basic y contiene los tributos necesarios para los combo box y de los checklist de visual basic.

| Nombre del atributo | Tipo | Longitud |
|---------------------|--------------|--------------|
| fiConcPadreId | Autonumérico | Entero largo |
| fiConcHijoId | Numérico | Entero largo |
| fcConcDetNombre | Texto | 50 |
| fcConcTipo | Texto | 1 |
| fiConcDetstatus | Si/No | Si/No |

4.2.15 RELACIÓN FOLIO

Los atributos de esta relación están relacionados con la ruta con la cual se va a dirigir y guardar la información del árbol de información dentro de una carpeta.

| Nombre del atributo | Tipo | Longitud |
|---------------------|--------------|--------------|
| Id_Folio | Autonumérico | Entero largo |
| DetalleFolio | Texto | 50 |
| Folio | Numérico | Entero largo |
| Ruta | Texto | 50 |

4.2.16 RELACIÓN USUARIO

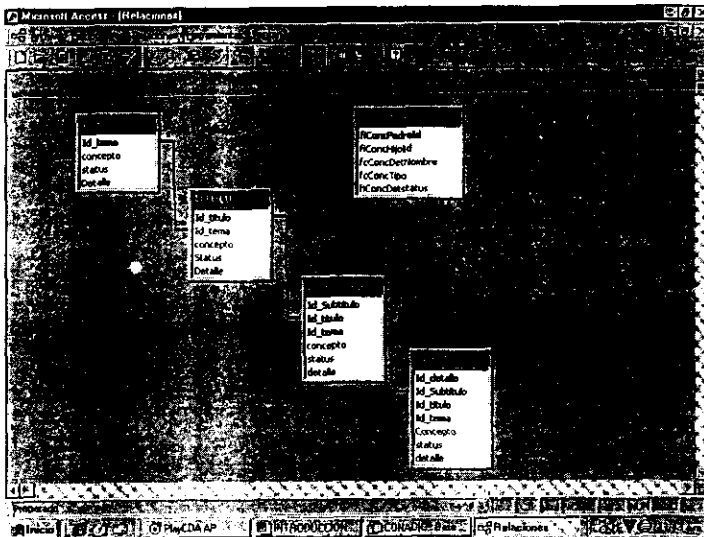
Esta relación contiene los atributos para dar acceso al sistema como son el password y nombre.

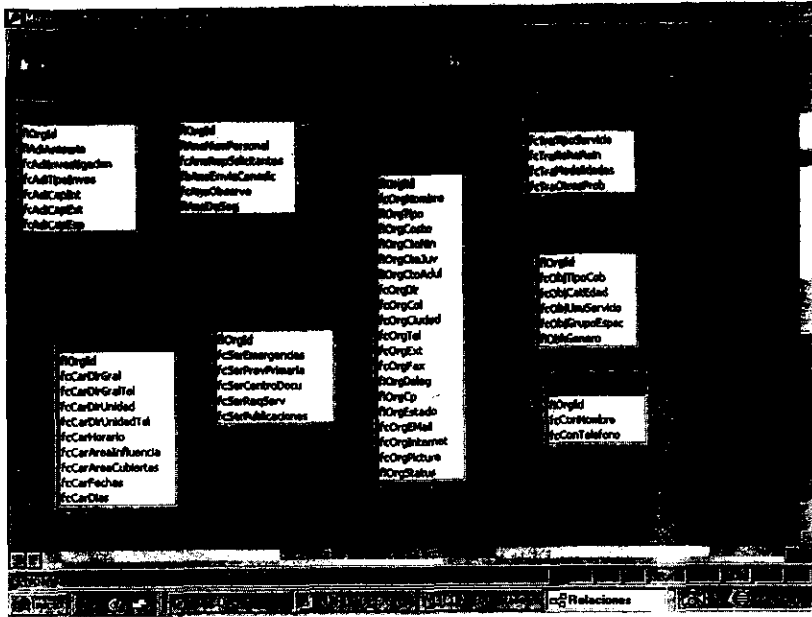
| | | |
|------------|-------|-------|
| Usuario_id | Texto | 6 |
| Password | Texto | 6 |
| Nombre | Texto | 60 |
| Acceso | Si/No | Si/No |
| Sistema | Si/no | Si/No |

Dentro del sistema existen dos partes a desarrollar una llamada árbol de información, que es un catálogo como su nombre lo dice de información que se utilizará para obtener información de las drogas.

La parte de Organismos, es la parte esencial del sistema ya que estos formularios no darán la información general de los mismos, así como, ofrecer un buen servicio de información a los solicitantes.

A continuación se muestran los dos diagramas de las relaciones y sus relaciones que pertenecen tanto al árbol de información, como al de organismos:





CAPÍTULO V
DESARROLLO DEL SISTEMA.

CAPÍTULO V

DESARROLLO DEL SISTEMA.

Microsoft Access es un programa que permite el almacenamiento, organización y búsqueda rápida de información, ya que, incluye asistentes y ayudas.

En Microsoft Access, es posible almacenar y clasificar la información de una sola base de datos en diversas relaciones, dependiendo de las características de los datos y de tema con el que estén relacionados.

El diseño del paquete Access se divide en seis partes importantes:

1. **Relaciones:** Una relación es una colección de datos sobre un tema específico, como orgcaracterísticas o orgcontacto. La utilización de una relación diferente para cada tema significa que se almacenan los datos sólo una vez, lo cual hace aumentar la eficacia de la base de datos, y reduce errores de entrada de datos.
2. **Consultas:** Se utilizan consultas para ver, modificar y analizar datos de formas diferentes. También pueden utilizarse como el origen de registros para formularios, informes y páginas de acceso a datos y también son llamado queries.
3. **Formularios:** Sirven para capturar la información en una pantalla especial.
4. **Informes:** Un informe es un método eficaz de presentar los datos en formato impreso. Dado que tiene el control sobre el tamaño y el aspecto de todo el informe, puede mostrar la información en la manera que desee verla.
5. **Macros:** Una macro es un conjunto de una o más acciones que cada una realiza una operación determinada, tal como abrir un formulario o imprimir un informe. Las macros pueden ayudar a automatizar las tareas comunes. Por ejemplo, puede

ejecutar una macro que imprima un informe cuando el usuario haga clic en un botón de comando.

6. **Módulos:** Que son los programas de aplicación para el usuario final escritos en un dialecto similar al Visual Basic.

Las relaciones organizan datos en columnas (denominadas campos) y filas (denominadas registros). Las relaciones pueden ser importadas de Dbase, paradox y SQL. Es una forma fácil de ver la información existente en el disco. Se puede editar directamente en la relación y se pueden modificar las llaves de los índices que Access se encarga de mantener actualizados.

Las consultas en Access sirven para seleccionar gráficamente la información por visualizarse, e imponen las condiciones que se requieran para determinar trabajo, es decir, una consulta permite formular preguntas sobre los datos almacenados en las relaciones. La forma en que se diseñe la consulta indicará a Access qué datos en particular debe recuperar. Las consultas más utilizadas son las de selección. Con las consultas de selección se pueden ver datos de las relaciones, analizar dichos datos e incluso hacer cambios en los datos de las relaciones. Es posible ver datos de una sola relación, o bien combinar datos de varias relaciones distintas.

Una consulta de selección es el tipo de consulta más habitual. Este tipo de consulta obtiene los datos de una o más relaciones y muestra los resultados en una hoja de datos en la que puede actualizar los registros (con algunas restricciones). También puede utilizar una consulta de selección para agrupar los registros y calcular sumas, cuentas, promedios y otros tipos de totales.

En Access también se pueden crear consultas de relaciones de referencias cruzadas presentan los datos en el formato propio de las hojas de cálculo, mientras que las consultas de acciones pueden utilizarse para hacer cambios en varios registros en una sola operación.

Una consulta SQL es una consulta creada mediante una instrucción SQL. La consulta de unión, la consulta de paso a través, la consulta de definición de datos y la subconsulta son ejemplos de consultas específicas de SQL.

Consulta de unión: combina campos (columnas) de una o más relaciones o consultas en un campo o columna del resultado de la consulta.

Consulta de paso a través: Este tipo de consulta envía comandos directamente a las bases de datos ODBC, como las de Microsoft FoxPro, utilizando comandos aceptados por el servidor.

Consulta de definición de datos: Este tipo de consulta crea, elimina o modifica relaciones, o crea índices en una base de datos, como relaciones de Microsoft Access o Microsoft FoxPro.

Subconsulta: consta de una instrucción SQL SELECT dentro de otra consulta de selección o consulta de acción. Puede introducir estas instrucciones en la fila **Campo** de la cuadrícula de diseño de la consulta para definir un campo nuevo o bien en la fila **Criterios** para definir criterios para un campo.

Las consultas que se usan en este sistema son las consultas con SQL, que nos permite realizar la combinación de campos de varias relaciones con pocas instrucciones SQL.

En suma, con las consultas se puede:

- Elegir campos
- Limitar registros
- Ordenar registros
- Formular preguntas sobre datos de varias relaciones

-
- Calcular totales
 - Crear formularios e informes basados en consultas
 - Crear consultas basadas en consultas
 - Crear gráficos basados en consultas
 - Formular preguntas sobre datos remotos

Por otro lado, la tercera parte fundamental de Access, son los formularios, son versátiles y muestran los datos que se desee recopilar.

Diseñar una forma de captura es algo que a los programadores les molesta mucho, ya que requiere de un tedioso trabajo de cálculo para que quede agradable a la vista, con buen gusto, con los colores adecuados, etc. Con Access se pueden diseñar formularios fáciles de utilizar y que presentan la información de la forma que usted desea. Aquí se tiene que mencionar una herramienta muy importante para la vista gráfica como lo es Visual Basic.

Visual Basic ofrece una rápida y fácil forma de crear aplicaciones para Microsoft Windows, una de las ventajas sobre Access son las tan variadas y excelentes Formularios de Visual Basic.

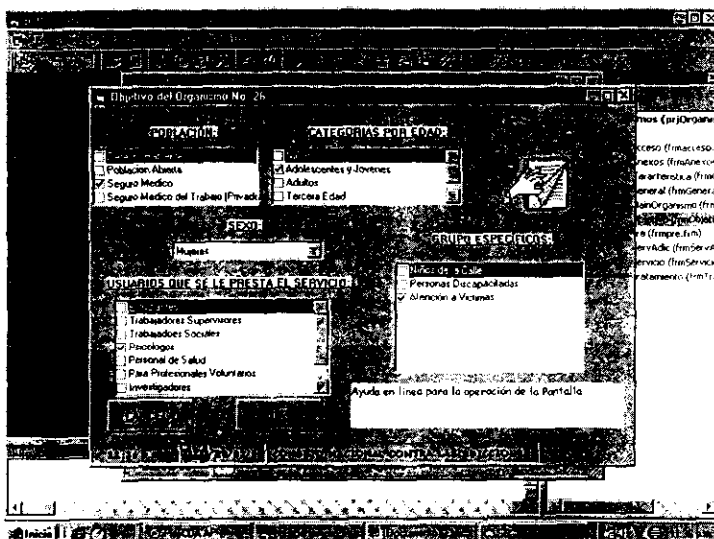
Un formulario sirve como una ventana que puede personalizar como la interfaz de su aplicación como un cuadro de diálogo que usa para obtener información del usuario. Un formulario puede existir individualmente o puede servir como un documento dentro de una interfaz de documento múltiple (MDI).

Cada formulario en su aplicación tiene un módulo de formulario asociado, estos son guardados con una extensión de archivo FRM y contienen:

- Los valores de las propiedades para el formulario y sus controles.
- Declaración de variables en el ámbito del formulario.

- Procedimientos de evento y procedimientos generales en el ámbito del formulario.

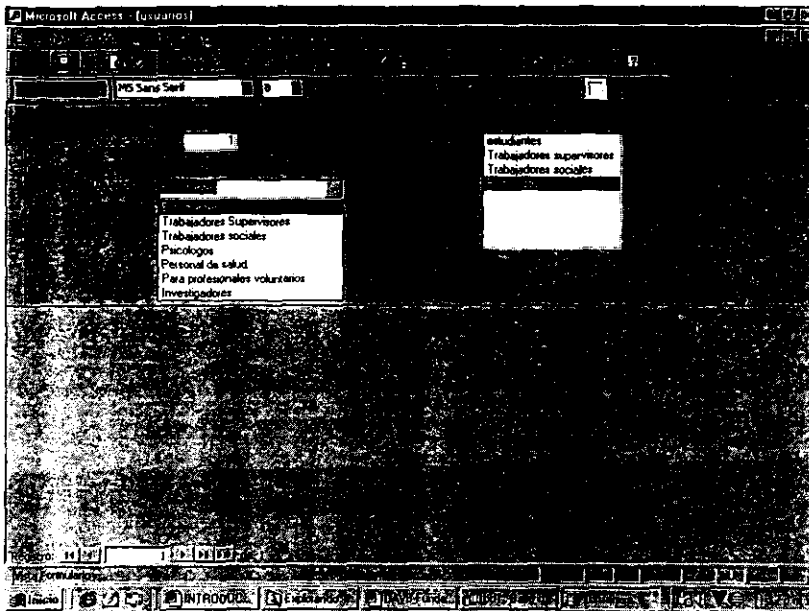
La principal importancia que se le dio a Visual Basic como un ambiente gráfico, es la excelente herramienta llamada listbox en la cual se pueden almacenar check box múltiples, la cual access no posee.



La decisión que se tomó con base a las ventajas que ofrece Visual Basic sobre Access, para la realización del sistema fueron:

- El sistema necesitaba tener listbox con checkbox y que estos fueran múltiples y que se pudieran elegir más de una opción.
- Visual Basic ofrece una enorme opción como la de construir archivos ejecutables, que es la herramienta de Visual Basic para la creación de los setup de instalación..

- Ofrece un ambiente gráfico mejorado a comparación de access.
- Visual Basic Ofrece una de las herramientas de enlace con manejadores de base de datos como es DAO (data Access Object).



Las funciones que desarrollan tanto Visual Basic como Access en el sistema son:

- Visual Basic actuando como una poderosa herramienta gráfica y por lo tanto una excelente presentación de pantallas, que hacen que el sistema tenga una excelente presentación.
- Access se convierte una herramienta de enlace, es decir, Access en el sistema actúa como el guardado de datos que se capturan en los formularios de Visual Basic.

El diseño de la base de datos es desarrollado en Access para este sistema y la presentación de pantallas se realiza por medio de Visual Basic.

Cabe hacer mención que el enlace de la base de datos con las pantallas es con el lenguaje de Visual Basic, que en realidad esta basado en el lenguaje Basic.

Para adentrarnos en el desarrollo del sistema daremos una breve explicación de cómo funciona el paquete Access:

Visual Basic reconoce tres categorías de Bases de Datos:

- *Bases de Datos Visual Basic.* También llamadas bases de datos *nativas*. Estas bases de datos utilizan el mismo formato que Microsoft Access y son creadas y manipuladas directamente por el motor de base de datos *Microsoft Jet*.
- *Bases de Datos externas.* En esta categoría se incluyen las bases de datos con métodos de acceso secuencial indexado (ISAM), como Microsoft Foxpro, Btrieve, dBASE, y Paradox, así como cada base de datos ODBC, o como el Servidor de SQL que reside en un Servidor Remoto.
- *Bases de datos ODBC.* Son las bases de datos cliente-servidor accedidas a través de estándar ODBC. ODBC es un protocolo estándar que permite a las aplicaciones conectarse con diferentes servidores de bases de datos o ficheros externos. Los controladores ODBC que utilizan el motor de bases de datos Microsoft Jet permiten tener acceso al Servidor SQL de Microsoft y a muchas otras bases de datos externas (SQL Server, Oracle, Informix, entre otras).

En el caso del sistema se utiliza bases de datos ODBC, es una especificación para una aplicación de programación de interfaces (API) que define las rutinas standard, con la cual una aplicación puede acceder a datos de alguna fuente.

Las aplicaciones ODBC para hacer referencias a funciones API directamente y también usando DAO, por lo tanto se pueden tener acceso a bases de datos externas.

En resumen las relaciones y consultas son realizadas en Access y los formularios son realizados en Visual Basic.

Cabe mencionar que el código que se escribe para obtener los listbox y checkbox en los formularios se hace en la ventana de código con la que cuenta Visual Basic.

En las aplicaciones tradicionales o procedurales, es la aplicación quien controla que porciones de código se ejecuta, y la secuencia en que este se ejecuta. La ejecución de la aplicación se inicia con la primera línea de código, y sigue una ruta predefinida a través de la aplicación, llamando procedimientos según sea necesario.

En las aplicaciones manejadas por eventos, la ejecución no sigue una ruta predefinida. En vez de esto, se ejecutan diferentes secciones de código en respuesta a eventos. Los eventos se desencadenan por acciones del usuario, por mensajes del sistema o de otras aplicaciones. La secuencia de eventos determina la secuencia en que el código se ejecuta. Es por esto que la ruta que sigue el código de la aplicación es diferente cada vez que se ejecuta el programa.

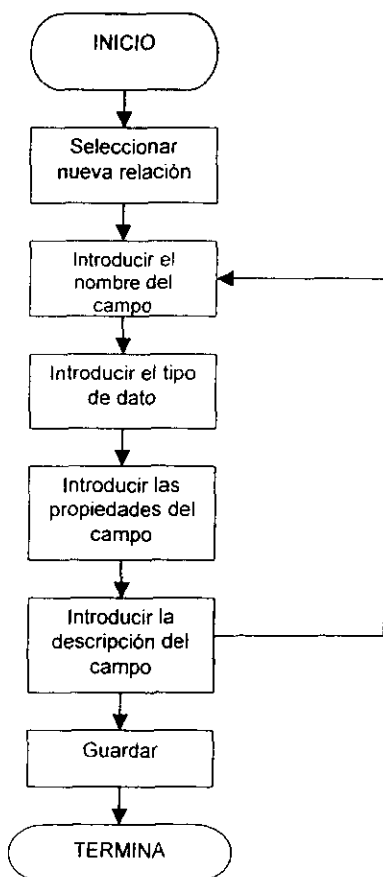
Una parte esencial de la programación manejada por eventos es el escribir código que responda a los posibles eventos que pueden ocurrir en una aplicación. Visual Basic facilita la implementación del modelo de programación manejada por eventos.

La *Ventana de Código* se usa para escribir, mostrar y editar el código de su aplicación. Puede abrir una ventana de código por cada módulo de su aplicación, de modo que puede fácilmente copiar y pegar entre ellos.

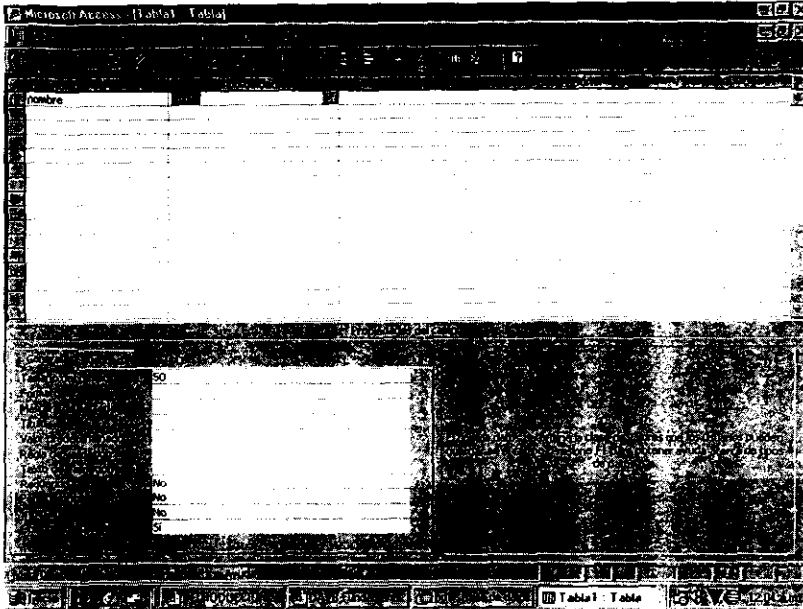
Habiendo definido brevemente lo que se puede realizar con el código y formularios de Visual Basic se procederá a explicar como fueron creadas las relaciones y consultas en Access de este sistema.

5.1 CREACIÓN DE RELACIONES.

A continuación se describe el proceso por el cual se crea una relación en Access. En principio, se muestra un diagrama del proceso, y después una breve explicación.



La relación que aparece en el manejador de bases de datos Access es la siguiente:



Al diseñar cada relación se especificaron los campos que se deseó que ésta contuviera. Esto se hace en la parte superior de la ventana de Relación

Después de asignar *nombre a un campo* se eligió el tipo de dato que almacenará dicho campo. Cuando se agrega un campo, Access le asigna automáticamente el tipo de dato "Texto". Si se desea que el campo tenga un tipo de datos diferente, se elige el tipo deseado en el cuadro de lista desplegable que aparece en la columna "Tipo de datos" para este campo.

Los tipos de datos se pueden seleccionar de la siguiente lista:

-
1. Texto
 2. Memo
 3. Numérico
 4. Fecha/Hora
 5. Moneda
 6. Autonumérico
 7. Sí/No
 8. Objeto OLE
 9. Hipervínculo
 10. Asistente para búsquedas

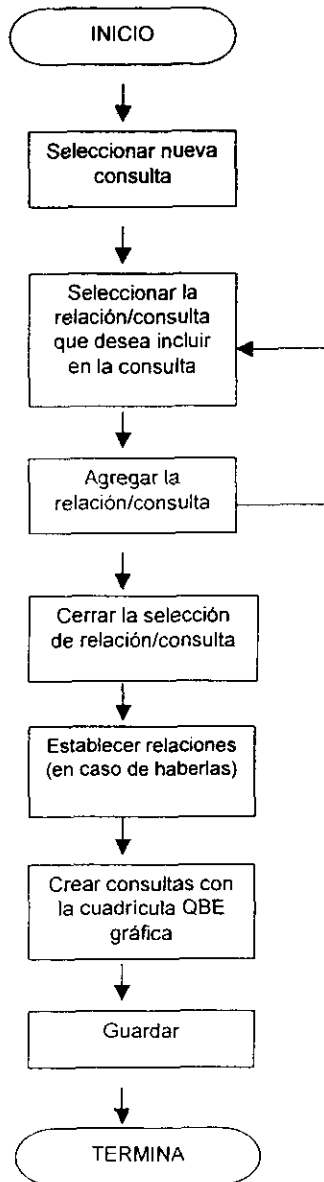
En la parte inferior de la ventana, se mejoró la relación estableciendo *propiedades* para cada campo. Las propiedades de los campos se enumeran a continuación:

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

Por último se guarda la relación con el nombre que uno elija y termina el proceso de creación de relación.

5.2 CREACIÓN DE CONSULTAS.

El proceso de la creación de una consulta se ejemplifica en el siguiente diagrama:

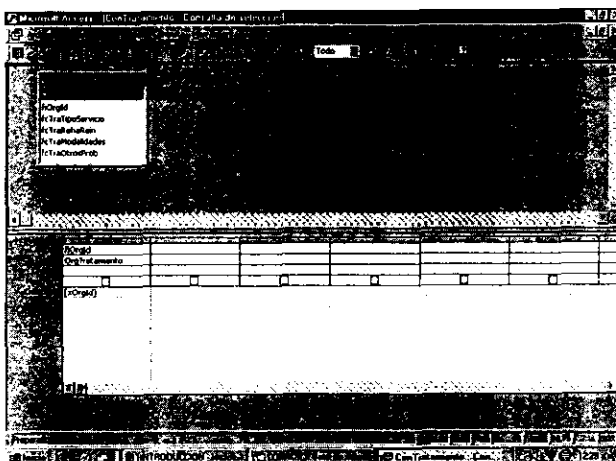


Al crear una consulta se abre una nueva ventana de consulta.

Después de haber agregado relaciones a la consulta, se hacen las relaciones entre éstas. Este proceso se realiza seleccionando un campo de una relación, y arrastrándolo hasta el campo equivalente de la otra relación.

Posteriormente, se seleccionan los campos que se desearon incluir en cada consulta. Los campos que se seleccionan, determinarán los datos que verá en la hoja de respuestas dinámica de las consultas (al presentarlos en forma de hoja de datos o formulario).

La lista de campos que aparece en la parte superior de la ventana de consulta muestra todos los campos disponibles para la consulta. En esta ventana, el diseño se realiza mediante una metodología denominada *consulta gráfica según ejemplo (QBE)*. Para crear consultas con la cuadrícula QBE gráfica, basta con arrastrar los campos deseados desde la parte superior de la ventana de consulta hasta la cuadrícula QBE (en la parte inferior de la ventana). En la cuadrícula QBE, cada columna contiene información sobre un campo incluido en la consulta, como se muestra en la siguiente figura:



Dentro de la cuadrícula QBE se tienen varios renglones de información, como:

Total: Donde se realizan cálculos sobre ciertos grupos de registros (Suma, Promedio, Mín, Máx, Cuenta, DesvEst, Var, Primero y Último), o condiciones como agrupar por, Expresión y Dónde.

Orden: Para especificar un tipo de ordenación para los registros (ascendente o descendente).

Mostrar: Para indicar si un campo debe aparecer o no en la hoja de respuestas dinámicas.

Criterios: Para que la hoja de respuestas dinámicas de la consulta se limite a ciertos registros. Al especificar criterios para una consulta se utiliza una expresión. Esta indica a Access los registros que debe incluir en la hoja de respuestas dinámicas.

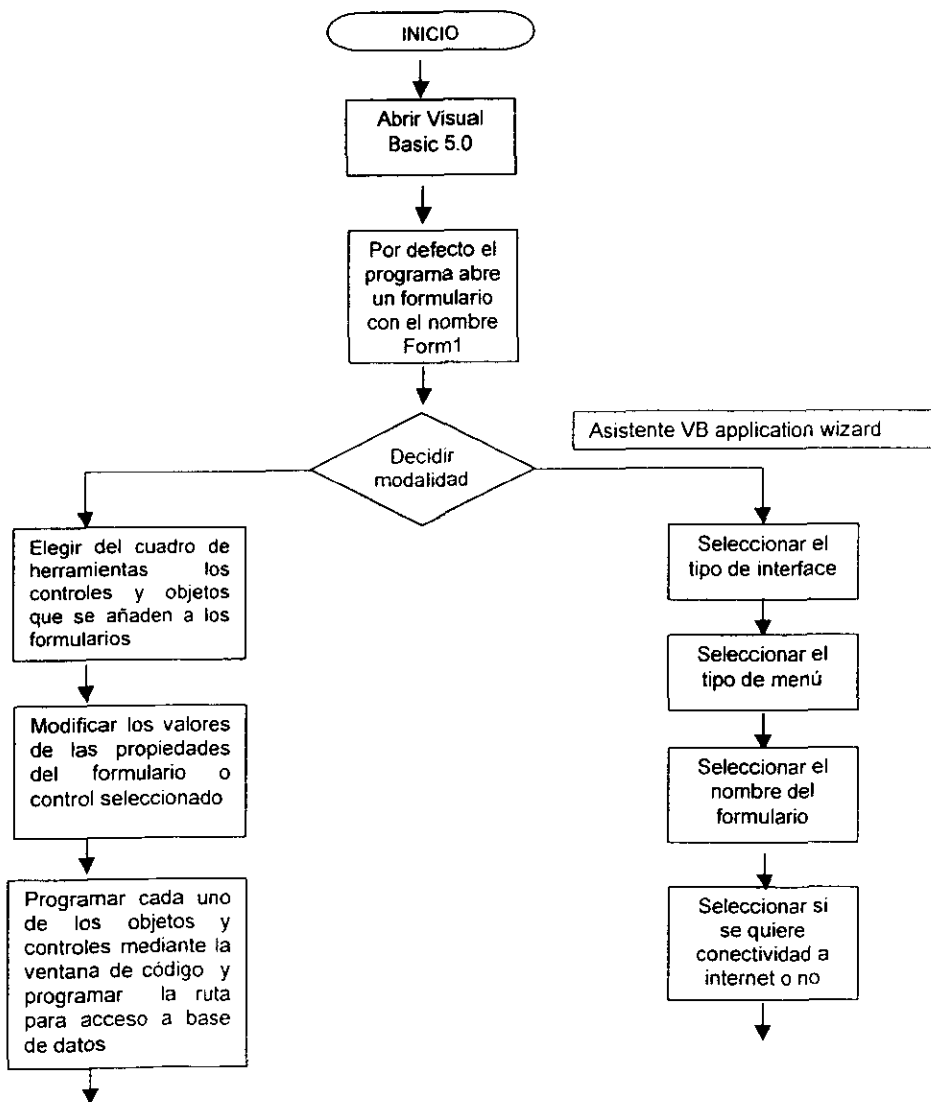
Finalmente, se guarda la consulta con un nombre significativo.

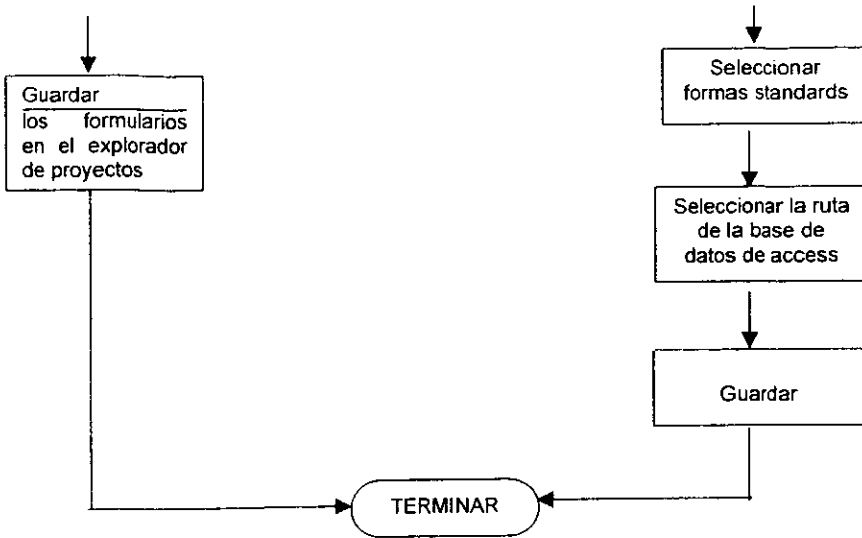
Cabe mencionar que una consulta **no puede** tener el mismo nombre que una relación.

Por otro lado, al abrir la ventana de consultas, se abre también su propia barra de herramientas, la cual ofrece métodos abreviados para ejecutar los comandos que se utilizan al diseñar una consulta.

5.3 CREACIÓN DE FORMULARIOS.

El proceso para crear un formulario en Visual Basic se describe mediante los siguientes pasos:





Una vez que se decidió hacer los formularios con el asistente para formularios, o empezar con uno en blanco, se proceden a tomar las siguientes decisiones:

En caso de decidirnos por el asistente para formularios, se puede crear un formulario que conste de:

- **Menús.-** En donde se incluyen menús de edición, archivo, ver, ventana y ayuda.
- **Controles y herramientas.-** En donde se agregan las herramientas como cuadros de textos, etiquetas, dbgrid, botones de comandos.
- **Control data.-** En donde se hace referencia a la base de datos, además de las relaciones y consultas.

En caso de decidirnos por formularios en blanco, se hace la creación propia del mismo.

En las cuales se deben tomar en cuenta los pasos siguientes:

- **Realizar el diseño de la base de datos** en Access o en otro manejador de base de datos compatible con Visual Basic.
- **Crear los formularios** con sus controles y objetos con ayuda de la barra de herramientas.
- **Ligar los objetos y eventos** con la ventana de código para que los formularios obtengan los datos de la base de datos.
- **Guardar los formularios** así como los proyectos.

Los módulos dentro de Visual Basic es la conexión a la base de datos y de ahí que se mencione el código más importante:

```
Public IsComando As String
Public db As Database
Public liOrganismo As Long
Public liOpcion As Integer
Public Sub Main()
Dim strBaseLocalizada As String

IsComando = Command
If IsComando = "" Then
    MsgBox "<< ERROR >> No Hay Parametros Para Ejecución !!!", vbCritical, "Inicio Prorama"
End If

Set db = DBEngine.Workspaces(0).OpenDatabase(IsComando, , True)
```

liOrganismo = 0

liOpcion = 0

frmacceso.Show

End Sub

Public Function ObtieneCadenaPorLista(PoObjeto As Object) As String

Dim lbPaso As Boolean

For i% = 0 To (PoObjeto.ListCount - 1)

If PoObjeto.Selected(i%) Then

If Not lbPaso Then

ObtieneCadenaPorLista = PoObjeto.ItemData(i%)

lbPaso = True

Else

ObtieneCadenaPorLista = ObtieneCadenaPorLista & "," & PoObjeto.ItemData(i%)

End If

End If

Next i%

End Function

```
Public Sub ValorCombo(PoObjeto As Object, PsCampo As String, PbSinRegistros)
```

```
    If PbSinRegistros Then
```

```
        PoObjeto.ListIndex = 0
```

```
    Else
```

```
        For i% = 0 To (PoObjeto.ListCount - 1)
```

```
            If PoObjeto.ItemData(i%) = Val(PsCampo) Then
```

```
                PoObjeto.ListIndex = i%
```

```
            Exit Sub
```

```
        End If
```

```
    Next i%
```

```
End If
```

```
End Sub
```

```
Public Sub ValorLista(PoObjeto As Object, PsCadena As String)
```

```
    Dim X As Integer
```

```
    Dim z As Integer
```

```
    Do While Len(PsCadena) <> 0
```

```
        X = InStr(1, PsCadena, ",")
```

```
        If X = 0 Then
```

```
            z = PsCadena
```

```
        Else
```

```
            z = Mid(PsCadena, 1, X)
```

```
        End If
```

```
        For i% = 0 To (PoObjeto.ListCount - 1)
```

```
            If PoObjeto.ItemData(i%) = z Then
```

```
                PoObjeto.Selected(i%) = True
```

```
            Exit For
```

```
End If
Next i%
If X = 0 Then
    Exit Do
Else
    PsCadena = Mid(PsCadena, X + 1, Len(PsCadena))
End If
Loop

PoObjeto.ListIndex = 0

End Sub

Public Sub LlenerCatalogo(PiPadre As Integer, _
    PoObjeto As Object, _
    PbCombo As Boolean)

Dim rsCatalogos As Recordset
Dim qrCatalogos As QueryDef

Set qrCatalogos = db.QueryDefs("LlenerCatalogos")
qrCatalogos.Parameters("xPadre") = PiPadre
Set rsCatalogos = qrCatalogos.OpenRecordset(dbOpenSnapshot)

If PbCombo = True Then
    'If pbobjeto.Name = "cboGenero" Then
        'No pone Sin Asignar
    'Else
        PoObjeto.AddItem "Sin Asignar"
        PoObjeto.ItemData(PoObjeto.NewIndex) = 0
    'End If
```

End If

If Not rsCatalogos.EOF Then

rsCatalogos.MoveFirst

End If

Do While Not rsCatalogos.EOF

PoObjeto.AddItem rsCatalogos("fcConcDetNombre")

PoObjeto.ItemData(PoObjeto.NewIndex) = rsCatalogos("fiConcPadreId")

rsCatalogos.MoveNext

Loop

PoObjeto.ListIndex = 0

qrCatalogos.Close

rsCatalogos.Close

Set qrCatalogos = Nothing

Set rsCatalogos = Nothing

End Sub

Dentro de los formularios los objetos recordset instancian a la base de datos y con una sola instrucción de SQL podemos llamar datos a un formulario de Visual Basic. Se debe mencionar que en este trabajo y sistema la instanciacion se realiza de la misma forma en todos los formularios.

If liOpcion = 0 Then

*lsComando = "SELECT * FROM OrgAnexo"*

Set rsObj = db.OpenRecordset(lsComando, dbOpenDynaset)

rsObj.AddNew

rsObj(0) = PiOrganismold

Else

*IsComando = "SELECT * FROM OrgAnexo WHERE fiOrgId = " &
LTrim(Str(PiOrganismold))*

Set rsObj = db.OpenRecordset(IsComando, dbOpenDynaset)

rsObj.Edit

End If

CAPÍTULO VI
PRUEBAS Y AJUSTES

CAPÍTULO VI

PRUEBAS Y AJUSTES

En este capítulo se exponen los tipos de pruebas aplicados al sistema, los resultados obtenidos durante la implantación de este, y cómo fueron resueltos los problemas, además de las ventajas y desventajas del sistema, las propuestas de mantenimiento y las mejoras.

El objetivo de hacer las pruebas es comprender la importancia de esta fase para garantizar la confiabilidad y asegura la calidad del sistema.

6.1 TIPOS DE PRUEBAS.

La prueba es uno de los pasos de la ingeniería de software que se puede ver como destructivo en lugar de constructivo, ya que es un proceso de ejecución de un programa con la intención de descubrir un error. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces, y tiene éxito si descubre un error no detectado hasta ese momento.

Las pruebas también demuestran hasta qué punto el software cumple con las especificaciones y requerimientos de rendimiento planteados desde el análisis; aunque estas no pueden asegurar la ausencia de defectos.

El desarrollo del sistema fue sometido a un proceso de verificación y validación con el fin de determinar el grado en el que se cumplieron las especificaciones de requisitos previas y mejorar, en lo posible, se utilizaron diversos tipos de pruebas, entre las que podemos citar:

Las pruebas de unidad y de integración.

La primera, también llamada de "Caja Blanca", se realiza al haber terminado de automatizar cada módulo pretendiendo descubrir los errores inherentes a este.

La prueba de caja blanca es un método de diseño que utiliza la estructura de control del diseño procedimental para derivar los casos de prueba. Mediante este método se derivaron los casos de prueba

- Garantizaron que se ejercitan por lo menos una vez todos los caminos independientes de cada módulo.
- Se ejercitaron todas las decisiones lógicas en sus caras verdaderas y falsas.
- Se ejecutaron todas las iteraciones en sus límites y con sus límites operacionales.
- Se ejecutaron las estructuras internas de datos para asegurar su validez.

La segunda prueba, de integración, se realiza al juntar varios módulos, pretendiendo descubrir los errores que surgieron al juntar dichos módulos, y se describe a continuación. Una vez finalizados los módulos, se llevó a cabo la integración del sistema. Esta integración fue realizada mediante la metodología de integración incremental descendiente, es decir, primero se integró el módulo de mayor nivel jerárquico, simulando cada uno de los módulos siguientes con la llamada de un formulario en blanco, después, cada módulo simulado fue reemplazado por su módulo real y sus submódulos, que a su vez fueron probados de la misma forma, para llegar a la parte inferior del sistema.

Por otra parte, el proceso de validación consistió en hacer pruebas de validación, también llamadas de aceptación, las cuales se realizan para evitar sorpresas cuando un

sistema pasa a la implantación y evalúa el trabajo de análisis; además, el resultado de esta prueba es binaria: Se acepta o se rechaza el sistema ya que no forma parte de un proceso de corrección de errores, sino que única y exclusivamente se deriva del documento de objetivos.

Pruebas de validación.

Son pruebas que se llaman comúnmente "Caja Negra" y se dividen en cinco tipos diferentes:

- **Pruebas de trayectoria normal.** Por ejemplo, para cada ítem se toma tres valores: los extremos y un valor medio, o las permutaciones de ítems que sean válidas.
- **Pruebas de excepción.** Por ejemplo, seleccionar valores inválidos para cada ítem.
- **Pruebas de estado transitorio.** Por ejemplo, secuencias de flujos de datos (altas, bajas, cambios, etc.).
- **Pruebas de rendimiento.** Es decir, se hará una prueba por cada factor de rendimiento a evaluar (tiempo de respuesta, capacidad de volumen, tiempo de CPU, espacio en disco, etc.).
- **Pruebas especiales.** Por ejemplo, impresiones.

En esta sección de pruebas de validación se aplicaron entradas al sistema para analizar y verificar que las salidas correspondieran a los resultados esperados. Estas entradas en un principio fueron seleccionadas de datos pruebas que tienen

características simples o problemáticas, como son entradas típicas y datos inválidos, posteriormente se introdujeron datos reales.

Una vez hechas todas las pruebas necesarias e integrado el sistema, este fue validado mediante una serie de pruebas de aceptación por parte del usuario.

6.2 PROBLEMAS QUE SE PRESENTARON EN EL DESARROLLO DEL SISTEMA.

Los principales problemas que se presentaron fueron:

- Llamar a la relación de concepto dentro de los listbox y combo box.
- El llamar las ramas del árbol de información para extraer su información, así como, su conceptualización.
- Realizar los programas ejecutables.
- El llamar a las relaciones por medio del Record set de Visual Basic.
- La utilización del Module de Visual Basic para realizar el OpenDatabase.

6.3 VENTAJAS Y DESVENTAJAS DEL SISTEMA.

1. El sistema es muy fácil de utilizar para aquellas personas que no poseen grandes conocimientos de computación, además, como se trabaja bajo plataforma Windows el ambiente gráfico es más accesible.
2. La información es compatible con otros tipos de formatos, ya sea para manejadores de bases de datos entre los que podemos mencionar FoxPro, Paradox, SQL ,Oracle, Sybase, etc. ; así como para el procesador de texto Word.

-
3. Los combo box, listbox y checklist; son una gran ventaja que tiene Visual Basic sobre otros manejadores de bases de datos y se hace uso de estos controles en todos los formularios, como es el caso de organismos.

 4. La ventaja de Visual Basic en el control de árbol en cual se puede obtener información de varios niveles, este es el caso del árbol de información.

CONCLUSIONES.

CONCLUSIONES.

Este trabajo fue realizado con la intención de resolver el problema de la rehabilitación ofreciendo diferentes opciones de acuerdo a las condiciones económicas del usuario para la línea de CONADIC, empleando conocimientos adquiridos durante la carrera de Ingeniería en Computación, además de ponerlos en práctica y emplear tecnología de vanguardia en el desarrollo de sistemas.

Para poder recolectar los requisitos de información referente a los bancos de datos, se realizaron con gran frecuencia, entrevistas con personal que labora en el CONADIC. Además se hizo la consulta de diferentes documentos generados por el personal y se examinó un sistema que en ese tiempo utilizaban.

El desarrollo del SISTEMA DE APOYO INFORMÁTICO PARA LA FUTURA LÍNEA DE INFORMACIÓN DEL CONADIC, ha permitido a los autores del mismo obtener una serie de experiencias tanto en el ámbito personal como profesional, mismas que generan las siguientes conclusiones:

La puesta en marcha del sistema mejora significativamente los tiempos requeridos para la captura, proceso y respuesta de canalizaciones a los usuarios.

La operación sistematizada de los datos favoreció al proceso de operaciones en forma manual, la confiabilidad de la información y sobre todo la importancia del banco de datos en cuanto al guardado de la operación.

Técnicamente, el desarrollo del sistema permitió comprobar el uso de tecnología de vanguardia como son las herramientas CHECKBOX múltiples y COMBO BOX múltiples en conjunto con un lenguaje de cuarta generación (VISUAL BASIC).

En la solución de este problema en específico, se logró lo siguiente:

- Un producto acorde a las necesidades del usuario.
- Reducción de costos y tiempos de desarrollo.
- Que el mantenimiento es más sencillo de realizar.
- Se cuenta con pantallas con un front gráfico para su mejor manejo y así tener compatibilidad con Microsoft.

El banco de datos que va generando el sistema es lo suficientemente flexible y dinámico para poder tener información al día y para ampliarla de acuerdo a las necesidades que aparezcan.

La recuperación de la información se realiza por medio de archivos con extensión .RTF para su mejor consulta en cualquier procesador de textos, presentada a detalle o resumida de acuerdo a los requerimientos del CONADIC.

El diseño modular con el que fue creado, permite la incorporación gradual de nuevas aplicaciones que habiliten operaciones relacionadas, lo cual asegura que el rango de operación del sistema no sólo permanecerá vigente, sino que se tendrá la capacidad de ampliar progresivamente sus alcances.

Por lo anterior, se concluye que el sistema cumple con los objetivos planteados al inicio del desarrollo, cubriendo los requerimientos de información para los usuarios como para los operadores y ofreciendo la sistematización adecuada para la reducción en tiempo de respuesta con un alto grado de confiabilidad en la información.

En el aspecto de desarrollo profesional, el sistema nos permitió obtener experiencia en el manejo de conocimientos adquiridos durante la información educativa en el área de Ingeniería en Computación, aplicados a la solución de problemas reales con requerimientos específicos.

Además nos permitió constatar que la información profesional en Ingeniería, amplía la perspectiva del profesional en otras áreas de conocimiento que no están directamente relacionadas con ella, pero que, sin embargo, requieren de sistemas como herramienta para la automatización y agilización de funciones y procesos.

Cabe hacer mención que dentro de una área, como lo es, la Ingeniería en Computación, siempre es importante la actualización en tecnologías de desarrollo.

DICCIONARIO DE DATOS.

DICCIONARIO DE DATOS.

RELACIÓN: ORGCARACTERÍSTICAS.

| Nombre | Tipo | Longitud |
|---------------------|------------|---------------|
| fiOrgId | N Numérico | Entero largo |
| FcCarDirGral | Texto | 50 |
| fcCarDirGralTel | Texto | 30 |
| fcCarDirUnidad | Texto | 50 |
| fcCarDirUnidadTel | Texto | 30 |
| fcCarHorario | Texto | 15 |
| fcCarAreaInfluencia | Texto | 150 |
| fcCarAreaCubiertas | Texto | 150 |
| fcCarFechas | Fecha/Hora | Fecha general |
| fcCarDias | Texto | 10 |

RELACIÓN: ORGCONTACTO

| Nombre | Tipo | Longitud |
|---------------|--------|----------|
| FiOrgId | Número | 50 |
| FcConNombre | Texto | 50 |
| fcConTelefono | Texto | 30 |

RELACIÓN: ORGGENERALES

| Nombre | Tipo | Longitud |
|---------------|--------------|-----------------|
| fiOrgId | Autonumérico | Entero largo |
| fcOrgNombre | Texto | 80 |
| fiOrgTipo | Numérico | Entero largo |
| fiOrgCosto | Numérico | Entero largo |
| fiOrgCtoNin | Numérico | Entero largo |
| fiOrgCtoJuv | Numérico | Entero largo |
| fiOrgCtoAdul | Numérico | Entero largo |
| fcOrgDir | Texto | 50 |
| fcOrgCol | Texto | 50 |
| fcOrgCiudad | Texto | 50 |
| fcOrgTel | Texto | 30 |
| fcOrgExt | Texto | 10 |
| fcOrgFax | Texto | 15 |
| fiOrgDeleg | Numérico | Entero largo |
| fiOrgCp | Numérico | Entero largo |
| fiOrgEstado | Numérico | Entero largo |
| fcOrgEMail | Texto | 50 |
| fcOrgInternet | Texto | 50 |
| fcOrgPicture | Texto | 50 |
| fiOrgStatus | Si/No | Si/No |

RELACIÓN: ORGOBJETIVO

| Nombre | Tipo | Longitud |
|------------------|-------------|-----------------|
| fiOrgId | Numérico | Entero largo |
| fcObjTipoCob | Texto | 150 |
| fcObjCatEdad | Texto | 150 |
| fcObjUsuServicio | Texto | 150 |
| fcObjGrupoEspec | Texto | 150 |
| fiObkGenero | Texto | 50 |

RELACIÓN: SERVADICIONAL

| Nombre | Tipo | Longitud |
|--------------------|-------------|-----------------|
| fiOrgId | Numérico | Entero largo |
| fiAdiAsesoria | Numérico | Entero largo |
| fcAdiInvestigacion | Texto | 150 |
| fcAdiTipInves | Texto | 150 |
| fcAdiCapInt | Texto | 150 |
| fcAdiCapExt | Texto | 150 |
| fcAdiCapEsp | Texto | 150 |

RELACIÓN: ORGSERVICIO

| Nombre | Tipo | Longitud |
|--------------------|----------|--------------|
| fiOrgId | Numérico | Entero largo |
| fcSerEmergencias | Texto | 150 |
| fcSerPrevPrimaria | Texto | 150 |
| fcSerCentroDocu | Texto | 150 |
| fcSerReqServ | Texto | 150 |
| fcSerPublicaciones | Texto | 150 |

RELACIÓN: ORGTRATAMIENTO

| Nombre | Tipo | Longitud |
|-------------------|----------|--------------|
| fiOrgId | Numérico | Entero largo |
| fcTraTipoServicio | Texto | 150 |
| fcTraRehaRein | Texto | 150 |
| fcTraModalidades | Texto | 150 |
| fcTraOtrosProb | Texto | 150 |

RELACIÓN: DETALLE

| Nombre | Tipo | Longitud |
|--------------|--------------|--------------|
| Id_detalle | Autonumérico | Entero largo |
| Id_Subtitulo | Numérico | Entero largo |
| Id_titulo | Numérico | Entero largo |
| Id_tema | Numérico | Entero largo |
| Concepto | Texto | 50 |
| status | Sí/No | Sí/No |
| detalle | Texto | 50 |

RELACIÓN: SUBTÍTULO

| Nombre | Tipo | Longitud |
|---------------|--------------|-----------------|
| Id_Subtitulo | Autonumérico | Entero largo |
| Id_titulo | Numérico | Entero largo |
| Id_tema | Numérico | Entero largo |
| concepto | Texto | 100 |
| status | Sí/No | Sí/No |
| detalle | Texto | 50 |

RELACIÓN: TEMA

| Nombre | Tipo | Longitud |
|---------------|--------------|-----------------|
| Id_tema | Autonumérico | Entero largo |
| concepto | Texto | 100 |
| status | Sí/No | Sí/No |
| Detalle | Texto | 50 |

RELACIÓN: TÍTULO

| Nombre | Tipo | Longitud |
|---------------|--------------|-----------------|
| Id_titulo | Autonumérico | Entero largo |
| Id_tema | Numérico | Entero largo |
| concepto | Texto | 100 |
| Status | Sí/No | Sí/No |
| Detalle | Texto | 50 |

RELACIÓN: ORGANEXO

| Nombre | Tipo | Longitud |
|----------------------|-------------|-----------------|
| fiOrgId | Numérico | Entero largo |
| fiAneNumPersonal | Numérico | Entero |
| fcAneRepSolicitantes | Texto | 150 |
| fbAneEnviaConadic | Sí/No | Sí/No |
| fcAneObserva | Texto | Texto |
| fiAneDaSeg | Sí/No | Sí/No |

RELACIÓN: ACCESO

| Nombre | Tipo | Longitud |
|---------------|-------------|-----------------|
| fcAccesold | Texto | 8 |
| fcAccNombre | Texto | 40 |
| fcAccPass | Texto | 10 |

RELACIÓN: CONCEPTO

| Nombre | Tipo | Longitud |
|-----------------|--------------|-----------------|
| fiConcPadreId | Autonumérico | Entero largo |
| fiConcHijold | Numérico | Entero largo |
| fcConcDetNombre | Texto | 50 |
| fcConcTipo | Texto | 1 |
| fiConcDetstatus | Sí/No | Sí/No |

RELACIÓN: FOLIO

| Nombre | Tipo | Longitud |
|---------------|--------------|-----------------|
| Id_Folio | Autonumérico | Entero largo |
| DetalleFolio | Texto | 50 |
| Folio | Numérico | Entero largo |
| Ruta | Texto | 50 |

RELACIÓN: USUARIO

| Nombre | Tipo | Longitud |
|---------------|-------------|-----------------|
| Usuario_id | Texto | 6 |
| Password | Texto | 6 |
| Nombre | Texto | 60 |
| Acceso | Si/No | Si/No |
| Sistema | Si/no | Si/No |

BIBLIOGRAFÍA.

BIBLIOGRAFÍA.

- C.J. Date. *Introducción a los sistemas de bases de datos Volumen 1*, Quinta edición, Addison-Wesley Iberoamericana, 1998.
- Gary W. Hansen. *DataBase Management And Design*. Segunda Edición, Prentice Hall, 1997
- Tom De Marco. *Structured Analysis and System Specification*, Prentice Hall, 1979.
- Roger S. Pressman. *Ingeniería de Software Un enfoque práctico*. Tercera Edición , Mc. Graw Hill, 1997.
- Timothy M. O'Brien. *Microsoft Access 97, Developer's Handbook*, Microsoft Press, 1997.
- William R. Vaughn. *Visual Basic & SQL-Server*, Quinta Edición, Microsoft Press, 1997.
- Fairley R. *Ingeniería de software*, Cuarta Edición, Mc. Graw Hill, 1998.
- Kendall K y Kendall J., *System analysis and design*, Segunda edición, Prentice Hall, 1997.
- James A. Senn, *Análisis y diseño de sistemas de información*, Quinta edición, Mc. Graw Hill, 1996.

-
- Alice Y.H. Tsai, *Sistemas de base de datos, administración y uso*, Cuarta Edición, Prentice Hall, 1991.
 - Jeffrey D. Ullman, *Introducción a los Sistemas de Bases de Datos*, Tercera Edición, Prentice - Hall , 1999.
 - Finkelstein C. , *An introduction to information engineering*, Tercera Edición, Addison-Welsey, 1997.

ANEXO 1.

Manual Técnico

MANUAL TÉCNICO.

Relaciones.

Una relación es un conjunto de datos sobre un tema concreto, como un inventario, o clientes. Usando una relación independiente para cada tema, evitará la duplicación de datos, y la Base de Datos resultará más eficiente, al tiempo que se reducirá el riesgo de errores en la entrada de datos.

En las relaciones los datos están organizados en filas (denominadas *registros*) y columnas (denominados *campos*). Esta organización queda patente en la vista hoja de datos

En Access 97 existen cuatro formas de crear una relación en blanco (vacía):

- Utilizar el asistente para Bases de Datos con el fin de crear en una sola operación todas las relaciones, formularios e informes necesarios para una Base de Datos completa. El Asistente para Bases de Datos crea una Base de Datos nueva; no puede usarse para añadir relaciones, formularios o informes a una Base de Datos existente.
- Utilizar el asistente para relaciones para elegir los campos de la relación entre una gran variedad de relaciones predefinidas.
- Introducir los datos directamente en una hojas de datos en blanco. Al guardar la nueva hoja de datos, Microsoft Access analiza los datos y le asigna automáticamente el tipo de datos y el formato apropiados para cada campo.
- Utilizar la vista Diseño para especificar todos los detalles de la relación partiendo después de cero.

Asignar nombres a los campos.

Para agregar un campo, el primer paso consiste en escribir un nombre para el campo en la columna "Nombre del campo". Los nombres de campo pueden tener 64 caracteres como máximo (letras o números), incluyendo espacios en blanco.

Para identificar fácilmente los campos, los nombres deben ser lo suficientemente explicativos, de forma que se puedan identificar fácilmente cuando se vean o editen registros.

Tipos de campos.

En la siguiente relación se resumen los tipos de campo contemplados en Access 97, así como la utilidad de cada uno de ellos.

- **Texto.** En este tipo de campo se almacena texto o combinaciones de texto y números, como por ejemplo, direcciones. También se guardan aquí números con los que no se van a realizar cálculos, como números de teléfonos o códigos postales. Para controlar el número máximo de caracteres que pueden introducirse, utilice la propiedad Tamaño del campo (en la vista Diseño).
- **Memo.** Se utiliza para almacenar texto de gran longitud, como notas o descripciones.
- **Numérico.** Se utiliza para almacenar datos numéricos que pueden ser utilizados en todo tipo de operaciones matemáticas, excepto operaciones monetarias (para este tipo de cálculos, utilice el tipo Moneda). Use la propiedad Tamaño del campo para definir el tipo Numérico específico.

-
- **Fecha/Hora.** Sirven para almacenar fechas y horas.
 - **Moneda.** Los campos de este tipo se utilizan para almacenar valores monetarios. Para los números almacenados en este tipo de campo se tiene una precisión de 15 dígitos a la izquierda de la coma decimal y 4 dígitos a la derecha.
 - **Autonumérico.** Este tipo de campo almacena exclusivamente números secuenciales (en incrementos de una unidad) o números aleatorios insertados automáticamente cuando se agrega un registro.
 - **Si/No.** Son campos que sólo contienen uno de dos valores, como Sí/No, Verdadero/Falso, Activado/Desactivado.
 - **Objeto Ole.** En este campo se pueden insertar objetos de cualquier tipo creados en otros programas (como documentos de Microsoft Word, hojas de cálculo de Microsoft Excel, imágenes, sonidos u otros datos binarios) mediante el protocolo OLE y pueden ser vinculados a, o incrustados en una relación de Access.
 - **Hipervínculo.** Tipo de campo que sirve para almacenar hipervínculos. Un hipervínculo puede ser una ruta de acceso a una ubicación en una red local, o una dirección URL para acceder a una página Web.
 - **Asistente para búsquedas.** Crea un campo que permite elegir un valor de otra relación o de una lista de valores mediante un cuadro combinado.

Al elegir esta opción en la lista de tipos de datos, se inicia un asistente que permite definirlo automáticamente.

Los tipos de campo *Numérico*, *Fecha/Hora*, *Moneda* y *Si/No*, disponen de formatos de visualización predefinidos. Utilice la propiedad Formato para seleccionar

uno de los formatos disponibles para cada tipo de datos. También puede crear un formato de visualización personalizado.

Propiedades de los campos.

En la siguiente relación se enumeran las propiedades de los campos y sus usos.

Tamaño Del Campo. Limitar a un cierto tamaño un campo con tipo de datos "Texto", o limita un campo con tipo de datos "Numérico" a valores incluidos en un rango de ese tipo de datos.

Formato. Presentar las fechas y números en un determinado formato, como "Fecha larga" o "Moneda".

Máscara de entrada. Puede utilizar la propiedad para facilitar la entrada de datos y para controlar los valores que los usuarios pueden introducir en un control de cuadro de texto.

Lugares Decimales. Presentar un cierto número de posiciones después del separador decimal (cuando se utiliza un formato para un campo de tipo "Numérico" o "Moneda").

Título. Especificar una etiqueta distinta de la etiqueta predeterminada (nombre del campo) para los nuevos formularios o informes.

Valor Predeterminado. Rellenar este campo automáticamente con un cierto valor, en los nuevos registros que agregue a la relación.

Regla De Validación, Texto De Validación. Limitar los datos introducidos en un campo a valores que cumplan un requisito determinado.

Requerido. Puede usar la propiedad para especificar si es necesario que exista un valor en un campo. Si el valor de esta propiedad es Sí, al introducir datos en un registro deberá especificar un valor en el campo en cualquier control independiente del mismo, y además el valor no podrá ser nulo.

Permitir longitud cero. Puede usar la propiedad para especificar si una cadena de longitud cero (" ") es una entrada válida para un campo de una relación.

Indexado. Acelerar las búsquedas en campos en los que se busca con frecuencia.

Campos Numéricos.

En el caso de un campo de tipo numérico, el valor de la propiedad TamañoDelCampo determina el rango de valores que pueden almacenarse en dicho campo, y también determina si el campo podrá o no valores fraccionario.

En la siguiente relación se muestran los valores de la propiedad TamañoDelCampo para los campos Numéricos y el rango de valores permitidos.

| Valor de TamañoDelCampo | Rango | Longitud de campo | Tamaño de campo |
|-------------------------|--|-------------------|-----------------|
| Byte | 0 a 255 | Ninguno | 1 byte |
| Entero | -32768 a 32767 | Ninguno | 2 bytes |
| Entero largo | -2147,483648 a 2147,483647 | Ninguno | 4 bytes |
| Simple | -3.4×10^{38} a 3.4×10^{38} | 7 | 4 bytes |
| Doble | -1797×10^{308} a 1797×10^{308} | 15 | 8 bytes |

Formatos tipo Numérico y la propiedad de LugaresDecimales.

En la siguiente relación se muestran los formatos disponibles para los campos de tipo Numérico o Moneda, así como la forma en que se presentarían los datos en un formulario o en una hoja de datos.

| Formato | Se muestra como | Se muestra como |
|----------------|-----------------|-----------------|
| Número general | 1234,5 | 1234,5 |
| Moneda | 1234,5 | 1.234,50 Pts. |
| Fijo | 1234,5 | 1235 |
| Estándar | 1234,5 | 1.234,50 |
| Porcentaje | 0,824 | 82,40% |
| Científico | 1234,5 | 1.23E+03 |

Formatos tipo Fecha/Hora.

En la siguiente relación se muestran los formatos disponibles para los campos que tienen el tipo de datos Fecha/Hora y la forma en que se presentan los datos en un formulario o una hoja de datos.

| Formato | Los datos/horas aparecen así |
|---------------|------------------------------|
| Fecha general | 31/01/2000 16:30:00 |
| Fecha larga | Viernes 31 de Enero de 2000 |
| Fecha mediana | 31-Ene-2000 |
| Fecha corta | 31/01/2000 |
| Hora larga | 16:30:00 |
| Hora mediana | 04:30 PM |
| Hora corta | 16:30 |

Formatos personalizados.

Para establecer la propiedad Formato de manera que indique un formato personalizado, basta con introducir los símbolos de formato que permitirán obtener la presentación deseada. En la siguiente relación se muestran algunos formatos personalizados útiles.

| Formato | Ejemplo | Resultado |
|----------------------|-----------|--------------------|
| "cuenta número" 0000 | 123 | Cuenta número 0123 |
| mmm aaaaa | 31-Ene-92 | Ene 1992 |
| hhmm "horas" | 4:00 PM | 1600 horas |

Partes de una expresión.

Las expresiones se utilizan para realizar cálculos para establecer criterios para consultas, para controlar macros, como argumentos de funciones y como elementos de módulos Access Basic. Una expresión puede incluir cualquier combinación de operadores, identificadores, funciones, valores literales y constantes.

| Elemento | Ejemplos | Descripción |
|---------------|--|--|
| Operador | +, =, >, &, Y, O, como | Indica una operación que se va a realizar sobre uno o más elementos. |
| Identificador | Formulario![Pedidos] [Núm de pedido], Informes![Factura].Visible | Se refiere al valor de un campo, un control o una propiedad. |

| | | |
|-----------|--------------------------------|--|
| Función | Fecha, Suma, DBúsq | Devuelve un valor, basándose en el resultado de un cálculo o de otra operación. Además, puede crear sus propias funciones utilizando Access Basic. |
| Literal | 100, #1-Ene-2000#, "Toledo" | Representa un número, una cadena o una fecha que Access utilizará tal y como se haya escrito. |
| Constante | Verdadero, Falso, Sí, No, Nulo | Representa un valor que no cambia |

Introducir expresiones de validación.

Las reglas de validación se especifican en forma de expresiones.

Una expresión indica a Access cómo debe determinar si se permite o no introducir un cierto valor en un determinado campo. Las expresiones pueden ser simples, como \leq , o más complejas como "EUA" O "España" O "México".

En la siguiente relación se muestran ejemplos de expresiones de validación frecuentes.

| Valor de Red de Validación | Valor de Texto de Validación |
|----------------------------|--|
| $\neq 0$ | Debe introducir un valor distinto de cero. |
| $0 \geq 100$ | El valor debe ser 0 o superior a 100 |
| Como "K???" | El valor debe tener cuatro caracteres y debe empezar por la letra K. |

| | |
|--------------------------|---------------------------------------|
| <#1/1/2000# | Introduzca una fecha anterior a 2000 |
| >=#1/1/2000#Y<=1/1/2000# | La fecha debe ser de 2000 |
| >=Fecha() | La fecha no puede ser anterior a hoy. |

Crear índices.

Si se crea un índice para una relación, entonces Access puede encontrar más fácilmente los valores contenidos en campos que usted busca u ordena con frecuencia. La propiedad indexado se establece para crear un índice de un solo campo. En la siguiente relación se muestran los posibles valores de esta propiedad y sus significados.

| Valor de la propiedad indexado | Significado |
|--------------------------------|---|
| No | No crear un índice en este campo (o eliminar el índice existente) |
| Sí (con duplicados) | Crear un índice en este campo |
| Si (con duplicados) | Crear un único índice en este campo |

Si se crea un índice único (sin duplicados), Access no permitirá introducir en ese campo un valor que ya existía para otro registro.

CONSULTAS

Las consultas son un tipo de objeto de Access que puede utilizarse para distintas tareas. Entre las más habituales están las de examinar, modificar y analizar los datos de una o varias relaciones de una base de datos. Por otro lado, las consultas pueden utilizarse también a modo de relaciones, como origen de registros para crear formularios e informes. Así, las relaciones originales conservan su función de almacén de datos básico, mientras que las consultas permiten crear relaciones virtuales personalizadas para cada tipo de tarea.

El tipo de consulta más común es la consulta de selección. Una consulta de selección recupera datos de una o más relaciones usando los criterios de selección especificados (de forma parecida a los filtros) y los muestra en un orden determinado en una hoja de datos, que puede utilizarse incluso para actualizar los datos de las relaciones originales (aquí hay una diferencia sustancial con los filtros donde no es posible editar los datos de las relaciones de origen). También puede utilizar una consulta de selección para agrupar los registros y calcular sumas, promedios y otros tipos de resultados.

CONSULTA DE SELECCIÓN

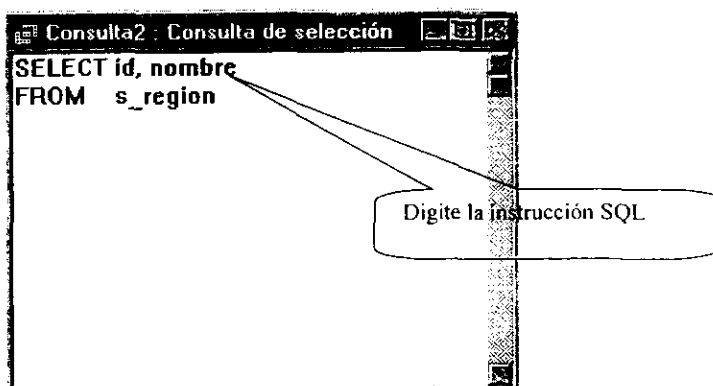
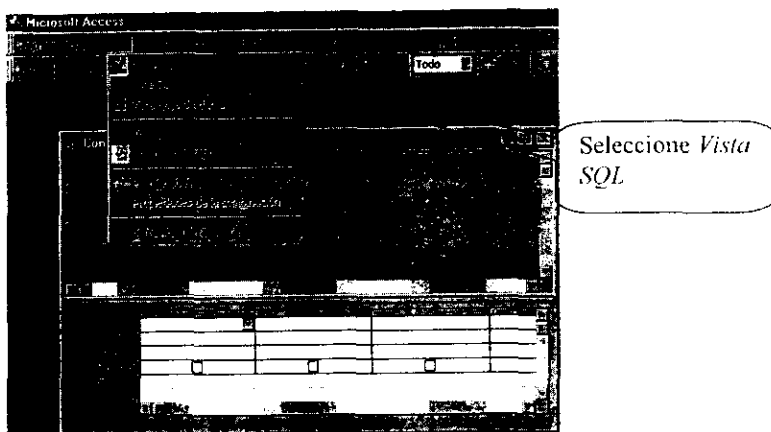
Microsoft Access permite crear consultas de forma automática, evitando de este modo que el usuario deba definirla desde el principio. Estos son los dos procedimientos básicos que proporciona Access para definir consultas de selección:

- Para crear fácilmente consultas para ejecutar de forma independiente o en las que desea basar formularios e informes, pruebe a utilizar uno de los asistentes para consultas.

Los asistentes para consultas realizan todo el trabajo básico automáticamente una vez obtenidas respuestas a una serie de preguntas.

Incluso aunque esté acostumbrado a crear consultas, tal vez desee utilizar un asistente para diseñar rápidamente la consulta. A continuación, puede cambiar a la vista Diseño para personalizarla.

- Para crear consultas sobre filtros creados usando Filtro por formulario, Filtro por selección o Filtro por entrada.



Los comandos siguientes se usan dentro de las consultas de selección del sistema, para instanciar estas consultas se utiliza la herramienta Querydef de Recordset dentro de Visual Basic.

Consulta de Selección ConAnexo

```
PARAMETERS xOrgId Value;  
SELECT *  
FROM OrgAnexo  
WHERE fiOrgId = xOrgId;
```

Consulta de Selección ConArbol

```
SELECT *  
FROM Arbol_Detalle;
```

Consulta de Selección ConCaracterísticas

```
PARAMETERS xOrgId Value;  
SELECT *  
FROM OrgCaracteristicas  
WHERE fiOrgId = xOrgId;
```

Consulta de Selección ConContacto

```
PARAMETERS xOrgId Value;  
SELECT *  
FROM OrgContacto  
WHERE fiOrgId = xOrgId;
```

Consulta de Selección ConModulo

```
SELECT fcModuloNombre AS Nombre, fiModuloId, fcModuloIcono AS Icono, fcModToolTip AS ToolTip
FROM ModuloAcceso
WHERE fcModOpcion = 'M'
ORDER BY fiModuloId, fiModHijo;
```

Consulta de Selección ConObjetivo

```
PARAMETERS xOrgId Value;
SELECT *
FROM OrgObjetivo
WHERE fiOrgId = xOrgId;
```

Consulta de Selección ConOpcArbol

```
PARAMETERS xmod Value;
SELECT fcNivelId
FROM Arbol_Detalle
WHERE fiArbolId = xmod;
```

Consulta de Selección ConOpcion

```
PARAMETERS xmod Value;
SELECT rtrim(fcModuloNombre) AS Nombre, fiModuloId, fcModuloIcono AS Icono, fcModToolTip AS
ToolTip
FROM ModuloAcceso
WHERE fiModHijo = xmod AND fcModOpcion = 'O'
ORDER BY fiModuloId, fiModHijo;
```

Consulta de Selección ConOperacion

```
PARAMETERS xmod Value;  
SELECT rtrim(fcModuloNombre) AS Nombre, fiModuloid, fcModuloIcono AS Icono, fcModToolTip AS  
ToolTip  
FROM ModuloAcceso  
WHERE fiModHijo = xmod AND fcModOpcion = 'P'  
ORDER BY fiModuloid, fiModHijo;
```

Consulta de Selección ConOperOpcion

```
PARAMETERS xoper Value;  
SELECT rtrim(fcModuloNombre) AS Nombre, fiModuloid, fcModuloIcono AS Icono, fcModToolTip AS  
ToolTip  
FROM ModuloAcceso  
WHERE fiModHijo = xoper AND fcModOpcion = 'O'  
ORDER BY fiModuloid, fiModHijo;
```

Consulta de Selección ConServicio

```
PARAMETERS xOrgId Value;  
SELECT *  
FROM OrgServicio  
WHERE fiOrgId = xOrgId;
```

Consulta de Selección ConSevAdicional

```
PARAMETERS xOrgId Value;  
SELECT *  
FROM OrgServAdicional  
WHERE fiOrgId = xOrgId;
```

Consulta de Selección ConSubtitulo

```
PARAMETERS Xtit Value, Xtema Value;  
SELECT *  
FROM SUBTITULO  
WHERE Id_titulo = XTit AND Id_tema = Xtema;
```

Consulta de Selección ConTodosOrg

```
SELECT *  
FROM Organismo_Generales  
ORDER BY fcOrgNombre;
```

Consulta de Selección ConTratamiento

```
PARAMETERS xOrgId Value;  
SELECT *  
FROM OrgTratamiento  
WHERE fiOrgId = xOrgId;
```

Consulta de Selección EjecutaExe

```
PARAMETERS id Value;  
SELECT fcModuloEjecuta AS Ejecuta  
FROM moduloacceso  
WHERE fiModuloId = Id;
```

Consulta de Selección InsArbol

```
PARAMETERS XHijo Value, XNivel Value, XNombre Value, XStatus Value, XTexto Text, XDib Value;  
INSERT INTO Arbol_Detalle ( fiInfHijoid, fcNivelId, fcInfDetNombre, fiInfDetStatus, fiInfDetTexto,  
fcArbolDib )
```

```
SELECT XHijo AS Expr1, XNivel AS Expr2, XNombre AS Expr3, XStatus AS Expr4, XTexto AS Expr5,  
XDib AS Expr6;
```

Consulta de Selección LlenaCatalogos

```
PARAMETERS xPadre Value;  
SELECT fiConcPadreId, fcConcDetNombre  
FROM Concepto  
WHERE fiConcHijoId = xPadre;
```

Consulta de Selección ObtenConc

```
PARAMETERS xvalor Value;  
SELECT fcConcDetNombre AS Nombre  
FROM Concepto  
WHERE fiConcPadreId = xvalor;
```

Consulta de Selección ObtenIdArbol

```
SELECT MAX(fiArbolId) AS Id  
FROM Arbol_Detalle;
```

Consulta de Selección ObtenOpcion

```
PARAMETERS xvalor Value;  
SELECT fcNivelId  
FROM Arbol_Detalle  
WHERE fiArbolId = xvalor;
```

Consulta de Selección ObtenRama

```
PARAMETERS xpadre Value;  
SELECT *  
FROM Arbol_Detalle  
WHERE fiArbolId = xpadre;
```

Consulta de Selección Paramemo

```
PARAMETERS xid Value;  
SELECT fiArbolId, fInfDetTexto  
FROM Arbol_Detalle  
WHERE fiArbolId = xid;
```

Existen en Access otras consultas que permiten realizar cálculos sobre grupos de registros. Los tipos de cálculos permitidos son:

Suma.- El total de los valores de un campo.

Promedio.- El promedio de los valores de un campo.

Min.- El menor valor de un campo.

Máx.- El mayor valor de un campo.

Cuenta.- El número de valores de un campo (sin contar los valores nulos).

DesvEst.- La desviación estándar de los valores de un campo.

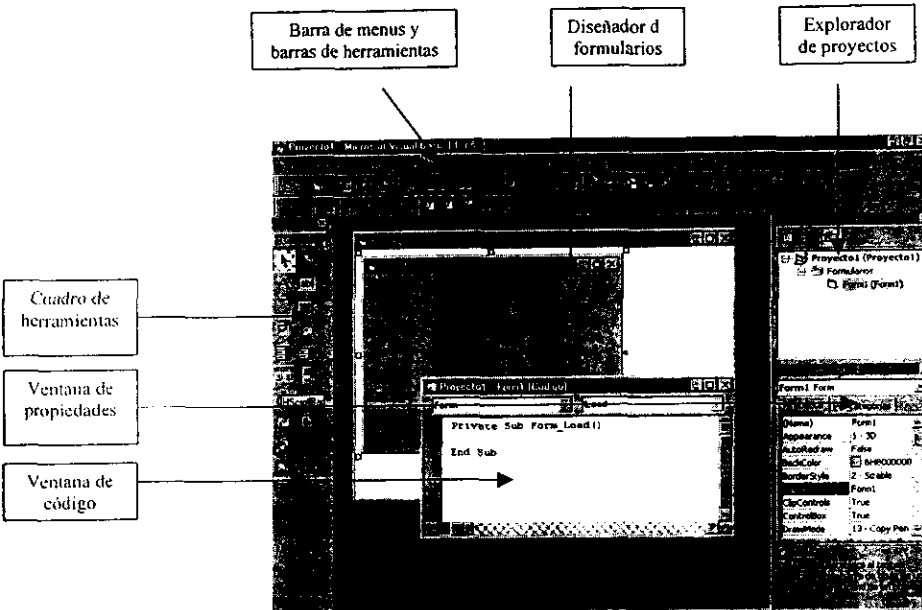
Var.- La varianza de los valores de un campo.

Primero.- El valor del campo correspondiente al primer registro de una relación o de una consulta.

Último.- El valor del campo correspondiente al último registro de una relación o de una consulta.

FORMULARIOS

El primer objeto Visual Basic con que nos encontramos es el FORMULARIO. De hecho, cada vez que iniciamos Visual Basic(VB) nos presenta en pantalla un nuevo formulario, que tiene por defecto el nombre de Form1.



El **Formulario** es un objeto, que sirve de **soporte** de otros objetos. El nombre de **FORMULARIO** lo toma precisamente porque, al igual que un formulario de papel contiene textos escritos, lugares donde podemos escribir, figuras, cajas donde podemos elegir una u otra opción, etc., en este cuadro gráfico que llamamos formulario, podemos introducir textos escritos, lugares donde podemos escribir, figuras, cajas donde podemos elegir una u otra opción. En realidad un formulario es lo que normalmente conocemos por **VENTANA**.

El nombre de formulario le viene muy bien cuando ese formulario es una ventana donde vamos a introducir datos alfanuméricos. Cuando en vez de introducir datos, lo que tenemos es, por ejemplo, una ventana donde se reproducen exactamente los controles de operación de una máquina, parece en principio que sería más correcto el nombre de "ventana". De alguna forma lo hay que llamar, y esa forma es FORMULARIO. Posiblemente a lo largo de estos apuntes nos referiremos al formulario con el nombre de ventana, o incluso, pantalla. Procuraremos usar el nombre apropiado: FORMULARIO.

BARRAS DEL ENTORNO DE VISUAL BASIC

Barra de Menús

Presenta los comandos que se usan para trabajar con Visual Basic. Además de los menús estándar **Archivo, Edición, Ver, Ventana y Ayuda**, contiene otros menús para tener acceso a funciones específicas de programación, como **Proyecto, Formato o Depuración**.

Barra de Herramientas

Permite un acceso directo (solo un clic) a muchas de las operaciones más frecuentes utilizadas durante el desarrollo de aplicaciones.

Cuadro de Herramientas

Contiene todos los objetos y controles que se pueden añadir a los formularios para crear aplicaciones.

Diseñador de Formularios

Funciona como una ventana en la que se puede personalizar el diseño de la interfaz de usuario (ventana) de una aplicación.

Explorador de Proyectos

Lista de los archivos (formularios, módulos, etc.) del proyecto actual. Un **Proyecto** es una colección de archivos que utiliza para construir una aplicación.

Ventana de Propiedades

Lista los valores de las propiedades del formulario o control seleccionado que pueden ser modificados durante el diseño del formulario o control.

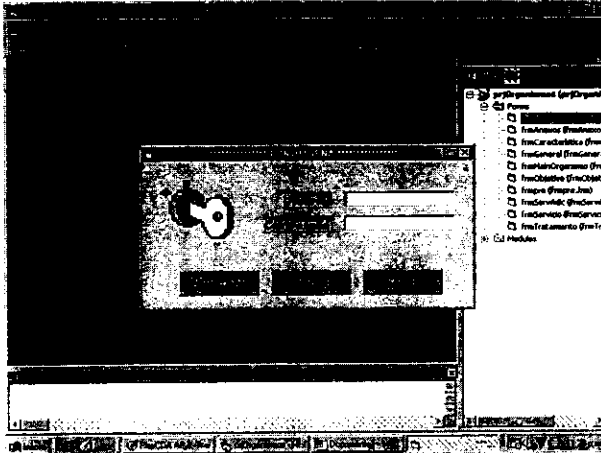
Ventana de Código

Funciona como un editor para escribir el código (sentencias) de la aplicación.

Los formularios que a continuación se ejemplificarán son aquellos que se realizaron en la parte de ORGANISMOS del sistema.

A continuación se muestra el formulario **frmacceso** y su código:

La función del formulario siguiente es la validar una clave de acceso para acceder a los demás formularios que componen el sistema.



El código del formulario **frmacceso** es el siguiente:

```
Private Sub cmdIngresar_Click()  
    If Len(Trim(txtUsuario)) = 0 Then  
        txtUsuario.SetFocus  
    ElseIf Len(Trim(txtContraseña)) = 0 Then  
        txtContraseña.SetFocus  
    ElseIf txtContraseña = "vicgtz" Then  
        frmpre.Show 1  
        Unload Me  
    Else  
        MsgBox "La clave ingresada no es válida", vbCritical  
        txtContraseña.SetStart = 0  
        txtContraseña.SetLength = Len(Trim(txtContraseña))  
        txtContraseña.SetFocus  
    End If  
End Sub  
  
Private Sub cmdLimpiar_Click()  
    txtUsuario.Text = ""  
  
    txtContraseña.Text = ""  
    txtUsuario.SetFocus
```

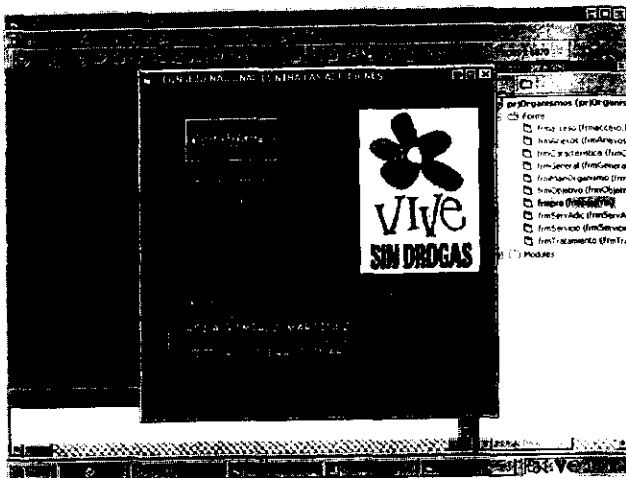
End Sub

```
Private Sub cmdsalir_Click()  
Unload Me  
End Sub
```

A continuación se muestra el formulario **frmpre** y su código:

Este formulario es la presentación del sistema en donde se muestra principalmente el logotipo de Vive sin Drogas.

El formulario anterior es una presentación en la que figura el logotipo de **vive sin drogas** una de las empresas que se encuentran en el convenio junto con la Secretaría de Salud. Con este formulario se puede entrar a la pantalla principal del Organismos.



El código del formulario **frmpre** es el siguiente:

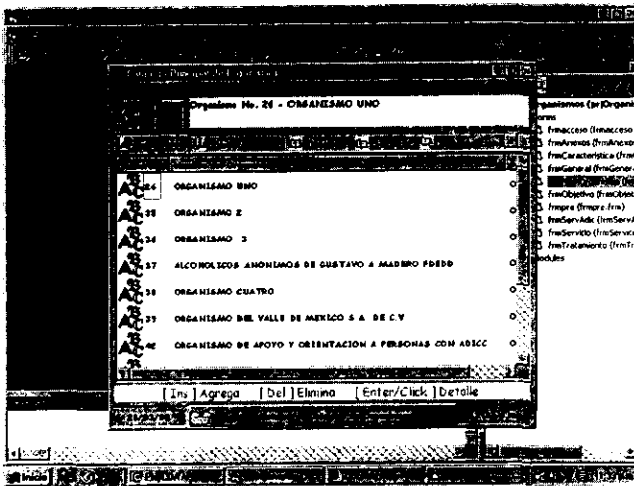
```
Private Sub Command1_Click()  
frmMainOrganismo.Show vbModal  
End Sub
```

```
Private Sub Command2_Click()  
Unload Me  
End Sub
```

A continuación se muestra el formulario **frmMainOrganismo** y su código:

En este formulario se observa los organismos que se encuentran disponibles dentro del list box pudiendo acceder al deseado.

Además este formulario es uno de los más importantes, ya que, por medio de él accedamos a las pantallas de captura de información, es decir, los organismos.



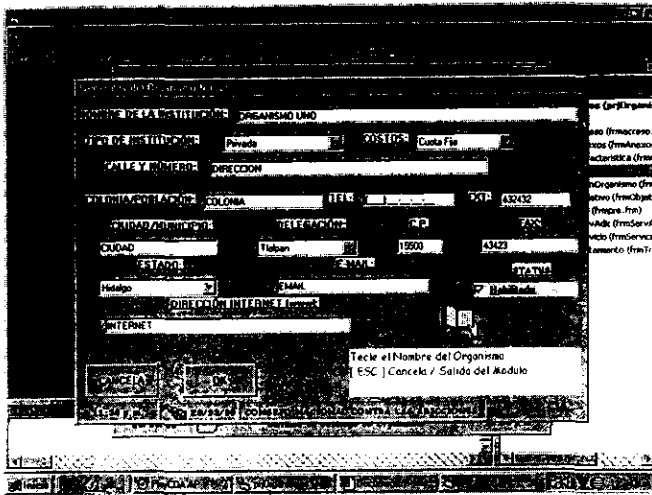
```
Private Sub TabOpcion_Click()  
If liOrganismo = 0 Then  
MsgBox "<< ERROR >> NO ha Seleccionado Ningún Organismo ...." & Chr(13) & _  
"Por Favor Selecciona Cualquiera para ver Detalle ..", vbInformation, "Selección de Organismo"  
Exit Sub  
End If  
Select Case TabOpcion.SelectedItem.Index  
Case 1 ' Generales  
liOpcion = 1  
DetalleOrg  
Case 2 ' Caracteristicas  
StatusBar1.Panels(2).Text = "Opción de << Caracteristicas >> .... !!"  
frmCaracteristica.Show 1  
Case 3 ' Objetivo  
StatusBar1.Panels(2).Text = "Opción de << Objetivo >> .... !!"  
frmObjetivo.Show 1
```

```
Case 4 ' Servicio
  StatusBar1.Panels(2).Text = "Opción de << Servicios >> .... !!"
  frmServicio.Show 1
Case 5 ' Servicio Adicional
  StatusBar1.Panels(2).Text = "Opción de << Servicios Adicionales >> .... !!"
  frmServAdic.Show 1
Case 6 ' Tratamiento
  StatusBar1.Panels(2).Text = "Opción de << Tratamiento >> .... !!"
  frmTratamiento.Show 1
Case 7 ' Anexos
  StatusBar1.Panels(2).Text = "Opción de << Anexos >> .... !!"
  frmAnexos.Show 1
End Select
End Sub
```

El objetivo de este código es el de mostrar las pantallas de captura por medio de una barra tabStrip , las cuales son mostradas por la instrucción **nombre_del_formulario.Show 1**.

A continuación se muestra el formulario **frmGeneral** y su código:

Dentro de este formulario se llenan los datos generales del organismo para tener una mejor ubicación en cuanto a su localización física.



```
Sub RegistraMov()  
Dim rsOrg As Recordset  
Dim lsComando As String  
Dim liRes As Integer  
Dim itmX As ListItem
```

```
lsComando = "SELECT * FROM OrgGenerales"  
Set rsOrg = db.OpenRecordset(lsComando, dbOpenDynaset)  
rsOrg.AddNew  
rsOrg(1) = txtnombre.Text  
rsOrg(2) = cboInstitucion.ItemData(cboInstitucion.ListIndex)  
rsOrg(3) = cboCostos.ItemData(cboCostos.ListIndex)  
rsOrg(4) = 0
```

```
rsOrg(5) = 0  
rsOrg(6) = 0  
rsOrg(7) = txtdireccion.Text  
rsOrg(8) = txtcolonia.Text  
rsOrg(9) = txtciudad.Text  
rsOrg(10) = txtTelefono.Text
```

```
rsOrg(11) = txttext.Text
rsOrg(12) = txtFax.Text
rsOrg(13) = cboDelegacion.ItemData(cboDelegacion.ListIndex)
rsOrg(14) = txtcp.Text
rsOrg(15) = cboEstado.ItemData(cboEstado.ListIndex)
rsOrg(16) = txtemail.Text
rsOrg(17) = txtwww.Text
rsOrg(19) = chkStatus.Value
rsOrg.Update
rsOrg.Requery
rsOrg.MoveLast
liOrganismo = rsOrg(0)
rsOrg.Close
Set rsOrg = Nothing
```

```
Set itmx = frmMainOrganismo.LViewOrganismo.ListItems.Add()
itmx.Text = LTrim(Str(liOrganismo))
itmx.SmallIcon = 1
itmx.SubItems(1) = txtnombre.Text
itmx.SubItems(2) = IIf(chkStatus.Value = 1, "Habilitado", "DesHabilitado")
Unload Me
End Sub
```

```
Sub ModificaMov()
Dim rsOrg As Recordset
Dim IsComando As String
Dim liRes As Integer
Dim itmx As ListItem
```

```
IsComando = "SELECT * FROM OrgGenerales WHERE fiOrgId = " & LTrim(Str(liOrganismo))
Set rsOrg = db.OpenRecordset(IsComando, dbOpenDynaset)
rsOrg.Edit
rsOrg(1) = txtnombre.Text
rsOrg(2) = cboInstitucion.ItemData(cboInstitucion.ListIndex)
rsOrg(3) = cboCostos.ItemData(cboCostos.ListIndex)
rsOrg(4) = 0
rsOrg(5) = 0
rsOrg(6) = 0
rsOrg(7) = txtdireccion.Text
rsOrg(8) = txtcolonia.Text
rsOrg(9) = txtciudad.Text
rsOrg(10) = txtTelefono.Text
rsOrg(11) = txttext.Text
rsOrg(12) = txtFax.Text
rsOrg(13) = cboDelegacion.ItemData(cboDelegacion.ListIndex)
```

```
rsOrg(14) = txtcp.Text
rsOrg(15) = cboEstado.ItemData(cboEstado.ListIndex)
rsOrg(16) = txtemail.Text
rsOrg(17) = txtwww.Text
rsOrg(19) = chkStatus.Value
rsOrg.Update
rsOrg.Close
```

Set rsOrg = Nothing

frmMainOrganismo.LViewOrganismo.SelectedItem.SubItems(1) = txtnombre.Text
frmMainOrganismo.LViewOrganismo.SelectedItem.SubItems(2) = IIf(chkStatus.Value = 1, "OK", "No")

Unload Me

End Sub

Private Sub cmdOpcion_Click(Index As Integer)

Select Case Index

Case 0 ' Cancela

Unload Me

Case 1 ' Registra

If liOpcion = 0 Then

RegistraMov ' Agrega Nuevo Organismo

Else

ModificaMov ' Modifica Organismo

End If

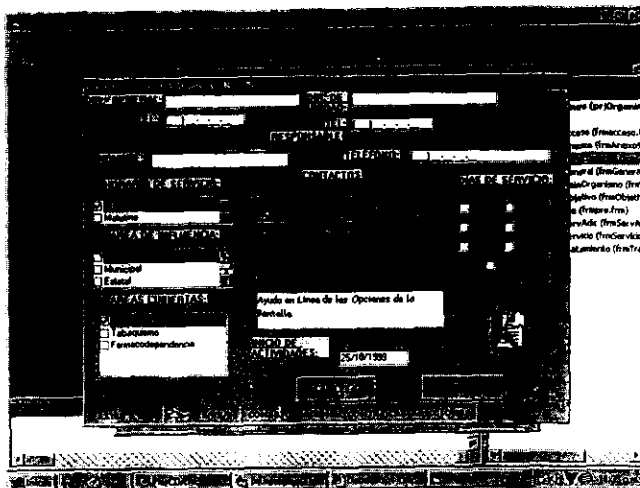
End Select

End Sub

Dentro de los command buttons se llama a las subrutinas RegistraMov y ModificaMov.

A continuación se muestra el formulario **frmCaracteristica** y su código:

Dentro de este formulario se llena la disponibilidad del organismo, los responsables del mismo y áreas que cubren.



Sub GrabaCaracteristicas(PiOrganismoid As Long)

Dim IsComando As String
Dim IsDirGral As String
Dim IsDirGralTel As String
Dim IsDirUnidad As String
Dim IsDirUnidadTel As String
Dim IsHorario As String
Dim IsAreaInfluencia As String
Dim IsAreaCubierta As String
Dim IsFecha As String
Dim IsDias As String
Dim IsPasoFecha As String
Dim rsCarac As Recordset

IsDirGral = Trim(txtDg.Text)
IsDirGralTel = Trim(txtTelDg.Text)
IsDirUnidad = Trim(txtDu.Text)
IsDirUnidadTel = Trim(txtTelDu.Text)
IsHorario = ObtieneCadenaPorLista(IsHorario)
IsAreaInfluencia = ObtieneCadenaPorLista(IsAreaInflu)
IsAreaCubierta = ObtieneCadenaPorLista(IsAreaCubie)

```

If txtFeclni.Text = " / / " Then
    IsPasoFecha = ""
Else
    IsPasoFecha = txtFeclni.Text
End If

```

```

IsFecha = IIf(Trim(IsPasoFecha) = "", Format(Trim(CStr(Date)), "dd/mm/yyyy"),
Format(Trim(IsPasoFecha), "dd/mm/yyyy"))

```

```

If IsFecha <> "" Then
    IsFecha = "#" & IsFecha & "#"
Else
    IsFecha = "null"
End If

```

```

IsDias = IIf(chkdias(0).Value = vbChecked, "D", "") 'Domingo
IsDias = IsDias & IIf(chkdias(1).Value = vbChecked, "L", "") 'Lunes
IsDias = IsDias & IIf(chkdias(2).Value = vbChecked, "M", "") 'Martes
IsDias = IsDias & IIf(chkdias(3).Value = vbChecked, "W", "") 'Miercoles
IsDias = IsDias & IIf(chkdias(4).Value = vbChecked, "J", "") 'Jueves
IsDias = IsDias & IIf(chkdias(5).Value = vbChecked, "V", "") 'Viernes
IsDias = IsDias & IIf(chkdias(6).Value = vbChecked, "S", "") 'Sabado

```

```

IsComando = "SELECT * FROM OrgCaracteristicas"
Set rsCarac = db.OpenRecordset(IsComando, dbOpenDynaset)
rsCarac.AddNew
rsCarac(0) = PiOrganismold
rsCarac(1) = txtDg.Text
rsCarac(2) = txtTelDg.Text
rsCarac(3) = txtDu.Text
rsCarac(4) = txtTelDu.Text
rsCarac(5) = IsHorario
rsCarac(6) = IsArealInfluencia
rsCarac(7) = IsAreaCubierta
rsCarac(8) = IsPasoFecha
rsCarac(9) = IsDias
rsCarac.Update
rsCarac.Close
Set rsCarac = Nothing

```

End Sub

Sub GrabaContactos(PiOrganismold As Long)

```

Dim IsComando As String
Dim IsNombre As String
Dim IsTelefono As String
Dim i%
Dim rsCarac As Recordset

```

```

If GrdCon.Rows > 1 Then
    IsComando = "DELETE FROM OrgContacto WHERE fiOrgId = " & LTrim(Str(PiOrganismold))

```

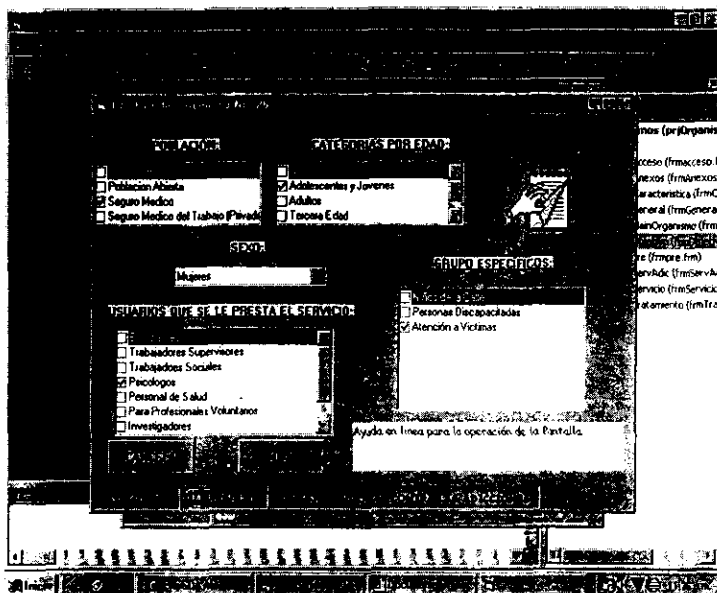
```
db.Execute IsComando
DBEngine.Idle dbRefreshCache
DoEvents
End If
```

```
If GrdCon.Rows > 1 Then
  For i% = 1 To (GrdCon.Rows - 1)
    GrdCon.Row = i%
    GrdCon.Col = 0
    IsNombre = GrdCon.Text
    GrdCon.Col = 1
    IsTelefono = GrdCon.Text
    IsComando = "SELECT * FROM OrgContacto"
```

```
Set rsCarac = db.OpenRecordset(IsComando, dbOpenDynaset)
rsCarac.AddNew
rsCarac(0) = PiOrganismoid
rsCarac(1) = IsNombre
rsCarac(2) = IsTelefono
rsCarac.Update
rsCarac.Close
Next i%
Set rsCarac = Nothing
End If
End Sub
```

A continuación se muestra el formulario **frmObjetivo** y su código:

Dentro de este organismo se llena la atención a las personas de acuerdo a su edad, tipo de población ó grupos específicos.



Sub GrabaObjetivo(PiOrganismoid As Long)

Dim IsComando As String

Dim IsTipoCob As String

Dim IsCatEdad As String

Dim IsUsuServicio As String

Dim IsGrupoEspec As String

Dim liGenero As Integer

Dim rsObj As Recordset

IsTipoCob = ObtieneCadenaPorLista(IsfCobertura)

IsCatEdad = ObtieneCadenaPorLista(IsfEdad)

IsUsuServicio = ObtieneCadenaPorLista(IsfUsuario)

IsGrupoEspec = ObtieneCadenaPorLista(IsfGrupo)

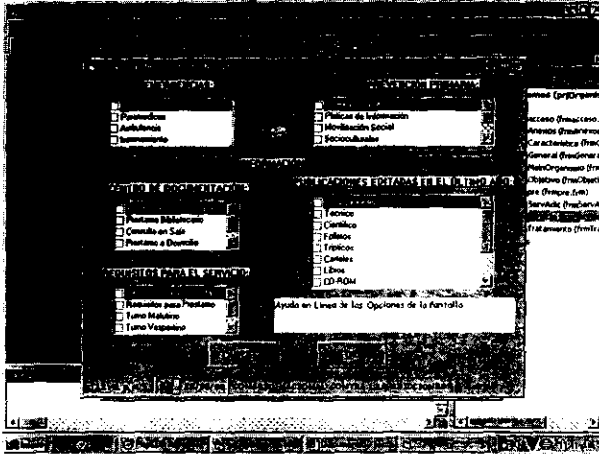
liGenero = cboGenero.ItemData(cboGenero.ListIndex)

```
If liOpcion = 0 Then
  IsComando = "SELECT * FROM OrgObjetivo"
  Set rsObj = db.OpenRecordset(IsComando, dbOpenDynaset)
  rsObj.AddNew
  rsObj(0) = PiOrganismold
Else
  IsComando = "SELECT * FROM OrgObjetivo WHERE fiOrgId = " & LTrim(Str(PiOrganismold))
  Set rsObj = db.OpenRecordset(IsComando, dbOpenDynaset)
  rsObj.Edit
End If

rsObj(1) = IsTipoCob
rsObj(2) = IsCatEdad
rsObj(3) = IsUsuServicio
rsObj(4) = IsGrupoEspec
rsObj(5) = liGenero
rsObj.Update
rsObj.Close
Set rsObj = Nothing
End Sub
```

A continuación se muestra el formulario **frmServicio** y su código:

En este organismo se registran servicios de emergencias, centros donde se realizan búsquedas bibliográficas, así como, orientación para la prevención y publicaciones.



```
Sub ConsultaServicios(PiOrganismoid As Long)
Dim rsConsulta As Recordset
Dim qrConsulta As QueryDef
```

```
Dim IsEmergencias As String
Dim IsPrimaria As String
Dim IsCentro As String
Dim IsRequisitos As String
Dim IsPublicaciones As String
```

```
Set qrConsulta = db.QueryDefs("ConServicio")
qrConsulta.Parameters("xOrgId") = LTrim(Str(PiOrganismoid))
Set rsConsulta = qrConsulta.OpenRecordset(dbOpenSnapshot)
```

```
If Not rsConsulta.EOF Then
    rsConsulta.MoveFirst
```

```

    liOpcion = 1
Else
    rsConsulta.Close
    qrConsulta.Close

    liOpcion = 0
    Exit Sub
End If

IsEmergencias = rsConsulta("fcSerEmergencias")
IsPrimaria = rsConsulta("fcSerPrevPrimaria")
IsCentro = rsConsulta("fcSerCentroDocu")
IsRequisitos = rsConsulta("fcSerReqServ")
IsPublicaciones = rsConsulta("fcSerPublicaciones")

'Emergencias
ValorLista IstEmer, Trim(IsEmergencias)
'Primaria
ValorLista IstPrevPrim, Trim(IsPrimaria)
'Centro
ValorLista IstCenDoc, Trim(IsCentro)
'Requisitos
ValorLista IstReqSer, Trim(IsRequisitos)
'Publicaciones
ValorLista IstPublicaciones, Trim(IsPublicaciones)

qrConsulta.Close
rsConsulta.Close

Set qrConsulta = Nothing
Set rsConsulta = Nothing

End Sub
Sub GrabaServicios(PiOrganismold As Long)
Dim IsComando As String
Dim IsEmergencias As String
Dim IsPrimaria As String
Dim IsCentro As String

Dim IsRequisitos As String
Dim IsPublicaciones As String
Dim rsObj As Recordset

IsEmergencias = ObtieneCadenaPorLista(IstEmer)
IsPrimaria = ObtieneCadenaPorLista(IstPrevPrim)
IsCentro = ObtieneCadenaPorLista(IstCenDoc)
IsRequisitos = ObtieneCadenaPorLista(IstReqSer)
IsPublicaciones = ObtieneCadenaPorLista(IstPublicaciones)

If liOpcion = 0 Then
    IsComando = "SELECT * FROM OrgServicio"
    Set rsObj = db.OpenRecordset(IsComando, dbOpenDynaset)
    rsObj.AddNew
    rsObj(0) = PiOrganismold
Else
    IsComando = "SELECT * FROM OrgServicio WHERE fiOrgId = " & LTrim(Str(PiOrganismold))

```

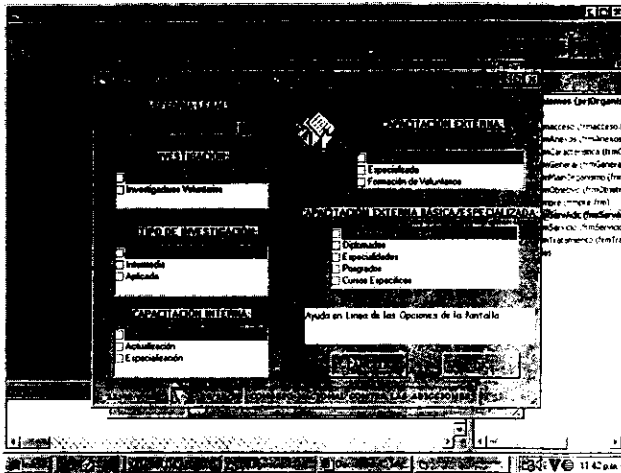
```
Set rsObj = db.OpenRecordset(IsComando, dbOpenDynaset)
rsObj.Edit
End If
```

```
rsObj(1) = IsEmergencias
rsObj(2) = IsPrimaria
rsObj(3) = IsCentro
rsObj(4) = IsRequisitos
rsObj(5) = IsPublicaciones
rsObj.Update
rsObj.Close
Set rsObj = Nothing
```

```
End Sub
```

A continuación se muestra el formulario **frmServAdic** y su código:

Dentro de este formulario se llenan los campos de asesorías, capacitaciones internas, externas y especializadas.



```
Sub GrabaAdicionales(PiOrganismold As Long)
Dim IsComando As String
Dim liAsesoria As Integer
Dim IsInvestigacion As String
Dim IsVoluntarios As String
Dim IsInterna As String
```

```
Dim IsExterna As String
Dim IsEspecial As String
Dim rsObj As Recordset
```

```
liAsesoria = cboLegal.ItemData(cboLegal.ListIndex)
IsInvestigacion = ObtieneCadenaPorLista(IsInvestigacion)
IsVoluntarios = ObtieneCadenaPorLista(IsInvVoluntarios)
IsInterna = ObtieneCadenaPorLista(IsCapInterna)
IsExterna = ObtieneCadenaPorLista(IsCapExterna)
IsEspecial = ObtieneCadenaPorLista(IsCapEspecial)
```

```
If liOpcion = 0 Then
IsComando = "SELECT * FROM OrgServAdicional"
Set rsObj = db.OpenRecordset(IsComando, dbOpenDynaset)
rsObj.AddNew
rsObj(0) = PiOrganismold
```

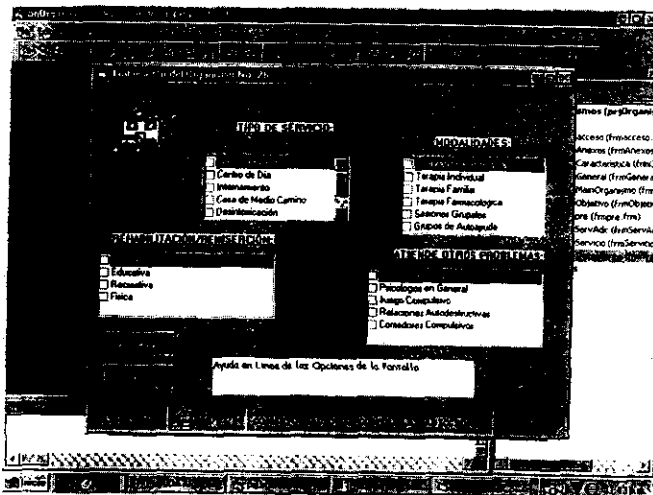
Else

```
IsComando = "SELECT * FROM OrgServAdicional WHERE fiOrgId = " & LTrim(Str(PiOrganismold))  
Set rsObj = db.OpenRecordset(IsComando, dbOpenDynaset)  
rsObj.Edit  
End If
```

```
rsObj(1) = liAsesoría  
rsObj(2) = IsInvestigación  
rsObj(3) = IsVoluntarios  
rsObj(4) = IsInterna  
rsObj(5) = IsExterna  
rsObj(6) = IsEspecial  
rsObj.Update  
rsObj.Close  
Set rsObj = Nothing  
End Sub
```

A continuación se muestra el formulario **frmTratamiento** y su código:

Dentro de este formulario se llenan los tipos de rehabilitación, las modalidades en terapias, los tipos de servicios que ofrecen los organismos, así como, otros problemas de adicciones.



Sub GrabaTratamiento(PiOrganismold As Long)

Dim IsComando As String

Dim IsServicio As String

Dim IsRehabilita As String

Dim IsModalidades As String

Dim IsAtiende As String

Dim rsObj As Recordset

IsServicio = ObtieneCadenaPorLista(IsServicio)

IsRehabilita = ObtieneCadenaPorLista(IsReha)

IsModalidades = ObtieneCadenaPorLista(IsModalidad)

IsAtiende = ObtieneCadenaPorLista(IsOtrosProb)

If liOpcion = 0 Then

*IsComando = "SELECT * FROM OrgTratamiento"*

Set rsObj = db.OpenRecordset(IsComando, dbOpenDynaset)

rsObj.AddNew

rsObj(0) = PiOrganismold

Else

*IsComando = "SELECT * FROM OrgTratamiento WHERE fiOrgId = " & LTrim(Str(PiOrganismold))*

Set rsObj = db.OpenRecordset(IsComando, dbOpenDynaset)

rsObj.Edit

End If

rsObj(1) = IsServicio

rsObj(2) = IsRehabilita

rsObj(3) = IsModalidades

rsObj(4) = IsAtiende

rsObj.Update

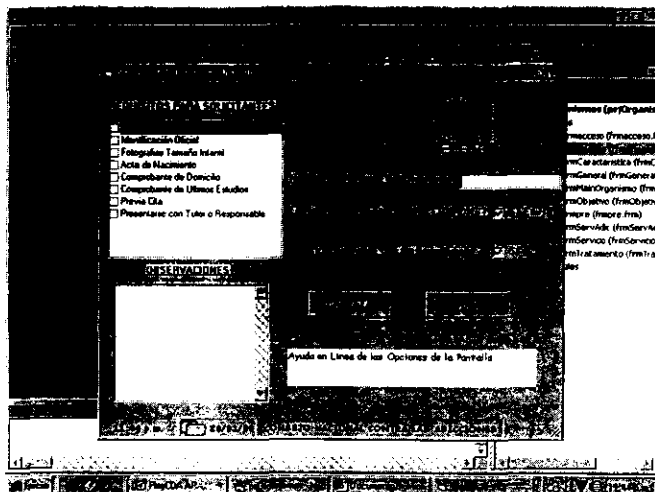
rsObj.Close

Set rsObj = Nothing

End Sub

A continuación se muestra el formulario **frmAnexos** y su código:

En este organismo se muestran los requisitos para el organismo a llenar, así como el contacto con el CONADIC para dar un seguimiento especial y las observaciones que haya que llenar.



```
Public Sub GrabaAnexos(PiOrganismold As Long)
Dim IsComando As String
Dim IsPersonal As String
Dim IsRequisito As String
Dim liEnvia As Integer
Dim IsObservaciones As String
Dim liSeguimiento As Integer
Dim rsObj As Recordset
```

```
IsPersonal = IIf(Trim(txtpersonal1.Text) = "", "0", Trim(txtpersonal.Text))
IsRequisito = ObtieneCadenaPorLista(1stRequisitos)
liEnvia = IIf(chkenvia.Value = 1, 1, 0)
IsObservaciones = Trim(txtobservacion.Text)
liSeguimiento = IIf(chkaut.Value = 1, 1, 0)
```

```
If liOpcion = 0 Then
IsComando = "SELECT * FROM OrgAnexo"
Set rsObj = db.OpenRecordset(IsComando, dbOpenDynaset)
rsObj.AddNew
rsObj(0) = PiOrganismold
Else
IsComando = "SELECT * FROM OrgAnexo WHERE fiOrgId = " & LTrim(Str(PiOrganismold))
Set rsObj = db.OpenRecordset(IsComando, dbOpenDynaset)
```

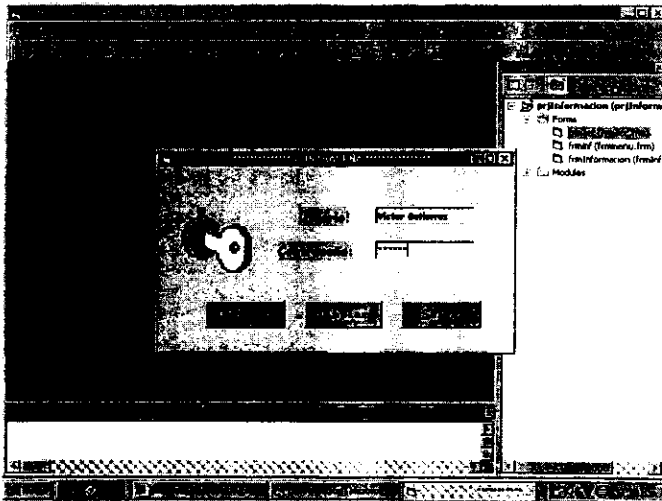
```
rsObj.Edit
End If

rsObj(1) = IsPersonal
rsObj(2) = IsRequisito
rsObj(3) = liEnvia
rsObj(4) = IsObservaciones
rsObj(5) = liSeguimiento
rsObj.Update
rsObj.Close
Set rsObj = Nothing

End Sub
```

La segunda parte del sistema es llamada **ÁRBOL DE INFORMACIÓN**, en el cual se guarda información de los tipos de drogas , sus características y sus consecuencias.

A continuación se muestra el formulario de **frmacc** y su código. Este formulario tiene la función de la seguridad a la entrada al sistema.



A continuación se muestra el código del formulario anterior:

```
Private Sub cmdIngresar_Click()
    If Len(Trim(txtUsuario)) = 0 Then
        txtUsuario.SetFocus
    ElseIf Len(Trim(txtContraseña)) = 0 Then
        txtContraseña.SetFocus

    ElseIf txtContraseña = "vicgtz" Then
        frmInf.Show vbModal
        Unload Me
    Else
        MsgBox "La clave ingresada no es válida", vbCritical
        txtContraseña.SelStart = 0
        txtContraseña.SelLength = Len(Trim(txtContraseña))
        txtContraseña.SetFocus
    End If
End Sub

Private Sub txtUsuario_keypress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        SendKeys "{tab}"
        KeyAscii = 0
    End If
End Sub

Private Sub txtContraseña_keypress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        SendKeys "{tab}"
        KeyAscii = 0
    End If
End Sub

Private Sub cmdLimpiar_Click()
    txtUsuario.Text = ""
    txtContraseña.Text = ""
    txtUsuario.SetFocus
End Sub

Private Sub cmdSalir_Click()
    Unload Me
End Sub
```

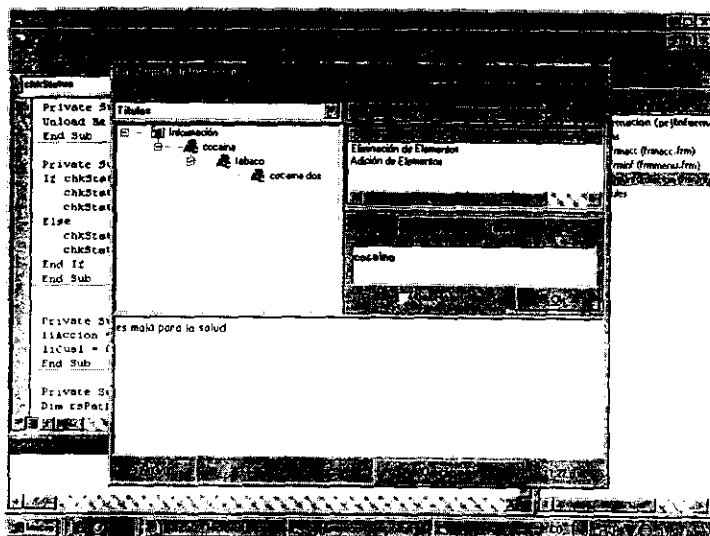
A continuación se muestra el formulario de `frmInf` y su código. Este formulario tiene la función de visualizar una pantalla de presentación.



```
Private Sub Command1_Click()  
frmInformacion.Show vbModal  
End Sub
```

```
Private Sub Command2_Click()  
Unload Me  
End Sub
```

A continuación se muestra el formulario **frmInformación** y su código, el cual tiene la función de guardar información de los diferentes tipos de drogas, características así como sus consecuencias en su consumo.



En esta parte del código se obtiene el ID del documento por medio de lenguaje SQL para realizar la instanciación.

```

Select Case liElem
Case 2 ' Tema
    liPaso = 1
    IsComando = "SELECT Folio, Ruta FROM Folio WHERE Id_Folio = " & LTrim(Str(liPaso))
    IsDoctos = "T"
Case 3 ' Titulo
    liPaso = 2
    IsComando = "SELECT Folio, Ruta FROM Folio WHERE Id_Folio = " & LTrim(Str(liPaso))
    IsDoctos = "U"
Case 4 ' Subtitulo
    liPaso = 3
    IsComando = "SELECT Folio, Ruta FROM Folio WHERE Id_Folio = " & LTrim(Str(liPaso))

```

```

IsDoctos = "V"
Case 5 ' Detalle

    liPaso = 4
    IsComando = "SELECT Folio, Ruta FROM Folio WHERE Id_Folio = " & LTrim(Str(liPaso))
    IsDoctos = "W"
End Select

```

En el siguiente código se obtiene el recordset de la relación en común y se realiza por medio del lenguaje SQL, es decir, se hace la instanciación a cada una de las relaciones que se utilizan en el Árbol de Información.

```

Select Case liElem
Case 2 ' Tema
    IsComando = "SELECT * FROM Tema"
Case 3 ' Titulo
    IsComando = "SELECT * FROM Titulo"
Case 4 ' Subtitulo
    IsComando = "SELECT * FROM SubTitulo"
Case 5 ' Detalle
    IsComando = "SELECT * FROM Detalle"
End Select

```

En esta parte del código se realiza la extracción y actualización de las relaciones tema, Titulo, Subtitulo y Detalle.

```
Set rsActualiza = Db.OpenRecordset(IsComando, dbOpenDynaset)
```

```
rsActualiza.AddNew
```

```
Select Case liElem
```

```
Case 2 ' Tema
```

```
rsActualiza(1) = txtNombre.Text
rsActualiza(2) = chkStatus.Value
rsActualiza(3) = IsDoctos
```

```
Case 3 ' Titulo
```

```
rsActualiza(1) = liTema
rsActualiza(2) = txtNombre.Text
rsActualiza(3) = chkStatus.Value
rsActualiza(4) = IsDoctos
```

```
Case 4 ' Subtitulo
```

```
rsActualiza(1) = liTitulo
rsActualiza(2) = liTema
rsActualiza(3) = txtNombre.Text
rsActualiza(4) = chkStatus.Value
rsActualiza(5) = IsDoctos
```

```
Case 5 ' Detalle
```

```
rsActualiza(1) = liSubTitulo
rsActualiza(2) = liTitulo
rsActualiza(3) = liTema
rsActualiza(4) = txtNombre.Text
rsActualiza(5) = chkStatus.Value
rsActualiza(6) = IsDoctos
```

End Select

```
txtTexto.SaveFile IsRuta & IsDoctos
rsActualiza.Update
rsActualiza.Close
txtTexto.Text = ""
txtNombre.Text = ""
chkStatus.Value = 0
CargaElementos
liCual = 0
End Sub
```

El código siguiente se agregan los elementos a las relaciones y de esta manera se pueden ejecutar consultas y devolver los datos más rápidamente con el Snapshot.

```
Sub AgregaElemento()
Dim rsActualiza As Recordset
Dim IsComando As String
Dim IsDoctos As String
Dim liDoctos As Long
Dim liPaso As Long
Dim Nod As Node
Dim IsLLavePad As String
Dim IsLLave As String
Dim IsPaso As String
```

```
Dim IsPaso1 As String
Dim IsPaso2 As String
Dim IsPaso3 As String
Dim liPaso1 As Integer
Dim liPaso2 As Integer
Dim liPaso3 As Integer
Dim IsRuta As String
```

' Obtiene el Id del Documento

```
Select Case liElem
Case 2 ' Tema
liPaso = 1
IsComando = "SELECT Folio, Ruta FROM Folio WHERE Id_Folio = " & LTrim(Str(liPaso))
IsDoctos = "T"
Case 3 ' Titulo
liPaso = 2
IsComando = "SELECT Folio, Ruta FROM Folio WHERE Id_Folio = " & LTrim(Str(liPaso))
IsDoctos = "U"
Case 4 ' Subtitulo
liPaso = 3
IsComando = "SELECT Folio, Ruta FROM Folio WHERE Id_Folio = " & LTrim(Str(liPaso))
```



```

IsDoctos = "V"
Case 5 ' Detalle
  liPaso = 4
  IsComando = "SELECT Folio, Ruta FROM Folio WHERE Id_Folio = " & LTrim(Str(liPaso))
  IsDoctos = "W"
End Select

Set rsActualiza = Db.OpenRecordset(IsComando, dbOpenSnapshot)
liDoctos = rsActualiza(0) + 1
IsRuta = Trim(rsActualiza(1))
rsActualiza.Close

IsComando = "UPDATE Folio SET Folio = " & LTrim(Str(liDoctos)) & " WHERE Id_Folio = " &
LTrim(Str(liPaso))
Db.Execute (IsComando)
DBEngine.Idle dbRefreshCache

IsDoctos = IsDoctos & String(7 - Len(LTrim(Str(liDoctos))), "0") & LTrim(Str(liDoctos)) & ".RTF"

' Obtiene el Recordset de la Relación en Común
Select Case liElem
Case 2 ' Tema
  IsComando = "SELECT * FROM Tema"
Case 3 ' Titulo
  IsComando = "SELECT * FROM Titulo"
Case 4 ' Subtitulo
  IsComando = "SELECT * FROM SubTitulo"
Case 5 ' Detalle
  IsComando = "SELECT * FROM Detalle"
End Select

Set rsActualiza = Db.OpenRecordset(IsComando, dbOpenDynaset)
rsActualiza.AddNew
Select Case liElem
Case 2 ' Tema
  rsActualiza(1) = txtNombre.Text
  rsActualiza(2) = chkStatus.Value
  rsActualiza(3) = IsDoctos
Case 3 ' Titulo
  rsActualiza(1) = liTema
  rsActualiza(2) = txtNombre.Text
  rsActualiza(3) = chkStatus.Value
  rsActualiza(4) = IsDoctos

Case 4 ' Subtitulo
  rsActualiza(1) = liTitulo
  rsActualiza(2) = liTema
  rsActualiza(3) = txtNombre.Text
  rsActualiza(4) = chkStatus.Value
  rsActualiza(5) = IsDoctos

```

Case 5 'Detalle

rsActualiza(1) = liSubTitulo

rsActualiza(2) = liTitulo

rsActualiza(3) = liTema

rsActualiza(4) = txtNombre.Text

rsActualiza(5) = chkStatus.Value

rsActualiza(6) = IsDoctos

End Select

txtTexto.SaveFile IsRuta & IsDoctos

rsActualiza.Update

rsActualiza.Close

txtTexto.Text = ""

txtNombre.Text = ""

chkStatus.Value = 0

CargaElementos

End Sub

El siguiente código carga los elementos a las ramas del árbol en este caso Treeview por medio del Snapshot.

Sub CargaElementos()

Dim rsCarga As Recordset

Dim IsComando As String

Dim Nod As Node

Dim IsLLavePad As String

Dim IsLLave As String

Dim IsPaso As String

Dim IsPaso1 As String

Dim IsPaso2 As String

Dim IsPaso3 As String

Dim liPaso As Integer

Dim liPaso1 As Integer

Dim liPaso2 As Integer

Dim liPaso3 As Integer

TreeView1.Nodes.Clear

LViewopciones.ListItems.Clear

Set Nod = TreeView1.Nodes.Add()

Nod.Image = 1

Nod.Text = "Información"

```
Nod.Key = "" & "1" & ""
Nod.Tag = "RRR"
```

```
' Carga los Temas
IsComando = "SELECT LTRIM(STR(Id_Tema)) & '-'; Concepto FROM Tema"
Set rsCarga = Db.OpenRecordset(IsComando, dbOpenSnapshot)
Do While Not rsCarga.EOF()
  IsLLavePad = "" & "1" & ""
  IsLLave = "1-" & rsCarga(0)
  Set Nod = TreeView1.Nodes.Add(IsLLavePad, tvwChild, IsLLave, Trim(rsCarga(1)), 2)
  Nod.Tag = "TEM"
  rsCarga.MoveNext
Loop
```

```
rsCarga.Close
```

```
' Carga los Títulos
IsComando = "SELECT LTRIM(STR(Id_Tema)) & '-' & LTrim(Str(Id_Titulo)) & '-'; Concepto FROM Titulo"
Set rsCarga = Db.OpenRecordset(IsComando, dbOpenSnapshot)
Do While Not rsCarga.EOF()
  IsLLavePad = "1-" & Mid(rsCarga(0), 1, InStr(rsCarga(0), "-") - 1) & "-"
  IsLLave = "1-" & rsCarga(0)
  Set Nod = TreeView1.Nodes.Add(IsLLavePad, tvwChild, IsLLave, Trim(rsCarga(1)), 2)
  Nod.Tag = "TIT"
  rsCarga.MoveNext
Loop
rsCarga.Close
```

```
' Carga los Subtítulos
IsComando = "SELECT LTRIM(STR(Id_Tema)) & '-' & LTrim(Str(Id_Titulo)) & '-' & LTrim(Str(Id_SubTitulo)) & '-'; Concepto FROM Subtitulo"
Set rsCarga = Db.OpenRecordset(IsComando, dbOpenSnapshot)
Do While Not rsCarga.EOF()
  liPaso = InStr(rsCarga(0), "-")
  IsPaso = Mid(rsCarga(0), InStr(rsCarga(0), "-") + 1, 50)
  liPaso1 = InStr(IsPaso, "-")
```

```
  IsLLavePad = "1-" & Mid(rsCarga(0), 1, (liPaso + liPaso1) - 1) & "-"
  IsLLave = "1-" & rsCarga(0)
  Set Nod = TreeView1.Nodes.Add(IsLLavePad, tvwChild, IsLLave, Trim(rsCarga(1)), 2)
  Nod.Tag = "STI"
  rsCarga.MoveNext
Loop
rsCarga.Close
```

```
' Carga los Detalles
IsComando = "SELECT LTRIM(STR(Id_Tema)) & '-' & LTrim(Str(Id_Titulo)) & '-' & LTrim(Str(Id_SubTitulo)) & '-' & LTrim(Str(Id_Detalle)) & '-'; Concepto FROM Detalle"
Set rsCarga = Db.OpenRecordset(IsComando, dbOpenSnapshot)
Do While Not rsCarga.EOF()
  liPaso = InStr(rsCarga(0), "-")
  IsPaso = Mid(rsCarga(0), InStr(rsCarga(0), "-") + 1, 50)
```

```

liPaso1 = InStr(IsPaso, "-")
IsPaso1 = Mid(IsPaso, liPaso1 + 1, 50)
liPaso2 = InStr(IsPaso1, "-")
IsLLavePad = "1-" & Mid(rsCarga(0), 1, (liPaso + liPaso1 + liPaso2) - 1) & "-"
IsLLave = "1-" & rsCarga(0)
Set Nod = TreeView1.Nodes.Add(IsLLavePad, tvwChild, IsLLave, Trim(rsCarga(1)), 2)

Nod.Tag = "DET"
rsCarga.MoveNext
Loop
rsCarga.Close

End Sub

```

Con el siguiente código se modifican los elementos del árbol de información , así como la ruta para la devolución de datos.

```

Sub ModificaElemento()
Dim rsActualiza As Recordset
Dim IsComando As String
Dim IsDoctos As String

Select Case TreeView1.SelectedItem.Tag
Case "TEM"
liElem = 2
Case "TIT"
liElem = 3
Case "STI"
liElem = 4
Case "DET"
liElem = 5
End Select

Select Case liElem
Case 2 ' Tema
IsComando = "SELECT * FROM Tema WHERE Id_Tema = " & LTrim(Str(liTema))
Case 3 ' Titulo
IsComando = "SELECT * FROM Titulo WHERE Id_Tema = " & LTrim(Str(liTema)) & _
" AND Id_Titulo = " & LTrim(Str(liTitulo))
Case 4 ' SubTitulo
IsComando = "SELECT * FROM SubTitulo WHERE Id_Tema = " & LTrim(Str(liTema)) & _
" AND Id_Titulo = " & LTrim(Str(liTitulo)) & _
" AND Id_Subtitulo = " & LTrim(Str(liSubTitulo))
Case 5 ' Detalle
IsComando = "SELECT * FROM Detalle WHERE Id_Tema = " & LTrim(Str(liTema)) & _
" AND Id_Titulo = " & LTrim(Str(liTitulo)) & _
" AND Id_Subtitulo = " & LTrim(Str(liSubTitulo)) & _
" AND Id_detalle = " & LTrim(Str(liDetalle))
End Select

```

```
Set rsActualiza = Db.OpenRecordset(IsComando, dbOpenDynaset)
```

```
rsActualiza.Edit
```

```
Select Case liElem
```

```
Case 2
```

```
IsDoctos = IsPathTema & Trim(rsActualiza!Detalle)
```

```
Case 3
```

```
IsDoctos = IsPathTitulo & Trim(rsActualiza!Detalle)
```

```
Case 4
```

```
IsDoctos = IsPathSubtitulo & Trim(rsActualiza!Detalle)
```

```
Case 5
```

```
IsDoctos = IsDetalle & Trim(rsActualiza!Detalle)
```

```
End Select
```

```
Select Case liElem
```

```
Case 2 ' Tema
```

```
rsActualiza(1) = txtNombre.Text
```

```
rsActualiza(2) = chkStatus.Value
```

```
Case 3 ' Titulo
```

```
rsActualiza(2) = txtNombre.Text
```

```
rsActualiza(3) = chkStatus.Value
```

```
Case 4 ' Subtitulo
```

```
rsActualiza(3) = txtNombre.Text
```

```
rsActualiza(4) = chkStatus.Value
```

```
Case 5 ' Detalle
```

```
rsActualiza(4) = txtNombre.Text
```

```
rsActualiza(5) = chkStatus.Value
```

```
End Select
```

```
txtTexto.SaveFile IsDoctos
```

```
rsActualiza.Update
```

```
rsActualiza.Close
```

```
TreeView1.SelectedItem.Text = Trim(txtNombre.Text)
```

```
End Sub
```

A continuación se muestra el código de los módulos del árbol de información y organismos:

MÓDULO DE ORGANISMOS:

```
Public IsComando As String
Public db As Database
Public liOrganismo As Long
Public liOpcion As Integer
Public Sub main()
Dim strBaseLocalizada As String

IsComando = Command
If IsComando = "" Then
    MsgBox "<< ERROR >> No Hay Parametros Para Ejecución !!!", vbCritical, "Inicio Programa"
End
End If

Set db = DBEngine.Workspaces(0).OpenDatabase(IsComando, , True)
liOrganismo = 0
liOpcion = 0

frmacceso.Show

End Sub

Public Function ObtieneCadenaPorLista(PoObjeto As Object) As String
Dim lbPaso As Boolean

For i% = 0 To (PoObjeto.ListCount - 1)
If PoObjeto.Selected(i%) Then
If Not lbPaso Then
ObtieneCadenaPorLista = PoObjeto.ItemData(i%)
lbPaso = True
Else
ObtieneCadenaPorLista = ObtieneCadenaPorLista & ", " & PoObjeto.ItemData(i%)
End If
End If
Next i%

End Function
```

```
Public Sub ValorCombo(PoObjeto As Object, PsCampo As String, PbSinRegistros)
```

```
    If PbSinRegistros Then
```

```
        PoObjeto.ListIndex = 0
```

```
    Else
```

```
        For i% = 0 To (PoObjeto.ListCount - 1)
```

```
            If PoObjeto.ItemData(i%) = Val(PsCampo) Then
```

```
                PoObjeto.ListIndex = i%
```

```
                Exit Sub
```

```
            End If
```

```
        Next i%
```

```
    End If
```

```
End Sub
```

```
Public Sub ValorLista(PoObjeto As Object, PsCadena As String)
```

```
    Dim X As Integer
```

```
    Dim z As Integer
```

```
Do While Len(PsCadena) <> 0
```

```
    X = InStr(1, PsCadena, ",")
```

```
    If X = 0 Then
```

```
        z = PsCadena
```

```
    Else
```

```
        z = Mid(PsCadena, 1, X)
```

```
    End If
```

```
    For i% = 0 To (PoObjeto.ListCount - 1)
```

```
        If PoObjeto.ItemData(i%) = z Then
```

```
            PoObjeto.Selected(i%) = True
```

```
            Exit For
```

```
        End If
```

```
    Next i%
```

```
    If X = 0 Then
```

```
        Exit Do
```

```
    Else
```

```
        PsCadena = Mid(PsCadena, X + 1, Len(PsCadena))
```

```
    End If
```

```
Loop
```

```
PoObjeto.ListIndex = 0
```

```
End Sub
```

```
Public Sub LienaCatalogo(PiPadre As Integer, _
```

```
    PoObjeto As Object, _
```

```
    PbCombo As Boolean)
```

```
    Dim rsCatalogos As Recordset
```

```
    Dim qrCatalogos As QueryDef
```

```
    Set qrCatalogos = db.QueryDefs("LienaCatalogos")
```

```
    qrCatalogos.Parameters("xPadre") = PiPadre
```

```
    Set rsCatalogos = qrCatalogos.OpenRecordset(dbOpenSnapshot)
```

```

If PbCombo = True Then
  'If pobjeto.Name = "cboGenero" Then
  'No pone Sin Asignar
  'Else
  PoObjeto.AddItem "Sin Asignar"
  PoObjeto.ItemData(PoObjeto.NewIndex) = 0
  'End If
End If

If Not rsCatalogos.EOF Then
  rsCatalogos.MoveFirst
End If

Do While Not rsCatalogos.EOF
  PoObjeto.AddItem rsCatalogos("fcConcDetNombre")
  PoObjeto.ItemData(PoObjeto.NewIndex) = rsCatalogos("fiConcPadreId")
  rsCatalogos.MoveNext
Loop
PoObjeto.ListIndex = 0

qrCatalogos.Close
rsCatalogos.Close

Set qrCatalogos = Nothing
Set rsCatalogos = Nothing

End Sub

```

ÁRBOL DE INFORMACIÓN:

Option Explicit

Public IsNombre As String

Public IsClave As String

Public Db As Database

Public Function Conecta() As Boolean

Dim lBol As Boolean

Conecta = True

On Error GoTo error

Set Db = DBEngine.Workspaces(0).OpenDatabase("c:\apoyo\CONADIC.mdb", False, False)

Exit Function

error:

Conecta = False

Resume Next

End Function

Sub Main()

If Conecta Then

frmacc.Show

Else

MsgBox "<< ERROR >> al Accesar la Base de Datos Verifique su Ubicación ...", vbCritical, "Acceso a la Base de Datos "

End

End If

End Sub

ANEXO 2.

Manual del Usuario

MANUAL DEL USUARIO

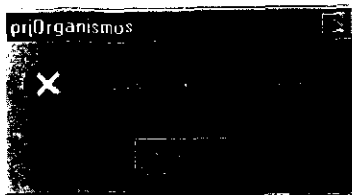
ORGANISMOS

Introduce usuario y contraseña en la posición indicada y presiona el botón ingresar para acceder al sistema

The image shows a screenshot of a login window. At the top, there is a title bar with the text "CONTRASEÑA" and standard window control icons. Below the title bar, on the left, is a logo consisting of a stylized gear and a key. To the right of the logo are two input fields: "Usuario:" and "Contraseña:". Below these fields are three buttons: "INGRESAR", "LIMPIAR", and "SALIR". Arrows from the text boxes above and below point to the input fields and the buttons respectively.

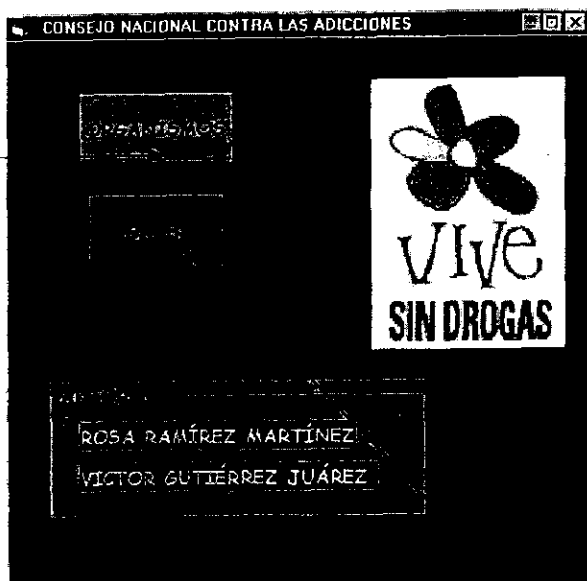
Presiona limpiar por si desea modificar y salir para terminar la cesión

Si al digitar la contraseña no es correcta el sistema desplegará el siguiente mensaje:



Para volver a teclear la contraseña damos un click en aceptar y el proceso se repetira.

Una vez ingresando a organismos mostrará la pantalla principal

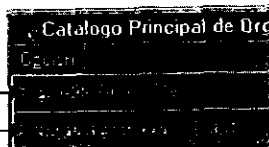


para acceder al menú principal de organismos, presionar el botón del mismo nombre

En caso contrario presionar salir

Una vez ingresado a organismos mostrará un catálogo de las diferentes instituciones que lo componen.

También podemos usar el menú correspondiente a OPCION dando click en él, desplegará: AGREGAR ORGANISMOS y SALIR, donde tendrá la función indicada en el nombre.



Para conseguir información de un determinado organismo lo seleccionamos y desplazamos hacia la pestaña correspondiente dando click en ella.

Catalogo Principal de Organismos

Opción

Organismo No. 26 - ORGANISMO UNO

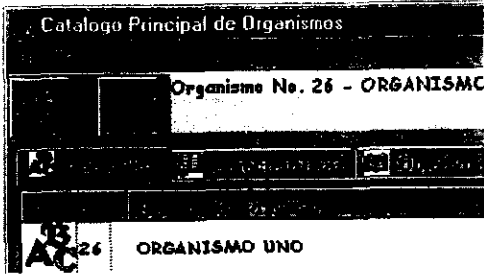
Generales | Características | Objetivo | Servicio | Serv. Adicionales

| Numero | Nombre del Organismo | S |
|--------|--|---|
| 26 | ORGANISMO UNO | 0 |
| 35 | ORGANISMO 2 | 0 |
| 36 | ORGANISMO 3 | 0 |
| 37 | ALCOHOLICOS ANONIMOS DE GUSTAVO A MADERO FBDD | 0 |
| 38 | ORGANISMO CUATRO | 0 |
| 39 | ORGANISMO DEL VALLE DE MEXICO S.A. DE C.V. | 0 |
| 42 | ORGANISMO DE APOYO Y ORIENTACION A PERSONAS CON ADICC... | 0 |

[Ins] Agrega [Del] Elimina [Enter/Click] Detalle

4/3/00

Para dar de alta un organismo seleccionar el botón de nombre ABC del catálogo principal

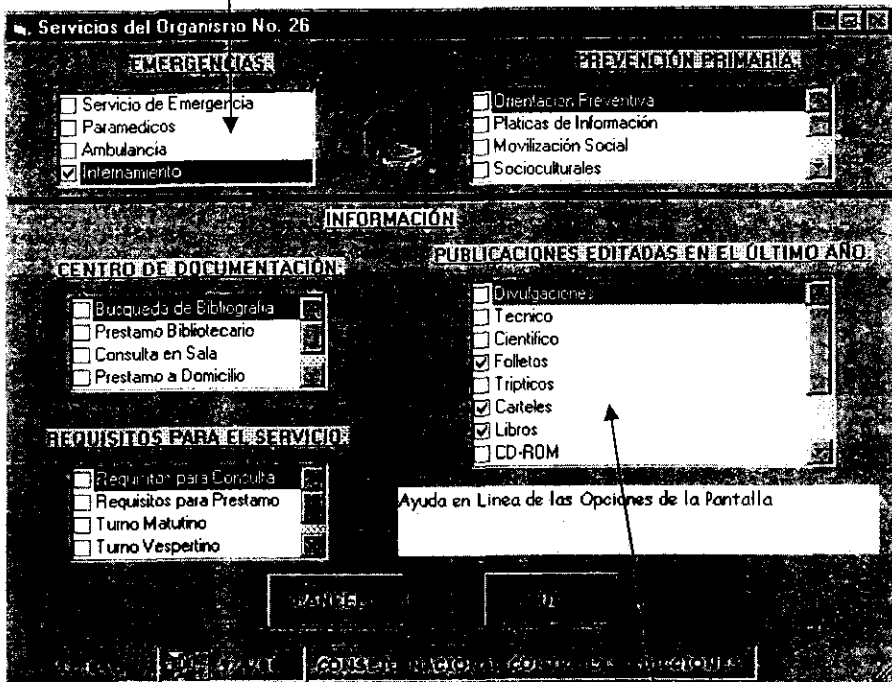


Donde podremos capturar la información correspondiente a cada organismo según nos sea solicitada por ejemplo en datos generales del organismo nuevo.

| | | | |
|---|-------------|------------|----------|
| Organismo Nuevo !!!! | | | |
| NOMBRE DE LA INSTITUCIÓN: Organismo Uno | | | |
| TIPO DE INSTITUCIÓN: | Publica | COSTOS: | Gratis |
| CALLE Y NÚMERO: Av. Reforma Agraria | | | |
| COLONIA/POBLACIÓN: | Tabacalera | TELEFONO: | 34255 |
| CIUDAD/MUNICIPIO: | DELEGACIÓN: | CP: | FAX: |
| D.F.: | Cuhtemoc | 23457 | 56478392 |
| ESTADO: | ESTAT: | STATUS: | |
| Distrito Federal | | Habilitado | |
| DIRECCION INTERNET: orgorganismouno@center.com.mx | | | |
| Teclee el Número de Fax del Organismo [ESC] Cancela / Salida del Modulo | | | |

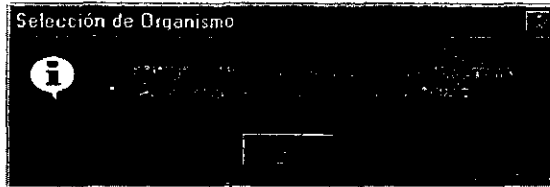
Una vez terminada la captura dar un click en ok para grabar la información.

En el caso de los formularios restantes solo hay que seleccionar el servicio dando un click en el recuadro que se requiera y de inmediato se marcara como se muestra en la opción internamiento de emergencias.



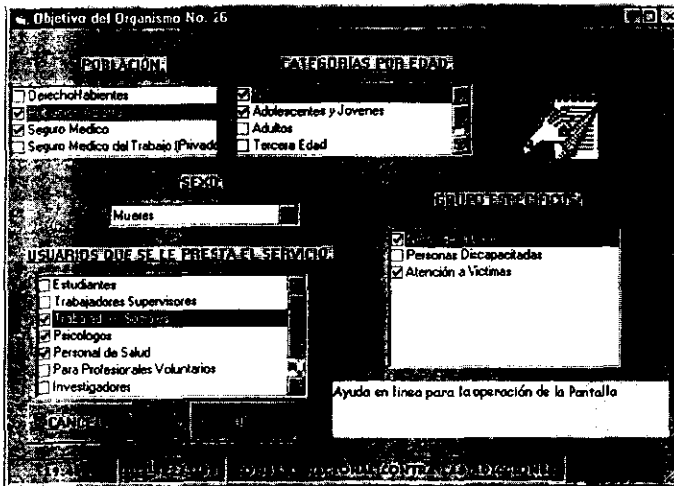
Podemos seleccionar más de una opción en el mismo servicio, por ejemplo en publicaciones editadas que seleccionamos folletos, carteles y libros

En caso de no seleccionar ningún organismo el sistema arrojará el siguiente mensaje:

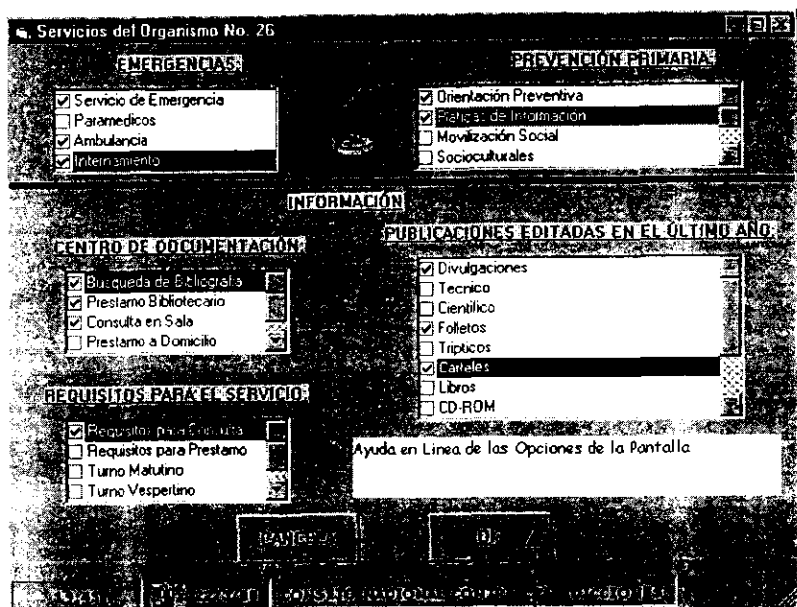


Dando click o enter en aceptar el cuadro se elimina mostrando el menú principal

En caso contrario, si seleccionamos objetivos del organismo uno, nos mostrará la pantalla correspondiente a él y sus diferentes opciones como son: población, categorías por edad, etc. Según la pantalla siguiente:

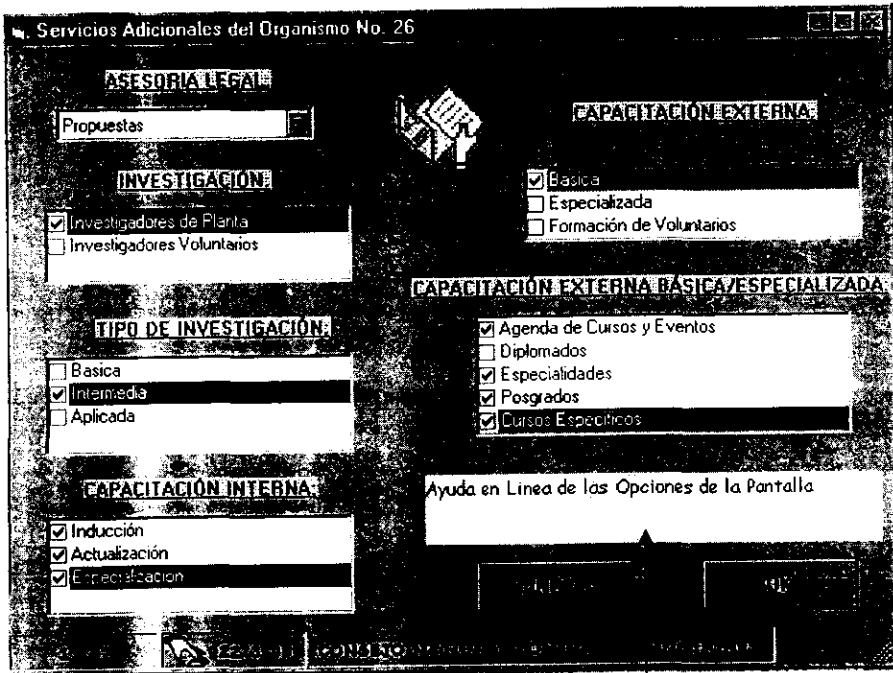


Lo mismo sucede con los servicios que proporcionan los organismos que muestra el menú de servicios del organismo (emergencias, prevención primaria, publicaciones editadas, etc.)



No olvidar que si se desea grabar la información en caso de captura de la misma dar un click en el botón correspondiente a ok

Al igual que en la anterior, podemos consultar los servicios que proporciona el organismo, en el ejemplo: asesoría legal, capacitación y tipo de investigación.



En cada una de las pantallas el sistema muestra un recuadro de ayuda en línea según sean seleccionadas cada una de las opciones

Para consultar tipo de servicio, modalidades, rehabilitación y otros problemas tenemos que consultar la opción de tratamiento del organismo, como se muestra en la siguiente pantalla.

Tratamiento del Organismo No. 26

TIPO DE SERVICIO

- Consulta Externa
- Centro de Día
- Internamiento
- Casa de Medio Camino
- Desintoxicación

MODALIDADES

- Orientación Psicológica
- Terapia Individual
- Terapia Familiar
- Terapia Farmacológica
- Sesiones Grupales
- Grupos de Autoayuda

REHABILITACIÓN/REINSERCIÓN

- Laboral
- Educativa
- Recreativa
- Física

ATIENDE OTROS PROBLEMAS

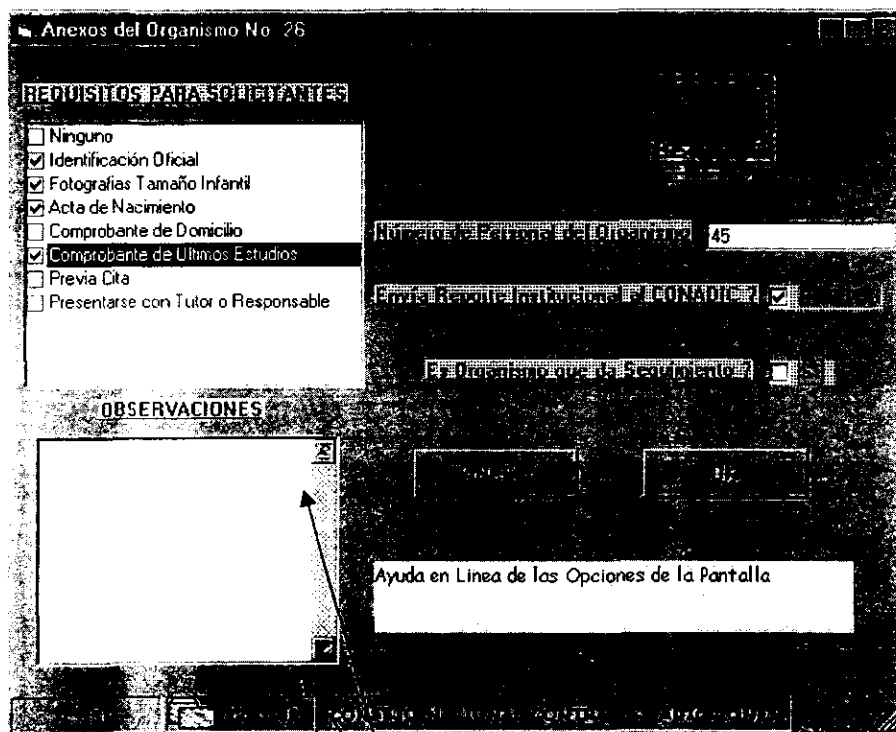
- Anorexia-Bulimia
- Psicólogos en General
- Juego Compulsivo
- Relaciones Autodestructivas
- Comedores Compulsivos

Ayuda en Línea de las Opciones de la Pantalla

CANCELAR ACEPTAR REINICIAR CONTINUAR AYUDA EN LÍNEA

Para volver al menú principal dar click en el botón cancelar de cada uno de los formularios.

Como última alternativa tenemos la opción de anexos del organismo donde consultaremos los requisitos para los solicitantes y algunas observaciones al mismo.

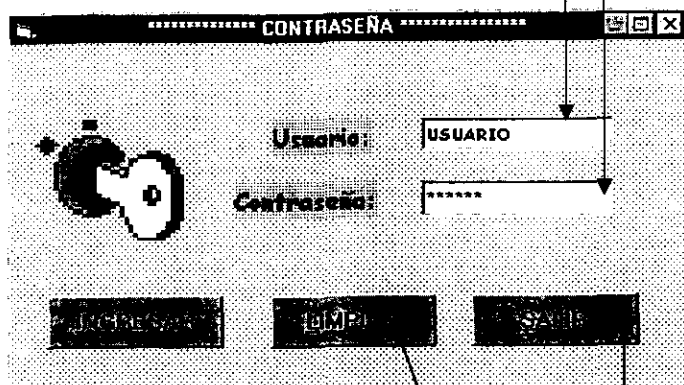


Para poder desplazarnos a lo largo del cuadro correspondiente a observaciones damos click a las flechas del extremo derecho del cuadro hacia arriba o hacia abajo según el requerimiento

MANUAL DEL USUARIO

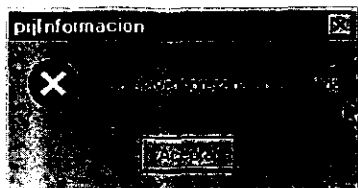
ÁRBOL DE INFORMACIÓN

Para acceder al menú de árbol de información teclear usuario y contraseña correcta en el lugar correspondiente, después dar click, en ingresar



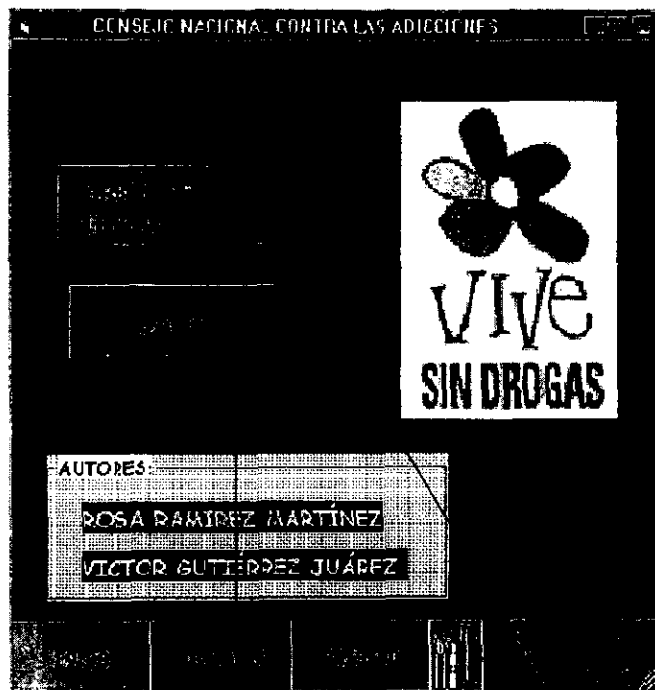
Dar un click en limpiar o en salir según se requiera

Si la clave ingresada es incorrecta desplegará el mensaje siguiente:



para repetir la
operación dar un
click en aceptar

Una vez ingresado al sistema desplegará la pantalla siguiente:

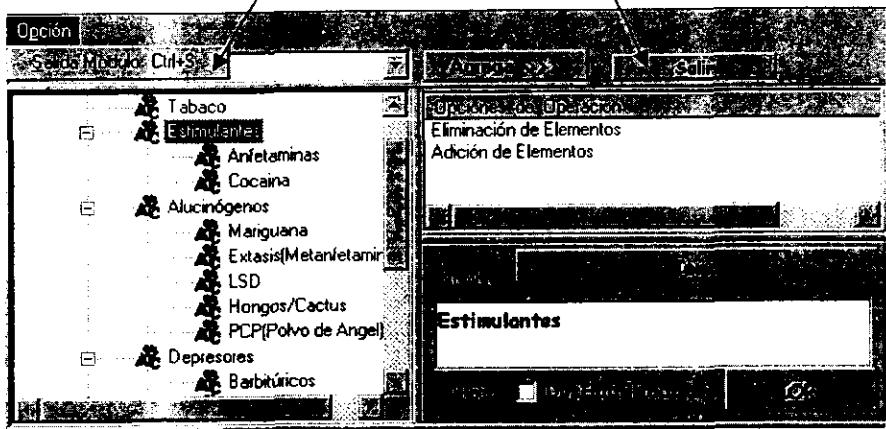


Dar un click en salir si deseamos no acceder al sistema y terminar la sesión en el mismo.

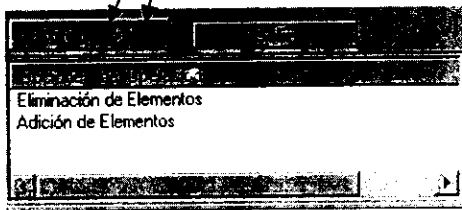
Dando click en el botón de árbol de información

Ya en el menú principal de árbol de información tenemos las siguientes opciones:

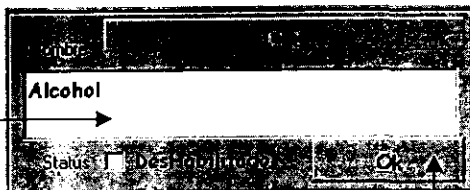
Si seleccionamos OPCIÓN desplegará un catálogo donde podemos seleccionar: SALIDA MODULO o con las teclas Ctrl + S para salir del sistema, también podemos seleccionar el botón Salir.



Para agregar un nuevo título, sub-título o detalle solo damos click en el botón **Agregar** o seleccionamos **Adición de Elementos**

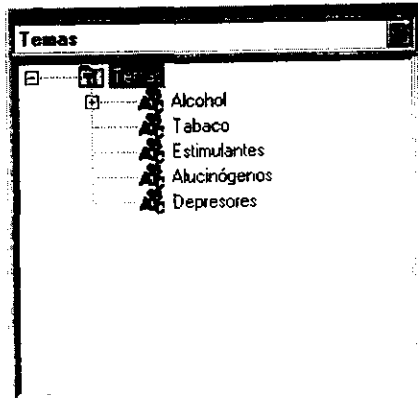


El cursor se posicionará en la pantalla donde teclearemos el elemento a agregar

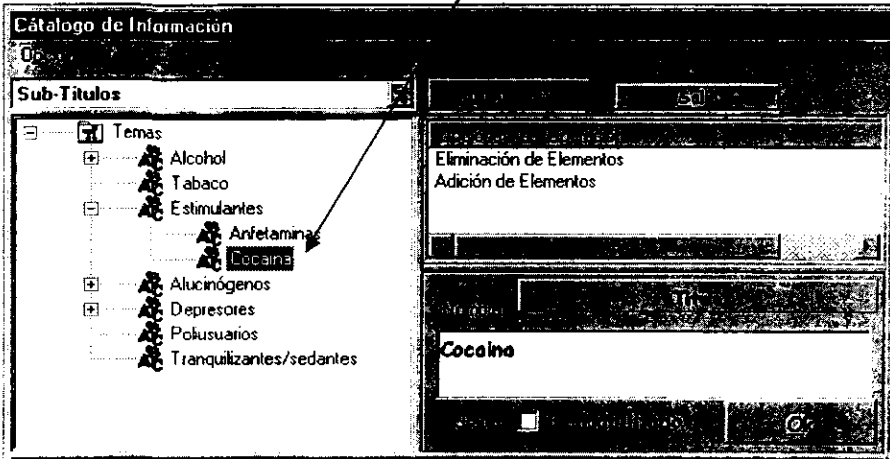


Seleccionamos ok y la información escrita quedará grabada.

En la siguiente pantalla se muestran los elementos agregados en este caso el elemento a agregar fue alcohol



El mismo procedimiento se sigue al agregar sub-Títulos o detalles



Si queremos consultar cualquier título, por ejemplo alcohol, seleccionamos el título correspondiente dando click y enseguida nos desplegará la información correspondiente

The screenshot shows a web application interface. At the top left, the title 'Cátalogo de Información' is visible. Below it is a 'Títulos' section with a tree view under 'Temas'. The tree view includes 'Alcohol', 'Tabaco', 'Estimulantes', 'Alucinógenos', and 'Depresores'. An arrow points from the text box above to the 'Alcohol' item in the tree. To the right of the tree view is a panel with a search bar and buttons for 'Eliminación de Elementos' and 'Adición de Elementos'. Below this is a large text area displaying the title 'Alcohol' and a scrollable text block. The text block contains the following content:

Alcoholismo

¿Qué es el alcoholismo?

El alcoholismo es una enfermedad crónica, progresiva y a menudo mortal; es un trastorno primario y no un síntoma de otras enfermedades o problemas emocionales. La química del alcohol le permite afectar casi todo tipo de célula en el cuerpo, incluyendo las que se encuentran en el sistema nervioso central. En el cerebro, el alcohol interactúa con centros responsables del placer y otras sensaciones deseables; después de la exposición prolongada al alcohol, el cerebro se adapta a los

At the bottom of the text area, there are navigation controls including a scroll bar and buttons for 'Inicio', 'Volver', and 'Salir'.

Donde podemos desplazarnos a lo largo de la información

El mismo procedimiento se sigue para consultar cualquier sub-título

Catálogo de Información

Sub-Títulos

- Alcohol
- Tabaco
- Estimulantes
 - Anfetaminas
 - Cocaína**
- Alucinógenos
- Depresores
 - Barbitúricos
 - Analgésicos
 - Tranquilizantes/sedantes
- Opiáceos
- Inhalables

Eliminación de Elementos
Adición de Elementos

Cocaína

COCAINA

CONSTITUCIÓN QUÍMICA
La cocaína es un alcaloide extraído de la *Eritroxillum coca*, arbusto originario de Perú y Bolivia cuyas hojas eran masticadas por los indígenas sudamericanos para aprovechar los efectos estimulantes y aligerar la fatiga de la faena diaria de trabajo. Este alcaloide fue aislado por primera vez de la hoja de coca en 1860 por el químico alemán Newman.

FORMAS DE PRESENTACIÓN
La presentación más común de la cocaína es el clorhidrato de cocaína el cual es un polvo blanco.

Para consultar **detalles** se selecciona de la misma manera y enseguida muestra la información.

The screenshot shows a software interface titled "Cátalogo de Información". On the left, there is a tree view under the heading "Detalles" with the following categories and sub-items:

- Estimulantes
 - Anfetaminas
 - Cocaína
- Alucinógenos
- Depresores
 - Barbitúricos
 - Analgésicos
 - Tranquilizantes/seda
- Opiáceos
 - Codeína
 - Heroína**
 - Morfina

On the right side of the interface, there are several panels. The top panel is titled "Operaciones de Operación" and contains "Eliminación de Elementos" and "Adición de Elementos". Below that, a panel displays the name "Heroína". At the bottom of the interface, a large text area contains the following information:

HEROINA

Efectos: Sensaciones exageradas a nivel fisico-emocional, ansiedad, disminución de la razón, el entendimiento y la memoria, retardo psicomotor

Consecuencias: Estados de confusión general, convulsiones, alucinaciones, contracturas musculares, disminución de la presión arterial

At the bottom of the window, there are buttons for "Inicio", "Imprimir", and "Salir".