

20
1ej.



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

FACULTAD DE INGENIERIA

**SISTEMA PARA LA ADMINISTRACION DEL
PATRIMONIO INMOBILIARIO DEL STC
METRO**

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION

P R E S E N T A N

EMILIO A. EYSSAUTIER MONTALVO

CLAUDIA URZUA ROSAS

CLAUDIA MARIA VELEZ VAZQUEZ

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



MEXICO, D. F.

**TESIS CON
FALLA DE ORIGEN**

269803

1997



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.

PAGINACION

DISCONTINUA

Al M.I. Juan Carlos Roa Beiza:

En agradecimiento al tiempo
que nos dedico, brindándonos
su apoyo incondicional para la
culminación de nuestra tesis.

Con aprecio sus alumnos que lo admiran

Claudia Urzúa Rosas,

Claudia María Vélez Vázquez,

Emilio A. Eyssautier Montalvo

INTRODUCCIÓN

Objetivo General :

El sistema será desarrollado a petición expresa en ACCESS utilizando como front-end algunas herramientas que posee esta base de datos y Visual Basic.

El sistema recabará toda la información que posea en papel de predios que serán o fueron afectados por la construcción del metro, después se hará un análisis de la información por parte del Departamento Jurídico, el cual determinará si ésta es suficiente o bien habrá de investigarse a quién corresponde la propiedad; en caso de ser insuficiente la información, se procederá a complementarla hasta que satisfaga los requisitos de Jurídico.

Una vez cubiertos los pasos anteriores se procederá a la captura de la información, la cual consistirá de fotografías, croquis, planos y copias de documentos que deberán ser compiladas en una carpeta que posea toda la información relacionada a cada predio investigado. Después, toda esta información, ya una vez autorizada, deberá ser digitalizada y capturada en la base de datos ACCESS para posteriormente ser explotada por el personal que así lo requiera.

La información será almacenada inicialmente en discos duros de gran capacidad para poder ser consultada en línea en todo momento. Posteriormente, ya una vez eliminados todos los posibles "bugs" que pudiera contener, se procederá a la grabación en CD-ROM, previa autorización de un comité de evaluación de ésta.

El sistema deberá poseer el concepto de **amigable**, es decir, que cualquier usuario con poca experiencia en computación pueda adentrarse rápidamente en el manejo del sistema a través del uso de menús gráficos.

ÍNDICE

1	FUNDAMENTOS TEORICOS	1
1.1	Conceptos de Bases de Datos Relacionales	1
1.2	Metodología de Diseño y Desarrollo de Sistemas	55
	1.2.1 Ciclo de vida de los sistemas	61
	1.2.2 Yourdon	73
	1.2.3 Gane and Sarson	76
	1.2.4 Técnica de SSADM (Structured Systems Analysis Design Methodology)	79
	1.2.5 Modelado de Merise	79
1.3	Redes	82
	1.3.1 Topologías más importantes, Ventajas y Desventajas	88
	1.3.2 Protocolos de Comunicaciones, TCP/IP, SNA, Token Ring	96
1.4	Características, ventajas y desventajas del manejador ACCESS	107
1.5	Características, ventajas y desventajas de los paquetes de soporte : Power Point, AutoCAD r13, PhotoStyler, Photo-End y Guía Roji (CD).	131
2.	ANÁLISIS DE LA PROBLEMÁTICA	202
2.1	Problemática actual	202
	2.1.1 Áreas de interés	210
	2.1.2 Consecuencias del problema	212

2.2	Recopilación de la información	216
2.3	Clasificación de la información	223
2.4	Requerimientos de usuario	236
2.5	Opciones de solución	238
2.6	Requerimientos mínimos necesarios para arrancar la aplicación	245
3.	DISEÑO E IMPLEMENTACIÓN DEL SISTEMA	249
3.1	Acotamiento del problema	249
3.2	Diseño y construcción de la base de datos	251
	3.2.1 Diagrama de flujo de datos	252
	3.2.2 Normalización de la base de datos	257
	3.2.3 Diagrama Entidad-Relación	263
	3.2.4 Diccionario de datos	265
3.3	Implementación del Back-End	266
3.4	Desarrollo de las pantallas de captura	289
3.5	Desarrollo de los reportes del sistema	307
3.6	Desarrollo de los módulos de administración del sistema	318
	3.6.1 Control de usuarios	322
	3.6.2 Mantenimiento a catálogos	324
	3.6.3 Depuración	330
3.7	Pruebas e Integración	333

CONCLUSIONES

BIBLIOGRAFÍA

APÉNDICE A **CÓDIGO DE PROGRAMACIÓN**

APÉNDICE B **GLOSARIO DE TÉRMINOS**

APÉNDICE C **DICCIONARIO DE DATOS**

APÉNDICE D **MANUAL DE USUARIO**

CAPITULO 1
FUNDAMENTOS TEORICOS

1. FUNDAMENTOS TEORICOS

1.1 CONCEPTOS DE BASES DE DATOS RELACIONALES

Una base de datos puede definirse como una colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es la de servir a una aplicación o más, de la mejor manera posible; los datos se almacenan de modo que resulten independientes de los programas que se usan; se emplean métodos bien determinados para incluir datos nuevos y para modificar o extraer datos almacenados.

1.1.1 Objetivos De La Organización De Las Bases De Datos

Las bases de datos pueden organizarse de muchas maneras. Los principios para la selección de técnicas de organización han sido estudiadas ampliamente. La comisión de sistemas CODASYL (Conference on Data Systems Languages) ha emitido varios informes sobre el tema, y ha emitido un informe clásico sobre los requerimientos que debe satisfacer un sistema de administración de bases de datos. A continuación se lista un resumen de sus opiniones:

- **Los datos podrán utilizarse de múltiples maneras.** Es decir, diferentes usuarios, perciben de manera distinta los mismos datos, y pueden emplearlos de diversas formas.
- **Se protegerá la inversión intelectual.** No será necesario rehacer los programas y las estructuras lógicas existentes (que representan muchos hombres-año de trabajo) cuando se modifique la base de datos.

- **Bajo costo.** Bajo costo de almacenamiento y el uso de los datos y minimización del costo de los cambios.
- **Desempeño.** Los pedidos de datos se atenderán con la rapidez adecuada según el uso que de ellos habrá de hacerse.
- **Claridad.** Los usuarios sabrán que datos se encuentran a su disposición y los comprenderán sin dificultad.
- **Facilidad de uso.** Los usuarios tendrán fácil acceso a los datos. Las complejidades internas son ajenas al usuario, gracias al sistema de administración de la base.
- **Flexibilidad.** Los datos podrán ser utilizados o explorados de manera flexible, con diferentes caminos de acceso.
- **Rápida atención de interrogantes no previstas.** Los pedidos espontáneos de información se atenderán sin necesidad de escribir un programa de aplicación sino utilizando un lenguaje de alto nivel para averiguación o generación de reportes.
- **Facilidad para el cambio.** La base de datos puede crecer y variar sin interferir con las maneras establecidas de usar los datos.
- **Precisión y coherencia.** Se utilizarán controles de precisión. El sistema evitará las versiones múltiples de los mismos datos con diferentes estados de actualización.
- **Seguridad de acceso.** Se evitará el acceso no autorizado de datos. Los mismos datos podrán estar sujetos a diferentes restricciones de acceso para distintos usuarios.
- **Integridad.** Los datos estarán protegidos contra fallas, catástrofes, personas incompetentes o que intenten falsificarlos.

- **Disponibilidad.** Los datos se hallarán inmediatamente disponibles para los usuarios casi todas las veces que los necesiten.
- **Independencia física de los datos.** El hardware de almacenamiento y las técnicas físicas de almacenamiento podrán ser alteradas sin obligar a la modificación de los programas de aplicación.
- **Independencia lógica de los datos.** Podrán agregarse nuevos datos, o expandirse la estructura lógica general, sin que sea necesario reescribir los programas de aplicación existentes.
- **Redundancia controlada.** Los datos serán almacenados una sola vez, excepto cuando existan razones técnicas o económicas que aconsejen el almacenamiento redundante.
- **Adecuada rapidez de acceso.** Los mecanismos de acceso y los métodos de direccionamiento serán lo suficientemente rápidos, para satisfacer las necesidades de los usos previstos.
- **Adecuada rapidez de exploración.** La conveniencia y necesidad de la exploración espontánea se incrementarán en la medida que se difunda el uso interactivo de los sistemas.
- **Normalización de los datos dentro de un organismo.** Se necesita un acuerdo interdepartamental sobre los formatos y las definiciones de datos. La normalización es indispensable porque de otro modo se crearían datos incompatibles.
- **Diccionario de datos.** Se necesita un diccionario de datos que defina las estructuras de las Base de Datos.

- **Interfaz de alto nivel con los programadores.** Los programadores de aplicaciones deben disponer de medios sencillos para pedir datos y estar aislados de las complejidades internas de organización y direccionamiento de los archivos.
- **Lenguaje del usuario final.** Un lenguaje de consulta de alto nivel o un lenguaje para la generación de reportes permitirán que los usuarios finales se vean libres de tener que escribir un programa de aplicación convencional.
- **Controles de integridad.** Siempre que sea posible, se recurrirá a revisiones de límites y otros controles para asegurar la exactitud de los datos.
- **Fácil recuperación en caso de falla.** Recuperación automática sin pérdida de información.
- **Afinación.** La base de datos debe ser afinable, para mejorar su desempeño sin exigir la reescritura de los programas de aplicación.
- **Ayudas para el diseño y la supervisión.** Ayudas que permitan al diseñador o al administrador de datos predecir y optimizar el desempeño del sistema.
- **Migración o reorganización automática.** Migración de datos u otra reorganización física previstas para la mejora del desempeño.

Los sistemas de base de datos se diseñan para manejar grandes cantidades de información. El manejo de los datos implica tanto la definición de estructuras para el almacenamiento como la creación de mecanismos para manejar la información.

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

de abstracción en los que puede considerarse a la base de datos: físico, conceptual y de visión.

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

Las bases de datos cambian con el tiempo al insertar información en ellas y eliminarla. El conjunto de información almacenada en la base de datos en determinado momento se denomina instancia de la base de datos. El diseño general de dicha base se conoce como esquema de la base de datos. La capacidad para modificar una definición de esquema en un nivel sin afectar la definición del esquema en el nivel inmediato superior se denomina independencia de datos.

1.1.2 Entidades y Atributos

Llamaremos entidades a los objetos sobre las cuales se almacena información. Una entidad puede representar algo real, tangible o abstracto usualmente a una persona, lugar, cosa o evento. Toda entidad tiene propiedades que eventualmente conviene registrar. A menudo, en el procesamiento de datos nos interesan las colecciones de entidades similares y necesitamos registrar información acerca de las mismas propiedades de cada una de ellas. A estas colecciones de objetos similares las llamaremos conjuntos de entidades.

Así como las entidades describen el mundo real, existen términos que se utilizan para describir la información acerca de ellas. Por lo general mantenemos un registro para cada entidad y agrupamos en conjuntos de registros de entidad todos los registros pertinentes a

entidades similares. Los registros se refieren a atributos de las entidades y contienen los valores de estos atributos.

Cuando hablamos de información podemos referirla a tres diferentes campos, y tendemos, a veces confusamente, a saltar de uno a otro sin advertencia previa. El primero de estos campos es el del mundo real, en el que hay entidades y estas exhiben ciertas propiedades. El segundo es el dominio de las ideas y la información existente en las mentes de las personas y los programadores. Aquí hablamos de los atributos de las entidades y nos referimos a éstos simbólicamente. Asignamos valores a los atributos. El tercer campo es el de los datos, en el que usamos caracteres o bits para codificar la información.

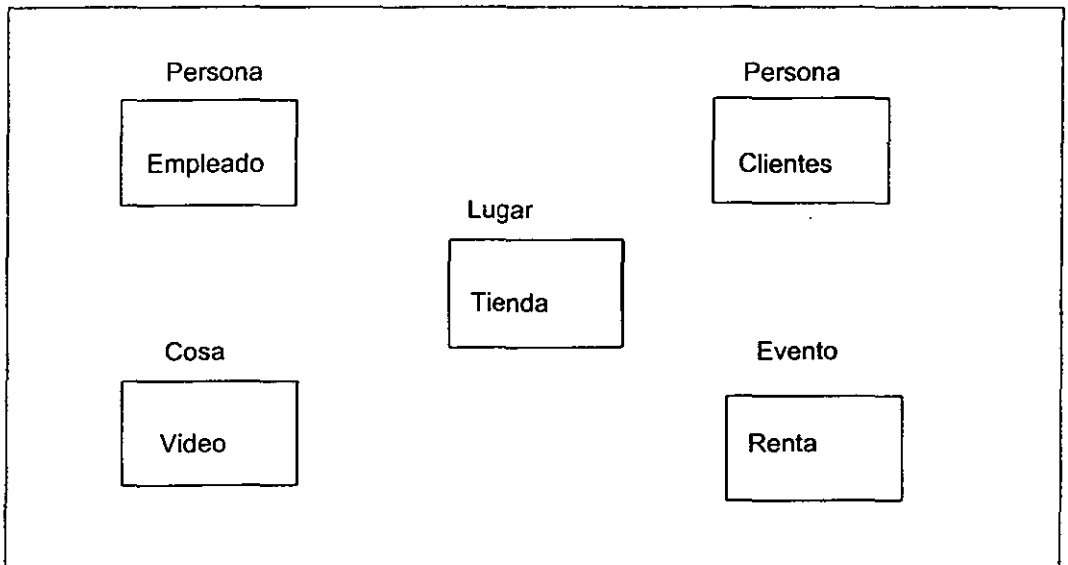


Figura 1.1.a Ejemplo de Entidades (Sistema de Renta de Videos)

La manera más común de asociar un valor con un dato y de asociar datos con atributos de entidad, consiste en almacenar juntos los datos en un secuencia fija, por ejemplo, en un registro de rentas. En este podría usarse un codificación para representar los valores

del atributo. Para esta entidad se indicará si el atributo es de longitud fija o de longitud variable. Los nombres de los atributos y las representaciones de valor no se registran en el archivo, aunque se deben encontrar registrados en alguna parte, por ejemplo, en el diccionario de datos, en el que se listen los nombres y tipos de los diferentes datos de la base.

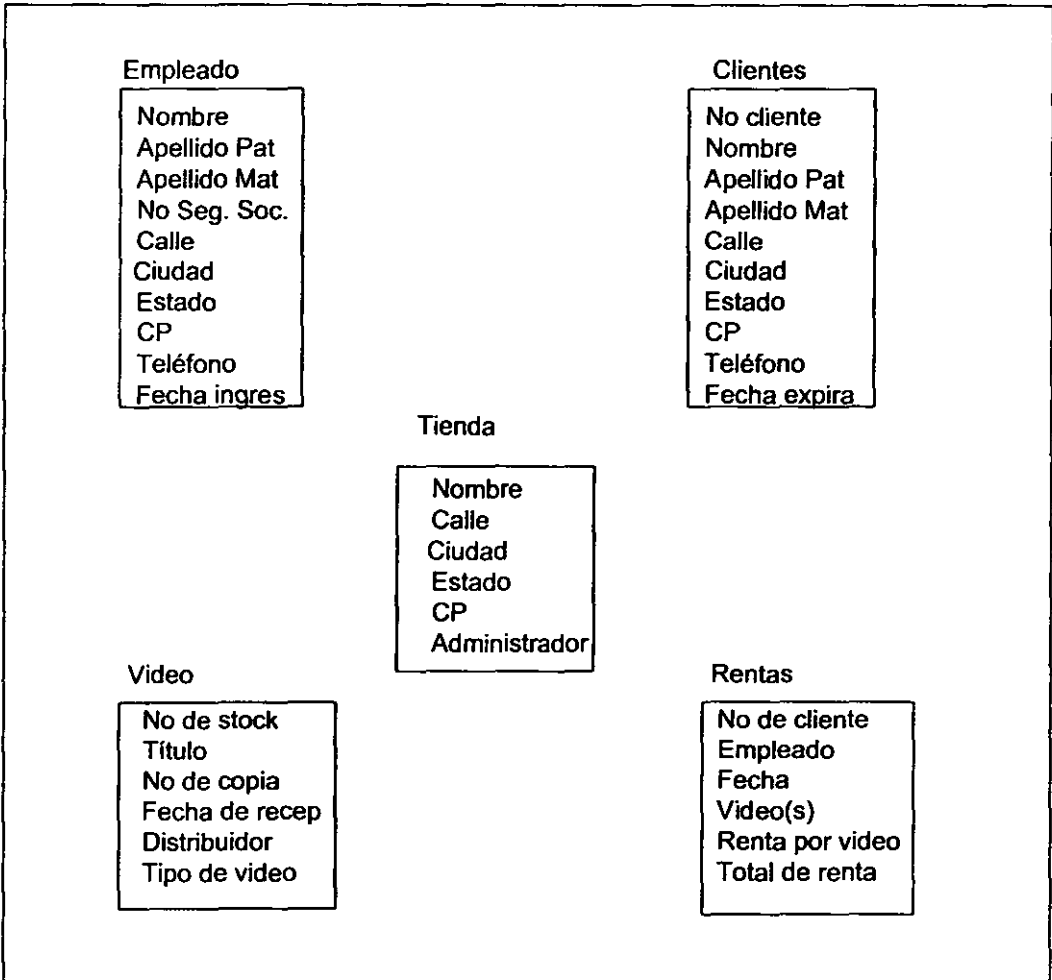


Figura 1.1b Ejemplo de Entidades con Atributos (Sistema de Renta de Videos)

Podemos determinar entonces un registro de entidad a cada elemento y los atributos que lo acompañan. A este tipo de distribución bidimensional de datos se le llama a menudo disposición plana.

Por otra parte un identificador de identidad es aquel atributo que es único a esa entidad y que sirve para reconocerlo, y en ocasiones se requiere más de un atributo para identificar un registro.

Una llave es el atributo o conjunto de estos que se utilizan para reconocer un registro. Una **llave primaria** es la que se utiliza para definir unívocamente un registro, es decir el identificador de entidad formado por uno o más atributos. También se usan llaves que no identifican registros únicos, sino todos aquellos que tienen cierta propiedad, a estas se les llama **llaves secundarias**.

1.1.3 Esquemas y Subesquemas.

Como es preciso describir la organización de los datos de una manera formal. Las descripciones, lógica y física, de la base de datos son indispensables para el software de administración con el fin de extraer de la base los datos que el usuario solicita.

Esquema

Llamamos esquema a la descripción lógica de la base de datos. El esquema es un diagrama de los tipos de datos que se usan, proporciona los nombres de las entidades y sus atributos y especifica las relaciones que existen entre ellos. Es un marco en el que se describen los valores de los datos. Cuando el marco del esquema se completa con los valores de los datos, entonces hablamos de una instancia del esquema.

Los esquemas se representan a menudo en forma de diagramas de bloques. Las líneas que unen ciertos bloques representan relaciones. Estas agregan una información que no es inherente a los datos indicados en el esquema.

Del mismo esquema se pueden derivar muchos subesquemas. Los programadores no tienen que conocer la totalidad del esquema, en cambio, el administrador de datos debe asegurar que los subesquemas que se usan se deriven del esquema principal. El software de administración de datos deriva estos de los subesquemas automáticamente, y los pone a disposición del programa de aplicación.

Ni los esquemas ni los subesquemas reflejan la forma como los datos se almacenan físicamente y hay muchos estilos de organización física para cada organización lógica dada, por lo que existen tres distintas descripciones de datos:

- El subesquema. Es el diagrama de una porción de los datos, orientado a satisfacer uno o más programas de aplicación, esto es, una organización de archivos del programador.
- La descripción lógica global de la base datos o esquema. Esta es la vista general de los datos como los contempla el administrador o los analistas que usan toda la base.
- La descripción de la base de datos. Es el diagrama de la distribución física de los datos en los dispositivos de almacenamiento. La vista de los datos que tienen los programadores y los diseñadores de sistemas que se ocupan del desempeño y de cómo se ubican los datos en el hardware, de cómo se indica o localiza, y de que técnicas de compactación se emplean.

Reglas para dibujar un esquema

- El diagrama debe describir claramente los nombres de registro, de datos y otros.
- Debe quedar clara la agrupación de datos en conjuntos de datos.
- Debe aclararse la distinción entre conjuntos de datos y registros.
- Deben mostrarse los identificadores de registro.
- El diagrama debe indicar claramente qué relaciones son simples y cuales complejas.
- No deben utilizarse nombres duplicados.

Independencia de Datos

Es la habilidad para modificar la definición de un esquema en un nivel sin afectar el esquema. El concepto de independencia de datos es similar en muchos aspectos al concepto de tipos abstractos de datos en los lenguajes modernos de programación. Ambos esconden los detalles de implementación a los usuarios, permitiéndoles concentrarse en la estructura general, en lugar de fijarse en detalles de implementación de bajo nivel. Hay dos niveles de independencia de datos:

- Independencia física de datos. Permite modificar el esquema físico sin tener que reescribir los programas de aplicación. Las modificaciones al nivel físico son necesarias ocasionalmente para mejorar el funcionamiento del sistema.
- Independencia lógica. Permite modificar el esquema conceptual sin que se tenga que alterar el programa de aplicación. Estos cambios a nivel conceptual son necesarios

cuando la estructura lógica de la base de datos es alterada. Esta independencia es más difícil de implementar, ya que los programas de aplicación dependen de la estructura lógica de los datos que accesan.

1.1.4 Abstracción de la información.

La búsqueda de la eficiencia conduce al diseño de estructuras de datos complejas para representar la información en la base de datos.

Niveles de abstracción en los que puede observarse la base de datos.

- **Nivel Físico.** Este es el nivel más bajo de la abstracción, en el que se describe cómo se almacenan realmente los datos. En este nivel se describen en detalle las estructuras de datos complejas del nivel más bajo.
- **Nivel Conceptual.** Este es el siguiente nivel más alto de abstracción, en el que se describe cuales son los datos reales que están almacenados en la base de datos y qué relaciones existen entre los datos. Este nivel contiene toda la base de datos en términos de unas cuantas estructuras relativamente sencillas. Aunque es posible que la implantación de las estructuras simples del nivel conceptual requiera estructuras complejas en el nivel físico, no es forzoso que el usuario del nivel conceptual se dé cuenta de ello. El nivel conceptual de abstracción lo utilizan los administradores de base de datos; quienes deciden qué información se guarda en la base de datos.
- **Nivel de Visión.** Este es el nivel de abstracción más alto, en el cual se describe solamente una parte de la base de datos. Aunque en el nivel conceptual se utilizan estructuras mas simples, todavía queda una forma de complejidad que resulta del gran tamaño de la base de datos. El sistema puede proporcionar muchas vistas diferentes de la misma base de datos.

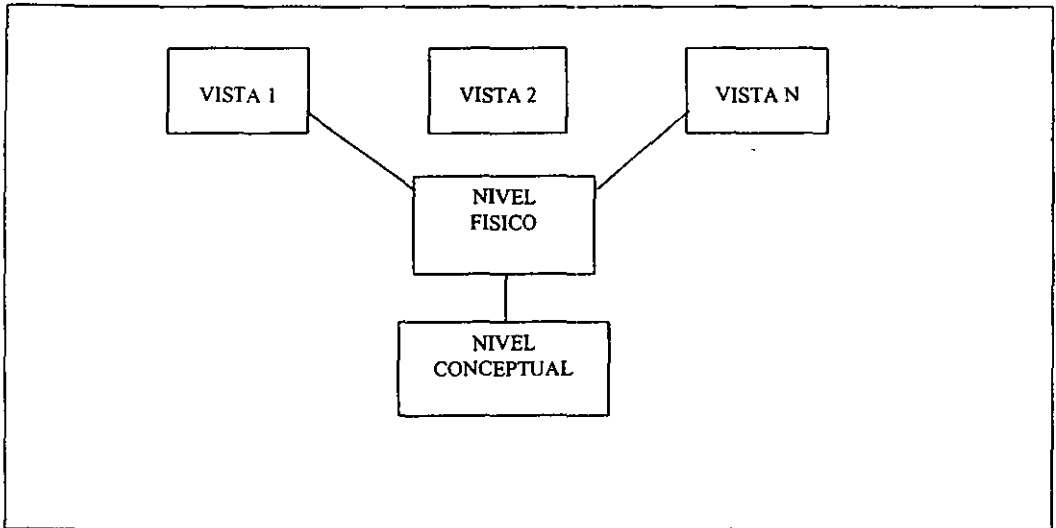


Figura 1.1.c Interrelación de los Niveles de Abstracción de la Información

1.1.5 Modelos de Datos

Para describir la estructura de una base de datos es necesario definir el concepto de modelo de datos. Este es un grupo de herramientas conceptuales para describir los datos, sus relaciones, su semántica y sus limitantes. Se han propuesto varios modelos de datos diferentes, los cuales pueden dividirse en tres grupos: los modelos lógicos basados en objetos y en registros, y los modelos físicos de datos.

Modelos Lógicos Basados en Objetos

Los modelos lógicos basados en objetos se utilizan para describir los datos en los niveles conceptual y de visión. Se caracterizan por el hecho de que permiten una estructuración bastante flexible y hacen posible especificar claramente las limitantes de los datos. Algunos de los más conocidos son:

- El modelo entidad-relación
- El modelo binario
- El modelo semántico de datos
- El modelo infológico

Se ha seleccionado al modelo entidad-relación como representativo de la clase de modelos lógicos basados en objetos. Se eligió éste porque ha tenido bastante aceptación como modelo de datos apropiado para el diseño de bases de datos porque se utiliza ampliamente en la práctica.

Modelos Lógicos Basados en Registros.

Los modelos lógicos basados en registros se utilizan para describir los datos en los niveles conceptual y de visión. A diferencia de los modelos de datos basados en objetos, estos modelos sirven para especificar tanto la estructura lógica general de la base de datos como una descripción en un nivel más alto de la implantación. Sin embargo, no permiten especificar en forma clara las limitantes de los datos.

- **Modelo Relacional.** Los datos y las relaciones entre estos se representan por medio de una serie de tablas, cada una de las cuales tiene varias columnas con nombres únicos.
- **Modelo de Red.** Los datos en el modelo de red se representan por medio de conjuntos de registros y las relaciones entre los datos se representan con ligas, que pueden

considerarse como apuntadores. Los registros de la base de datos se organizan en forma de conjuntos de gráficas arbitrarias.

- **Modelo Jerárquico.** Es similar al modelo de red en cuanto a que los datos y las relaciones entre los datos se representan por medio de registros y ligas, respectivamente, pero difiere del de red en que los registros están organizados como conjuntos de árboles en vez de gráficas arbitrarias.

Modelos Físicos de los Datos

Los modelos físicos de los datos sirven para describir los datos en el nivel más bajo. A diferencia de los modelos lógicos de los datos, son muy pocos los modelos físicos utilizados. Algunos de los más conocidos son:

- El modelo unificador
- La memoria de cuadros

Los modelos físicos de los datos capturan aspectos de la implantación de los sistemas de base de datos.

1.1.6 Estructura General de un Sistema de Base de Datos

Un sistema de base de datos se divide en módulos que se encargan de cada una de las tareas del sistema en general. Algunas de las funciones del sistema de base de datos pueden ser realizadas por el sistema operativo. En la mayor parte de los casos, el sistema operativo proporciona únicamente los servicios más elementales y la base de datos debe partir de ese fundamento. Así, el diseño de la base de datos debe incluir una consideración de la interfaz entre el sistema de base de datos y el sistema operativo.

Un sistema de base de datos consiste en varios componentes funcionales, entre los que se cuentan:

- **Manejador de Archivos.** Controla la distribución de espacio de almacenamiento en disco y las estructuras de datos usadas para representar la información almacenada en el disco.
- **Manejador de Base de Datos.** Suministra la interfaz entre los datos de bajo nivel almacenados en la base de datos y los programas de aplicación y queries que se envían al sistema.
- **Procesador de consultas.** Traduce instrucciones de un lenguaje de query a instrucciones de bajo nivel que sean entendidas por el manejador de la base de datos. En adición, el procesador de queries intenta transformar los pedidos del usuario a una forma equivalente pero más eficiente buscando una buena estrategia para ejecutar el query.
- **Precompilador DML.** Convierte las instrucciones de DML incluidas en un programa de aplicación a llamadas normales de procedimientos en el lenguaje de programación. El precompilador debe interactuar con el procesador de consulta para generar el código apropiado
- **Compilador DDL.** Transforma las instrucciones de DDL a un conjunto de tablas que contienen metadatos. Estas tablas son almacenadas en el diccionario de datos.

Adicionalmente se requieren también varias estructuras de datos como parte de la implementación del sistema físico, que incluyen:

- **Archivos de Datos.** Los cuales almacenan la base de datos.

- **Diccionario de Datos.** Almacena información sobre la estructura de la base de datos. El diccionario de datos es usado frecuentemente, por los componentes funcionales por esto se debe dar gran énfasis al logro de un buen diseño y una implementación eficiente del diccionario.
- **Indices.** Proporcionan un acceso rápido a los datos con valores particulares.

Lenguaje de Definición de Datos

Un esquema de base de datos se especifica por medio de una serie de definiciones que se expresan en un lenguaje de definición de datos (DDL, Data Definition Language). El resultado de la compilación de las proposiciones en DDL es un conjunto de tablas que se almacenan en un archivo especial llamado diccionario de datos.

Un diccionario de datos es un archivo que contienen, "datos acerca de los datos". Este archivo se consulta antes de leer o modificar los datos reales en el sistema de base de datos.

La estructura de almacenamiento y los métodos de acceso empleados por el sistema de base de datos se especifican por medio de un conjunto de definiciones de un tipo especial de DDL llamado lenguaje de almacenamiento y definición de los datos. El resultado de la compilación de estas definiciones es una serie de instrucciones que especifican los detalles de implantación de los esquemas de datos que normalmente no pueden ver los usuarios.

Lenguaje de Manejo de Datos.

El manejo de los datos consiste en:

- La recuperación de información almacenada en la base de datos.

- La inserción de la información nueva en la base de datos.
- La eliminación de información de la base de datos.

En el nivel físico, deben definirse algoritmos que permitan tener acceso a los datos en forma eficiente. En los niveles de abstracción más altos lo importante es la facilidad de uso. El objetivo es lograr una interacción eficiente entre las personas y el sistema.

Un lenguaje de manejo de datos (DML, Data Manipulation Language) permite a los usuarios manejar o tener acceso a los datos que estén organizados por medio del modelo apropiado. Existen básicamente dos tipos de DML.

- De procedimientos, necesitan que el usuario especifique cuáles datos quiere y cómo deben obtenerse.
- Sin procedimientos, requieren que el usuario especifique cuales datos quiere sin especificar como obtenerlos.

Una consulta es una proposición que solicita la recuperación de información. La parte de un DML que implica la recuperación de información se conoce como lenguaje de consultas o lenguaje de manejo de datos como sinónimos.

Manejador de Base de datos.

Generalmente las bases de datos requieren una gran cantidad de espacio de almacenamiento. Las bases de datos de las empresas comúnmente se miden en términos de gigabytes (mil millones de bytes) de información. Puesto que la memoria principal de la computadora no puede almacenar esta información, se guarda en discos. Los datos se

transfieren entre el almacenamiento en disco y la memoria principal, según se requiera. Ya que el movimiento de los datos del disco y al disco es lento comparado con la velocidad de la unidad central de procesamiento de las computadoras, es imperativo que el sistema de base de datos estructure la información de tal manera que se reduzca la necesidad de transferir datos entre el disco y la memoria principal.

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

El manejador de base de datos es responsable de las siguientes tareas:

- **Interacción con el manejador de archivos.** Los datos sin procesar se almacenan en el disco mediante el sistema de archivos proporcionado normalmente por un sistema operativo convencional. El manejador de base de datos traduce las diferentes proposiciones en DML a comandos de sistema de archivos de bajo nivel. Así, el manejador de base de datos se encarga realmente del almacenamiento, recuperación y actualización de los datos en la base de datos.
- **Implantación de la integridad.** Los valores de los datos almacenados en la base de datos deben satisfacer ciertos tipos de limitantes de consistencia.
- **Puesta en práctica de la seguridad.** No es preciso que todos los usuarios de la base de datos tengan acceso a todo su contenido. Es labor del manejador de la base de datos hacer que se cumplan estos requisitos de seguridad.

- **Respaldo y recuperación.** Un sistema de cómputo, como cualquier otro dispositivo mecánico o eléctrico, está sujeto a fallas. Existen muy diversas causas de estas fallas, entre ellas el aterrizaje de las cabezas lectoras del disco duro, la interrupción del suministro de energía y los errores de software. En cada uno de estos casos se pierde información de la base de datos. Es responsabilidad del manejador de la base de datos detectar estas fallas y restaurar la base de datos al estado que existía antes de presentarse la falla. Esto se logra normalmente iniciando diversos procedimientos de respaldo y recuperación.
- **Control de concurrencia.** Cuando varios usuarios actualizan la base de datos en forma concurrente, es posible que se conserve la consistencia de los datos. Es necesario que el sistema controle la interacción entre los usuarios concurrentes; lograr dicho control es una de las tareas del manejador de la base de datos.

Administrador de Base de Datos

Una de las principales razones por las cuales es necesario contar con un sistema de manejo de bases de datos es tener un control central de los datos y los programas que los accesan. La persona que lleva el control del sistema es el administrador de la base de datos (DBA, Data Base Administrator). Las funciones de éste son las siguientes:

- **Definición de esquema.** La creación del esquema original de la base de datos. Esto se lleva a cabo escribiendo un conjunto de definiciones que son traducidas por el compilador de DDL a un conjunto de tablas que son almacenadas permanentemente en el diccionario de datos.
- **Estructura de almacenamiento y definición de métodos de acceso.** La creación de estructuras adecuadas de almacenamiento y métodos de acceso. Esto se logra escribiendo un conjunto de definiciones que son traducidas por el compilador.

- **Modificación de la organización física y el esquema** . Estos cambios se llevan a cabo escribiendo un conjunto de definiciones que son usadas por el compilador de DDL o el compilador de definición del lenguaje para generar modificaciones a las tablas internas apropiadas.
- **Permitir la autorización para el acceso de datos**. Diferentes niveles de acceso a los diferentes usuarios. Esto le permite regular qué partes de la base pueden acceder los diferentes usuarios.
- **Especificación de las restricciones de Integridad**. Estas restricciones se mantienen en una estructura especial del sistema que son consultadas por el manejador cuando se hace una actualización.

Usuario de la Base de Datos

Uno de los objetivos de los sistemas de bases de datos es el de proporcionar un ambiente para recuperar y almacenar información en la base de datos. Hay tres tipos diferentes de usuarios de este sistema, y se diferencian por la manera en que se espera que interactúen con el sistema:

- **Programadores de Aplicación**. Son los profesionales de la computación, los cuales interactúan con el sistema a través de llamadas de DML, las mismas que están escritas en algún lenguaje. Estos programas son referidos comúnmente como programas de aplicación. Como la sintaxis de DML es generalmente diferente del lenguaje, por lo regular estas llamadas están precedidas de un caracter especial, de modo que el código apropiado pueda ser generado por un preprocesador llamado precompilador de DML. Este convierte las instrucciones de DML a llamadas de procedimientos normales en el lenguaje de programación. El programa resultante se corre entonces con el compilador del mismo, el cual genera el código objeto apropiado. Existen tipos especiales de lenguajes

de programación los cuales combinan estructuras de control para el manejo de objetos de bases de datos.

- **Usuario Final.** Estos son los usuarios que interactúan con el sistema sin escribir programas, actualizan los datos y validan que se hayan registrado en forma correcta. Sus consultas son atendidas a través de pantallas y reportes predefinidos, cuando ha adquirido cierta experiencia es capaz de formular un **query (consulta)**.
- **Usuarios Especializados.** Estos usuarios escriben aplicaciones que no encajan en la estructura tradicional de procesamiento de datos. Entre estas aplicaciones están los sistemas de CAD, sistemas expertos que almacenan tipos de datos complejos (por ejemplo audio y gráficos), y sistemas de modelaje de ambiente.

1.1.7 Bases de Datos En Ambientes Multiusuarios

Para entender mejor lo que es un ambiente de bases de datos multiusuario podríamos imaginarnos que estamos escribiendo un libro en colaboración con otra persona, donde cada uno es responsable de hacer ciertos capítulos. Como resultado de este trabajo "concurrente", el trabajo avanza rápidamente. Trabajando de esta manera, podemos leer los capítulos del otro, pero no podemos editar o escribir en el capítulo que no nos corresponde, ya que de hacerlo así, el resultado sería un caos y un manuscrito que tal vez quedaría incompleto. Cuando necesitemos generar un índice, la otra persona debe dejar de escribir y darnos control absoluto del manuscrito hasta que la generación del índice haya terminado.

Si comparamos este ejemplo con una base de datos multiusuario, el libro sería la base, los capítulos serían los registros, y nosotros los usuarios.

Manejo de las Bases de Datos Multiusuarios

Para los sistemas de Bases de Datos Multiusuarios los DBMSs proporcionan un balance entre dos objetivos conflictivos: la concurrencia y la integridad de los datos. Asegurar la integridad de los datos implicaría bloquear (lock) los archivos y los registros, lo cual reduce la habilidad de trabajar concurrentemente.

A diferencia de las bases de datos de un solo usuario, un aplicación multiusuario debe estar preparada para manejar las situaciones en las cuales un archivo no esté disponible cuando se le necesita, y en el cual un archivo puede contener datos modificados la próxima vez que se le necesite. Al convertir las aplicaciones de un solo usuario a aplicaciones multiusuario, el principal objetivo será el de maximizar la concurrencia tratando de mantener al mínimo el nivel de bloqueo de archivos que se requieran, y solamente bloquearlos mientras dure la operación.

Solamente el usuario que ha bloqueado un archivo puede escribir en él. Otros usuarios solo pueden leer los registros o archivos. Un archivo bloqueado puede ser accesado de nuevo dándole un comando para desbloquearlo (unlock). Este comando deberá liberar todos los bloqueos que se encuentren en el archivo. Los bloqueos deben desaparecer también cuando se cierre el archivo o cuando el programa termine normalmente.

Procesamiento Alternado

En el ambiente multiusuario, es común que varios usuarios necesiten usar un dato al mismo tiempo, por lo cual un programa multiusuario necesita saber que hacer cuando un usuario no puede obtener el bloqueo deseado sobre un recurso. El programa podría seguir intentándolo hasta que se produzca una interrupción, o bien deberá reintentar un cierto número de veces antes de reportarlo al usuario. La aplicación debe incluir un procedimiento de error para manejar actividades de procesamiento alterno cuando un usuario no pueda acceder el archivo.

Control de Concurrencia.

Los sistemas multiusuarios bloquean los archivos y registros para prevenir que dos o más usuarios actualicen el mismo dato al mismo tiempo, lo cual causaría actualizaciones perdidas. En una red típica, los pedidos de datos del usuario requieren que se copie el archivo de disco a la memoria de la estación de trabajo. Esto puede causar actualizaciones erróneas aún si los archivos tienen bloqueos, porque varios usuarios pueden tener copias de los mismos datos en la memoria de sus estaciones de trabajo. Supongamos que dos usuarios tienen un archivo en sus estaciones. Un usuario modifica los datos y los vuelve a escribir a disco, pero el otro usuario puede actualizar la copia anterior de los datos que aún está en su memoria. Cuando este usuario lo escriba nuevamente a disco, éste va a reemplazar la actualización anterior. Algunos de los métodos utilizados para manejar las actualizaciones concurrentes son:

- La apertura de archivos exclusivos. Este es el método más sencillo para controlar la concurrencia de actualizaciones y consiste en abrir los archivos involucrados en la actualización para uso exclusivo mientras dure la operación de lectura/escritura. Esto impide que otros usuarios puedan obtener copias de los archivos mientras estén ocurriendo las actualizaciones. Este método controla la concurrencia efectivamente pero restringe el acceso de otros usuarios a archivos enteros por períodos largos e innecesarios. Por esta razón es altamente ineficiente.
- Marcar los archivos en uso. Como parte de una aplicación se puede crear un archivo maestro que almacene en cada registro la llave primaria y una bandera que indique si un registro está en uso, se niega el acceso de otros usuarios a éste. Esto es más eficiente que el uso de archivos exclusivos, ya que solamente pone bloqueo a los registros en uso y permite que otros usuarios accedan los registros restantes. Su inconveniente es la complejidad de programación, especialmente para una aplicación existente, porque cada pedido de acceso a un registro debe llamar primero a una rutina

que revise la disponibilidad del registro. Si un registro está disponible, la bandera debe ser actualizada para indicar que está en uso, cuando se libera un registro, debe actualizarse nuevamente la bandera.

- **Uso de semáforos.** Se puede implementar una función para controlar el acceso de registros basados en sus llaves primarias. Un semáforo es un mecanismo que asocia una etiqueta definida por el usuario (representando un recurso) con un contador que puede tomar un valor de 0 a 127. Se usa para controlar acceso a un recurso. Se definen operaciones de **OpenSemaphore (abrir semáforo)**, **ExamineSemaphore (examinar semáforo)**, **WaitOnSemaphore (esperar semáforo)**, **SignalSemaphore (señalar semáforo)** y **CloseSemaphore (cerrar semáforo)**.

Cada semáforo tiene un contador abierto asociado a él. Un contador abierto es el número de procesos que tiene un semáforo abierto, es decir el número de aplicaciones que ha usado la operación **OpenSemaphore** y que todavía no han llamado la de **CloseSemaphore**. La información de contador de apertura tiene poco uso, ya que solamente indica el número de procesos que pueden pedir acceso a los recursos.

El valor del semáforo es el número de recursos disponibles. Este número es puesto inicialmente por el primer proceso que llama a **OpenSemaphore** para un recurso particular. El valor del semáforo es mantenido por el sistema, y es el elemento crucial para controlar el acceso a un recurso. Cuando el valor se hace negativo, el proceso debe esperar a que se libere el registro, es decir debe esperar a que uno o más recursos sean liberados con la instrucción **SignalSemaphore**.

Una aplicación que use semáforos para controlar el acceso concurrente debe llamar a **OpenSemaphore** con la llave primaria como etiqueta, y un contador inicial de 1. Debe llamar a **WaitOnSemaphore** cuando esté listo para acceder un registro; si el recurso no está disponible, la operación de **WaitOnSemaphore** pondrá al programa en cola hasta que el recurso esté disponible o a que expire el tiempo límite. Cuando una aplicación deje de usar un registro, debe llamar a **SignalSemaphore** para incrementar el contador

y permitir a otro usuario acceder el recurso. Finalmente la aplicación debe llamar a CloseSemaphore antes de terminar.

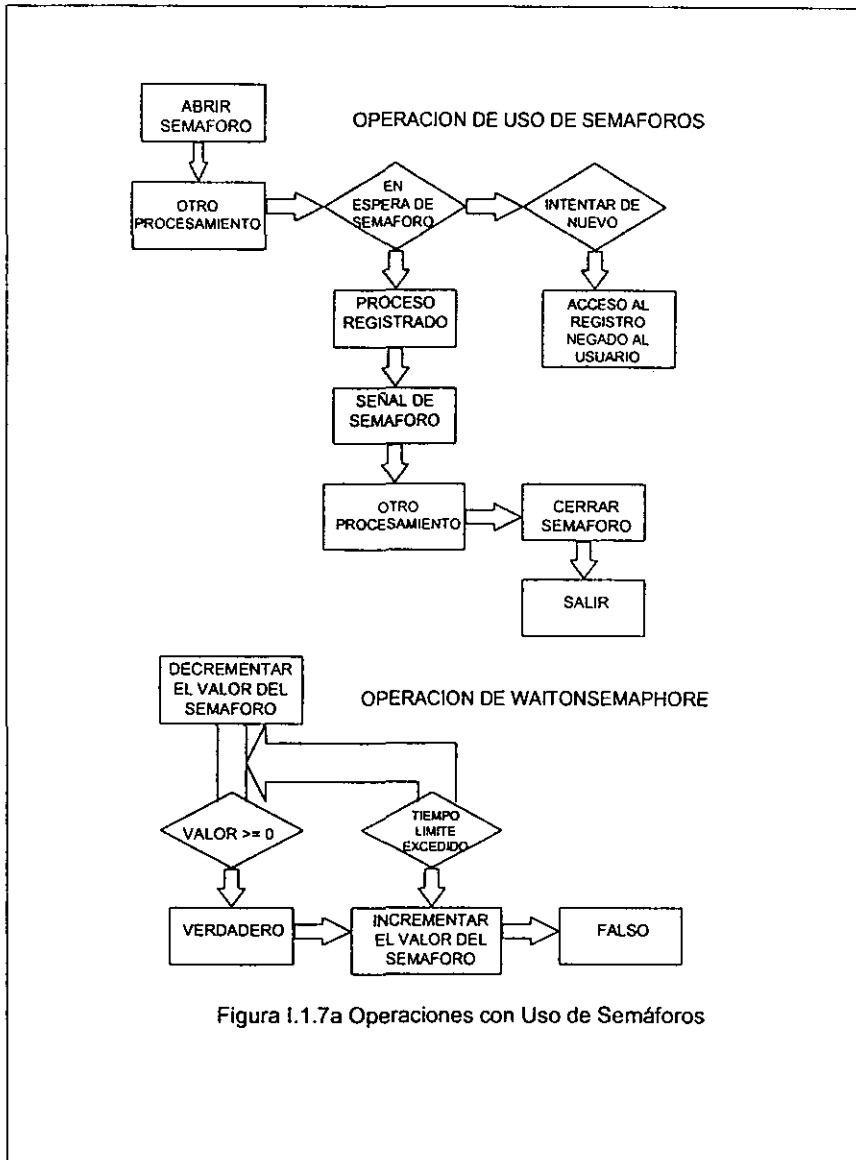


Figura 1.1.7a Operaciones con Uso de Semáforos

- Comparar los valores antes y después en los arreglos. Se pueden implementar procedimientos para verificar que un registro no haya sido modificado por otro usuario entre el momento en que el registro es subido a la memoria de la estación de trabajo y el momento en que se actualiza. Esto es posible en productos que manejen arreglos.
- En el momento en que el registro es leído (arreglo base), su contenido debe ser copiado en un arreglo (arreglo copia). Luego se copia el registro Base a un archivo temporal en el cual un usuario puede modificar los datos (arreglo de trabajo). Antes de escribir el registro nuevamente (arreglo de trabajo) se compara el arreglo copia con el arreglo base. Si hay diferencias, es porque el registro base se cambió entre el momento en que el usuario lo tomó y lo volvió a escribir. La edición que completó el último usuario queda desactualizada y necesita volverse a capturar.

Prevención de Estancamientos.

A menos de que se tenga cuidado, una aplicación multiusuario que lleva a cabo operaciones de bloqueo de registros y archivos podría causar un problema de contención de recursos llamado estancamiento (**deadlock**).

En el ejemplo mostrado a continuación (Figura 1.1.9.b) El proceso **A** mantiene bloqueado el registro **x**, y desea bloquear un segundo registro, el registro **y**. El renglón **y** se encuentra actualmente bloqueado por otro usuario, que está ejecutando el proceso **B**. El proceso **A** estará esperando a que el proceso **B** libere el registro **y**, pero el proceso **B** mantendrá el bloqueo sobre el registro **y** hasta que el proceso **A** libere el registro **x**. En otras palabras **A** está esperando por **B** y **B** está esperando por **A**. A esta situación se le conoce como **deadlock**; cuando esto ocurre los procesos esperarán indefinidamente a menos que el estancamiento sea detectado y resuelto por el sistema

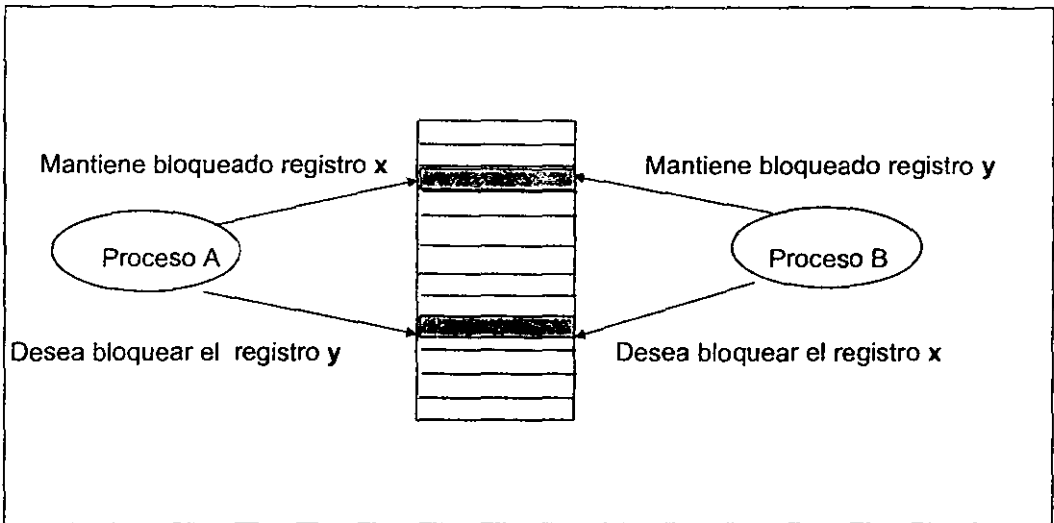


Figura 1.1.7b Situación de Estancamiento

Algunas opciones para prevenir estos estancamientos son:

- **Abrir archivos en un orden preestablecido.** Una aplicación puede adoptar un estándar que pida que todos los archivos de la base sean abiertos y que se les ponga un bloqueo en un orden predeterminado, sin importar el orden en el que serán usados. Una desventaja de esta opción es que requiere que la aplicación abra los archivos en un orden que probablemente no vaya con sus necesidades. El diseño de la aplicación puede ser ineficiente debido al requerimiento de que los archivos sean puestos en bloqueo de manera secuencial, además, algunos archivos pueden permanecer en bloqueo por periodos prolongados si son abiertos mucho antes de que sean necesitados.
- **Preasignación de archivos en áreas de trabajo específicas.** Esta opción también requiere que se creen estándares en la aplicación. Cada archivo debe ser abierto solo

en áreas de trabajo preasignadas, las cuales deben ser seleccionadas en secuencias numéricas.

Esta alternativa tiene varias desventajas. En primer lugar no es apta para aplicaciones grandes que usen un gran número de archivos que excedan el número máximo de áreas de trabajo. En segundo lugar, el estándar puede ser más difícil de aplicar que la apertura de archivos en orden preestablecido. Cada programador debe tener acceso a una lista que indique el área de trabajo asignada a cada archivo en el sistema. Mantener este sistema puede ser un verdadero problema cuando se tengan que cambiar las áreas preasignadas o cuando se agreguen nuevos archivos. Finalmente, este método evita los estancamientos cuando se trabaja con archivos, pero no con registros.

- **Poner bloqueo a los recursos necesarios desde la inicialización del programa.** Otra alternativa para evitar los estancamientos es hacer que cada programa de aplicación ponga un bloqueo a todos los archivos o registros que va a necesitar al comienzo de cada procedimiento. Si no se puede lograr el bloqueo en todos los archivos/registros, el programa debe liberar los otros. Esto evita los estancamientos resultantes de poner bloqueo a los archivos y registros porque un programa no esperará a que estén disponibles.

Esta solución obliga a la aplicación a saber que recursos va a necesitar, lo cual puede presentar problemas si el procesamiento varía dependiendo de los datos proporcionados por los usuarios. También hace el mantenimiento más difícil. Esta solución tiene el potencial de monopolizar recursos porque estos son puestos en bloqueo mientras dure el proceso. Si se requieren muchos recursos, un programa puede esperar mucho para que todos los recursos estén disponibles simultáneamente.

- **Implementar servicios de sincronización.** Este esquema se centra en una tabla que mantiene una lista de todos los registros a los que se va poner bloqueo. Una aplicación

puede requerir que al conjunto de archivos contenidos en la tabla se les ponga bloqueo, de modo que todo el conjunto de archivos quede con bloqueo, o ninguno de ellos. La aplicación debe liberar los archivos antes de que termine. Este concepto es útil porque el conjunto de archivos puede ser liberado pero mantenido en la tabla para uso posterior. Esto se puede hacer físicamente especificando los registros en un archivo DOS con el offset de inicio y la longitud del registro. Los servicios de bloqueo lógico presentan una solución más razonable, realizando un seguimiento de los bloqueos lógicos colocados en los nombres de los registros. Esto coordina el acceso multiusuario a los datos compartidos sin poner bloqueos físicos. Las aplicaciones necesitan un mayor grado de control interno para implementar efectivamente el bloqueo lógico. La clave de esto es establecer un estándar para asignar nombres a los registros a los que se va a poner bloqueo.

Borrado de registros en una red

El borrado de registros en una sistema multiusuario presenta otros problemas. Cuando se borren registros, se debe planear empacar el archivo para remover los registros permanentemente, recapturando el espacio que ocupan. Como esta operación incluye el uso exclusivo del archivo, presenta un problema de concurrencia en un ambiente multiusuario en el cual los usuarios frecuentemente requieren acceso a varios archivos.

En el momento en que se está empacando un archivo, los usuarios no tienen acceso a éste, por lo tanto, se necesita una mejor alternativa. Una opción sería la de reciclar primero un registro, es decir ponerlo en blanco, y luego marcarlo como borrado. Cuando la aplicación necesite agregar un registro, puede "desborrar" el borrado y regresarlo a la aplicación como un registro nuevo. Cuando se acaben estos registros reciclados se necesitarán más registros nuevos. Lo que se recomienda en estos casos es anexar un lote de registros en blanco, y marcarlos como borrados, exceptuando los que se van a regresar a la aplicación.

1.1.8 El Enfoque Relacional

Un sistema de Base de Datos debe ser capaz de representar y manipular entidades (registros o segmentos) y sus relaciones de manera fácil y conveniente. En el enfoque jerárquico o de árbol, se representa la relación entre dos segmentos por la posición relativa de arriba hacia abajo y de derecha a izquierda de los tipos de segmentos involucrados. En el enfoque de red, las relaciones se representan mediante mecanismos de set (conjunto) con uso de apuntadores, los cuales enlazan a un tipo de registro propietario con un tipo de registro miembro. Para el caso de una Base de Datos grande y complicada, el modelo lógico y la manera como pueden tener acceso a él los usuarios, por medio de un lenguaje de manipulación de datos (DML) pueden volverse muy complejos. Aún más, el acceso a los datos se inclina demasiado a las rutas de acceso, en término de los enlaces o posiciones jerárquicas que estableció el diseñador. De esta manera puede ocurrir que muchos cambios a la Base de Datos violen la independencia de los datos o afecten los programas de aplicación.

El enfoque relacional a Base de Datos concebido por E.F. Codd, constituye un enfoque muy diferente para la descripción y manipulación "lógica" de los datos. Se esfuerza por evitar muchas de las desventajas que se han mencionado. En forma concisa, visualiza la Base de Datos lógica como una simple colección de tablas bidimensionales llamadas "Relaciones". Estas tablas son planas, en el sentido de que no hay grupos repetitivos. Los usuarios las comprenden y manejan fácilmente con poco o ningún entrenamiento en programación, y no implican consideración alguna sobre aspectos posicionables, de apuntadores o de rutas de acceso.

1.1.8.1 El Modelo Entidad- Relación.

El modelo entidad relación (E-R) está basado en una percepción del mundo real, que consiste en un conjunto de objetos básicos llamados "entidades" y las "relaciones" entre estos objetos. Fué desarrollado para facilitar el diseño de las bases de datos permitiendo la especificación de un esquema. Tal esquema representa la estructura lógica de la base de datos.

Entidades y Conjuntos de Entidades

Una entidad es un objeto que existe y puede distinguirse de otros objetos. Un conjunto de entidades es un grupo de entidades del mismo tipo. No es necesario que las entidades pertenezcan exclusivamente a un solo conjunto. Una entidad está representada por un conjunto de atributos. Un conjunto de relaciones es un grupo de relaciones del mismo tipo. Formalmente es una relación matemática de $n \geq 2$ (posiblemente idénticos) conjuntos de entidades.

Tipos de Relaciones

La relación que existe entre dos conjuntos de datos puede ser simple o compleja. Por simple queremos decir que hay una correspondencia biunívoca (uno a uno) entre los datos. Cuando a cada dato le corresponden más de uno del otro conjunto, entonces la correspondencia es compleja.

- Relación uno a uno (1:1). Una entidad en A está asociada, cuando más, con una entidad en B, y una entidad en B está asociada cuando más con una entidad A.

- Relación uno a muchos (1:N). Una entidad en A está asociada con cualquier número de entidades en B. Una entidad en B, sin embargo, puede estar asociada, cuando más con una entidad en A.
- Relación muchos a uno (N:1). Una entidad en A está asociada, cuando más con una entidad en B. Sin embargo, una entidad en B, puede estar asociada con cualquier número de entidades en A.
- Relación muchos a muchos (N:M). Una entidad en A está asociada con cualquier número de entidades en B y una entidad en B está asociada con cualquier número de entidades en A.

La relación cardinal para un conjunto de relación obviamente depende del mundo real para el que sea modelado por el conjunto de relación.

Las dependencias de existencia constituyen una limitante importante; específicamente, si la existencia de la entidad A depende de la existencia de la entidad B, entonces se dice que A es dependiente por existencia de B. Funcionalmente, esto quiere decir que si se elimina B, también se eliminará A. Se dice que la entidad B es una entidad dominante y que A es una entidad subordinada.

Dentro de la modelación de base de datos es importante especificar cómo se van a distinguir las entidades y las relaciones. Conceptualmente, las entidades individuales y las relaciones son distintas entre sí, pero desde el punto de vista de una base de datos la diferencia entre ellas debe expresarse en términos de sus atributos.

Es posible que existan varios conjuntos de atributos distintos que pudieran servir como llaves candidato.

Se utilizará el término llave primaria para referirse a la llave candidato que elija el diseñador de la base de datos como la forma principal de identificar a las entidades dentro de un conjunto de éstas.

Los conjuntos de relaciones también tienen llaves primarias. Sus llaves primarias se forman tomando todos los atributos que constituyen las llaves primarias de los conjuntos de entidades que definen al conjunto de relaciones.

Diagrama Entidad-Relación

El modelo Entidad-Relación, está asociado con su respectivo diagrama Entidad-Relación, para representar las entidades y las relaciones entre éstas. Esta es la mejor manera de expresar la vista global de la base de datos. Las entidades están representadas por un rectángulo y una relación se representa con un rombo.

Las entidades y relaciones poseen propiedades. Estas propiedades pueden expresarse en términos del valor de sus atributos. Por lo tanto, opcionalmente, estas pueden representarse dentro del Diagrama de Entidad-Relación mediante círculos y los valores de estos se expresan en los arcos que van de la entidad al respectivo atributo.

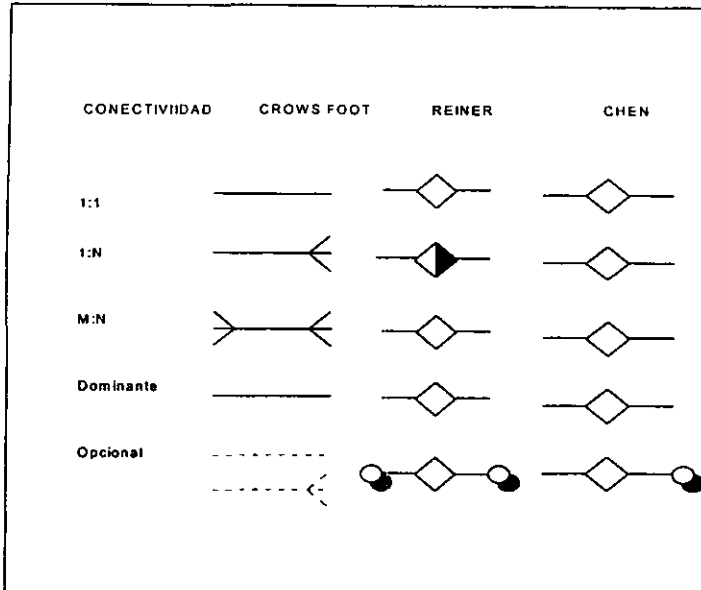


Figura 1.1.8.1 Algunos Estilos utilizados en el Diagrama E-R

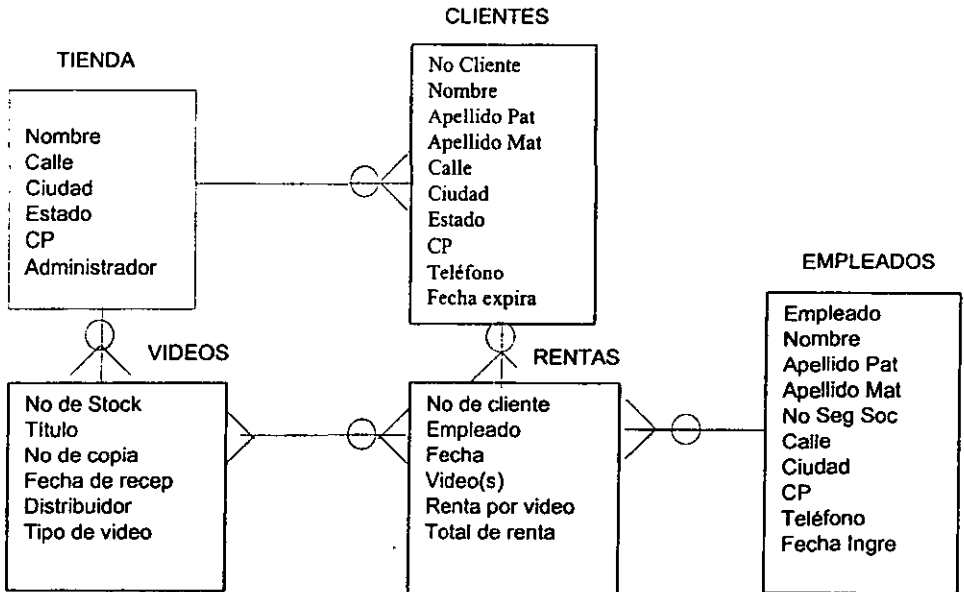


Figura 1.1.8.1b Ejemplo de Diagrama E-R (Sistema de Renta de Videos)

I.1.8.2 El Modelo Relacional

El modelo relacional es en la actualidad el más popular en los sistemas de manejo de una base de datos, puesto que es conceptualmente sencillo y comprensible por los profesionales de los sistemas de información y muchos otros usuarios finales; puede evolucionar, ya que las relaciones entre los datos no necesitan estar predefinidas, además utiliza valores de los datos para implicar las relaciones. El modelo relacional de datos, desarrollado en 1970 por E.F. Codd, como ya se mencionó se basa en una relación (tabla bidimensional). Los renglones de la tabla representan los registros y las columnas muestran los atributos de la entidad (Fig. I.1.8.2a). Las **bases de datos relacionales** utilizan un modelo para mostrar cómo se relacionan lógicamente los datos de un registro.

El orden de los datos en la tabla no es significativo y tampoco implica un orden cuando los registros están incluidos en la relación. Análogamente, los detalles físicos de almacenamiento (ya sea una organización aleatoria, indexada o secuencial) no son de interés para el analista. Las tablas relacionales muestran las relaciones lógicas, no físicas.

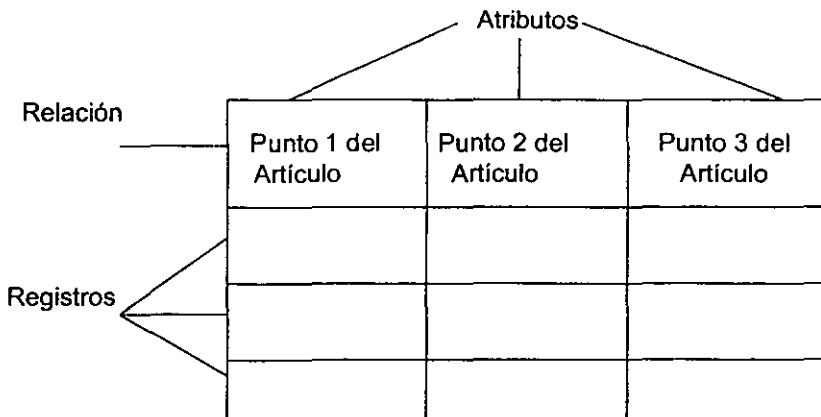


Figura 1.1.8.2a Componentes de Relación

Al hacer una solicitud de información, el sistema produce una tabla que contenga la información. En un ejemplo de un sistema de una lavandería, si se desea determinar quién utiliza sábanas, el sistema producirá una tabla que contenga los nombres de todos los usuarios de sábanas.

Los datos de la figura 1.1.8.2b se muestran en forma relacional para la entidad pedido. Si queremos saber los nombres de todos los clientes que hicieron pedidos el primero de diciembre, podemos rastrear la relación de pedidos, observando el orden de los datos. Lo atractivo de las bases de datos relacionales es que el modelo relacional se puede comprender rápidamente.

Estructura de Registro					Grupo de Repetición				
No Pedido	No Cliente	Nombre del Cliente	Dirección del Cliente	Fecha solicitud	No Artículo	Descripción del Artículo	Precio del Artículo	Cantidad Solicitada	Costo Total
101456	812	Harper University	Rosboro, NY 13514	12/1	T101	Manteles	.95	100	185.40
					B16	Cobijas	1.55	50	
					C118	Sábanas	.29	30	
					B14	Toallas	.42	10	
102721	1319	McCraw Manufact	Endicott, NY 13760	12/1	C118	Sábanas	.29	20	5.80
103654	2107	Union Hospital	6th Avenue, NY 10021	12/2	B14	Toallas	.42	60	118.20
					B16	Cobijas	1.55	60	
105489	824	Hara Enterprises	Buffalo, NY 13240	11/30	N38	Servilletas	.84	100	162.50
					T101	Manteles	.95	50	
					B16	Cobijas	1.55	20	

Figura 1.1.8.2b Estructura de registro no normalizada para los datos de los pedidos

Definición de una base de datos. En una base de datos relacional, un esquema, el cual describe la base de datos que utilizará un sistema, se comunica con el DBMS. Según el DBMS que se esté usando, las instrucciones pueden variar. La mayoría de las bases de datos relacionales utilizan un lenguaje de definición de una base de datos, como SQL.

Para crear una tabla para una relación, las instrucciones de definición de datos nombran a la relación de forma que se pueda añadir al directorio de la base de datos. Entonces, cada dato de la relación debe definirse describiendo el dato, tipo de dato y longitud. Los campos para crear la relación PEDIDO son éstas.

```
CREATE TABLE PEDIDOS
( NUMERO_DE_PEDIDO INTEGER,
  NUMERO_CLIENTE CHAR(24),
  NOMBRE_CLIENTE CHAR(24),
  DIRECCION_CLIENTE CHAR(24),
  FECHA_DEL_PEDIDO CHAR(24),
  NUMERO_ARTICULO CHAR(10),
  DESCRIPCION_ART CHAR(24),
  PRECIO_POR_ART INTEGER
  CANTIDAD_SOLIC INTEGER,
  COSTO_TOTAL INTEGER)
```

El comando anterior crea la relación pero no agrega datos. Esto se hace mediante una aplicación que acepta datos, posiblemente desde el teclado de una estación de trabajo en un sistema en línea y los agrega en la tabla. Para alimentar una base de datos, se utilizan instrucciones equivalentes a la siguiente:

```
INSERT INTO PEDIDOS VALUES ('101456','812','UNIVERSIDAD DE
HARPER','33','ROSBORO, NY 13514','12/1','T101','MANTELES','.95','100','185.40')
```

Usualmente, este enunciado estará dentro de un programa hecho en un lenguaje huésped. De esta manera se insertarán datos en la tabla.

El modelo de datos relacional representa, como ya se mencionó, a la base de datos como una colección de tablas. Aunque las tablas son simples y de noción intuitiva, están en correspondencia directa entre el concepto de tabla y el concepto matemático de relación.

En años posteriores a la introducción del modelo relacional, se desarrolló una teoría para bases de datos relacionales. Esta teoría asiste en el diseño de bases de datos relacionales y en el procesamiento eficiente de los requerimientos de información de los usuarios desde la base de datos.

Una base de datos relacional consiste en una colección de tablas, a cada una de las cuales se le asigna un nombre único, Cada tabla tiene una estructura similar a las representadas en el modelo Entidad-Relación. Un renglón en una tabla representa una relación entre un conjunto de valores. Dado que una tabla es una colección de tales relaciones, encierra el concepto de tabla y el concepto matemático de relación, del cual el modelo de datos relacional toma su nombre.

En términos menos formales, una relación es una tabla bidimensional de n columnas constituidas por un conjunto de tuplos de n elementos (n -tuplos). Cada una de las columnas en una relación es un conjunto de valores de elementos de datos (tipo de atributo o campo) al que se le denomina el dominio.

Observe la siguiente tabla de videos:

NoVideo	Título	No_copias	Fecha_recep	Distribuidor	Tipo_Video
54213	La Cenicienta	6/7	12/06/96	Producciones A	A
26458	Batman	1/3	10/02/95	Producciones B	B
23641	El Exorcista	4/8	03/04/94	Producciones C	C

En esta tabla se encuentran seis atributos: NoVideo, Título, No_copias, Fecha_recep, Distribuidor, tipo_video. Para cada atributo hay un conjunto de valores permitidos, llamado dominio de un atributo. Por ejemplo para el atributo NoVideo, el dominio debe ser el conjunto de todos los números de video. Denotemos a este conjunto con D_1 , D_2 al conjunto de todos los títulos (Título), D_3 al conjunto de todas las fechas (Fecha_recep), y así sucesivamente. Cada renglón debe consistir de 6 tuplos ($v_1, v_2, v_3, v_4, v_5, v_6$) donde v_1 es el número de video (esto es, v_1 está en el dominio de D_1), v_2 es el título y está en el dominio de D_2 , así para cada tuplo.

Una relación o tabla es un arreglo bidimensional con las siguientes características:

- Cada entrada en la tabla es un elemento de datos o dato elemental; no hay grupos repetitivos. Es decir, cada dominio debe representar a una sola relación. Se dice que una relación está normalizada si no tiene grupos repetitivos.
- A cada columna, se le asigna un nombre diferente y está constituido por valores del mismo dato elemental.
- Todos los tuplo son distintos; no se permiten duplicados.
- Las hileras y columnas pueden ordenarse en cualquier secuencia en cualquier momento, sin que esto afecte el contenido de la formación o la semántica aplicada.

•

Cada tuplo o relación debe poseer una llave que lo identifica unívocamente y lo diferencia de otros tuplos de esa relación. La llave es un dominio simple o una combinación de dominios. Una llave es un dominio simple o una combinación de dominios. Una llave constituida por una combinación de dominios es no redundante si ninguna identidad de la llave puede eliminarse o borrarse sin destruir la habilidad de identificar unívocamente cada tuplo. Puede existir más de un conjunto de dominios que pueden constituir una llave; es decir que identifica unívocamente a un tuplo y que es no redundante. A estos conjuntos se les denomina llaves candidatas. La llave primaria es el conjunto de dominios que se selecciona para identificar a los tuplos. Normalmente debería ser el que tuviera el número mínimo de dominios.

1.1.8.3 Estructuración de Datos

Al planear la organización de los datos que van a almacenarse, el analista debe prever la necesidad de acceder los datos para cumplir con requerimientos inesperados, objetivo que se puede alcanzar mediante la normalización de los datos.

Normalización

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

La normalización es el proceso de simplificar la relación entre los campos de un registro. Por medio de la normalización, un conjunto de datos en un registro se reemplaza por varios registros que son más simples y predecibles y, por tanto, más manejables. La normalización se lleva a cabo por cuatro razones:

- Estructurar los datos de forma que se puedan representar las relaciones pertinentes entre los datos.
- Permitir la recuperación sencilla de los datos en respuesta a las solicitudes de consultas y reportes.
- Simplificar el mantenimiento de los datos actualizándolos, insertándolos y borrándolos.
- Reducir la necesidad de reestructurar o reorganizar los datos cuando surjan nuevas aplicaciones.

El proceso de normalización tiene como objetivos:

- Eliminar en lo posible todos los datos que mantengan anomalías
- Conservar toda la información
- Maximizar la flexibilidad

Se han realizado investigaciones extensas con el fin de desarrollar métodos para normalización.

Los analistas de sistemas deben familiarizarse con los pasos de la normalización, ya que este proceso puede mejorar la calidad del diseño de una aplicación.

- Descomponer todos los grupos de datos en registros bidimensionales.
- Eliminar todas las relaciones en las que los datos no dependan completamente de la llave primaria del registro.
- Eliminar todas las relaciones que contengan dependencias transitivas.

La estructura debe ser tal que haya lugar para todos los datos requeridos.

La redundancia que pueda existir deberá ser causada por los elementos que son identificadores o llaves. Por lo que se deben elegir aquellos que no estén sujetos a actualizaciones.

Los efectos indeseables son las anomalías que pueden presentarse en las operaciones de actualización, inserción y eliminación.

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

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

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

- Esta capacidad de adaptabilidad de los cambios maximizan la independencia de uso particular de los datos.

La normalización requiere de tres acciones sobre un atributo de una entidad. Estas son las siguientes:

- Primera Forma Normal
- Segunda Forma Normal
- Tercera Forma Normal

El poder definir si una relación se encuentra en la primera, segunda o tercera forma normal, se basa en las dependencias funcionales que existan entre los atributos y los dominios particulares a esa relación. Las dependencias funcionales las determina directamente el significado o la semántica del contenido de la base de datos según la interpretación del diseñador de la base de datos.

Cuando se hizo la descripción del modelo E-R, se mencionó la existencia de una entidad independiente y otra subordinada. Esta subordinación está determinada por la relación que guardan entre ellas. El modelo relacional se basa en este concepto para establecer sus relaciones funcionales entre atributos. De esta forma, una definición formal de dependencia funcional en el modelo relacional sería el siguiente:

Dada la relación **R** se dice que el atributo **B** es funcionalmente dependiente del atributo **A** si en cualquier instante de tiempo cada valor de **A** no tiene más de un valor **B** asociado a él en la relación **R**. El indicar que **B** es funcionalmente dependiente de **A** es equivalente a indicar que **A** identifica o determina a **B**, lo cual se denota como $A \Rightarrow B$. Esto último concuerda con la lógica matemática en la que, $A \Rightarrow B$ significa que **A** identifica a **B**, es decir, que si **A** tiene un cierto valor "a" entonces **B** debe tener un valor "b".

Primera Forma Normal

Una de las mejoras básicas que el analista puede hacer es diseñar la estructura de un registro de manera que todos los registros de un archivo tengan la misma longitud. Los registros de longitud variable crean problemas especiales, ya que el sistema debe verificar siempre en dónde se encuentran los extremos de un registro (por ejemplo, buscando marcas especiales o leyendo un indicador de la longitud). Al fijar la longitud del registro se elimina este problema.

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

La primera forma normal se alcanza cuando se quitan todos los grupos de repetición, de forma que un registro tenga longitud fija. Un grupo de repetición, es decir, la aparición

repetida de un dato o grupo de datos dentro de un registro, es en realidad otra relación. Por lo tanto, se quita del registro y se le considera como una parte del mismo o como una relación adicional.

Consideremos la información contenida en el pedido de un cliente (Figura 1.1.8.2b) : número de pedido, número de cliente, dirección del cliente, fecha del pedido, al igual que el número de artículo, descripción del artículo, precio y cantidad del artículo ordenado. El diseño de una estructura de registro para manejar un pedido que contenga tales datos no es difícil.

Sin embargo, el analista debe considerar cómo manejar el pedido. Existen cuatro números de artículo, cuatro precios de artículo y cuatro especificaciones de cantidad. El pedido se puede considerar como cuatro registros separados, en cada uno de los cuales se incluya la información sobre el pedido y el cliente. Sin embargo, al considerar cada registro como un pedido aparte se aumenta la complejidad para el cambio de los detalles de cualquier parte del pedido y utiliza espacio adicional (hallar el artículo correcto para cambiar valores es en sí más difícil).

Otra opción es la de diseñar el registro con longitud variable. Cuando un pedido especifica un artículo, los detalles de éste se establecen sólo una vez. Cuando se piden cuatro, los detalles del artículo se repiten cuatro veces. La parte del registro de los datos que se repite se denomina grupo de repetición.

La primera forma normal se alcanza cuando un registro se diseña de longitud fija. Esto se lleva a cabo quitando el grupo de repetición y creando un archivo o relación aparte que contenga al grupo de repetición. El registro original y el nuevo se interrelacionan mediante un punto común de los datos.

La figura 1.1.8.2c muestra la normalización a la primera forma normal para el registro de pedidos. Conviene observar cómo los datos incluidos en la nueva relación incluyen el

número de pedido, número de artículo, descripción del artículo, precio y cantidad. El número de pedido es común a los registros de pedido y pedido-artículo. La aplicación utiliza ahora estas estructuras de registro para describir los detalles de un pedido de uno o más artículos. Aún así, la longitud del registro de pedidos no se ve alterada.

Primera Forma Normal

Registro de pedidos

No Pedido	No Cliente	Nombre del Cliente	Dirección del Cliente	Fecha solicitud	Costo Total
101456	812	Harper University	Rosboro, NY 13514	12/1	185.40
102721	1319	McCraw Manufact	Endicott, NY 13760	12/1	5.80
103654	2107	Union Hospital	6 th Avenue, NY 10021	12/2	118.20
105489	824	Hara Enterprises	Búffalo, NY 13240	11/30	162.50

Registro de Artículos

No Pedido	No Artículo	Descripción del Artículo	Precio del Artículo	Cantidad Solicitada
101456	T101	Manteles	.95	100
101456	B16	Cobijas	1.55	50
101456	C118	Sábanas	.29	30
101456	B14	Toallas	.42	10
102721	C118	Sábanas	.29	20
103654	B14	Toallas	.42	60
103654	B16	Cobijas	1.55	60
105489	N38	Servilletas	.84	100
105489	T101	Manteles	.95	50
105489	B16	Cobijas	1.55	20

Figura 1.1.8.2c Cada registro tiene una longitud fija y no contiene grupos repetitivos.

Segunda Forma Normal

La prueba para determinar si una entidad está en segunda forma normal es que el valor de cualquier atributo que no es llave dependa de todos los atributos que forman la llave. La segunda forma normal se alcanza cuando un registro está en la primera forma normal y cada campo depende totalmente de la llave del registro (en el almacenamiento y

La segunda forma normal se alcanza cuando un registro está en la primera forma normal y cada campo depende totalmente de la llave del registro (en el almacenamiento y recuperación). En otras palabras, el analista busca la dependencia funcional: un campo es funcionalmente dependiente si su valor está asociado de manera única con un campo específico. Aunque este concepto suena complejo, en realidad es muy sencillo, como lo mostrará este ejemplo.

Los departamentos estatales de control vehicular trabajan arduamente para garantizar que a un vehículo en el estado se le asigne un número de placa específico (los diseñadores de bases de datos llaman a esto una relación uno a uno). El número de placa identifica de manera única a un vehículo específico; un número de serie de un vehículo está asociado con uno y sólo un número de placa. Así, si se conoce el número de serie de un vehículo, se puede determinar el número de placa. Esto es la dependencia funcional. En contraste, si el registro de un vehículo contiene los nombres de todas las personas que lo manejan, se pierde la dependencia funcional (y el diseño del archivo no cumple el objetivo de la segunda forma normal). ¿Por qué? Si conocemos el número de licencia, no conocemos quién es el conductor (pueden ser muchos). Y si conocemos el nombre de un conductor, no conocemos el número de placa o de serie del vehículo específico, ya que un conductor puede estar asociado con más de un vehículo en el archivo.

Para alcanzar la segunda forma normal, cada campo del registro que no dependa de la llave primaria del registro debe quitarse y utilizarse para formar una relación aparte.

Consideremos el ejemplo de los pedidos. El registro de los pedidos en la figura 1.1.8.2c está en primera forma normal, ¿pero también está en la segunda forma normal? Intentemos esta prueba. Si conocemos el nombre del proveedor, ¿conocemos el número de pedido?, ¿existe una relación uno a uno o muchos a uno entre el nombre del proveedor y el número del artículo?

Puesto que el nombre del cliente depende del número del cliente y no del número de pedido, y ya que la relación entre la llave primaria del número de cliente y número de artículo no es uno a uno, sabemos que no se ha alcanzado la segunda forma normal. Por lo tanto, se diseñan dos nuevas estructuras de registro, como se muestra en la figura 1.1.8.2d. El registro de proveedores contiene el número de proveedor, nombre del proveedor y otros detalles, los cuales dependen en su totalidad de la llave del registro, es decir, del número de proveedor. Análogamente, el registro de partes contiene la llave de número de parte y todos los demás datos en el registro dependen funcionalmente de él.

Segunda Forma Normal

Registro de pedidos

No Pedido	No Cliente	Fecha solicitud	Costo Total
101456	812	12/1	185.40
102721	1319	12/1	5.80
103654	2107	12/2	118.20
105489	824	11/30	162.50

Registro de proveedores

No Cliente	Nombre del Cliente	Dirección del Cliente
812	Harper University	Rosboro, NY 13514
1319	McCraw Manufact	Endicott, NY 13760
2107	Union Hospital	6th Avenue, NY 10021
824	Hara Enterprises	Buffalo, NY 13240

Figura 1.1.8.2d Cada dato en un registro depende funcionalmente de la llave.

Tercera Forma Normal

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

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

Para poner una entidad en Tercera Forma Normal, se crea una entidad con los atributos que no dependen de ningún atributo que forma la llave, siendo el identificador de la nueva entidad el atributo del cual era dependiente.

La tercera forma normal se alcanza cuando se quitan todas las dependencias transitivas de un diseño de registro. El caso general es el siguiente (Figura 1.1.8.2e):

- A, B y C son tres datos en un registro
- Si C es funcionalmente dependiente de B y
- B es funcionalmente de A,
- entonces C es funcionalmente dependiente de A.
- Por lo tanto, existe una dependencia transitiva.

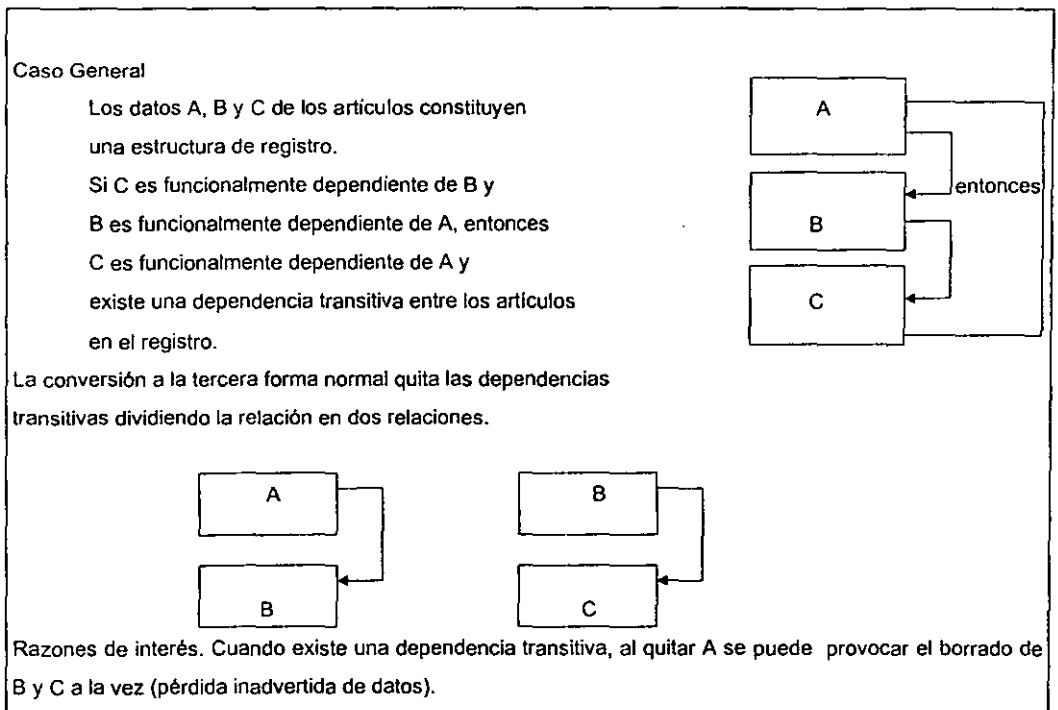


Figura 1.1.8.2e Tercera Forma Normal

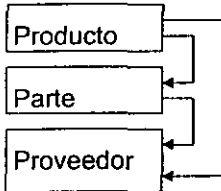
En el manejo de datos, la dependencia transitiva es una preocupación, ya que los datos pueden perderse de manera inadvertida cuando la relación está oculta. En el caso anterior, si se quita A, entonces también se quitan B y C, sea o no ésta la intención. Este problema se elimina diseñando el registro para la tercera forma normal. La conversión a la tercera forma normal quita la dependencia transitiva dividiendo la relación en dos relaciones separadas.

El registro de la figura 1.1.8.2d se halla en la segunda forma normal. Pero puesto que no contiene dependencias transitivas, también está en tercera forma normal.

Consideremos otro ejemplo. Al utilizar la información sobre el proveedor de la figura 1.1.8.2f, podemos ver que PARTE depende de PRODUCTO, y PROVEEDOR depende de PARTE. Si quitamos un cierto producto de la base de datos, no sería adecuado eliminar las partes y los proveedores (los cuales podrían estar asociados con otros productos). O bien, podríamos desear un registro de cada uno en la base de datos para un uso futuro.

Para quitar la dependencia transitiva en esta situación, se crean los registros PRODUCTO, PARTE y PROVEEDOR (Figura 1.1.8.2f). Los registros adicionales dan mayor flexibilidad para cubrir los requerimientos futuros a la vez que eliminan las dificultades mencionadas arriba.

No Producto	No Proveedor	Nombre Proveedor	Dirección Proveedor	No de Parte	Descripción de parte	Cantidad	Descripción del producto	Costo del producto
-------------	--------------	------------------	---------------------	-------------	----------------------	----------	--------------------------	--------------------



Tercera Forma Normal

Producto	No del producto	Descripción producto	Costo del producto
Componentes product	No de Producto	No de parte	Cantidad utilizada
Parte	No de parte	Descripción parte	No de proveedor
Proveedor	No de "proveedor	Nombre proveedor	Dirección proveedor

Figura 1.1.8.2.f Tercera Forma Normal para los datos de los productos

De lo anterior, podemos ver que la comprensión de cómo eliminar las dependencias transitivas requiere de un buen conocimiento de la relación entre los datos y las actividades empresariales en las que se utilizan. Pero éste es exactamente el punto. Los diseños del archivo y base de datos deben modelar la empresa a la que soportan. Estas decisiones no se pueden hacer en busca de satisfacciones técnicas o de almacenamiento.

La tabla siguiente resume las tres formas de normalización analizadas. Si la base de datos se diseña de acuerdo con los principios de normalización, la manipulación de datos, la cual se describe abajo, será más fácil (de otra forma, el analista debe trabajar con el administrador de datos para realizar los ajustes necesarios).

TABLA I.1.8.2a Resumen de los pasos en la normalización de los datos	
Forma	Pasos
Primera Forma Normal	Cambiar todas las estructuras que no sean bidimensionales (es decir, grupos repetitivos) en estructuras de registro bidimensionales.
Segunda Forma Normal	Eliminar los datos que no dependan totalmente de las llaves del registro.
Tercera Forma Normal	Eliminar los datos que dependan transitivamente de las llaves primarias

Manipulación de los datos.

La manipulación de los datos para preparar reportes o responder a consultas utiliza uno de los tres operadores relacionales (Tabla I.1.8.2b).

Operaciones SELECT (seleccionar)

La operación SELECT produce una nueva tabla en respuesta a una consulta o solicitud de reporte creada a partir de los renglones de la tabla inicial que cumplan los criterios de la solicitud. La consulta ¿ Cuáles son los nombres y direcciones de los clientes que hicieron pedidos el 12/1? es satisfecha al elegir los renglones de la relación CLIENTE, donde el campo de la fecha del pedido tenga como valor 12/1. Se crea una nueva tabla que contenga dos renglones y dos columnas como respuesta. Únicamente los campos solicitados se incluyen en la nueva relación.

TABLA 1.1.8.2b Operaciones Relacionales: Manipulación de datos	
Operador Relacional	Descripción de la Operación
SELECT	Crea una nueva relación extrayendo renglones que cumplan con los criterios establecidos
PROJECT	Crea una nueva tabla extrayendo columnas que cumplan con los criterios establecidos
JOIN	Crea una nueva relación a partir de los renglones de dos tablas que tengan atributos que cumplan los criterios establecidos

Operaciones PROJECT (proyectar)

El comando PROJECT crea una nueva tabla a partir de los datos extraídos, utilizando atributos especificados en la pregunta. En otras palabras, PROJECT escoge columnas de una relación. La pregunta ¿ Cuáles son los números de cuenta de los clientes que rentan sábanas o cobijas ? producirá la tabla que se muestra en la Figura 1.1.8.2g.

No de cuenta
812
836
1924
2107

Figura 1.1.8.2g Resultado de PROJECT

Operaciones JOIN (Unir)

La operación JOIN crea una nueva selección combinando dos tablas existentes, eligiendo los registros que cumplan los criterios establecidos en la pregunta y removiendo después los registros duplicados. (Conceptualmente, una base de datos relacional no contiene entradas duplicadas. Sin embargo, en muchas implantaciones del modelo relacional, los duplicados se eliminan sólo cuando el usuario elige la opción).

La pregunta siguiente demuestra la operación JOIN:

¿ Cuáles son los números de pedido y números de cuenta de los clientes que solicitan toallas ?.

Estos operadores relacionales son la base de todas las operaciones en los datos. Por medio de éstos se pueden revisar las operaciones de adición, borrado y cambio:

- **Adición**

Se añade un registro a la base de datos sin afectar los demás registros.

```
INSERT INTO CLIENTES VALUES
```

```
'1245','Acme Supply', '7856 Wilson High, Sandy Springs, GA 30338'
```

- **Borrado**

Se borra el renglón de la base de datos que contiene al registro especificado

```
DELETE FROM CLIENTE WHERE CLIENTE-ID = '836'
```

- **Cambio**

El registro con la llave especificada, se localiza y los valores de los datos en el atributo secundario se cambian de acuerdo a los nuevos datos.

```
UPDATE CLIENTE  
SET DIRECCION = '4675 WEST MAIN'  
WHERE CLIENTE-ID = '836'
```

Conviene hacer notar que en cada ejemplo, los pasos necesarios para responder a la consulta o procesar las solicitudes de actualización se pueden visualizar examinando los datos. Aun cuando una solicitud particular no se haya previsto al diseñar la base de datos, ésta puede todavía ser satisfecha debido a las poderosas operaciones disponibles mediante el modelo relacional. Este no es necesariamente el caso con los otros modelos de datos.

Observemos que no hicimos referencia a las llaves del registro físico, estructuras de archivo o rutas de acceso al manipular las tablas. Sólo se mencionaron datos comunes para obtener los resultados deseados.

1.2 METODOLOGÍAS DE DISEÑO Y DESARROLLO DE SISTEMAS

El acelerado ritmo en el crecimiento de algunas empresas pone de manifiesto la necesidad de contar con sistemas que hagan frente a las demandas de los clientes por mejores productos y servicios.

En nuestro ámbito laboral día a día vivimos la creciente necesidad de contar con mejores herramientas de desarrollo de software que nos permitan la implementación de sistemas de manera oportuna y cubrir las crecientes demandas por sistemas más eficientes y adecuados a las necesidades organizacionales.

Establecer una metodología orientada al desarrollo de software es el objetivo de la Ingeniería de Software como disciplina científica. Veamos pues los conceptos inherentes a esta disciplina.

SOFTWARE

Podríamos anotar tantas definiciones de software como autores consultemos; sin embargo, podemos resumir que

Software: Es el conjunto de programas o instrucciones y datos que "HACEN OPERAR" una computadora.

En contrapartida al concepto de software encontramos que **hardware** denomina en conjunto a los componentes físicos de una computadora.

CARACTERÍSTICAS DEL SOFTWARE

El software es un elemento "lógico" (intangible) del sistema y no físico. Cuando se construye el hardware, el proceso creativo humano (análisis, construcción, prueba, ...) se traduce finalmente en una forma física: El gabinete, teclado, tarjetas, circuitos, cables, ... Todo es tangible; Cuando se desarrolla software todo se reduce a la manera (programas) de cómo se resuelve un problema a través de la computadora.

El software es desarrollado; no es fabricado en un sentido clásico. El desarrollo de software conlleva actividades de análisis, diseño, implantación, pruebas y mantenimiento orientado hacia el qué y cómo hacerlo y no al con qué hacerlo.

El software no se "deteriora". Naturalmente que no existen factores físicos ni temporales para el desgaste del software como ocurre con el hardware (uso, edad, variación de voltaje, clima). Cuando un componente hardware se deteriora, se sustituye por una pieza de repuesto. Sin embargo, puede ocurrir que el software se vuelva obsoleto con el paso del tiempo, aunque mantiene su rendimiento y sigue cubriendo las necesidades para las cuales fue desarrollado originalmente.

El diseño del software se traduce a un programa que especifica la estructura de datos, los procedimientos y requerimientos del sistema deseados. Un traductor (interprete o compilador) procesa este programa convirtiendo el código fuente en instrucciones ejecutables por una computadora.

ÁREAS DE APLICACIÓN DEL SOFTWARE

En toda situación en la que pueda definirse un algoritmo para computadora (conjunto específico de pasos procedurales para resolver un problema), es aplicable el desarrollo de software. Las siguientes son algunas de las grandes áreas de aplicación del software.

Sistemas. El software de sistemas es una colección de programas escritos para servir a otros programas (sistemas operativos, compiladores, editores, controladores de dispositivos periféricos, procesadores de telecomunicaciones, etc.). El área del software de sistemas se caracteriza por la fuerte interacción con el hardware de la computadora; su gran utilización por múltiples usuarios; la operación concurrente que requiere planificación, el compartir recursos y una sofisticada gestión de procesos; estructura de datos complejas y múltiples interfaces externas.

Software de tiempo real. Es software que mide, analiza y/o controla sucesos del mundo real conforme ocurren. Los elementos del software de tiempo real incluyen un componente de acumulación de datos que recibe y formatea la información desde un entorno externo, un componente de análisis que transforma la información según requiera la aplicación; un componente de control/salida que responda al entorno externo y un componente de monitoreo que coordina a los componentes anteriores, de forma que pueda mantenerse la respuesta en tiempo real (según la aplicación, en el rango de 1 milisegundo a 1 minuto).

Contable y Administrativa. Hoy en día, el procesamiento de información comercial constituye la mayor de las áreas de aplicación del software: Nóminas, inventarios, ventas, bases de datos, ...

Ingeniería y Ciencias. El software de ingeniería y científico se ha caracterizado por los algoritmos numéricos, de simulación y diseño asistido por computadora.

Inteligencia Artificial. El software de inteligencia artificial (IA) hace uso de algoritmos no numéricos para resolver problemas complejos que no son adecuados para el cálculo o análisis directo. Actualmente, el área más activa de la IA es la de los sistemas expertos. Otras áreas de aplicación para el software de IA es el reconocimiento de patrones (imágenes y voces), prueba de teoremas y juegos.

Software de computadoras personales. El procesamiento de palabras, las hojas de cálculo, los gráficos por computadora, juegos, bases de datos, aplicaciones financieras, comerciales y personales, son sólo unos cuantos de los cientos de aplicaciones para PC's.

Software empotrado. Éste reside en circuitos de memoria de solo lectura y se utiliza para controlar productos y sistemas comerciales e industriales; puede ejecutar funciones muy limitadas y variadas, como por ejemplo, el control de las teclas de un horno de microondas, tacómetros, inyectores de combustible, etc.

CRISIS DEL SOFTWARE.

La crisis del software se refiere a un conjunto de problemas encontrados en el desarrollo del software de computadoras. Los problemas no están limitados al software que no funciona adecuadamente. Sino que la crisis del software abarca los problemas asociados con cómo desarrollar el software, cómo mantener un volumen creciente de software existente y cómo podemos esperar satisfacer la demanda creciente de software. Aunque la referencia a una crisis del software puede ser criticada por ser algo

melodramática, la frase sirve como propósito útil para alumbrar los problemas reales encontrados en todas las áreas de desarrollo del software.

Son tres los grandes problemas que han caracterizado la crisis del software: (1) la planificación y estimación del costo suelen ser imprecisas; (2) la productividad del analista de sistemas no corresponde con la demanda; y (3) la calidad del software no es adecuada.

No hay un método que dé solución, de la noche a la mañana, a la crisis del software. Sin embargo, se ha ido adoptando una disciplina para el desarrollo del software —la Ingeniería de Software—, combinando métodos para establecer fases en el desarrollo del software, nuevas herramientas para automatizar estos procesos, construcción de bloques de código reutilizable y una filosofía predominante de coordinación, control y administración.

INGENIERÍA DE SOFTWARE

La ingeniería de software es una disciplina para desarrollar software de alta calidad para sistemas basados en computadora. Abarca un conjunto de tres elementos claves —métodos, herramientas y procedimientos— que facilitan al líder de proyecto controlar el proceso de desarrollo de software y suministrar a los que practiquen dicha ingeniería las bases para construir software de alta calidad de una forma productiva.

Los **métodos** de la ingeniería de software (**metodologías**) suministran el "cómo" construir técnicamente el software. Los métodos abarcan un amplio espectro de tareas que incluyen: planificación y estimación de proyectos; análisis de requerimientos del sistema y del software; diseño de estructuras de datos; arquitectura de programas y procedimientos algorítmicos; codificación; prueba de mantenimiento. Los métodos de la

ingeniería de software introducen frecuentemente una notación especial orientada a lenguaje o gráfica y un conjunto de criterios para la calidad de software.

Las **herramientas** de la ingeniería de software suministran un soporte automático o semiautomático para los métodos. Hoy, existen herramientas para soportar cada uno de los métodos mencionados anteriormente. Cuando se integran las herramientas de forma que la información creada por una herramienta pueda ser usada por otra, se establece un sistema para el soporte de desarrollo de software, llamado Ingeniería de Software Asistido por Computadora (CASE: computer-aided software engineering). CASE combina el software, hardware y bases de datos de la ingeniería de software (una estructura de datos que contenga la información relevante sobre el análisis, diseño, codificación y prueba) para crear un entorno de ingeniería de software, análogo al diseño asistido por computadora (CAD: computer-aided design).

Los **procedimientos** de la ingeniería de software son la cola que pega los métodos y herramientas y facilita un desarrollo racional y oportuno de software de computadora. Los procedimientos definen la secuencia en la que se aplican los métodos, las entregas (documentos, informes, formas, etc.) que se requieren, los controles que ayudan a asegurar la calidad y coordinar los cambios, y las guías que facilitan a los gestores del software establecer su desarrollo.

La ingeniería de software está compuesta de pasos que abarcan los métodos, herramientas, y procedimientos. Estos pasos se denominan paradigmas de la ingeniería del software. Un paradigma se elige basándose en la naturaleza del proyecto y de la aplicación.

A continuación se describen tres de los paradigmas más utilizados:

1.2.1 EL CICLO DE VIDA CLÁSICO.

Exige un enfoque sistemático, secuencial, del desarrollo del software que comienza en el nivel del sistema y progresa a través del análisis, diseño, codificación, prueba y mantenimiento. Algunas veces llamado "modelo en cascada"

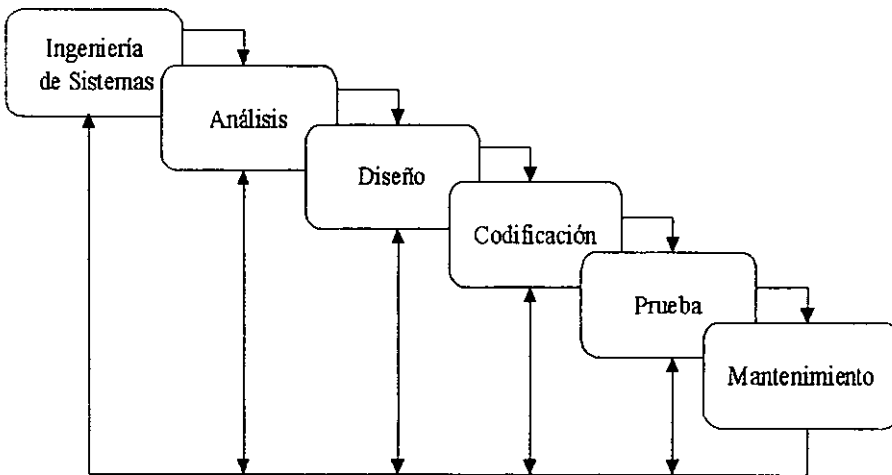


FIG. 1.2.1 Ciclo de vida clásico

Ingeniería y análisis de sistemas. Partiendo de la definición del problema se describen las interrelaciones necesarias entre software, hardware, datos y usuarios. El flujo del proceso a automatizar y/o resolver debe comprenderse plenamente.

Análisis de los requerimientos del software. Definen la determinación y contenido de la información así como las interfaces requeridas. La determinación de información se

refiere a la necesidad de predecir el orden y tiempo de datos de entrada. El contenido se refiere al significado y la forma de la información de entrada y salida. Una interface es la forma a través de la cual interactuamos con el sistema: teclado, mouse, lectores ópticos, ambientes gráficos, programas, ...

Diseño. El proceso de diseño traduce los requerimientos del sistema a un conjunto de representaciones (gráficas, pseudocódigos, tablas, esquemas, ...) que describen la estructura de datos, la arquitectura del software y los procedimientos inherentes. Un buen diseño se acerca mucho al código fuente.

Codificación. El diseño debe traducirse a una forma legible para la computadora. La codificación ejecuta esta tarea, a través de un lenguaje de programación. Si el diseño se ejecuta en una forma detallada, la codificación puede realizarse mecánicamente.

Prueba. Una vez codificado el sistema, es ejecutado a modo de prueba para garantizar que la entrada definida producirá la salida requerida.

Mantenimiento. El software sufrirá invariablemente cambios; éstos ocurrirán debido a que se tiene un nuevo sistema operativo o periférico, o debido a que se requieren aumentos funcionales o de rendimiento.

CONSTRUCCIÓN DE PROTOTIPOS

Normalmente un cliente definirá un conjunto de objetivos generales para el sistema, pero no identificará los requerimientos detallados de entrada, procesamiento o salida. En otros casos el programador puede no estar seguro de

la funcionalidad de un algoritmo, la adaptabilidad de un sistema operativo o lo amigable de una interface. En este tipo de situaciones, puede ser mejor realizar un prototipo.

La construcción del prototipo es un proceso que facilita al programador la creación de un modelo del software a construir. El modelo tomará una de las tres formas siguientes: un **prototipo en papel** que describe la interface usuario-máquina, sirve para obtener

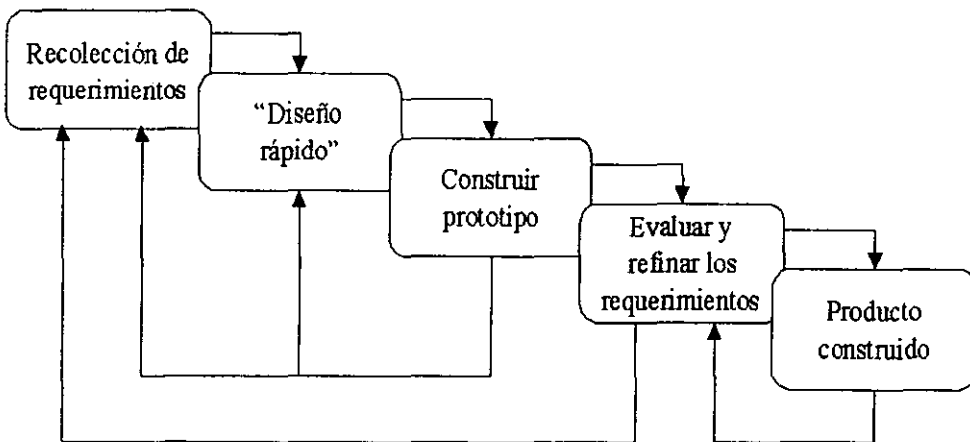


FIG. 1.2.2 Construcción de prototipos

una comprensión clara del sistema; un **prototipo** que implementa algunos subconjuntos de las funciones requeridas del sistema; o un **programa existente** que ejecute parte o toda la función deseada, pero que tenga otras características que deban ser mejoradas en el nuevo trabajo de desarrollo.

Como en todos los métodos de desarrollo de software, la construcción de prototipos comienza con la **definición de requerimientos**. El técnico y el cliente se reúnen para

definir los objetivos globales del sistema, identifican los requerimientos y perfilan las áreas donde será necesario una mayor definición.

Luego se produce un **diseño rápido**, enfocándose sobre las interfaces de usuario (formatos de entrada y salida). El diseño rápido conduce a la **construcción de un prototipo**. El prototipo es **evaluado** por el cliente/usuario y se utiliza para **refinar los requerimientos** del sistema a desarrollar. Se produce un proceso interactivo en el que el prototipo es afinado para que satisfaga las necesidades del cliente, al mismo tiempo que facilita al que lo desarrolla una mejor comprensión de lo que hay que hacer. Idealmente, el prototipo sirve como un mecanismo para identificar los requerimientos del sistema. Si se construye un prototipo que funciona, el desarrollador hace uso de fragmentos de programas existentes o aplica herramientas que faciliten la rápida generación de programas que funcionen.

La construcción de prototipos es un paradigma efectivo para la ingeniería del software; tenga en cuenta que sirve sólo como un mecanismo de definición de los requerimientos. Posteriormente ha de ser descartado (al menos en parte) y debe construirse el software real, con miras en la calidad y mantenimiento.

TÉCNICAS DE CUARTA GENERACIÓN

El término técnicas de la cuarta generación (T4G) abarca un amplio espectro de herramientas de software que facilitan al programador especificar algunas características del sistema a alto nivel. Luego, la herramienta genera automáticamente el código fuente basándose en la especificación dada.

Actualmente, un entorno para el desarrollo de software que soporte el paradigma T4G incluye algunas de las siguientes herramientas: lenguajes no procedurales para consulta de bases de datos, generación de informes, manipulación de datos, interacción y definición de pantallas y generación de código; capacidades gráficas; y capacidad de hoja de cálculo.

Estas herramientas existen sólo para dominios de aplicación muy específicos, como los manejadores de bases de datos.

COMBINACIÓN DE PARADIGMAS

Los paradigmas de la ingeniería de software pueden y deben combinarse de forma que puedan utilizarse las ventajas de cada uno en un único proyecto. El trabajo comienza con la definición del problema y la especificación de los requerimientos. El método puede seguir con el ciclo de vida clásico (ingeniería de sistemas y análisis de los requerimientos) o puede ser la definición menos formal del problema usada en la construcción de un prototipo. Independientemente, debe producirse la comunicación cliente-desarrollador del software.

La naturaleza de la aplicación sugerirá la elección del método adecuado. Si los requerimientos para la función y rendimiento del sistema están razonablemente bien comprendidos, pueden ser aplicables los métodos de especificación recomendados para el ciclo de vida clásico. Por otra parte, si la aplicación del software exige una fuerte interacción hombre-máquina o requiere algoritmos no probados o técnicas de control de salidas, puede realizarse un prototipo. En tales casos puede usarse a veces un L4G para desarrollar rápidamente un prototipo. Una vez que se haya evaluado y refinado el prototipo, pueden aplicarse los pasos de diseño e implementación del ciclo de vida clásico para desarrollar el software formalmente.

No hay necesidad de ser dogmático en la elección de los paradigmas para la ingeniería del software; la naturaleza de la aplicación debe dictar el método a elegir. Mediante la combinación de paradigmas, el todo debe ser mejor que la suma de las partes.

PROCESO GENERAL DEL DESARROLLO DE SOFTWARE

El proceso de desarrollo de software contiene tres fases genéricas independientemente del paradigma de ingeniería elegido. Las tres fases, definición, desarrollo y mantenimiento, se encuentran en todos los desarrollos de software, independientemente del área de aplicación, tamaño del proyecto o complejidad.

La **fase de definición** se enfoca sobre el qué. Esto es, durante la definición, el que desarrolla el software intenta identificar qué información será procesada, qué función y rendimiento se desea, qué interfaces han de establecerse, qué restricciones de diseño existen y que criterios de validación se necesitan para definir un sistema correcto. Por tanto han de identificarse los requerimientos claves del sistema y del software. Aunque los métodos aplicados durante la fase de definición varían dependiendo del paradigma de ingeniería de software aplicado (o combinación de paradigmas), de alguna forma se producirán tres pasos específicos:

Análisis del sistema. Definir el papel de cada elemento de un sistema informático, asignando el rol que jugará el software.

Planificación del proyecto de software. Una vez que está asignado el ámbito del software, se asignan los recursos, se estiman los costos y se definen las tareas de planificación del trabajo.

Análisis de requerimientos. El ámbito definido para el software da la dirección, pero antes de comenzar a trabajar es necesario disponer de una información más detallada del dominio de la información y de la función del software.

La **fase de desarrollo** se enfoca sobre el cómo. Esto es, durante el desarrollo, el que desarrolla el software intenta descubrir cómo han de diseñarse las estructuras de datos y arquitectura de software, y cómo han de implementarse los detalles procedurales, cómo ha de trasladarse el diseño a un lenguaje de programación (o lenguaje no procedural) y cómo ha de realizarse la prueba. Los métodos aplicados durante la fase de desarrollo varían dependiendo del paradigma de ingeniería de software aplicado. Sin embargo, de alguna forma se producirán tres pasos concretos:

Diseño del software. El diseño traslada los requerimientos de software a un conjunto de representaciones (algunas gráficas, otras tabulares o basadas en lenguajes) que describen la estructura de datos, arquitectura y procedimiento algorítmico.

Codificación. Las representaciones de diseño deben traducirse a un lenguaje artificial (un lenguaje de programación convencional o un lenguaje no procedural usado en el contexto del paradigma T4G) que da como resultado unas instrucciones ejecutables por la computadora. El paso de la codificación ejecuta esta traducción.

Prueba de Software. Una vez que el software se ha implementado en una forma ejecutable por la maquina, debe ser probado para descubrir los defectos que puedan existir en la función, lógica e implementación.

La **fase de mantenimiento** se enfoca sobre el cambio que va asociado con una corrección de errores, adaptaciones requeridas por la evolución del entorno del software y modificaciones debidas a los cambios de los requerimientos del cliente para

reforzar o aumentar el sistema. La fase de mantenimiento replica los pasos de las fases de definición y desarrollo, pero en el contexto del software existente. Durante la fase de mantenimiento se encuentran tres tipos de cambios:

Corrección. Incluso con las mejores actividades para garantizar la calidad, es probable que el cliente descubra defectos en el software. El mantenimiento correctivo cambia el software para corregir los defectos.

Adaptación. Con el paso del tiempo es probable que cambie el entorno original (por ejemplo, CPU, sistema operativo, periféricos) para el cual se desarrolló el software. El mantenimiento adaptativo se traduce en modificación del software para acomodarlo a los cambios de su entorno externo.

Aumento. Conforme se utilice el software el cliente/usuario reconocerá funciones adicionales que podría ser benéfico añadirlas. El mantenimiento perfectivo aumenta el software para más allá de sus requerimientos funcionales originales.

Las fases y pasos relacionados descritos en nuestra visión genérica de la ingeniería de software se complementan con varias actividades protectoras. Las revisiones se realizan durante cada paso para asegurar que se mantiene la calidad. La documentación se desarrolla y controla para asegurar que toda la información sobre el sistema y software estará disponible para un uso posterior.

El control de los cambios se instituye de forma que éstos puedan ser mejorados y registrados.

En el paradigma clásico del ciclo de vida, las fases y pasos descritos en esta sección se definen explícitamente. En los paradigmas de construcción de prototipos y T4G

están implicados algunos de los pasos, pero no explícitamente identificados. El método de cada paso puede variar de un paradigma a otro pero el enfoque global que exige la definición, desarrollo y mantenimiento permanece invariable. Uno puede realizar cada fase con disciplina y métodos bien definidos o de forma completamente desordenada. Pero habrá que realizarlos de alguna forma.

ESPECIFICACIÓN DE REQUERIMIENTOS

Para realizar bien el desarrollo del software es esencial realizar una especificación completa de los requerimientos. Independientemente de lo bien diseñado o codificado que esté, un programa pobremente especificado decepcionará al usuario y hará fracasar el desarrollo.

Tanto el programador como el cliente tienen un papel activo en la especificación de los requerimientos. El cliente intenta reformular su concepto de la función y comportamiento de los programas en detalles concretos. El que desarrolla el software actúa como interrogador, consultor y el que resuelve los problemas.

La tarea de análisis de los requerimientos es un proceso de descubrimiento y refinamiento. El ámbito del programa, establecido inicialmente durante la ingeniería del sistema, es refinado en detalle. Se analizan y asignan a los distintos elementos de los programas las soluciones alternativas.

El análisis de requerimientos facilita al ingeniero de sistemas especificar la función y comportamiento de los programas, indicar la interface con otros elementos del sistema y establecer las restricciones de diseño que debe cumplir el programa. El análisis de requerimientos permite al ingeniero de sistemas (analista) refinar la asignación de

software y representar el dominio de la información que será tratada por el programa. El análisis de requerimientos da al diseñador la representación de la información y las funciones que pueden ser traducidas en datos, arquitectura y diseño procedural. Finalmente, la especificación de requerimientos suministra al técnico y al cliente, los medios para valorar la calidad de los programas, una vez que se haya construido.

El análisis de requerimientos puede dividirse en cuatro áreas:

- 1) Reconocimiento del problema
- 2) evaluación y síntesis
- 3) especificación
- 4) revisión

Es importante comprender el contexto del sistema y revisar el ámbito de los programas para, primeramente, generar la estimación de la planificación y, posteriormente, establecer la comunicación necesaria para el análisis, de forma que se asegure *el reconocimiento del problema*.

La evaluación del problema y síntesis de la solución se enfocan a que el analista debe evaluar el flujo y estructura de la información, definir a detalle todas las funciones del programa, establecer las características de la interface del sistema y describir las restricciones de diseño. Cada una de las tareas sirven para describir el problema de forma que pueda sintetizarse un enfoque o solución global.

Una vez que se hayan descrito las funciones básicas, comportamiento, interface e información, se especifican los criterios de validación para demostrar una comprensión de una correcta implementación de los programas. Estos criterios sirven como base para hacer la prueba durante el desarrollo de los programas. Para definir las

características y atributos del software se escribe una *especificación de requerimientos* formal.

En este momento, el cliente y el analista inician una *revisión* de los requerimientos, observando las posibles modificaciones en la función, comportamiento, representación de la información, interrelaciones y criterios de validación.

ANÁLISIS DE REQUERIMIENTOS

Se han desarrollado varios métodos de análisis y especificaciones de requerimientos. Los investigadores han identificado los problemas y sus causas y desarrollado reglas y procedimientos para resolverlos. Cada método de análisis tiene una única notación y punto de vista. Sin embargo, todos los métodos de análisis están relacionados por un conjunto de principios fundamentales:

1. *El dominio de la información*, Considera a) el flujo de información; b) el contenido de la información y c) la estructura de la información.
2. *El problema debe subdividirse* de forma que se descubran los detalles de una manera progresiva o jerárquica.
3. Deben desarrollarse las *representaciones lógicas y físicas* del sistema.

Aplicando estos principios, el analista enfoca el problema sistemáticamente. Se examina el dominio de la información de forma que pueda comprenderse su función. La partición se aplica para reducir la complejidad. La visión lógica y física del software, son necesarias para acomodar las restricciones lógicas impuestas por los requerimientos de procesamiento, y las restricciones físicas impuestas por otros elementos del sistema.

Como resultado del análisis se desarrolla la Especificación de Requerimientos del Software. La revisión es esencial para asegurar que el desarrollador y el cliente tienen la misma percepción del sistema; en este punto es muy útil el diseño de prototipos. Los requerimientos se representan gráficamente o a través del lenguaje natural, de forma que conduzcan a una correcta implementación del software.

METODOLOGÍAS DE ANÁLISIS DE REQUERIMIENTOS

El papel de los métodos de análisis de requerimientos, es asistir al analista en la construcción de una descripción precisa del sistema; y aunque cada método introduce su propia notación y heurística de análisis, los métodos pueden ser evaluados en el contexto de las siguientes características comunes: 1) mecanismos para el análisis del dominio de la información; 2) método de representación funcional; 3) definición de interfaces; 4) mecanismos para subdividir el problema; 5) Soporte de la abstracción, y 6) representación de las visiones físicas y lógicas.

Todos los métodos se enfocan (directa o indirectamente) al flujo de datos y al contenido o estructura de datos. El flujo de datos se caracteriza por las transformaciones (funciones) que se aplican a los datos de entrada para obtener "predeterminados" datos de salida. El contenido de los datos puede representarse explícitamente usando un mecanismo de diccionario o, implícitamente, enfocando primero la estructura jerárquica de los datos.

Las funciones se describen normalmente como transformaciones o procesos de la información. Cada función puede ser representada usando una notación específica. Una descripción de la función puede desarrollarse usando el lenguaje natural, un lenguaje procedural con reglas sintácticas informales (Pseudocódigo) o un lenguaje de especificación formal.

Una descripción de las interfaces se da generalmente como un añadido a las representaciones funcionales y de información. El flujo hacia y desde una función específica puede ser identificado con flujos de información a otras funciones. Las interfaces se derivan del examen del flujo de información.

1.2.2 MÉTODOS DE ANÁLISIS ORIENTADOS AL FLUJO DE DATOS. (YOURDON)

El sistema acepta entrada de distintas formas; aplica hardware, software y elementos humanos para transformar la entrada en salida. Conforme la información se mueve a través del software, se modifica mediante una serie de transformaciones. Un diagrama de flujo de datos (DFD), es una técnica gráfica que describe el flujo de información y las transformaciones que se aplican a los datos, conforme se mueven de la entrada a la salida.

La función global del sistema es una transformación sencilla de la información, representada en la figura 1.2.3 en una burbuja.

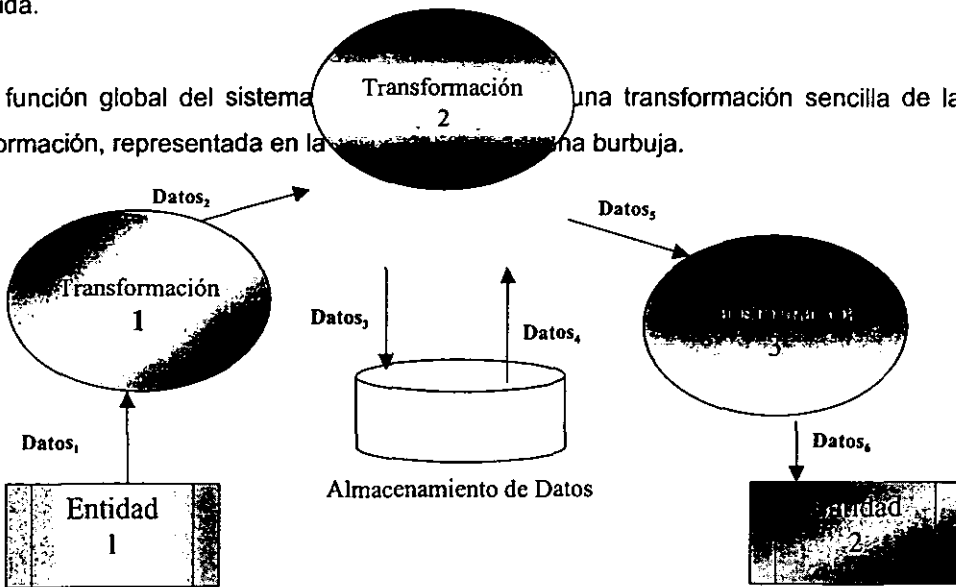


Fig. 1.2.3 Un diagrama de flujo de datos

El DFD puede usarse para representar un sistema a cualquier nivel de abstracción. De hecho, los DFD pueden particionarse en niveles que representan flujo incremental de información y detalle funcional. Un nivel 0 de un DFD, también llamado modelo de sistema fundamental, representa un sistema como un único proceso con datos de entrada y salida, indicados por flechas de entrada y hacia afuera, respectivamente. Transformaciones adicionales y caminos de flujo de

información se representan conforme el nivel 0 del DFD es subdividido para revelar más detalles.

El diagrama de flujo de datos es una herramienta gráfica muy útil durante el análisis de requerimientos del software. Sin embargo, el diagrama puede causar confusión si su función se confunde con la de un diagrama de flujo de procesos. Un diagrama de flujo de datos describe el flujo de la información sin una notación explícita de control (condiciones y ciclos).

Para derivar un diagrama de flujo de datos orientado al software/sistema debemos observar los siguientes criterios:

1. El primer nivel del diagrama de flujo de datos debe describir al sistema como un proceso sencillo (caja negra).
2. Los archivos de entrada/salida principales deben ser anotados cuidadosamente.
3. Todas las flechas y procesos son etiquetados con nombres significativos.
4. La continuidad del flujo de información debe ser mantenida.
5. Cada vez debe refinarse un solo proceso.

Durante el análisis de requerimientos se puede descubrir que ciertos aspectos del sistema "están sujetos a cambios" o "se aclararán en el futuro" o están vagamente definidos por el cliente; el diagrama de flujo de datos permite un fácil aislamiento del

dominio del cambio. Al comprender el flujo de información a lo largo del dominio de los límites del cambio, puede hacerse una mejor preparación para futuras modificaciones o puede conducirse una modificación actual sin modificar otros elementos del sistema.

Un análisis del dominio de la información puede ser incompleto si sólo se considera el flujo de datos. Cada flecha de un diagrama de flujo de datos representa uno o más elementos de información. Por lo tanto, el analista debe disponer de un método para representar el contenido de cada flecha de un DFD.

Se ha propuesto *el diccionario de datos* como una gramática casi-formal para describir el contenido de los elementos de información y ha sido definido de la siguiente forma: "El diccionario de datos contiene las definiciones de todos los datos mencionados en el DFD, en una especificación del proceso y en el propio diccionario de datos". Los datos compuestos (datos que pueden ser además divididos) se definen en términos de sus componentes; los datos elementales (datos que no pueden ser divididos) se definen en términos del significado de cada uno de los valores que puede asumir. Por tanto, el diccionario de datos está compuesto de definiciones de flujo de datos, archivos (datos almacenados) y datos usados en los procesos (funciones o transformaciones).

Una vez que ha sido representado el dominio de la información (usando un DFD y un diccionario de datos), el analista describe cada función (transformación) representada, usando el lenguaje natural o alguna otra notación estilizada. Una de tales notaciones es el "Pseudocódigo", que incorpora construcciones procedurales básicas —secuencia, selección, y repetición— junto con frases del lenguaje natural, de forma que pueden desarrollarse descripciones procedurales precisas de las funciones representadas dentro de un DFD.

1.2.3 MÉTODOS DE ANÁLISIS ORIENTADOS A LA ESTRUCTURA DE DATOS (GANE & SARSON).

Conocido como SSA (Structured Systems Analysis). Utiliza un lenguaje gráfico para construir los modelos de los sistemas, incorpora conceptos de bases de datos; sin embargo, SSA no proporciona una gran variedad de mecanismos estructurales. Hay cuatro elementos básicos en SSA: diagramas de flujo de datos, diccionario de datos, representaciones lógicas de procedimientos y técnicas de estructuración de almacenamiento de datos.

Las características que distinguen a estos métodos son:

1. Identificar los objetos de información clave (entidades o items)
2. Definir una estructura jerárquica del sistema
3. Definir el diccionario de datos
4. Transformar una estructura de datos jerárquica en una estructura de programa

Desarrollo de Sistemas Estructurados de Datos (DSED: Gane & Sarson), examina el contexto de la aplicación, esto es, como se mueven los datos entre productores y consumidores de la información. A continuación, se establecen las funciones de la aplicación, que describen los elementos de la información y el procesamiento que debe ejecutarse sobre ellos. Finalmente, se modelan los resultados de la aplicación usando una estructura jerárquica. Usando este método, (DSED) comprende todos los atributos del dominio de información: flujo de datos, contenido y estructura de datos.

Para determinar el contexto de la aplicación en DSED, debe establecerse el problema de forma que nos facilite responder a tres preguntas:

1. ¿Cuáles son los elementos de información que han de procesarse?

2. ¿Quién/qué son los productores y consumidores de información?
3. ¿Cómo ve la información cada productor/consumidor en el contexto de otros grupos?

DSED propone un **diagrama de entidades** como mecanismo para responder a estas preguntas.

El diagrama de entidades utiliza una notación muy parecida al diagrama de flujo de datos; por un lado encontramos *entidades* las cuales se interrelacionan a través de *acciones*. Al derivar un diagrama de entidades para el análisis de una base de datos se conforma el llamado diagrama entidad-relación.

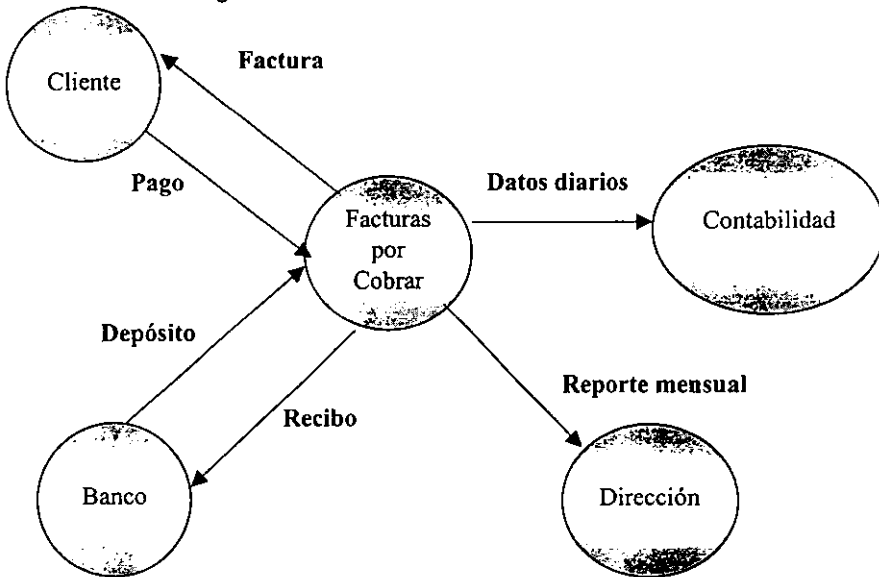


Fig. 1.2.4 Diagrama de entidades

Los métodos de análisis de requerimientos del software fueron desarrollados para ser aplicados manualmente. Sin embargo, la aplicación manual del análisis estructurado (DSED) puede ser pesado y propenso a error cuando se analizan grandes sistemas.

Hoy, los métodos de análisis, junto con una clase de técnicas específicamente diseñadas para el procesamiento automático, están disponibles en un formato de

“ayuda por computadora”. Las herramientas para automatizar el análisis son un primer paso hacia la ingeniería de software asistida por computadora (CASE) y están mejorando la calidad y productividad de los analistas.

SADT® es una técnica de análisis y diseño estructurado, que ha sido ampliamente usada como herramienta para la definición de sistemas, análisis de requerimientos de software y diseño de sistemas. Desarrollado originalmente como un método manual, SADT consiste en procedimientos que permiten al analista descomponer las funciones del sistema; una notación gráfica, el actigrama y datograma de SADT, que muestra las relaciones de la información con las funciones dentro del software; y unos criterios de control del proyecto para aplicar la metodología.

Usando SADT, el analista desarrolla un modelo que comprende muchos actigramas y datogramas definidos jerárquicamente.

La metodología SADT comprende herramientas automáticas que soportan procedimientos de análisis y unos utensilios de organización bien definidos mediante los cuales pueden aplicarse las herramientas. Las revisiones y recorridos son especificados, permitiendo la validación de la comunicación técnico/cliente. Las responsabilidades del equipo son similares a las que se encuentran en un equipo con un programador en jefe, acentuando un trabajo en equipo para el análisis, diseño y revisión.

1.2.4 SSADM (Structured Systems Analysis and Design Method)

Es una técnica sistemática (metodología de desarrollo) para el análisis y diseño de aplicaciones. Fue patrocinado por el Gobierno del Reino Unido en los 80's.

Conjuga tanto los métodos orientados al flujo de datos como los métodos orientados a la estructura de datos. Se enfoca hacia el establecimiento de cómo los datos pasan a través del dominio de la información y suministra procedimientos para aislar el contenido de los datos y las funciones de procesamiento. Por otra parte, identifica los elementos de información y las acciones (procesos) y los modela de acuerdo con la jerarquía (estructura) de información del problema.

1.2.5 MODELADO DE MERISE

La concepción del método de Merise es doble, por un lado, representa un método de diseño de sistemas de información, y por otra parte, propone una metodología de desarrollo.

Las principales ventajas del método de diseño de Merise son:

- Una solución global del sistema de información llevada en paralelo sobre los datos y el procesamiento
- Una descripción del sistema de información por niveles: el nivel conceptual, el nivel organizacional o lógico y el nivel operacional o físico.
- Una descripción del sistema de información usando una representación formal: precisa, simple y rigurosa para la descripción de los datos.
- Esta **formalidad** es estándar internacionalmente por el ISO bajo el nombre de modelo "ENTIDAD-RELACIÓN"

- Una descripción muy rica del nivel conceptual fundada sobre la invariabilidad de los sistemas de información, lo que permite construir un nuevo sistema sobre bases sólidas, independientemente de la organización y opciones técnicas de automatización
- Finalmente, la representación visual, en particular de los modelos conceptuales, contribuye al total establecimiento de un dialogo constructivo entre todos aquellos que participan en la concepción del nuevo sistema.

Como método de desarrollo de sistemas de información:

- Divide el proceso de desarrollo en cuatro fases:
 - Estudio de factibilidad
 - Estudio detallado
 - Realización
 - Implantación

Esta división fue estandarizada en el plan nacional de Francia por AFNOR (estándar Z67-101 recomendaciones para el control de proyectos de procesamiento de datos). Esta división corresponde al ciclo de vida de un sistema de información.

- Una descripción detallada de la estructura del trabajo es conformada para concluir el desarrollo del sistema de información. Esta estructura es realizada por un comité de Administración, de un grupo de proyecto y un comité de usuarios. La misión y composición de cada grupo de participantes está especificada dentro del marco del método.

Merise es un método de desarrollo y diseño de sistemas de información, muy usado en Francia. Fue definido a fines de los 70's; está fuertemente orientado a los puntos de vista de los "actores" (quienes participan en el desarrollo del sistema) y su definición y uso de la información. Provee para el modelado de eventos, sincronización y múltiples operaciones en varios niveles.

1.3 REDES

Definición

¿Qué es una RED?: En el campo de la computación se puede decir que una RED, es un conjunto de computadoras enlazadas entre si y/o con otros equipos, cuya configuración permita que esté sea un medio para transmitir, recibir, compartir y manejar información.

Conceptos básicos

¿Qué hace una RED?: una RED tiene como objetivo principal; compartir recursos físicos (equipos y sus periféricos) y recursos lógicos (archivos de datos y programas), actualizándolos, organizándolos y explotándolos.

¿Por qué una RED?: porque la RED es la repuesta correcta a una necesidad de compartir entre usuarios los recursos más costosos de equipo y la información centralizada y/o dispersa de un organismo, obteniendo con esto la tan necesaria organización y economía en la informatica.

Sin mucha pretensión, se puede aseverar que los tres puntos anteriores, vienen a ser el "A,B,C " de las REDES LOCALES.

Normalmente las microcomputadoras necesitan distintos recursos (periféricos) como son: impresores, graficadores, discos duros, unidades de respaldo en cinta magnetica, programas de aplicación, paquetería, etc. Que se tienen que adquirir a costos adicionales.

En una RED estos recursos en una sola micro se van a compartir con las demás, mediante un canal de comunicación que por lo general, es un cable dedicado a las comunicaciones. Las micros se conectan a este canal por medio de una interface, que es una tarjeta electrónica que se coloca en una de las ranuras de expansión de cada micro.

La microcomputadora que cuenta con los recursos periféricos recibe el nombre de servidor (SERVER) de la RED, que auxiliado por el sistema operativo de la RED viene a ser virtualmente el "cerebro" dedicado a administrar los recursos y las comunicaciones entre las demás micros, mismas que trabajando así, reciben el nombre de ESTACIONES DE TRABAJO.

Componentes de una red local

Los componentes principales de una RED son:

- **Servidor**

El Servidor, que puede ser **DEDICADO** o **NO DEDICADO**.

- **DEDICADO**, exclusivamente administra los recursos de la RED
- **NO DEDICADO**, además de administrar los recursos de la RED, funciona como Estación de Trabajo

Las características y configuración de la computadora que sea posible definir como SERVER, están en función de los requerimientos del caso, generalmente se trata de un equipo robusto tanto en hardware como en software.

- **Estaciones de Trabajo**

Están representadas por cada una de las microcomputadoras conectadas en RED.

En la RED, tanto el Server como las Estaciones de Trabajo, pueden ser PC's XT o AT's, equipos 386, 486 o Pentium, los modelos PS/2 de IBM, sus Value Point, e inclusive microcomputadoras no compatibles como es el caso de Macintosh.

En la actualidad se fabrica hardware exprefeso para REDES LOCALES como es el caso de los Servidores y Estaciones de Trabajo de fábrica, con ventajas que posteriormente analizaremos. En el mercado nacional podemos encontrar que la mayoría de los fabricantes de productos de marca como IBM, HP, COMPAQ, DIGITAL, ACER, etc., ofrecen productos de estas características.

- **Interface de RED.**

Dispositivo que permite la interconexión de los nodos de la RED. Debe instalarse en cada equipo que conformará la RED. Generalmente es una tarjeta que va instalada dentro de cada computadora que se conectará a una RED, de ahí que se le denomine inadecuadamente Tarjeta de RED. En la actualidad las interfaces de RED también se pueden conectar a un puerto (Paralelo, Serial, PCMCIA, etc.) del nodo de la RED.

Según su especificación y normas, cada interface de RED determina los protocolos de comunicación y la forma de interconexión (TOPOLOGIA) de cada RED. Existen tres estándares de interfaces de RED que dominan el mercado a nivel internacional:

- **ARCNET:** Que tiene una relación costo-beneficio favorable, con un sistema de cableado sencillo de amplio rango.

- **ETHERNET:** La de uso más generalizado por su alto rendimiento facilidad de interconexión de equipos heterogeneos.
- **TOKEN-RING:** Muy costosa, pero con el respaldo técnico v promocional de IBM, esta tarjeta puede conectar toda la línea de equipos IBM, desde una PC hasta un 309X ó 93XX en una sola RED de este tipo.

Es importante recalcar que empresas mexicanas, como el caso de Digital Data y Micron, producen con tecnología propia, tarjetas bajo estos tres estandares.

- **Canal de Comunicación**

Es el medio físico por el cual se comunican los nodos de una RED. Por lo general es un cable dedicado a las comunicaciones, mismo que puede ser:

- a) De tipo telefónico
- b) De par trenzado (Twisted Pair)
- c) Coaxial
 - Broadband Lento, varios canales
 - Baseband Rápido, un canal
- d) Fibra óptica: Más rápido y varios canales

Este canal de comunicación determina la velocidad máxima de transferencia de información que va desde 2.5 Mb/s hasta 100 Mb/s, dependiendo del tipo de cable que se utilice.

Actualmente se están desarrollando nuevas tecnologías para que el medio de comunicación sea inalámbrico.

A partir de 1990 se comercializan interfaces de RED inalámbricas, con tecnologías de radio frecuencia, microondas, rayo laser, etc.

- **Repetidores**

Elementos que permiten incrementar las distancias del medio de comunicación, reforzando su señal sin importar la topología; pueden ser tarjetas internas o cajas externas. Se dividen en activos y pasivos.

- **Sistemas de Cableado**

Cuya forma de conexión entre los equipos (TOPOLOGIA), está en función de la interface de red que se haya seleccionado.

- **Cajas y elementos de Conexión**

Son los elementos adicionales de conectorización, los cuales dependen del tipo de sistemas de cableado que se utilice.

- **Sistema Operativo de RED**

Es el software que se instala en el servidor de la RED, permitiendo la compartición de recursos y el control y administración de la información de la RED.

Los principales sistemas operativos de RED en el mercado internacional, son:

- NETWARE de Novell. En diferentes versiones
- LAN MANAGER de Microsoft
- Windows NT y Windows para grupos de trabajo de Microsoft.
- Lantastic.
- Todos los NETBIOS compatibles.
- IBM LAN SERVER
- VINESS
- NETWORK
- QNX de quantum Software System Ltd
- TAPESTRY

- **Software de Aplicación**

Son las aplicaciones disponibles en la RED, para los usuarios. Las más importantes son:

- Suites de Productividad Personal, (Procesadores de Texto, Hojas de Cálculo, Manejadores de Archivos, Presentaciones, etc.)
- Manejadores de Bases de Datos.
- Correo Electrónico.

Después de haber mencionado algunas de sus características es importante mencionar que el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) es una organización que entre sus funciones esta la de desarrollar normas en diferentes áreas y las LAN no han sido la excepción. La IEEE ha desarrollado la norma 802 que se ha convertido en el estándar para el desarrollo de redes de área local (a nivel software).

La IEEE define a la red de área local como:

Un sistema que comunica datos que permite un número de dispositivos independientes conectados directamente entre sí, con un margen de distancia moderada sobre canales físicos de comunicación, [MAR89].

1.3.1. Topologías más importantes. Ventajas y desventajas.

Para protocolos, simplemente se adoptará la definición que por extensión se le da a éste término, es decir un conjunto de reglas aplicadas a las comunicaciones.

En REDES LOCALES, prácticamente existen tres tipos básicos de Topologías, a saber:

- **Estrella**
- **Bus**
- **Anillo**

Se puede sumar a estos tipos básicos la topología de ARBOL que es una conexión compuesta. Para el estudio de la Topología se deben de considerar dos tipos:

- **Física**
- **Lógica**

La **TOPOLOGIA FISICA** es determinada por la disposición de los elementos conectados a la RED, mientras que la **TOPOLOGIA LOGICA** la determina el Protocolo de Comunicación operando en la RED, no importando la disposición física de los elementos, en otros términos se puede implementar un anillo lógico en un bus físico.

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

La topología de una red se refiere a la configuración física en que los dispositivos son conectados mediante un canal (usualmente cable).

Topología de árbol

Desde el punto de vista físico, esta conexión como se dijo anteriormente, es combinada y es una opción más para implementar REDES, según las necesidades del usuario

Trabaja el Protocolo TOKEN PASSING, tarjeta ARCNET y repetidores tanto PASIVOS como ACTIVOS.

1.3.1.1 Topología de estrella.

En este tipo de conexión, el elemento central es el Servidor con sus Periféricos, como se puede apreciar en la figura No. 1.3.1.1 se mantiene preguntando constantemente a cada estación de trabajo mediante comunicación exclusiva y por turno si desea transmitir información, y en caso afirmativo, la atiende y al terminar prosigue con otra su interrogatoria permanente.

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

En el despertar de las REDES, esta topología fué la que se utilizó primero, pero resultaba una de las más caras.

La gran desventaja de esta topología es depender en su totalidad del nodo central por lo que al fallar éste, falla toda la red. Esta topología es usada en gran escala en los sistemas telefónicos.

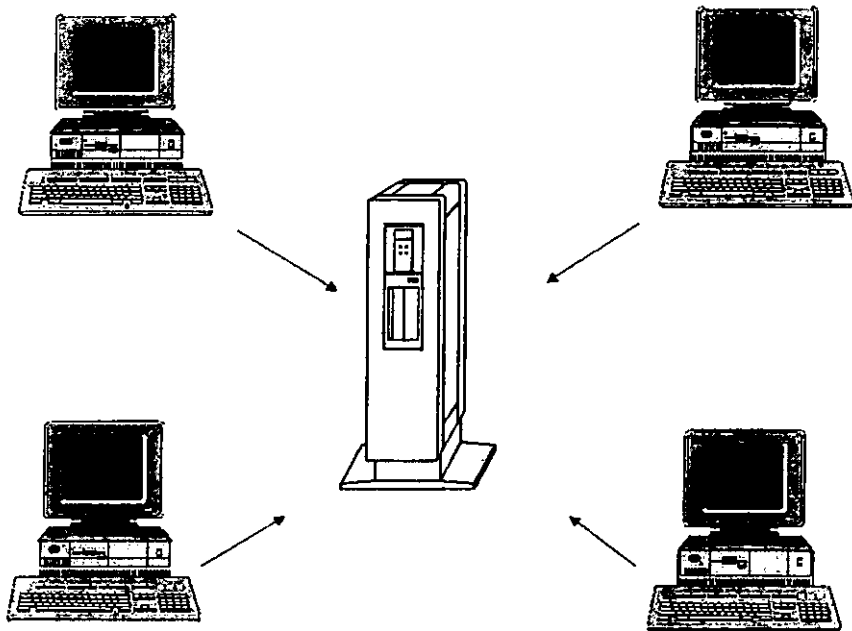


Figura 1.3.1.1 Topología Estrella

1.3.1.2 Topología de bus

Esta conexión se considera que es la más sencilla de todas, donde las Estaciones de Trabajo y el Servidor, están enlazadas por un solo cable (coaxial o par trenzado), y la información viaja en ambos sentidos, por lo que es necesario prevenir las colisiones.

Por ello el Protocolo apropiado es el CSMA/CD (Carrier Sense Multiple Access/ Collision Detection).

Con este protocolo, cualquier NODO transmite y espera a que se le confirme que la información fue recibida correctamente, de otra forma, se detecta la posible colisión, espera un tiempo a que el canal esté desocupado y la información se transmite nuevamente.

Es decir en la topología de bus, cada dispositivo esta directamente conectado a un canal de comunicación común, como se puede observar en la figura 1.3.1.3. Las señales que son transmitidas sobre el canal son llamados mensajes. Todos los mensajes pasan a lo largo del canal, por lo cual todas las computadoras se dan cuenta de lo que se transmite sobre el canal. El mensaje contiene una dirección la cual indica a que dispositivo se traslada en caso de que el nodo al que pasa el mensaje es el indicado por la dirección se procesa en caso contrario se ignora y el mensaje sigue su camino.

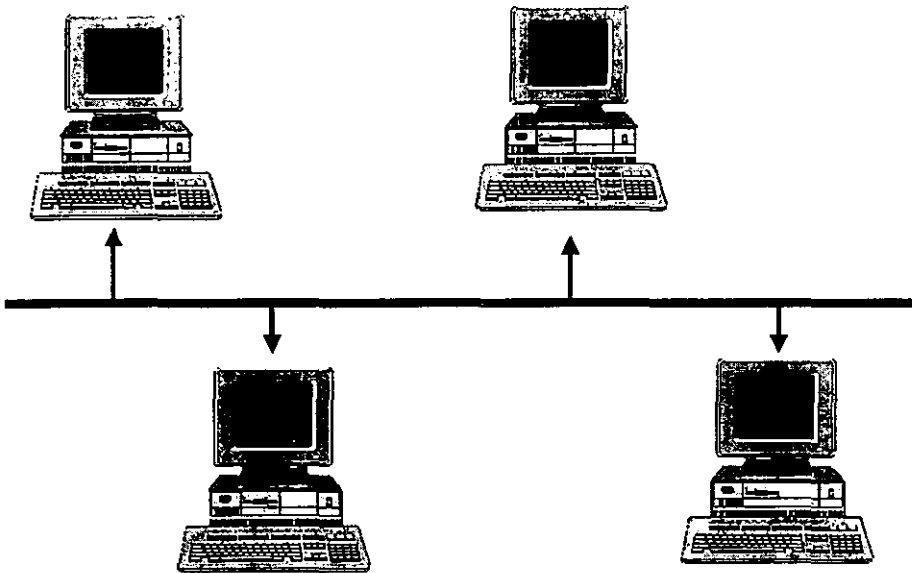


Figura 1.3.1.2 Topología Bus

1.3.1.3 TOPOLOGIA DE ANILLO

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

Una señal llamada TOKEN (Receptáculo, a modo de estafeta) va circulando por la RED y pasando por cada estación, si la primera resultó ser la solicitante, previa identificación entrega la información, de lo contrario la deposita en "sobre cerrado" para que ésta a su vez la envíe a la siguiente, llevando la consigna de entregarla hasta identificar a la solicitante.

Cada estación de paso, cuando más, colecta información adicional enviándola a la siguiente y así se la pasa la señal cerrando ciclos "circulares" por ello el protocolo apropiado para este caso se conoce como TOKEN PASSING.

Si observamos la figura 1.3.1.3, en una configuración de anillo las computadoras se conectan entre si, como su nombre lo indica. Los mensajes que se mandan alrededor del anillo deben estar específicamente dirigidos al nodo destino. debido a que las computadoras conectadas tienen acceso al mensaje, de similar forma que la configuración de bus.

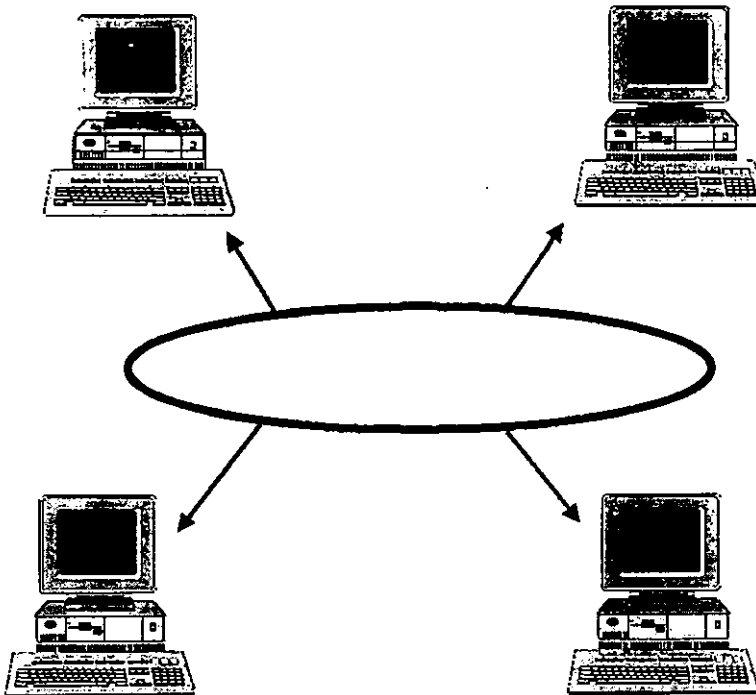


Figura 1.3.1.3 Topología Anillo

1.3.1.4 Malla.

La topología se puede llevar a cabo en forma local sin embargo con frecuencia este tipo se utiliza en las redes amplias o metropolitanas (MAN, WAN) que necesitan conectar diferentes puntos en forma remota. Se utilizan ruteadores para seleccionar el mejor trayecto de un punto a otro punto a través de la malla, figura 1.3.1.4.

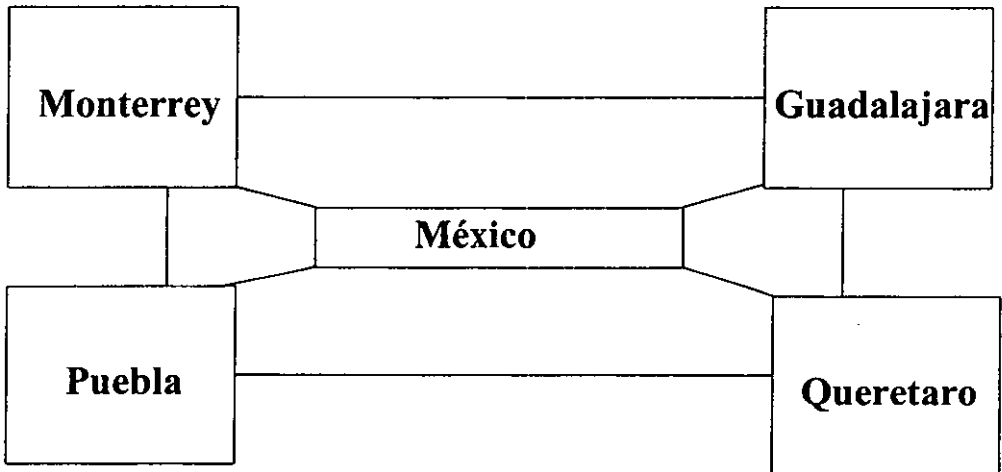


Figura 1.3.1.4 Malla

1.3.1.5 Soporte (backbone)

Los ruteadores o puentes administran el flujo de tráfico entre las subredes y el soporte. Se encuentran por lo general en grandes consorcios de oficinas o campus (el de C.U. es un ejemplo), figura 1.3.1.5.

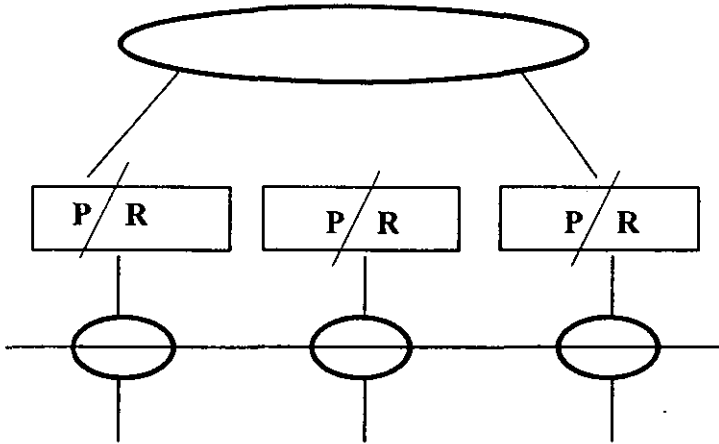


Figura 1.3.1.5 Topología Soporte

1.3.1.6 Estrella jerárquica.

Los concentradores, de algunas áreas o departamentos a los cuales se conectan las computadoras en forma de estrella, se conectan a un concentrador central a su vez, también en forma de estrella el cual administra el tráfico entre ellos, figura 1.3.1.6.

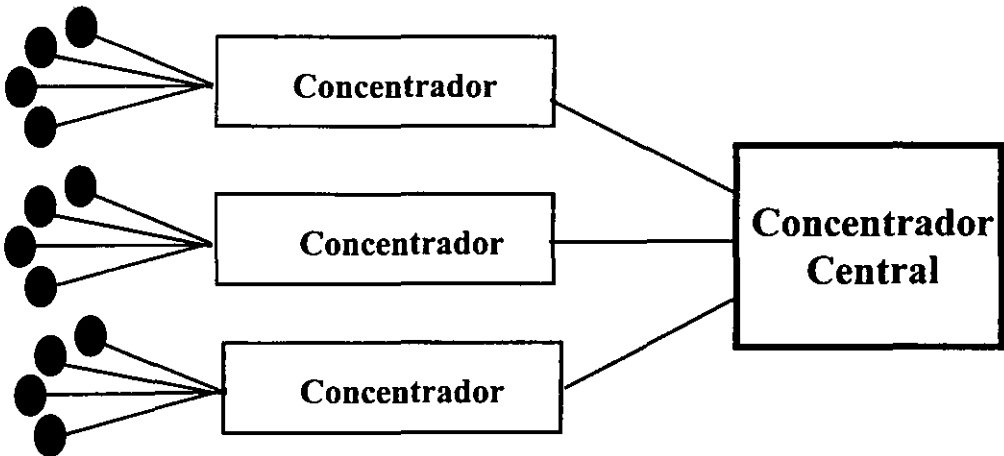


Figura 1.3.1. 6.- Topología estrella

1.3.2 PROTOCOLOS DE COMUNICACIÓN.

1.3.2.1 Estándar IEEE 802.

Debido a la gran variedad de protocolos de acceso que pueden construirse y los cuales pueden ser completamente incompatibles en 1980 en los Estados Unidos el Instituto de Ingenieros Electricistas y Eléctricos mejor conocido como el IEEE se dio a la tarea de normalizar un solo protocolo de acceso denominándole "802". Esta normalización trajo como consecuencia el gran desarrollo y auge de las redes locales.

1.3.2.1.1 Estándar IEEE 802.3, CSMA/CD

Protocolo de Acceso: CSMA/CD [TAN91]. Los protocolos son usados en las topologías de bus y anillo debido a que los dispositivos están conectados a un mismo medio de transmisión, por lo cual si no existieran dichos protocolos, en el caso de que dos o más computadoras transmitan al mismo tiempo la interferencia en el medio puede causar que una o varias transmisiones resulten dañadas. Es necesario seguir un protocolo de transmisión para la seguridad de la información.

Este protocolo propone que las computadoras conectadas a la LAN deben escuchar el medio físico antes de transmitir y esperar si existe una señal en el bus. CSMA/CD quiere decir Acceso Múltiple por Sensado y Portadora con Detección de Colisión (Carrier Sense Multiple Access/ Collision Detection). En este protocolo, las computadoras escuchan el bus mientras transmiten para detectar una posible colisión, en caso de ser positiva la colisión abortan la transmisión y esperan un tiempo aleatorio antes de volver a sensar el bus. Para mayor explicación contemplemos lo siguiente: Si en un tiempo t_1 una computadora A inicia una transmisión, poco tiempo después en t_2 la computadora B desea enviar información pero encuentra el bus ocupado y espera. Más adelante, en t_3 , otra computadora C también desea transmitir y encuentra

ocupado el bus por la computadora A, por lo que también espera. En t_4 , A termina de transmitir y el bus queda libre, esto lo notan las computadoras B y C e inician su respectiva transmisión. Como se ha mencionado anteriormente en CSMA/CD, la computadora que transmite también escucha la señal en el cable. Si lo que recibe es lo que esta enviando todo va bien, en caso contrario, supone que ha ocurrido una colisión. Al detectar la colisión, la computadora transmisora aborta su información y en su lugar envía un pequeña señal reforzando la colisión para garantizar que las demás computadoras involucradas en la interferencia también puedan detectar la colisión, después de enviar la señal espera durante un intervalo de tiempo aleatorio y vuelve a intentar la transmisión de su información escuchando en el bus.

Formato de la trama.

La información que se transmite a través de la red debe de contener además algunos campos a este conjunto se le llama trama. La trama no debe ser menor a 64 bytes eliminado el preámbulo y el delimitador de inicio de trama. La trama contiene lo siguiente, figura 1.3.2.1.1:

Preámbulo	SFD	Destino	Fuente	Longitud	Info	FCS
7	1	2 o 6	2 o 6	2		4

Figura 1.3.2.1.1.- Trama CSMA/CD.

- Preámbulo (7 bytes). Es un campo de 7 bytes con el código 10101010. Al transmitir esta byte en codificación Manchester, se genera una señal cuadrada que sirve para sincronizar a los receptores en la red.

- Delimitador de inicio de trama (SFD: Start Frame Delimiter, 1 byte). Un byte formado por el patrón 10101011. El último par de bits interrumpe la onda cuadrada formada por el preámbulo y los primeros bits de este byte. Esta interrupción sirve para indicar dónde en realidad inician los campos con información útil.
- Dirección Destino (2 o 6 bytes). Cada computadora en la red tiene un identificador único que es su dirección. Este campo contiene la dirección del sitio al que envía la trama. Se utilizan 6 bytes para los campos de direcciones. Con esto 2^{48} valores con la intención de garantizar que ningún otro dispositivo en el mundo pueda tener una misma dirección.
- Dirección Fuente (2 o 6 bytes). Contiene la dirección de la computadora que generó la trama.
- Tipo de trama o longitud (2 bytes). En la mayor parte de los casos esto permite identificar cómo interpretar el campo de datos, pues indica el tipo de protocolo de capas superiores que lleva la trama. También este campo puede indicar el número de bytes válidos en el campo de información, tomando valores de 0 a 1500.
- Información. En este campo viaja la información que se transfiere de un sitio a otro. Como se ha mencionado anteriormente las tramas deben tener un tamaño mínimo de 64 bytes. Para determinar el tamaño de la trama se toman en cuenta todos los campos excepto el preámbulo y el delimitador, por lo que para la LAN con direcciones de 6 bytes, el tamaño mínimo del campo de información debe ser de 46 bytes. En caso de que la computadora transmisora quiera enviar menos de 46 bytes, se deben agregar caracteres de relleno (padding).
- Secuencia de Verificación de la trama (FCS: Frame Check Sequence, 4 bytes). El último campo es un código cíclico de redundancia (CRC) de 32 bits, calculado con los campos de dirección fuente y destino, tipo o longitud e información para tratar

de predecir la integridad de la trama: la computadora calcula el CRC y lo añade a la trama cuando lo envía por la red, la computadora destino recibe la trama y repite el cálculo, si el valor calculado es diferencial recibido, se supone que la trama ha sido alterada por el ruido en el medio por lo que se descarta. CSMA/CD ha sido apoyado desde sus inicios por Xerox, Intel y DEC.

1.3.2.1.2 Estándar IEEE 802.4, token bus.

El paso de testigo en bus trabaja de la siguiente forma: el token es transmitido de un equipo al siguiente. Cuando un equipo recibe el token este puede transmitir datos en un tiempo límite. Una vez terminada su transmisión o terminado su tiempo pasa el token al siguiente equipo. Si el equipo que recibe el token no tiene datos que transferir para el token de inmediato al siguiente equipo.

Hay que tomar en cuenta que existen en token bus prioridades por lo que las tramas intrascendentes esperan mientras que las importantes terminan su transmisión.

Aunque la red físicamente es un bus, lógicamente se configura como un anillo para el paso de testigo de un equipo al otro.

Formato de la trama.

Preámbulo	SD	FC	Destino	Fuente	Info	FCS	ED
1-n	1	1	2 o 6	2 o 6	0-819	4	1

- Preámbulo (1-n bytes). Cada trama comienza con un preámbulo que es usado por la estación receptora para la sincronización. La longitud varía dependiendo del método de modulación.

- Delimitador inicial (SD: Starting Delimite, 1 byte). Su longitud es de un byte y contiene una señal siempre distintiva de un dato.
- Control de Trama (FC: Frame Control, 1 byte). Este identifica el tipo de trama que esta siendo enviada.
- Dirección destino (DA: Destination Address, 2 o 6 bytes). La dirección destino puede ser tan solo una dirección o un grupo de direcciones.
- Dirección Origen (SA: Source Address, 2 o 6 bytes). La dirección origen a diferencia de la destino es única. Ambas direcciones pueden ser de 2 o 6 bytes.
- Campo de Información (IF: Information Field, 0-819 Bytes). Este campo puede contener: una unidad de enlace (LLC), controlador de señal, manejador de datos o datos de propósito especial.
- Secuencia de Verificación de Trama (FCS: Frame Check Sequence, 4 bytes). Cuando el equipo emisor conjunta una trama realiza una revisión redundante cíclica (CRC), esta formula fue descrita por la IEEE la cual requiere 4 bytes para el resultado. El equipo emisor almacena este resultado en este campo. Cuando el equipo receptor recibe la trama realiza la misma formula que el emisor y el resultado lo compara con el de este campo (FCS). Si los resultados no son iguales, el equipo receptor asume que la transmisión tuvo algún error y nuevamente requiere que la trama sea retransmitida.
- Delimitador final (ED: Ending Delimiter, 1 byte). Es similar al delimitador inicial y marca el fin de la trama.

1.3.2.1.3 Estándar IEEE 802.5, token ring.

Protocolo de Acceso: Paso de Testigo (Token Ring), [MAR89]. Hoy en día hay diversas técnicas para que en una topología en anillo se pueda controlar el acceso al medio, pero la más conocida se le denomina Token Ring. La IEEE en 1989 (revisión) dio a conocer el estándar 802.5 describiendo las características para una LAN del tipo Token Ring como: la topología en anillo, sobre cable de par trenzado, aunque el uso de otros medios esta sujeto a consideración, el tipo de codificación Manchester diferencial con velocidad de 4 a 16 Mbps y protocolo de acceso por paso de token.

Las unidades de datos son transmitidas de una computadora a la próxima en una secuencia física a lo largo del anillo. Cada computadora transmite la unidad de datos a la próxima computadora actuando como un repetidor. La transmisión de unidades de datos esta controlada por un token. Cuando una computadora recibe el token, puede transmitir unidades de datos mientras un tiempo límite esta corriendo. Un token con una configuración que indica libre para transmitir es llamado free token (testigo libre). Cuando una computadora recibe un free token y tiene unidades de datos para transmitir, cambia la configuración del token de free token a busy token (testigo ocupado) e incluye el propio busy token con la unidad de datos que transmita. La unidad de datos viaja de computadora en computadora alrededor del anillo. Cada computadora que recibe una unidad de datos realiza un chequeo, en esta verificación se analiza el campo de control (FC) en caso de tratarse de una unidad de datos de tipo MAC, la información será procesada por todas las computadoras del anillo, si es de tipo LLC y coincide con la dirección destino es procesada. En caso de que no vaya dirigida a la computadora que en instancia tenga el busy token lo retransmite a la siguiente computadora, y así sucesivamente hasta que encuentre su destino. Para evitar que una computadora trasmita indefinidamente, se define un tiempo máximo de posesión del token, después del cual deberá emitirse un nuevo token que proporcione la oportunidad de transmitir tramas a las demás computadoras. Cuando una unidad de

datos retorna a la computadora que originalmente lo mando, esta remueve la unidad de datos de la red y manda un free token a través de la red a la siguiente computadora.

Dos condiciones de error pueden seriamente afectar la operación de una red mediante esta técnica: la pérdida del token y la persistencia del busy token. Para la detección y corrección de estas fallas, una de las computadoras conectada a la rd debe de tener la función de Monitor Activo. La computadora asignada como Monitor Activo continuamente monitorea la red en caso de que el período de tiempo predeterminado transcurra y no se detecta inicio de token el monitor asume que el token ha sido perdido e inicia un nuevo token. Para verificar una circulación indefinida de busy token, el monitor utiliza un bit de la trama conocido como bit M. Si el recibe una trama con el bit M=1, determina que la computadora que transmite no eliminó la trama del anillo y por lo tanto no liberó un nuevo token, por lo cual el Monitor Activo realiza estas dos últimas funciones.

Formato de la trama

A lo largo de la explicación del protocolo por paso de testigo se ha hablado mucho sobre las unidades de datos, la cual es sinónimo de trama, el cual contiene un encabezado que consiste de 5 partes (Delimitador de inicio, Control de Acceso, Control de Trama, Dirección destino y fuente), un campo de información y una secuencia de fin de trama que consiste de tres partes (Secuencia de Verificación de trama, Delimitador final, Estado de trama). Todas las partes de la trama serán detalladas a continuación en la figura 9.

SD	AC	FC	Destino	Fuente	Info	FCS	ED	FS
1	1	1	2 o 6	2 o 6		4	1	1

- Delimitador inicial (SD: Starting Delimiter, 1 byte). Es una señal única que identifica el inicio de la trama y esta formado por la siguiente secuencia de bits: "JK0JK000".
- Control de Acceso (AC: Access Control, 1 byte). Este byte esta formado por los siguientes bits "PPPTMRRR". Los bits PPP indican la prioridad del token o de la trama. El bit T tiene un valor de "0" en caso de tratarse de un free token. Cuando una computadora desea transmitir espera el free token y cambia el valor del bit a "1". El bit M se transmite con un valor de "0". Cuando el monitor activo lo retransmite, le cambia el valor a "1". Esto ayuda a la detección y corrección de situaciones en las que una trama de alta prioridad se transmite indefinidamente. Los bits RRR sirven para reservar la prioridad del siguiente free token.
- Control de Trama (FC: Frame Control, 1 byte). Esta formado de la siguiente cadena de bits "FFZZZZZZ". Los bits FF indican el tipo de trama (00=MAC, 01=LLC), los bits ZZZZZZ identifican un subtipo de la trama.
- Dirección destino (DA: Destination Address, 2 o 6 bytes). Todas las direcciones de la red deberán tener la misma longitud. El primer bit indica si se trata de una dirección individual o de grupo. mientras que el segundo bit, para el caso de direcciones de 6 bytes indica si se trata de una dirección local o universal.
- Dirección Origen (SA: Source Address, 2 o 6 bytes). Debe tener la misma forma y longitud que la dirección destino, y el primer bit debe valer "0".
- Campo de Información (IF: Information Field). Contiene la información destinada a las capas MAC o LLC. La longitud máxima es de 4.488 Kbytes en el caso de una red que soporte 4 Mbps y de 17.984 Kbytes para una red diseñada para 16 Mbps.

- Secuencia de Verificación de Trama (FCS: Frame Check Sequence, 4 bytes). Este campo contiene información para detectar errores de transmisión. La secuencia se calcula a partir del FC, DA, SA y de información.
- Delimitador de fin (ED: Ending Delimiter, 1 byte). Este byte contiene la siguiente cadena de bits "JK1JK1IE", El bit I puede usarse para determinar el fin de transmisión de una estación. Para la transmisión de una serie de tramas el bit I=1 en la primera trama e intermedias; I=0 cuando se transmita una sola trama o cuando se transmite la última trama. El bit E se transmite con valor igual a "0", cuando se detecta un error el bit E=1.
- Estado de la Trama (FS: Frame Status, 1 byte). Este byte contiene los bits de reconocimiento de la dirección y la trama copiada que son usados para indicar si la trama fue o no fue exitosamente recibida por la computadora destino.

TERMINOLOGIA

A efecto de estar familiarizados con los términos básicos que pueden resultar "no muy conocidos" para algunos de los participantes, se hacen los siguientes breves comerciales. En el medio, las REDES LOCALES también son llamadas LAN's (de Local Area Network), término que se menciona más por la asociación de ideas, que por el protocolo formal de una traducción de inglés.

De los vocablos RED y LOCAL, diremos que el primero se asocia a una conexión entre equipos de cómputo y el segundo a la cercanía física entre estos, que va de unos cuantos metros hasta unos cuantos kilómetros, en ocasiones de hasta 10 Km, distancia que ya más bien es un parámetro de enlace remoto.

Se habló al principio de las microcomputadoras compatibles, llamadas así por pretender ser "clonos" de las producidas hasta hace poco por la IBM. Los distintos

fabricantes de las primeras, comercializaron sus equipos con la "etiqueta" de PC's/IBM, término que es muy familiar en el mundo de la computación.

También se mencionaron los términos HARDWARE y SOFTWARE cuyos significados son ya muy conocidos, no obstante, será saludable recordar que HARDWARE implica todo aquello que sea electrónica física (como la propia C.P.U; con sus circuitos integrados, conductores, drives, discos, periféricos, cableado, etc.), y que SOFTWARE implica todo aquello que sea programas (como sistemas operativos, programas de aplicación, paquetería etc.).

Otro término que se menciona, y que es necesario conocer bien, es la palabra TOPOLOGIA:

Entre los matemáticos que estudiaron esta disciplina, está A. Listing quien le dió el nombre y la definió como la parte de las Matemáticas que estudia la disposición de agrupaciones de elementos.

Por lo tanto en el ambiente de REDES, y en congruencia con la definición anterior, en adelante se entenderá simplemente que: TOPOLOGIA, es la forma en que estan conectados el grupo de elementos que conforman una RED.

Otro concepto inherente a las REDES es el término de NODO, el cual desde el punto de vista matemático es el punto de unión o enlace de al menos dos ramas de una topologia, pero desde el punto de vista de las REDES es cualquier dispositivo que se encuentre conectado a la RED, por ejemplo: Servidores, Estaciones de Trabajo, Impresoras, Ruteadores, etc.

Las redes de área local (LAN) son en la actualidad productos comerciales que son utilizados en gran medida por diferentes dependencias públicas y empresas privadas en nuestro país y en la mayor parte del mundo.

También las redes cuentan con la capacidad de uso del concepto de distribución como es en el control de procesos de tiempo real, las estructuras de procesamiento en paralelo y las mismas bases de datos.

1.4 Características, ventajas y desventajas del manejador ACCESS

¿Qué es Microsoft Access?

Microsoft Access es un programa para el manejo de bases de datos relacionales que permite el almacenamiento, agrupamiento y búsqueda rápida de todo tipo de datos indispensables en las labores diarias de una empresa que se combina con la facilidad del uso que permite Windows.

Microsoft Access, cuenta con herramientas sencillas e intuitivas, que proporcionan la potencia de una base de datos de características complejas para tareas cotidianas de administración de datos.

Las bases de datos de Microsoft Access están constituidos por diversos objetos: tablas, consultas, formularios, informes, macros y módulos. Estos objetos se listan en la ventana de datos, donde se crean, se abren y se utilizan. (Ver figura 1.4.1.)

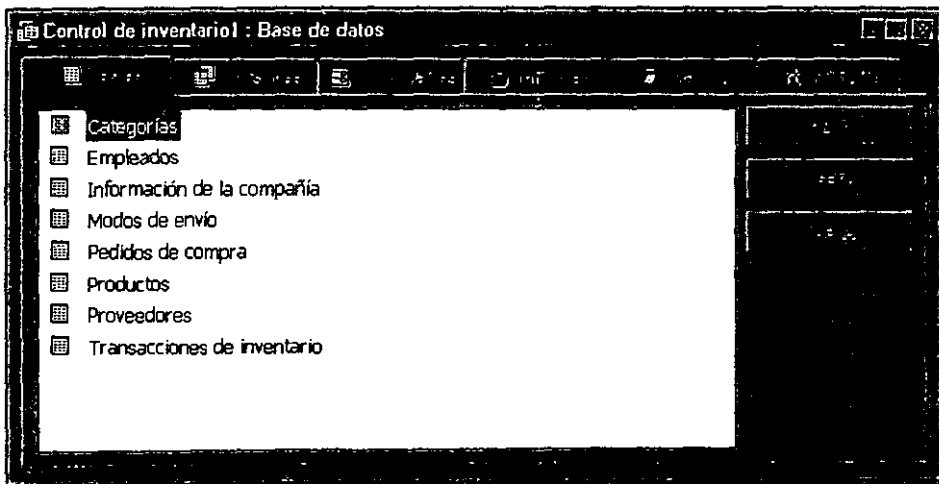


Figura 1.4.1. Objetos de Access.

La estructura de Access, no es muy compleja, sin embargo requiere de varios elementos que lo hacen que sea potente y al mismo tiempo fácil de manejar por los usuarios, desde bases de datos sencillas como una agenda o directorio telefónico hasta aquellas grandes y complejas como un inventario o una base de datos que controle las compras, ventas, inventario y nómina de una empresa. La figura 1.4.2. nos muestra la estructura de Access.

Diseño de una base de datos.

Para poder obtener el máximo de provecho de las herramientas con que cuenta Access, es importante saber diseñar correctamente una base de datos.

Los pasos a seguir para el diseño de una base de datos son:

1. Determinar el propósito de la base de datos.
2. Determinar las tablas.
3. Determinar los campos.
4. Determinar las relaciones entre tablas.
5. Depurar el diseño.

Podemos mencionar además que una base de datos está formada por una o mas tablas. Por lo general, varias tablas que se relacionan entre sí de alguna manera conforman una base de datos. Una tabla contiene filas y columnas. Ver figura 1.4.3.

Al utilizar Microsoft Access, se puede administrar toda la información desde un único archivo de base de datos. Dentro de este archivo, se dividen los datos en contenedores de almacenamiento separados denominados tablas; en las cuales se pueden ver, agregar y actualizar los datos mediante formularios en línea; buscar y recuperar solamente los datos que se necesiten mediante consultas, analizar o imprimir los datos con el esquema deseado mediante informes.



Bases de datos: Una base de datos es un conjunto de información relacionada con un asunto o con una finalidad



Tablas: Una tabla es un conjunto de datos sobre un tema específico. Las tablas organizan los datos en columnas (denominados campos) y filas (denominados registros).



Consultas: Las consultas se usan para ver, cambiar y analizar datos de distintas maneras. También se pueden usar como origen de registros para formularios e informes.



Formularios: Los formularios pueden crearse para la entrada de datos a una tabla, para ejecutar acciones con los datos introducidos en el formulario, o para abrir otros formularios o informes.




Informes: Un informe es una forma efectiva de presentar los datos en formato impreso. Como tiene control sobre el tamaño y el aspecto de todos los elementos de un informe, puede presentar la información en la forma que se desee verla.

Access



Macros: Una macro es un conjunto de una o más acciones para realizar una operación determinada, tal como abrir un formulario o imprimir un informe.



Módulos:	Un módulo es una colección de declaraciones y procedimientos de Access Basic para aplicaciones que se almacenan juntos como una unidad.
Eventos:	Un evento es una acción determinada que se produce en, o con, un objeto particular. Microsoft Access puede responder a una variedad de eventos: clics del mouse, cambios en los datos, formularios que se abren o se cierran, y muchos otros. Los eventos son normalmente el resultado de una acción del usuario.

Figura 1.4.2. Estructura de Access.

El poder de manejo de una base de datos relacional como Access incide en su rápida búsqueda y extracción de información almacenada en tablas separadas.

Características de las tablas.

Antes de crear cualquier otro objeto de una base de datos, se deben crear primero las tablas y la definición de tabla se ha proporcionado en la figura 1.4.2, la cual hace referencia a una colección de datos acerca de un tema en particular y los datos son representados en forma matricial en columnas y renglones o campos y registros respectivamente.

Cada fila de una tabla es un registro. El registro está formado por toda la información relacionada con una entrada de la tabla.

Cada columna de la tabla es un campo. Un campo es la porción más pequeña de información que contiene un registro, tal como se muestra en la figura 1.4.3.

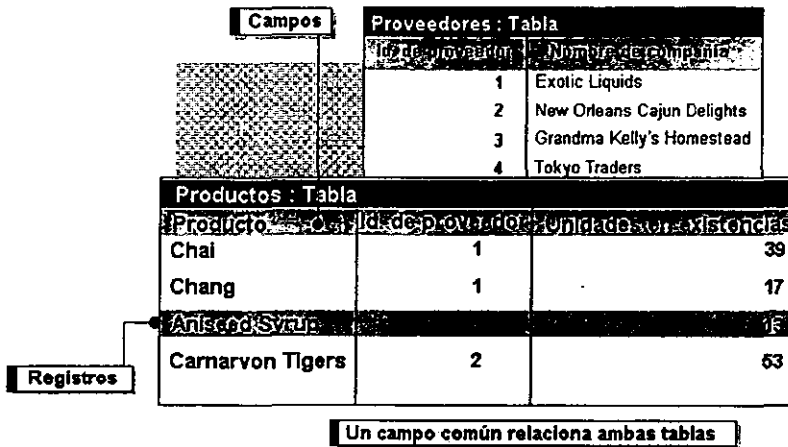


Figura 1.4.3. Estructura de una tabla de Microsoft Access.

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

Se definen relaciones entre las tablas para recuperar datos de varias tablas en una consulta, formulario o informe.

Al diseñar las tablas es necesario especificar un identificador exclusivo de registros, mismo que se le denomina llave primaria. Esta consiste en uno o más campos que identifican a cada registro almacenado en la tabla.

Las reglas de validación que establezca para las tablas y sus campos siempre estarán en vigor, tanto si cambian los datos en un formulario, como si importa nuevos datos o utiliza una consulta de acciones, una macro o código de Access para cambiar datos.

Estableciendo las propiedades de los campos en el modo de presentación Diseño de tabla, es posible controlar la apariencia de los datos, evitar su introducción incorrecta, especificar valores predeterminados y acelerar la búsqueda y la ordenación en la tabla.

Características de las relaciones entre tablas

Cuando se crea una tabla, ésta no está relacionada con las demás tablas de la Base de datos. Para crear una relación, se debe abrir la ventana de relaciones y añadir las tablas o consultas que se desee relacionar.

Microsoft Access, cuenta con una ventana gráfica llamada Relaciones, que permite realizar de forma visual un boceto del diseño de la base de datos y observar simultáneamente todas las relaciones de la base de datos.

Gracias a las actualizaciones y eliminaciones en cascada, ahora puede actualizar un campo de una tabla y hacer que Microsoft Access actualice automáticamente los campos relacionados de las tablas relacionadas. También puede eliminar un registro y hacer que Microsoft Access elimine automáticamente los registros relacionados de las tablas relacionadas.

Características de las consultas.

Una consulta es la respuesta a una serie de preguntas acerca de datos almacenados en tablas e incluso en consultas.

Microsoft Access, cuenta con el asistente para consultas, que nos ayuda a crear consultas complejas para tareas frecuentes de administración de datos.

Se crean consultas para buscar y recuperar exactamente aquellos datos que cumplen unas determinadas condiciones que se especifican. Una consulta también puede actualizar o eliminar múltiples registros al mismo tiempo, así como realizar sobre los datos diversos cálculos incorporados o personalizados. Figura 1.4.4.

Aunque no se hayan establecido relaciones entre tablas, Access crea automáticamente combinaciones en una consulta entre tablas que contienen campos coincidentes.

Cuando una consulta incluye campos de varias tablas relacionadas, pueden actualizarse los datos a ambos lados de la combinación. También se puede definir una consulta de selección para limitar el número de registros devueltos.

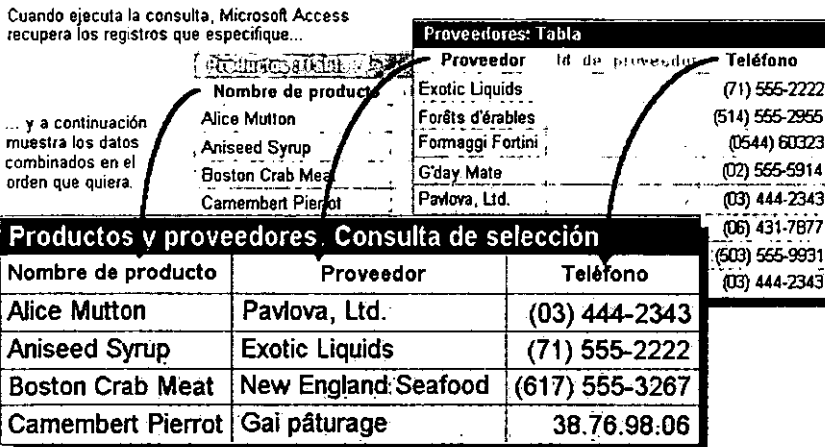


Figura 1.4.4. Consulta de Microsoft Access.

Existen tres modos de presentación de consultas:

- i) Diseño: se utiliza para crear nuevas consultas o modificar las ya existentes
- ii) Hoja de datos: se usa para la presentación de datos recuperados de una consulta y en algunos casos agrega y edita datos.

- iii) Presentación preliminar: para ver la apariencia que tendrán los datos de la consulta una vez impreso.
- iv) Presentación SQL: para cuando se desee crear o modificar una consulta mediante inscripción SQL.

Access cuenta con los siguientes tipos de consultas:

- Selección: son las más comunes, aquí se plantean preguntas sobre datos almacenados en las tablas y la respuesta es una hoja de datos, en donde se puede ver y cambiar los datos de las tablas subyacentes.
- Referencias cruzadas: presenta los datos con títulos en las filas y columnas, como en una hoja de cálculo. Con una consulta de tabla de referencias cruzadas se puede resumir gran cantidad de información en un formato de fácil lectura.
- Consulta de unión: combina campos coincidentes de dos o más tablas.
- Consultas de definición de datos: crea, modifica o elimina tablas de una base de datos Access utilizando inscripción de SQL.
- Acción: modifican muchos registros en una sola operación. Se utiliza para crear una nueva tabla, eliminar, agregar o modificar registros. A partir de la selección actualizan, agregan y eliminan datos. Access provee cuatro tipos de consultas de este tipo que son: consultas de creación de tablas, de eliminación, de agregación de datos y de actualización.

A la hora de diseñar las consultas, veremos que son muy flexibles, ya que podremos:

- Elegir campos de una o más tablas.
- Elegir registros mediante criterios que deben cumplir los registros para que se incluyan en la hoja de respuesta dinámica de la consulta.

- Ordenar registros.
- Formular preguntas sobre datos de varias tablas.
- Realizar cálculos, es decir, crear nuevos campos que contengan el resultado de un operación aritmética sobre algún campo de todos o algunos registros.
- Usar una consulta como origen de los datos para formularios, informes y otras consultas. Utilizando una consulta se pueden incluir datos de varias tablas y establecer criterios para mostrar sólo un conjunto limitado de datos. También se puede introducir nuevos datos o modificar los existentes directamente en la hoja de datos de la consulta, o bien en un formulario que esté basado en esta consulta.
- Modificar los datos de las tablas, con las consultas de acciones puede actualizar, eliminar o agregar un grupo de registros simultáneamente. También puede utilizarlas para crear una nueva tabla que contenga registros de otra tabla o de un grupo de tablas existentes.

Características de formularios.

La creación de un formulario nos permite ver, introducir o cambiar datos directamente en una tabla de una manera sencilla. Cuando se abre un formulario, Microsoft Access recupera los datos de una o más tablas y los muestra en la pantalla usando el esquema que se eligió en el Asistente para formularios o usando un esquema creado desde el principio. Figura 1.4.5.

Con Microsoft Access se pueden diseñar formularios fáciles de utilizar y que presenten la información de la forma deseada. Se pueden utilizar diversos elementos de diseño (texto, datos, imágenes, líneas y colores) para crear exactamente el formulario deseado.

El uso de formularios también es un modo eficiente de introducir los datos, pues ahorra tiempo y evita errores de mecanografía.

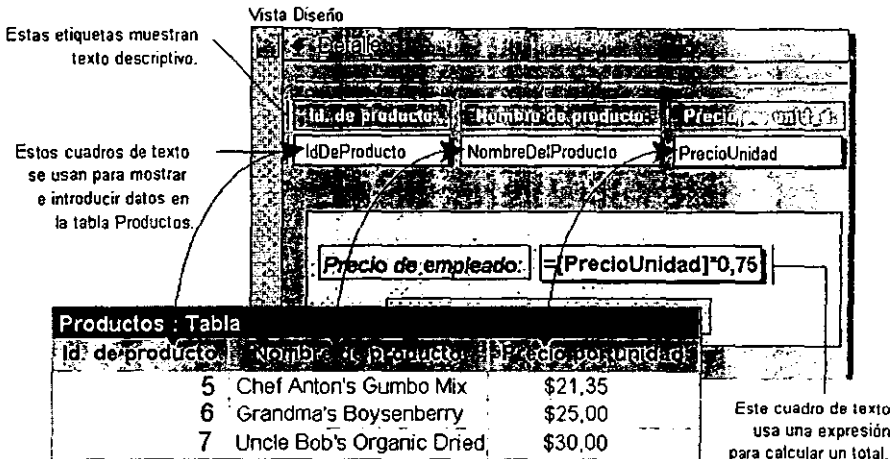


Figura 1.4.5 Formulario de Microsoft Access.

Los formularios se pueden presentar de tres modos diferentes:

- El modo de Diseño se utiliza para crear nuevos formularios o modificar los ya existentes.
- El modo de formulario se utiliza para introducir, modificar y ver datos. En este modo generalmente se pueden ver al mismo tiempo todos los campos correspondientes a un registro.
- El modo de hoja de datos se utiliza para introducir, modificar y ver datos utilizando para ello una hoja de datos. En este modo de presentación se pueden ver muchos registros al mismo tiempo, pero puede que la hoja de datos sea mas ancha que la pantalla, en cuyo caso no se podrán ver todos los campos correspondientes a cada registro sin desplazarse por el formulario.

- El modo de presentación preliminar se utiliza para ver la apariencia que tendrá un formulario cuando se imprima.

Los formularios proporcionan una forma sencilla de ver los datos. Se pueden ver todos los valores de un registro en el modo presentación formulario o bien se puede cambiar al modo de hoja de datos para ver todos los registros del formulario.

El formulario ofrece el diseño más cómodo para introducir, modificar y ver los registros de la base de datos. Las herramientas de diseño para formularios de Access ayudan a diseñarlos de forma fácil y útil para:

- Presentar los datos en un formato atractivo, con fuentes especiales y otros efectos gráficos, tales como colores y sombreado.
- Ofrece una apariencia similar a los formularios en papel.
- Calcular totales y presentarlos.
- Contener gráficas de los datos presentados.
- Mostrar datos de varias tablas.
- Automatizar tareas que se realizan habitualmente.

Otros elementos de los formularios son los subformularios que son un método para incluir en un formulario información que proviene de más de una tabla. Un subformulario es un formulario dentro de otro formulario.

En Access, el formulario primario se denomina formulario principal y el formulario incluido recibe el nombre de subformulario.

Al utilizar un subformulario es sencillo ver la relación existente entre los registros de dos o más tablas.

Características de los informes

Un informe es un conjunto de información que se puede organizar según determinados criterios y al que se le aplica formato de acuerdo con determinadas especificaciones.

Un informe constituye una forma de recuperar y presentar los datos como información significativa que se puede usar y distribuir. Ver figura 1.4.6.

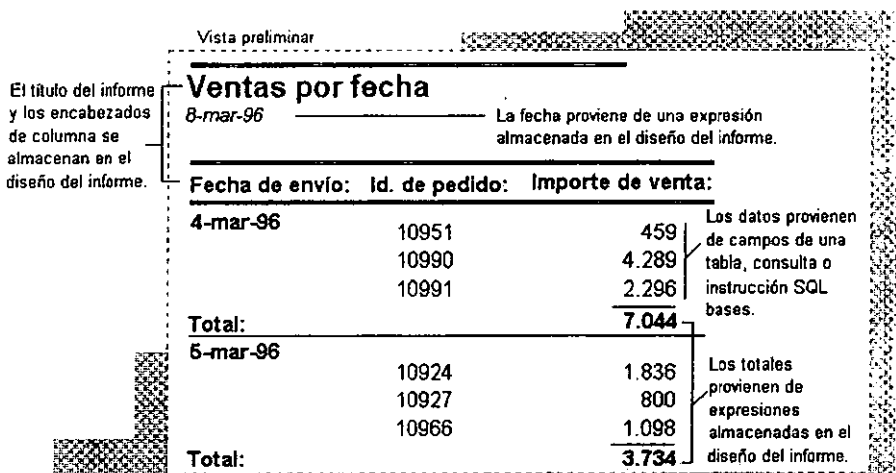


Figura 1.4.6. Informe de Microsoft Access.

Con Access se pueden diseñar informes que presenten la información de la forma deseada. Se puede utilizar numerosos elementos de diseño (texto, datos, imágenes, líneas, cuadros y gráficos) para crear exactamente el informe que necesita.

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

Los informes, al igual que otros elementos de Access se presentan de diversos modos:

- El modo de diseño, se utiliza para crear un informe o modificar alguno ya existente.
- El de presentación preliminar se utiliza para comprobar los datos de todo el informe.
- El modo muestra preliminar se utiliza para comprobar la fuente, el tamaño de la fuente y el diseño general de un informe.

Access ofrece varios métodos para obtener información de la base de datos: utilizar una consulta, formulario o un informe, por lo que se deberá seguir el método mas adecuado para la tarea que se requiera realizar.

Los informes se pueden crear para:

- Organizar y presentar los datos en grupos.
- Calcular totales parciales, totales de grupo, sumas de totales y porcentajes de totales.
- Incluir subformularios, subinformes y gráficas.
- Presentar los datos en un formato atractivo, con imágenes, líneas y fuentes especiales.

Características de las macros

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

Access ofrece una lista de acciones, entre las que se puede elegir una o más para crear las macros. Cuando se ejecuta la macro, Access realizará todas las acciones en el mismo orden en que figuran dentro de la macro, utilizando los objetos o los datos especificados.

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

Las macros también son útiles para automatizar tareas rutinarias o repetitivas, como la impresión de informes semanales. Además, las macros ofrecen versátiles posibilidades de personalización tales como realizar complejas validaciones de los datos introducidos en formularios.

Las macros nos permiten:

- Hacer que los formularios e informes funcionen conjuntamente.
- Buscar y filtrar registros automáticamente.
- Establecer la exactitud de los datos, las macros también son idóneas para el manejo de datos y para su validación en formularios.

- Establecer propiedades de formularios, informes y controles. Se pueden utilizar las macros para establecer la mayor parte de las propiedades de formularios, informe y control.
- Automatizar transferencias de datos, es decir para importar o exportar datos entre diferentes formatos de archivos.
- Crear nuestro propio entorno de trabajo personalizado.

Características de los módulos

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

Para mayor información del código, consultar el manual de usuario de Microsoft Access.

Desarrollo de Access

La primera versión de Microsoft Access apareció en 1992, como el primer sistema manejador de bases de datos relacionales (RDBMS) diseñado para el sistema operativo Windows. Durante su vida, Microsoft Access ha continuado mejorando sus características y sobre todo en rapidez, facilidad y manejo de datos.

Cada versión que aparece de Access, ha introducido nuevas tecnologías revolucionadas que han cambiado el aprendizaje de los usuarios y del manejo de las bases de datos. Access es el primer manejador de bases de datos que soporta OLE

Automation (Automatización OLE) que permite a los usuarios compartir información entre aplicaciones. Access versión 2.0 introduce herramientas tales como relaciones gráficas y constructores de consultas. Las formas de Access y el Asistente para Reportes ayudan a los usuarios con poca o nada de experiencia a ser productivos rápidamente.

Como usuarios actualizados a Windows 95, Access es el primer RDBMS de 32 bits e introduce tecnologías tales como las tablas y **Performance analyzers** (Analizadores de funcionamiento) y **Briefcase replication** (Portafolio de replicación), quienes manejan algunas de las tareas más complejas en el manejo de bases de datos automáticamente. Estas características crearon dificultades históricas de la tecnología de acceso a bases de datos para que los usuarios generaran negocios.

Microsoft Access continua su tradición de innovación. Aunque no se sabe si los usuarios estén conectados por una red de área local o a través de Internet o estén trabajando en Microsoft, Access se asegura que los usuarios puedan darse cuenta de los beneficios de usar una base de datos rápidamente. Con estas nuevas tecnologías integradas Access fue diseñado para hacer más fácil a todos los usuarios encontrar respuestas, compartir información al mismo tiempo y crear soluciones rápidas.

Nuevas Características de Access

Las nuevas características de Access se enfocan en las herramientas que lo forman, las cuales están organizadas en tres grupos:

- Herramientas que ayudan a los usuarios a encontrar respuestas fácilmente.

Con las herramientas que ayudan a los usuarios a encontrar respuestas fácilmente, Microsoft Access introduce nuevas características que hacen a los usuarios obtener respuestas a sus preguntas rápida y fácilmente.

Históricamente, las bases de datos han sido una herramienta poderosa de los usuarios, al igual que los manejadores de información y desarrolladores de soluciones, también, cabe mencionar que las computadoras personales han sido trasladadas de los laboratorios de computadoras a los hogares y pequeños negocios. Sin embargo, las necesidades de bases de datos de los usuarios han cambiado.

Hoy los usuarios de bases de datos manejan muchos grupos de datos o uno simple, que deben ser compartidos entre muchos usuarios. Los datos han ido creciendo, no solamente con texto o números, sino datos visuales y audibles. Los usuarios demandan una aplicación intuitiva y fácil de usar que ayude a obtener el mejor valor de su información.

Elementos nuevos:

Asistente de Office: Es un paso tecnológico que unifica el asistente en línea del usuario, permite al usuario realizar preguntas en palabras relacionadas y proveer interactivamente sugerencias útiles.

El asistente es controlado a través de la automatización del lenguaje Microsoft Visual Basic para construcción de aplicaciones desarrolladas basadas sobre Access pudiendo tener ventajas sobre su funcionalidad.

Elementos mejorados:

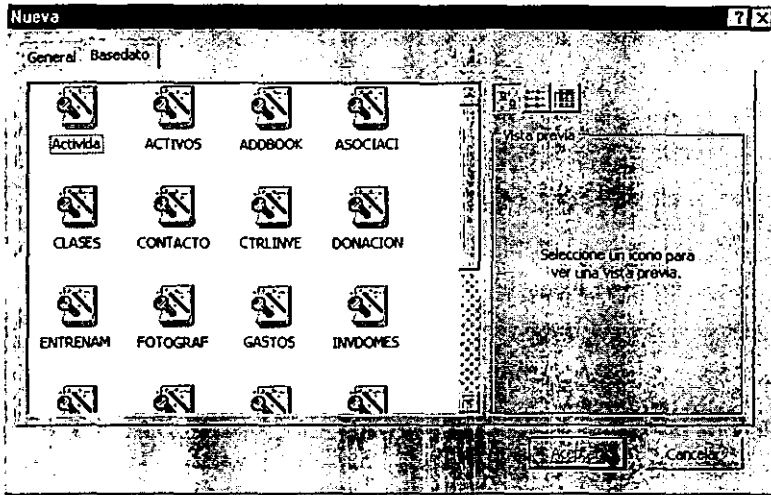


Figura 1.4.7 Asistente para crear bases de datos empleando plantillas.

Asistente de Consultas: Esta ayuda de usuario localiza exactamente la información que se busca, no importando el nivel de experiencia en bases de datos en una forma rápida.

Asistente de Base de Datos: Esta herramienta permite a los usuarios sin ninguna experiencia en bases de datos crear una o más de 20 aplicaciones de bases de datos funcionalmente completas.

Filtros por Forma: Los usuarios pueden localizar la información que buscan para visualizarla, usando las formas, hojas de datos o formularios que trabajan diariamente.

Asistente Analizador de Tablas: Esta herramienta analiza la estructura de los archivos de datos llanos, es el encargado del proceso de normalización, además de que realiza y brinda recomendaciones para mejorar la base de datos relacional que el usuario puede aceptar o rechazar.

Asistente de Formas y Reportes: Estas herramientas crean tablas y reportes de información de bases de datos con pequeñas intervenciones de los usuarios.

Además dentro de Microsoft Access, pueden utilizarse máscaras de entrada para definir patrones que deban seguir los datos cuando se introduzcan en formularios y hojas de datos, lo que confiere una mayor rapidez y precisión a la introducción de estos.

Microsoft Access, puede ordenar rápidamente los registros de tablas, consultas y formularios, en forma ascendente o descendente. Puede ajustar la anchura de una columna de la hoja de datos para que se adapte a su valor mas ancho. Además puede imprimir información detallada sobre el diseño de cualquier objeto de la base de datos.

Herramientas para compartir información al mismo tiempo

Hoy en día acceder a la información correcta en el momento correcto puede significar la diferencia entres sucesos y fallas. La receta surge con la popularidad de Internet e Intranets. Como resultados Microsoft Access incluyó nuevas características que permiten a los usuarios obtener mas de sus ambientes computacionales ya instalados.

Desde un principio, las bases de datos han sido herramientas que pueden ser usadas por mas de una persona. El propósito de las bases de datos es proveer una localización central para almacenar piezas de información relacionadas. Esta información puede ser organizada o vista en muchas formas como aplicaciones para datos.

Las extensiones naturales de la funcionalidad de base de datos están disponibles en Microsoft Access ayudando a compartir a los usuarios información con un grupo de trabajo, en una Intranet o sobre Internet.

Microsoft Access es una de las primeras aplicaciones de bases de datos que soporta el almacenamiento de hiperligas como un tipo nativo, la cual ofrece muchos beneficios como ir a los URLs o bien para localizar otros documentos de Office 97.

Elementos nuevos:

Hiperligas: los usuarios pueden almacenar hiperligas en todas las bases de datos de Access para conectarse fácilmente a información no encontrada en el lugar donde reside.

Salvar a HTML: los usuarios pueden publicar vistas estáticas de datos para su grupo de trabajo o para el WEB.

Asistente para publicar en el WEB: Esta herramienta automatiza las publicaciones de información de bases de datos dinámicamente en el WEB

Importación y ligado de archivos HTML: Los usuarios pueden ahora importar o atar sus bases de datos a páginas HTML y usar esta información dentro de las aplicaciones de Microsoft Access.

Replicación en Internet: Los usuarios pueden extender la capacidad de replicación introducida en Access 95 bajo Internet usando FTP (Protocolo de transferencia de archivos).

- **Herramientas para la creación de soluciones rápidas**

En suma para engrandecer el rendimiento, Access ha mejorado substancialmente el ambiente en Access y trae nuevas herramientas orientadas al desarrollo. Estas mejoran el camino de los desarrolladores en rapidez y cuando crean sus soluciones.

Nuevas herramientas:

Mejora el rendimiento: cambia el diseño de aplicaciones y la estructura de objetos, resultando mejoras del rendimiento del sistema en más del 50%.

Replicación parcial de tablas: Los usuarios ahora pueden replicar un grupo seleccionado de registros de cualquier tabla.

Control de código fuente: Múltiples desarrolladores pueden crear ahora soluciones y mantener código y objetos consistentes usando el control de código fuente.

Herramientas mejoradas:

Asistente Analizador de Rendimiento: esta herramienta analiza la estructura de la base de datos y sugiere las áreas para mejorarla.

Visual Basic para aplicaciones: Engrandecimiento del ambiente de diseño integrado haciendo de Access un ambiente de desarrollo más robusto para las aplicaciones de base de datos.

Como parte de Microsoft Office 97, Access ofrece la posibilidad de manejar bases de datos obtenidas a través de Internet. Copiadas de la red a un equipo en particular. Lo interesante es que permite ligar la base de datos a una hoja WEB, haciendo que cada vez que sea consultada, el sistema la actualice automáticamente accedando la página correspondiente en WEB, sin importar si ocupa una o más páginas.

Comentando algunas desventajas de Microsoft ACCESS

Este manejador de bases de datos posee algunas desventajas las cuales se comentarán :

- No maneja objetos o librerías compartidas: Los objetos compartidos entre aplicaciones son almacenados en módulos que tienen que ser llamados o cargados desde cada aplicación, por lo que se pierden los beneficios de un solo punto de control.
- Uso de funciones no estandarizadas : El soporte de SQL esta disponible únicamente a través del dialogo del query, los controladores ODBC no proveen el performance (desempeño) que proporcionan los controladores nativos, por lo que no se pueden explotar todas las características únicas de los servidores de las bases de datos. Entre las que destacan: vistas, procedimientos interconstruidos sinónimos y números de secuencia.
- Vulnerabilidad de las bases de datos e índices: Las bases de datos e índices de Access, son susceptibles de corromperse debido a desconexiones de la red o problemas de los equipos de computo, además de que las operaciones necesarias para reparar las bases de datos y reconstruir los índices son extremadamente lentas.
- No puede generar programas ejecutables por si mismo; requiere de un módulo adicional para hacerlo.

Especificaciones de uso de Microsoft Access.

Para utilizar Microsoft Access se necesita:

- Computadora personal procesador 486 o superior (con multimedia de preferencia).
- Sistema operativo Windows 95, Windows NT como estación de trabajo 3.51 o superior.

- 12 MB de memoria RAM para Windows 95 y 16 para estaciones de trabajo Windows NT.
- De 28 a 60 MB de espacio en disco duro dependiendo de la configuración y 40 para la estación típica.
- Manejador de CD-ROM
- CD-ROM que contiene Internet Explorer, manejadores adicionales y archivos con extensión AVI.
- Adaptador de vídeo VGA mínimo (se recomienda super VGA a 256 colores).
- Microsoft Mouse. Microsoft IntelliMouse o cualquier puntero compatible.

Opciones o servicios adicionales para usar ciertas características:

- Módem con una velocidad de 9600 bauds (se recomienda a 14400 bauds)
- Computadora multimedia, cuando se requieren efectos de sonido u otros efectos en Access.
- La característica de publicación al WEB requiere Microsoft Information Server para Windows NT o Microsoft Personal WEB Server para Windows 95.
- Alguna funcionalidad de Internet puede requerir acceso a Internet y el pago del servicio por separado.

Formatos de datos soportados:

- Directamente importados, exportados o ligados a : Excel versión 3.0 o superior, Microsoft FoxPro versión 2.x o superior, Microsoft SQL Server, Dbase III plus, Dbase IV o 5.0 de Borland, Paradox 3.0 a 5.0 de Borland, texto ASCII y todas las bases de datos relacionados por ODBC.
- Directamente importados y exportados a Microsoft Visual FoxPro versión 3.0 y Lotus 123.

1.5. Características, ventajas y desventajas de los paquetes de soporte: Power Point, AutoCAD r13, Photo Styler, Photo Enhancer, y CD-ROM Guía Roji.

El desarrollo de aplicaciones con interface gráfica de usuario (medio de comunicación entre el usuario y la aplicación) son aplicaciones en el entorno de Windows que generalmente cuenta con una interface gráfica consistente e intuitiva, por lo que permite disponer de más tiempo para dominar la aplicación sin preocuparse por memorizar comandos y opciones. Dando pie a un gran número de paquetes de software, de los cuales se utiliza una parte para el desarrollo del sistema.

POWER POINT

Es un paquete completo de herramientas para presentaciones gráficas, que consta de: procesamiento de texto, esquemas, dibujos, gráficos, imágenes prediseñadas, etc. En la figura 1.5.1 se identifica cada parte que compone la ventana de Power Point.

Barras de desplazamiento

Mueven la ventana de edición horizontal o verticalmente, o cambian a la diapositiva siguiente o anterior cuando se arrastra el elevador o se hace clic en los botones con flechas.

Barra de estado

Proporciona la información actual sobre la diapositiva seleccionada, incluyendo lo que hacen los botones o los comandos, qué diapositiva se está mostrando actualmente y qué Plantilla de diseño se ha aplicado a la diapositiva.

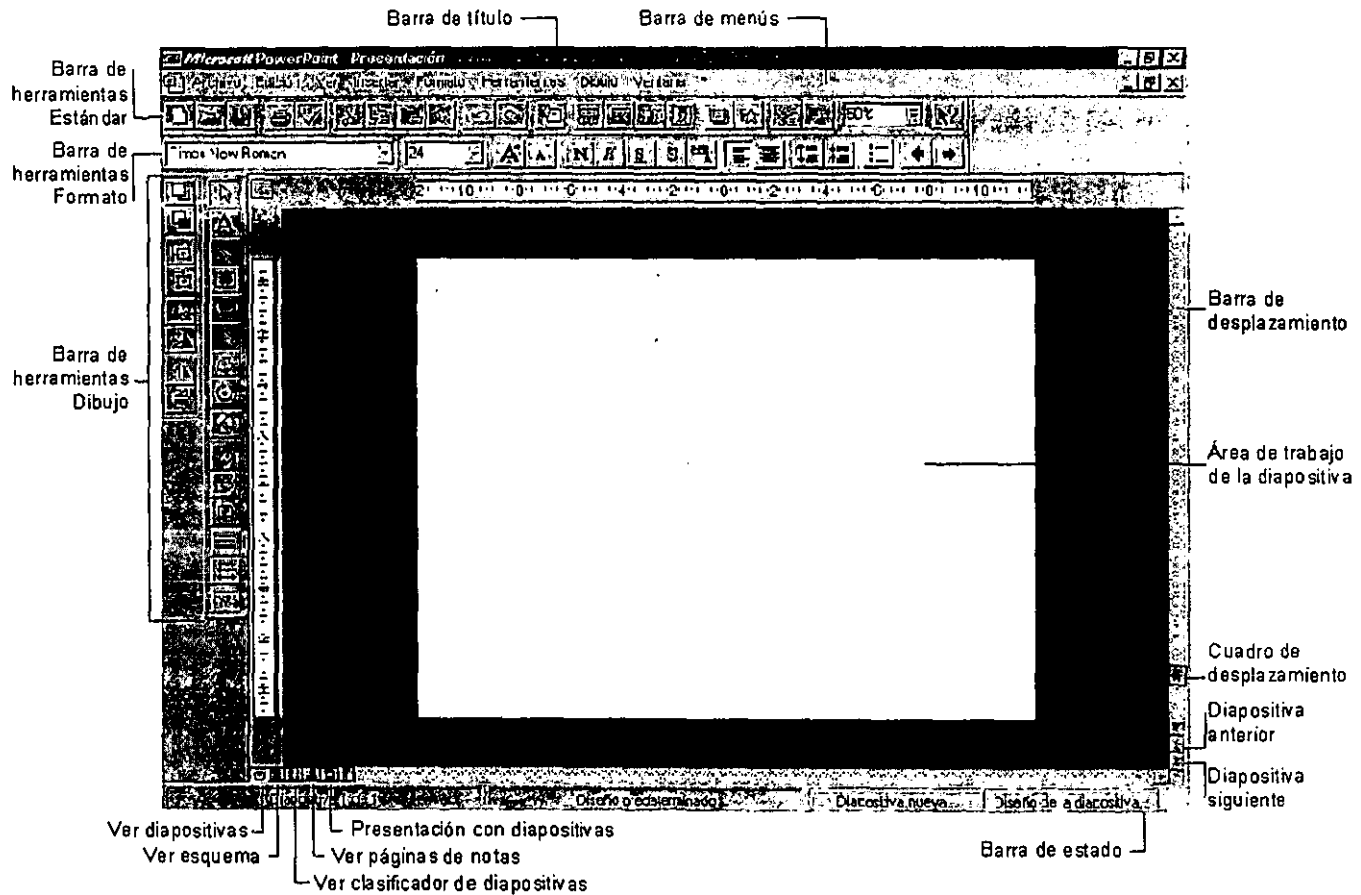


Figura 1.5.1. Ventana principal de Power Point.

Barras de herramientas

Las barras de herramientas contienen botones que le permiten acceder rápidamente con el mouse a numerosos comandos y características.

Barra de herramientas Estándar (fig. 1.5.2.)

Presentación nueva. Crea una nueva presentación en blanco sin presentar el cuadro de dialogo.

Abrir. Abre una presentación existente.

Guardar. Guarda los cambios en la presentación actual.

Imprimir. Imprime la presentación en el formato actual.

Ortografía. Corrige la ortografía de la presentación completa.

Cortar. Quita el texto u objetos seleccionados y los coloca en el portapapeles para poder pegarlos en otro sitio.

Copiar. Realiza una copia del texto u objetos seleccionados y los coloca en el portapapeles para poder pegarlos en otro sitio.

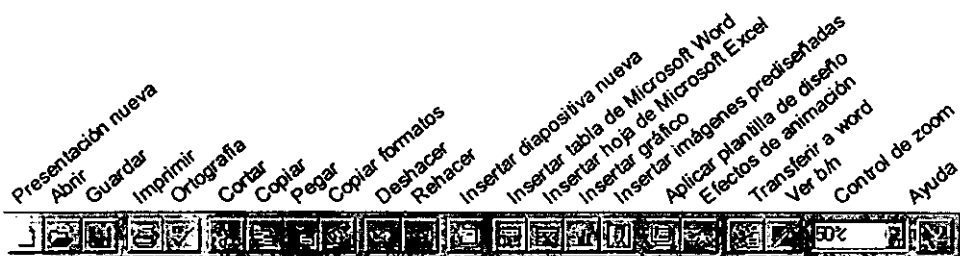


Figura 1.5.2. Barra de herramientas Estándar

Pegar. Inserta el texto u objeto que se encuentra en el portapapeles.

Copiar formatos. Copia el formato del texto u objeto seleccionado y lo aplica a otro.

Deshacer. Anula la última acción.

Rehacer. Anula la última acción de deshacer.

Insertar diapositiva nueva. Para insertar una nueva diapositiva con el diseño que prefiera, después de la diapositiva actual.

Insertar tabla de Microsoft Word. Le permite insertar una tabla en blanco de Microsoft Word con el número de filas y columnas que especifique.

Insertar tabla de Microsoft Excel. Agrega una hoja de cálculo en blanco de Microsoft Excel a su diapositiva en modo de ver diapositivas.

Insertar gráfico. Para incrustar una gráfica en una diapositiva.

Insertar imágenes prediseñadas. Existe una Galería de imágenes prediseñadas para seleccionar la imagen a insertar en la presentación o para actualizar dichas imágenes.

Aplicar plantilla de diseño. Para aplicar en una presentación o para usar una de sus propias presentaciones como plantilla.

Efectos de animación. Se crean progresiones, objetos y textos animados para una presentación con diapositivas.

Transferir a Word. Exporta un esquema desde Microsoft PowerPoint a Microsoft Word, donde se puede modificar usando los comandos de Word.

Ver B/N. Muestra la presentación activa en blanco y negro.

Control de zoom. Controla con qué tamaño aparecerá en la pantalla una presentación.

Ayuda. Permite obtener ayuda contextual acerca de los diferentes elementos de la pantalla y comandos del programa.

Barra de herramientas de Formato (fig. 1.5.3.)

Fuente. Permite cambiar la fuente del texto seleccionado o texto del marcador de posición de texto activo.

Tamaño de la fuente. Permite cambiar el tamaño del texto seleccionado o texto del marcador de posición de texto activo.

Aumentar el tamaño de la fuente. Para aumentar el tamaño de la fuente del texto seleccionado en el 10 por ciento del valor actual.

Disminuir el tamaño de la fuente. Para disminuir el tamaño de la fuente del texto seleccionado en el 10 por ciento del valor actual.

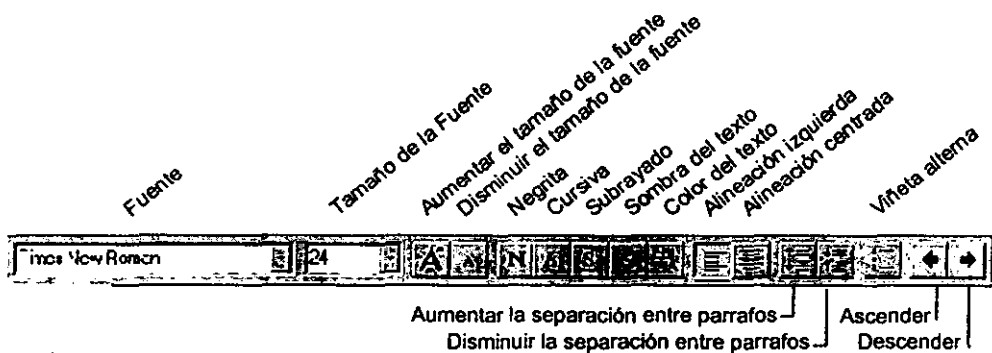


Figura 1.5.3. Barra de herramientas de Formato.

Negrita. Para poner el texto seleccionado en negrita o para eliminar el formato de negrita.

Cursiva. Para poner el texto seleccionado en cursiva o para eliminar el formato de cursiva.

Subrayar. Para subrayar el texto seleccionado o para eliminar el subrayado.

Sombra del texto. Para agregar una sombra o eliminar una sombra del texto seleccionado.

Color de texto. Para cambiar el color del texto seleccionado.

Alineación izquierda. Ajusta el texto seleccionado al margen izquierdo, dejando el derecho irregular.

Alinear centro. Centra cada línea del texto seleccionado.

Aumentar la separación entre párrafos. Agrega más espacio entre los párrafos seleccionados.

Disminuir la separación entre párrafos. Reduce la cantidad de espacio entre los párrafos seleccionados.

Viñeta (alterna). Para mostrar u ocultar las viñetas de los párrafos seleccionados.

Ascender. Para mover un párrafo seleccionado a la izquierda, es decir, un nivel hacia arriba.

Descender. Para mover un párrafo seleccionado a la derecha, es decir, un nivel hacia abajo.

Barra de herramientas de Formato (fig. 1.5.4.)

Selección. Le permite seleccionar objetos gráficos y de texto.

Texto. Para agregar texto fuera de un marcador de posición o de una Autoforma. Para agregar texto que no se ajuste automáticamente (como una leyenda).

Línea. Para dibujar una línea.

Rectángulo. Para dibujar un rectángulo.

Elipse. Para dibujar una elipse.

Arco. Para dibujar un arco.

Forma libre. Para dibujar polígonos cerrados o abiertos, dibujos a mano alzada o combinaciones de polígonos y dibujos a mano alzada.

Girar libremente. Gira un objeto seleccionado en cualquier grado.

Autoformas. Para dibujar una forma.

Color de relleno. Rellena un objeto seleccionado con un relleno de color, textura, entramado o sombreado. También puede eliminar un relleno o cambiar un color de relleno.

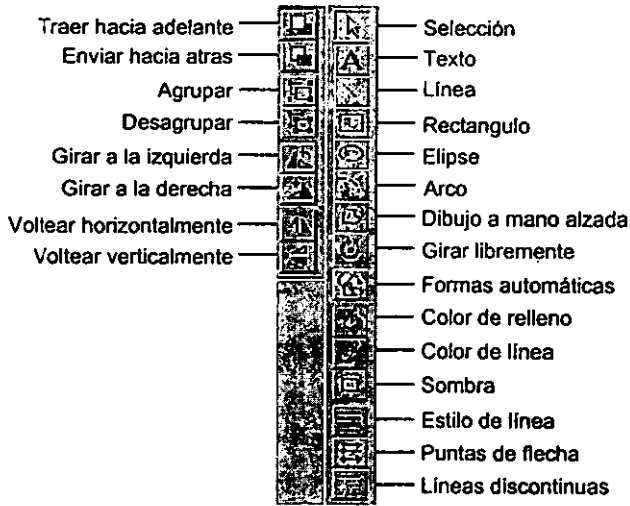


Figura 1.5.4. Barra de herramientas de Dibujo.

Color de línea. Para cambiar el color de la línea seleccionada o para quitar el color de la línea.

Sombra. Para agregar o eliminar una sombra de objeto.

Estilo de línea. Para cambiar el grosor de la línea o transformarla en una línea compuesta.

Puntas de flecha. Para agregar puntas de flecha a la línea, el arco o la forma libre abierta seleccionada.

Líneas discontinuas. Cambia el estilo de la línea, el arco o el borde seleccionado a un estilo de guiones.

Traer adelante. Para mover un objeto o un grupo un lugar hacia la parte superior de una pila de objetos.

Enviar atrás. Para mover un objeto o un grupo un lugar hacia la parte inferior de una pila de objetos.

Agrupar. Para crear un conjunto de objetos con el que pueda trabajar como si fuera con un único objeto.

Desagrupar. Para separar un conjunto de objetos agrupados.

Girar a la izquierda. Para girar un objeto o grupo de objetos seleccionado 90 grados a la izquierda.

Girar a la derecha. Para girar un objeto o grupo de objetos seleccionado 90 grados a la derecha.

Voltear horizontalmente. Para voltear un objeto o grupo de objetos seleccionado 180 grados horizontalmente.

Voltear verticalmente. Para voltear un objeto o grupo de objetos seleccionado 180 grados verticalmente.

Ventajas:

- Por ser un software desarrollado en Windows puede vincularse con otras aplicaciones.
- Nuevo asistente para ayuda que emplea tecnología IntelliSense, para determinar el tipo de ayuda necesaria.
- Contiene un nuevo autocorrector para corregir errores de escritura típicos.
- Cuenta con un sistema de autoimagen, el cual le permite que Power Point le sugiera imágenes prediseñadas.
- Power Point puede corregir sus presentaciones por medio del corrector de estilos.
- Puede configurar criterios de búsqueda para localizar una presentación.
- Contiene una opción para observar nuestra imagen en blanco y negro, utilizado para observar la imagen antes de ser impresa (Impresoras de una sola tinta).
- Contiene una nueva ayuda para notas de la presentación.
- Power Point incluye sonidos, música, textura, fotos, y elementos de multimedia.
- Contiene nuevas herramientas de presentación con diapositivas.
- Nuevas opciones para dar fondo una presentación.
- Contiene varias opciones de deshacer.

Desventajas:

- Maneja una cantidad mínima de formatos de archivos.
- En edición de imágenes importadas su manipulación es mínima.
- Para esta versión requiere un sistema mayor del procesador 80486, con una memoria mínima de 16 Mb., y espacio libre en disco duro de 14 Mb., Y en ambiente Windows'95.

AutoCAD R13

Las imágenes visuales para expresar ideas se utilizaron antes que el lenguaje escrito. Los primeros ingenieros diseñaron la construcción de antiguas estructuras por medio de dibujos, actualmente han cambiado las herramientas para crear dibujos. Una de estas nuevas herramientas es CAD, que significa diseño asistido por computadora. En la figura 1.5.5, se muestra cómo es la ventana principal de AutoCAD.

Barra de menús

Contiene varios de los comandos que se utilizarán al trabajar, se muestra una breve lista general de los menús:

File (archivo). Contiene comandos para abrir, guardar e imprimir dibujos, además de comandos para importar y exportar datos de dibujo. Los últimos cuatro archivos que se utilizaron también se listan.

Edit (editar). Ofrece funciones de edición Windows, como cortar y pegar desde la ventana de dibujo hacia el portapapeles. Los comandos de vinculación e incrustación de objetos (OLE) también se localizan aquí.

View (vista). Proporciona comandos para controlar el despliegue de la ventana de dibujo. Los controles Pan (panorámica) y Zoom (ampliación) también se localizan aquí, además de los comandos para guardar y restituir vistas.

Data (datos). Incluye comandos que controla la manera como se crean los objetos en AutoCAD. Se tiene acceso a todas las configuraciones para capas, estilos de texto, de dimensión y de multilíneas.

Options (opciones). Presenta configuraciones que controlan la apariencia y comportamiento de AutoCAD. Es factible controlar las configuraciones para cuadrícula, sujetadores, referencias a entidades y manijas, así como volver a configurar AutoCAD y controlar la apariencia de la ventana principal.

Tools (herramientas). Da acceso a varias herramientas de productividad dentro de AutoCAD. Se pueden utilizar algunas características como barras de herramientas, Aerial View (vista aérea), Spell Check (verificación de ortografía) y Calculator (calculadora).

Help (ayuda). Da acceso a las diferentes formas de ayuda en línea disponibles en AutoCAD.

Barra de estado

Despliega información actual del ambiente AutoCAD. Existen siete indicadores (o botones) en la barra de estado: el despliegue de coordenadas, Snap, Grid, Ortho, Tile, y la hora actual.

Area de línea de comando

Cuando se selecciona un ítem de menú o una herramienta de una barra de herramientas, AutoCAD desplegará el comando en la línea de comando.

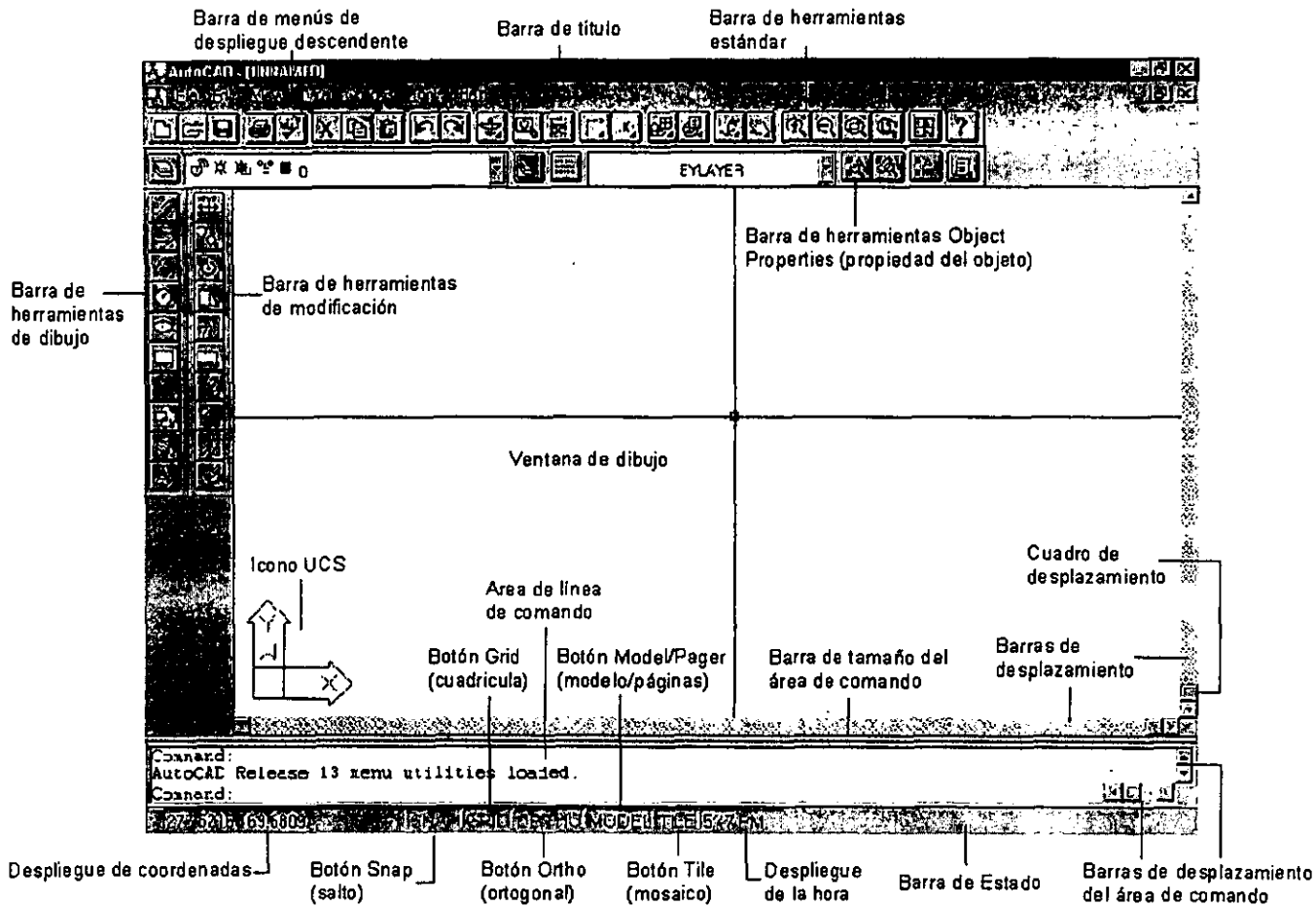


Figura 1.5.5. Ventana principal de AutoCAD.

Ventana de texto AutoCAD

Además del área de línea de comando, AutoCAD tiene aparte una ventana de texto, ésta, como el área de línea de comando, muestra un historial de todo lo que ha sucedido desde que comenzó la sesión actual. Es activada y desactivada con F2.

Barras de herramientas

Proporcionan acceso directo a la mayoría de los comandos permitiendo mayor facilidad en el desarrollo del dibujo; además a diferencia del menú, se controla la ubicación, el tamaño y la orientación de las barras de herramientas. A continuación se explican las herramientas que se despliegan por omisión.

Standard (estándar). Incluye comandos que se utilizan en la mayor parte de la sesión. Hay comandos para abrir y guardar archivos, modificar el despliegue de pantalla, seleccionar objetos e imprimir, además del comando Help de AutoCAD (fig. 1.5.6.).

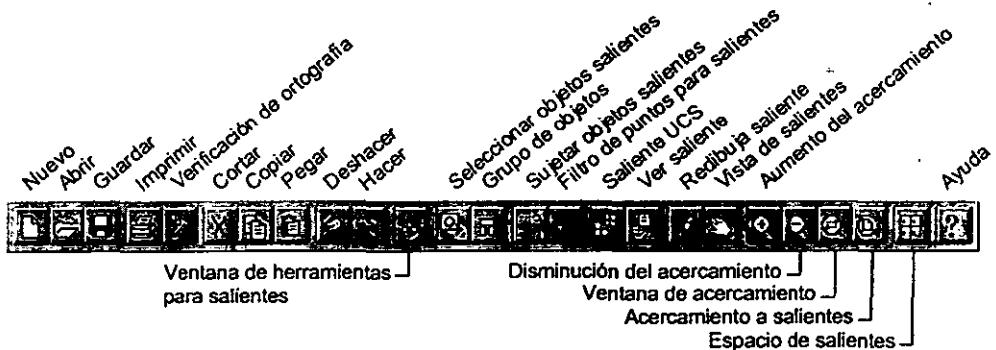


Figura 1.5.6. Barra de herramientas estándar.

Nuevo. Crea un nuevo archivo de dibujo.

Abrir. Abre un dibujo de un archivo existente.

Guardar. Guarda el dibujo actual.

Imprimir. Imprime un dibujo a una impresora, plotter, o archivo.

Verificación de ortografía. Corrige la ortografía de objetos del texto en un dibujo, creada con el TEXTO, DTEXT y órdenes de MTEXT.

Cortar. Copia la selección actual y la pone en el portapapeles borrando el objeto del dibujo hasta colocarlo en su nueva posición.

Copiar. Para copiar objetos seleccionados en el portapapeles en todos los formatos posibles.

Pegar. Para pegar objetos, textos, y varios formatos del archivo como el metafile, bitmap, y multimedios en un dibujo.

Deshacer. Marcha atrás a la instrucción o funcionamiento más reciente.

Hacer. Invierte los efectos de un solo deshacer. Debe seguir inmediatamente o deshacer la orden.

Ventana de herramientas para salientes. Es una herramienta de la navegación. Despliega una vista del dibujo en una ventana separada para que pueda localizar rápidamente y puede moverse a una área específica.

Seleccionar objetos salientes. AutoCAD requiere que se seleccione el objeto para ser procesado.

Grupo de objetos. Crea un juego de la selección nombrado de objetos.

Sujetar objetos salientes. Juegos que ejecutan modos Instantáneos en el objeto también llamados modos de Osnap permiten especificar un punto a una situación exacta en un objeto; cada modo, determina la situación.

Filtros de puntos para salientes. Construir puntos al seleccionar las coordenadas X, Y y Z de objetos existentes.

Salientes UCS. Selecciona un Sistema de Coordenadas prefijado por el usuario para la colocación conveniente de objetos.

Ver saliente. Crea y restaura vistas.

Redibuja saliente. Se vuelve a dibujar quitando marcas que no pertenecen en la ventana de dibujo actual.

Vista de salientes. Despliega del dibujo en la ventana de dibujo actual.

Aumento del acercamiento. Aumenta el tamaño del objeto en la ventana de dibujo actual.

Disminución del acercamiento. Disminuye el tamaño del objeto en la ventana de dibujo actual.

Ventana de acercamiento. Aumenta el tamaño del objeto seleccionando el área a ver en la ventana de dibujo actual.

Acercamiento a salientes. Disminuye el tamaño del objetos al mínimo en la ventana de dibujo actual.

Espacio de salientes. Los mandos acceden para empapelar espacio, así como la conducta de viewports de AutoCAD.

Ayuda. Despliegue de la ayuda en línea.

Object Properties (propiedades del objeto). Contiene herramientas que le permiten controlar la manera de crear, almacenar y desplegar los objetos en AutoCAD (fig. 1.5.7.).

Capas. Maneja capas, siendo una propiedad asignada a cada entidad del dibujo AutoCAD

Control de capas. Contiene toda la información de capas en el dibujo.

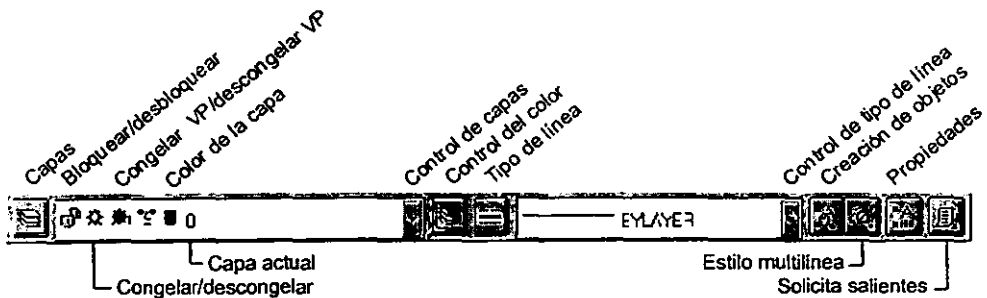


Figura 1.5.7. Barra de herramientas Object Properties.

Bloquear/desbloquear. Bloque o desbloquea la capa o capas seleccionadas.

Congelar/descongelar. Congela o descongela la capa o capas seleccionadas.

Congelar VP/descongelar VP. Congela o descongela la capa o capas seleccionadas en el puerto de vista de la ventana de dibujo actual.

Encendido/apagado. Activa o desactiva la capa o capas seleccionadas.

Color de la capa. Asigna un color a la capa o capas seleccionadas.

Capa actual. Esta opción convierte la capa seleccionada en la capa actual.

Control del color. Selección de color para los nuevos objetos.

Tipo de línea. Asigna tipos de líneas a una capa.

Control de tipo de línea. Muestra el tipo de línea en la capa actual.

Creación de objetos. Propiedades para los nuevos objetos ajustando: capa, color, tipo de línea, estilo de texto y escala de tipo de línea.

Estilo multilínea. Define de 1 a 16 elementos de multilínea, además de especificar un color y tipo de línea para cada elemento, así como las capas de inicio y fin y agregar un relleno sólido.

Propiedades. Modifica y establece cualquier característica de un objeto.

Solicita salientes. Información de banco de datos de despliegues para los objetos seleccionados.

Draw (dibujar). Contiene la mayoría de las herramientas para dibujar objetos (fig. 1.5.8.).



Figura 1.5.8. Barra de herramientas de dibujo.

Salientes de línea. Crea segmentos de la línea rectos.

Salientes de polilínea. Crea polilíneas bidimensionales.

Salientes de arco. Crea un arco.

Salientes de círculo. Crea un círculo.

Salientes de elipse. Crea una elipse o un arco elíptico

Salientes de polígono. Dibuja un rectángulo (poligonal).

Salientes de punto. Crea un objeto del punto, uniendo cada uno como nodos.

Salientes de cuadrilátero. Inserta un bloque de referencia dentro del dibujo.

Salientes de anchura. Llena una área incluida de un modelo de la compuerta asociada.

Salientes de texto. Agrega un objeto de texto de párrafo al dibujo.

Modify (modificar). Contiene las herramientas de edición utilizadas con más frecuencia (fig. 1.5.9.).

Mover. Cambia de sitio objetos a una distancia y dirección específica.

Copia salientes. Copia de los objetos.

Rotar salientes. Da un giro al objeto sobre un punto.



Figura 1.5.9. Barra de herramientas de modificación.

Dimensiona salientes. Amplia o contrae los objetos seleccionados en la ventana de selección.

Ajustar salientes. Arregla objetos a un margen de corte definido por otros objetos.

Edición especial de salientes. Borra partes de objetos o lo divide en dos.

Característica de salientes. Revisa las polilíneas y redes tridimensionales del polígono.

Separa salientes. Descompone un objeto en sus componentes.

Borra. Quita objetos de un dibujo.

Ventana de dibujo.

Es donde se desarrolla, edita y modifica los dibujos o gráficas. La barra vertical y horizontal son barras de desplazamiento que permiten mover el dibujo.

Ventajas:

- Es de arquitectura abierta, permite a sus usuarios crear menús personalizados y programas de aplicación.
- La compatibilidad ascendente con versiones interiores es garantizada, así también el poder ser guardado en versiones anteriores (11 , 12 y AutoCAD LT).
- En el entorno Windows, la versión 13 soporta OLE, tanto en modo servidor como cliente, realizando vínculos y inclusiones de datos a otros programas.
- Por ser un software de diseño, contiene una variedad grande en herramientas.
- Además esta versión cuenta con un sistema para manipular la imagen en tercera dimensión.

Desventajas:

- Al abrir archivos sólo permite de formato .DWG.
- Al importar archivos sólo permite los de formato: .WMF, .DXF, .SAT, .EPS, .PCX, .TIF, .GIF, y .3DS.
- Al exportar archivos sólo permite los de formato: .WMF, .DXF, .SAT, .EPS, .3DS, .DXX, .BMP, y .DWG.
- Para ser instalado necesita un procesador mayor del 80386, 16 Mb. de memoria en RAM, y un espacio libre en disco duro de 80 Mb.
- Si se manejan imágenes de alto diseño en equipos 80386, disminuye la velocidad del sistema.

Photo Styler.

Es un procesamiento de imagen gráfica. Las operaciones típicas incluyen escaneo de imagen, control para aumento o disminución del contraste, balanceo del color, control de exposición y trazado o esquematización de objetos, de modo que la imagen pueda

ser sometida a otros procesos (fig. 1.5.10). Photo Styler integra en su barra de menús: de archivo, edición, selección, imagen, efectos, vista, ventana, y de ayuda..

Menú de archivo.

Muestra opciones en la administración de archivos:

Nuevo. Crea una nueva imagen.

Abrir. Abre uno o más archivos que se han creado previamente.

Restaurar. Reemplaza la imagen activa, selección, o enmascara con la versión recientemente guardada.

Cierre. Cierra la ventana de la imagen activa.

Cierre Todo. Cierra todas las ventanas.

Salvar. Salva la imagen activa en el disco.

Salvar como. Asigna un nuevo archivo de diferente formato.

Importación. Abre imágenes de otros paquetes o programas de diferente formato.

Exportación. Permite usar programas de utilidad del software para convertir datos de la imagen a los formatos diferentes.

Escáner. Se selecciona el escáner a utilizar.

Impresión. Imprime el archivo activo.

Demostración. Permite ver la presentación de impresión.

Preferencias. Muestra opciones específicas en la imagen (general, archivo de inicialización, selección de herramientas, y Asociar extensiones.

Salida. Termina la sección, saliendo del paquete.

Menú de Edición

Muestra opciones en el manejo de imágenes:

Deshacer/Rehacer. deshace o rehace los cambios en la imagen.

Habilitar deshacer. Habilita o no la opción deshacer.

Borrar memoria deshacer. Borra la elección deshacer memorizada.

Corte. Quita el área de selección que es colocado en el portapapeles para ser después pegado.

Copia. Copia el área seleccionada en el portapapeles.

Pegar. Agrega el área seleccionada del Portapapeles a la imagen activa o en una nueva (selección, nuevo documento, dentro de la selección, fuera de la selección, y máscara).

Limpiar / Borrar. Quita los objetos o áreas seleccionadas.

Portapapeles. Permite ver cortes o copias realizadas en Windows y en Photo Styler (leer, guardar, contenido, y borrar).

Duplicar. Crea una copia de toda la imagen actual.

Ampliar. Amplia la imagen entera o simplemente el área seleccionada con un color especificado.

Modelo. Permite crear y definir modelos de imagen para uso continuo.

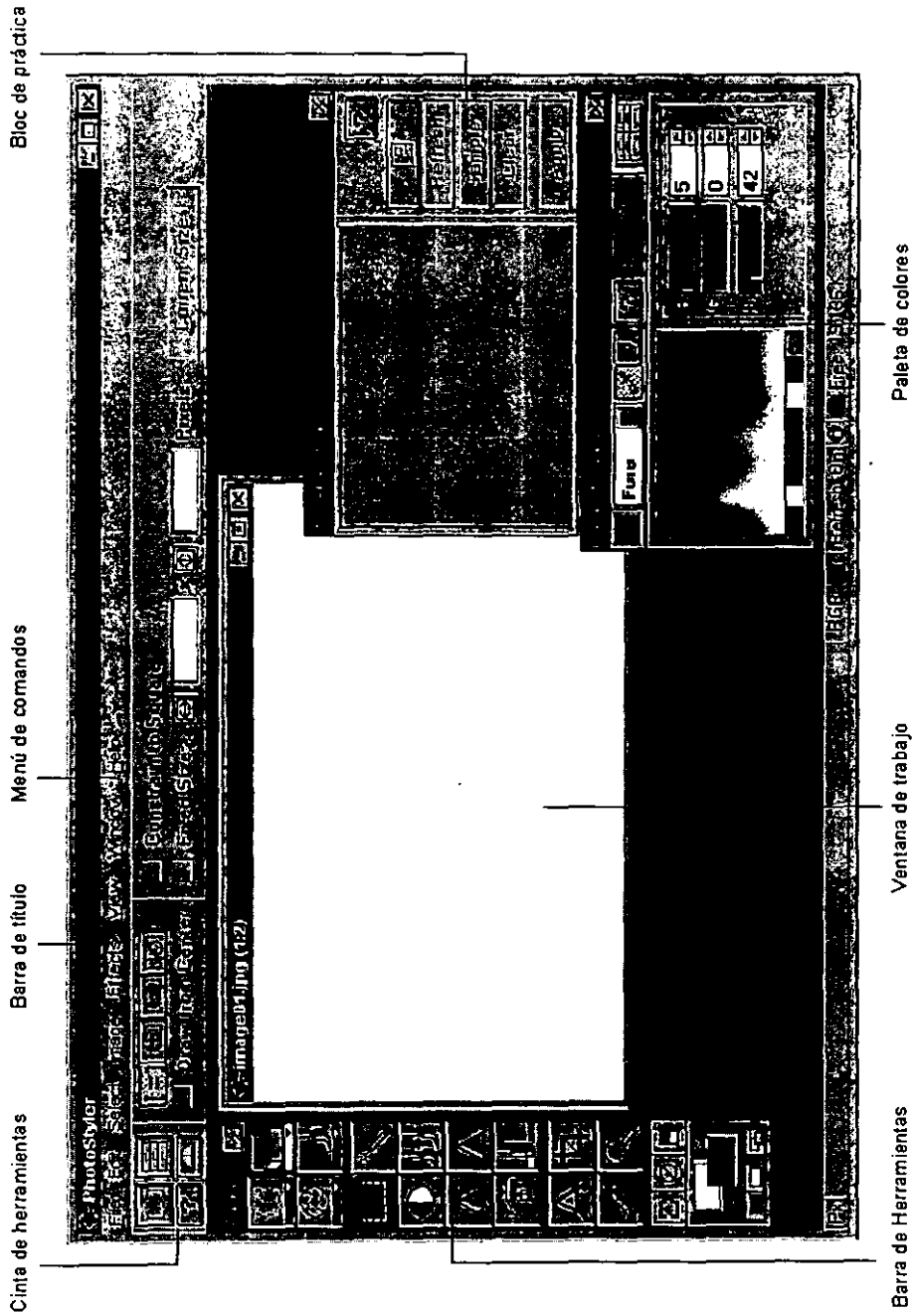


Figura 1.5.10. Ventana principal de Photo Styler

Menú de Selección

Muestra opciones de selección en la imagen:

Todo. Selecciona toda la imagen.

Ninguno. Deshace toda la selección en la imagen.

Inverso. Se invierte la selección en la imagen activa.

Extender. Se extiende la selección actual para incluir píxeles en la imagen.

Borde suave. Define una área alrededor del límite de la selección aplicando pintura parcial y efectos.

Borde. Amplia el límite del área de la selección actual en una nueva selección.

Remover aberturas. A veces una selección puede contener muchas áreas no pertenecientes (abertura).

Crear capa. El área seleccionada la convierte en capa sobre el resto de la imagen.

Capa. Une la selección de capa con la imagen subyacente.

Conservar imagen. Determina si la imagen subyacente se altera cuando una área seleccionada se mueve.

Esconda Marquesina. Muestra o esconde la marquesina que perfila una área de la selección.

Control de unión. Le permite controlar una selección de capa (imagen de capa) y el fondo de la imagen (imagen designada) para unión.

Menú de Imagen

Muestra opciones en detalle de imágenes:

Tono. Contiene funciones para detalle de imagen (enfoque, brillo y contraste, corrección tonal, balance en gris, balance en color, tinte y saturación, negativo, igualación).

Muestra. Permite cambiar las dimensiones.

Resolución. Redefine la imagen sin cambiar el tamaño del archivo.

Redimensión. Modifican las dimensiones de la imagen libremente o por medida.

Mover muestra. Mueve la imagen vertical, horizontal, y hacia abajo.

Rotación. Permite rotar la imagen (libre, por grados, horizontal, y vertical).

Transformar. Se modifica la imagen en inclinación, perspectiva, y distorsión.

Mover selección. Mueve la imagen dentro del área seleccionada.

Convertir a. Realiza la conversión a opciones de color, influyendo en el tamaño de los bits.

Dividir a. Realiza la división en escala de grises de la imagen (canales: RGB, HSB, HLS, y CMYK).

Combinar a. Realiza la función inversa de "Dividir a" uniendo la imagen en escala de grises hasta obtener nuevamente el color (canales: RGB, HSB, HLS, y CMYK).

Canal. Permite visualizar un solo canal de color en RGB o CMYK (verdadero - color) en la imagen (RGB, rojo, verde, y azul).

Menú de Efectos

Muestra opciones en efectos para las imágenes:

Suavizar. Permite manipular la imagen a grado de producir un efecto opaco (borrón, borrón Gaussian, promediando, máximo, medio, y mínimo).

Afilar. Permite la agudeza de la imagen de aumentos del contraste en los pixeles (afilado, perfeccionamiento del borde, encontrar borde, y contorno del trazo).

Ruido. Realiza un tipo de retocado en la imagen (despeckle y añadir ruido).

Especial. Son efectos de mayor realce en la imagen (realce, difuso, monocromático, borrón del movimiento, y mosaico).

Menú de Vista

Muestra opciones en el manejo de vistas de las imágenes:

Añadir vista. Abre otra vista de la imagen actual.

Añadir vistas para los Canales. Abre una nueva imagen para cada canal de la imagen activa.

Vista real (1:1). Despliega la imagen a su tamaño real.

Aumento. Aumento de imagen hasta 16 niveles de amplificación.

Disminución. Disminución de la imagen hasta 16 niveles de reducción.

Ajuste en ventana. Ajuste en el tamaño de la imagen en la ventana.

Pantalla llena. Visualiza la imagen sin la barra de título y/o utiliza toda la pantalla sin mostrar los aditamentos del paquete (edición y sólo vista).

Mostrar/Ocultar. Permite mostrar o esconder cualquiera de las paletas individuales de PhotoStyler.

Reglas. Muestras o esconde las reglas ventana de la imagen.

Información de la imagen. Despliegue de Información sobre la imagen.

Información del sistema. Proporciona información sobre su configuración del sistema.

Tabla de colores. Despliega la tabla para imágenes de 16 o 256 colores.

Menú de Ventanas

Muestra opciones de presentación de o ventanas activas:

Cascada. Coloca las ventanas en un modelo solapado para que la barra del título de cada ventana permanezca visible.

Azulejo. Coloca las ventanas del lado-por-lado de las imágenes abiertas para que todas las imágenes sean visibles en el espacio de trabajo.

Colocar Iconos. Coloca iconos de la imagen minimizados en una fila horizontal al fondo del espacio de trabajo.

Menú de Ayuda

Muestra la ayuda del paquete.

Barra de herramientas en uso.

Cada herramienta tiene una barra con sus respectivas opciones (fig.1.5.11).

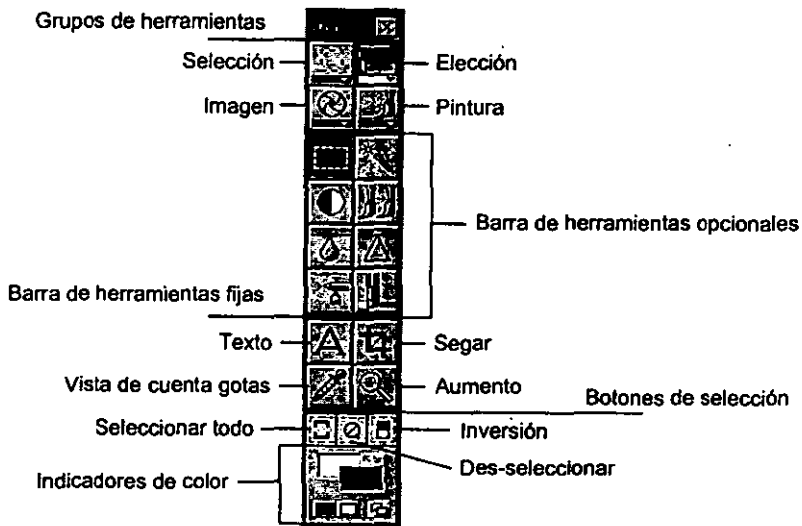


Figura 1.5.11. Barra de herramientas

Grupos de Herramientas

Las herramientas de Photo Styler se sitúan en grupos; conteniendo una categoría específica (selección, pintura, elección y imagen); conformándolo con herramientas que aparecen en los grupos (texto, segar, vista a cuenta gota, y aumento).

Herramientas de grupo de selección (fig.1.5.12).

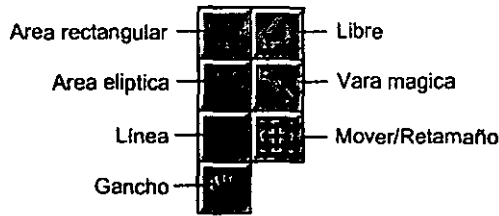


Figura 1.5.12. Grupo de selección.

Area rectangular. Selecciona una área rectangular o cuadrada en la imagen actual.

Libre. Selecciona una área irregularmente formada en una imagen.

Area elíptica. Selecciona una área elíptica o redonda en la imagen actual.

Vara mágica. Automáticamente selecciona una sola área de la imagen que contiene pixels con el mismo (o similar) escala de grises o valores de color como el punto o región que usted especifica.

Línea. Selecciona una línea horizontal o vertical de pixeles.

Mover/Re-tamaño. Mover o re-tamaño la marquesina de la selección con o sin afectar los volúmenes de la selección.

Gancho. Mueve la imagen en cualquier dirección para poder ver cuando es demasiado grande por la ventana de trabajo.

Herramientas de pintura (fig.1.5.13)

Pincel. Pintar una línea a pulso usando el color del primer plano.

Pistola. Simula un efecto de rocío liso pintando el color del primer plano.

Lápiz. Dibuja una línea a pulso con el color del primer plano actual.

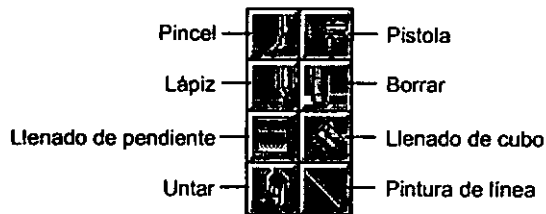


Figura 1.5.13. Grupo de pintura.

Borrar. Le permite cambiar pixeles en la imagen al negro, blanco, el color del fondo actual o restaura a la última versión guardada de la imagen.

Llenado de pendiente. Le permite llenar una imagen o área de la selección de colores que gradualmente se mezclan de uno a otro.

Llenado de cubo. Llena una área de la imagen con el color del primer plano.

Untar. Unta la pintura en una imagen por una distancia especificada.

Pintura de línea. Dibuja una línea recta en el color del primer plano actual.

Herramientas de imagen (fig. 1.5.14).

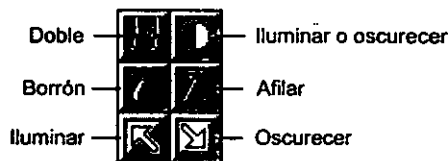


Figura 1.5.14. Grupo de imagen.

Doble. Lo permite doblar una área de una imagen a otra parte de la misma imagen o a una imagen separada.

Iluminar o Oscurecer. Ilumina o oscurece en resalto, medios tonos, y áreas de la sombra de la imagen.

Borrón. Disminuciones de contraste en pixeles, toca y hace el área seleccionada menos enfocada.

Afilar. Aumenta el contraste en pixeles elegida y hace el área afectada más enfocado.

Iluminar. Ilumina la imagen.

Oscurecer. Oscurece la imagen.

Herramientas Fijas

Texto. Inserta texto en la imagen activa.

Segar. Selecciona una área de una imagen y desecha el resto.

Aumento. Reduce o aumenta la imagen permitiendo realizar modificaciones a detalle.

Vista de cuenta gota. Toma el color directamente de la imagen actual.

Botones de selección.

Todos. Selecciona la imagen activa entera.

Ninguna. Verificar las áreas elegidas.

Inverso. Invierte la selección en la imagen activa.

Indicadores de color.

Muestra el color del primer plano actual y del fondo. El color del primer plano es el color que se aplica cuando usted pinta hacia la imagen, el color del fondo se usa cuando usted borra de la imagen.

Paleta de colores

Selección en un primer plano, fondo, o color de una paleta predefinida de valores o colores que se utilizan.

Bloc de práctica

Permite probar herramientas en una muestra de la imagen activa. Los cambios que se realicen en el bloc de Práctica no afecta la imagen original a menos que se pulse la opción de aplicar.

Ventajas:

- Este tipo de software permite abrir archivos en diferentes formatos, dicha versatilidad nos permite manejarlos en diferentes aplicaciones.

BMP (Bitmap)

DCS (Desktop Color Separation protocol)

EPS (Encapsulated PostScript)

GIF 89a (Graphic Interchange Format)

JPG (JPEG compressed)

MAC (Mac paint)

PCD (Photo CD)

PCT (Mac PICT)

PCX (ZSoft)

PSD (Adobe Photoshop)

RLE (Run Length Encoded)

SCT (Scitex)

TGA (Targa)

TIF (Tag Image File Format)

Facilidad en la edición de imágenes.

- Gran variedad de herramientas para rediseño y mejoramiento de imágenes.

- Es compatible con cualquier aplicación de windows, permitiendo la importación y exportación de imágenes.
- Se obtiene una mejor calidad de impresión.
- Permite su manejo en ambientes bajo red.
- Muestra una gran variedad de efectos.
- Proporciona ayuda en cuatro ventanas para proceso de imágenes.
- Utiliza la precisión Kodak el Sistema de Dirección Colorido (CMS), un traductor colorido universal, produciendo color consistente, exacto.

Desventajas:

- Solo es operado en sistemas de procesadores mayores del 80386, con un mínimo de 6 Mb. en RAM, en ambiente Windows, capacidad de espacio libre en disco duro de 20 Mb.

PhotoEnhancer

PhotoEnhancer es un paquete en edición de imágenes (fig. 1.5.15), conformando los siguientes puntos:

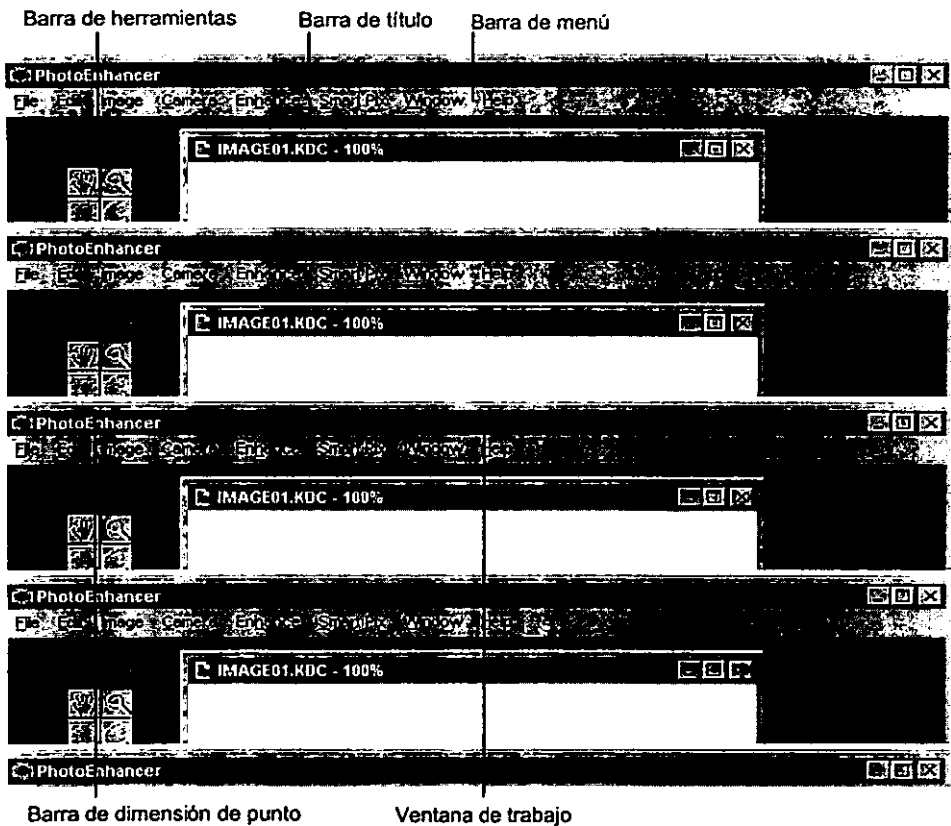


Figura 1.5.15. Ventana principal de PhotoEnhancer

Compatibilidad de formato de archivos

PhotoEnhancer se ha diseñado para poder ver los formatos estándar de archivos Windows. Abra o guarde imágenes en cualquiera de los siguientes formatos: TIFF, BMP, EPS y el formato original de la cámara digital Kodak, KDC.

Tablas de diapositivas

Elija Abrir tabla de diapositivas en el menú Archivo para crear una tabla de diapositivas de imágenes en el disco (fig. 1.5.16). En el cuadro de diálogo Abrir, seleccione el

directorio o la carpeta de imágenes. Los nombres de archivos aparecerán en gris puesto que no se pueden abrir imágenes individuales estando en este diálogo. Utilice las diapositivas para hojear varias imágenes de una sola vez, así como para ver información de las imágenes y cambiarle el nombre a los archivos. Utilice los comandos de clasificación del menú de Imagen para reorganizar la presentación visual. Desde la tabla de diapositivas podrá hacer correcciones a las imágenes o bien hacer doble clic en una diapositiva para abrir la imagen en una ventana de edición. Elija Ver diapositivas en cámara en el menú Cámara para crear una tabla de diapositivas de las imágenes que se encuentren en cámara en ese momento.

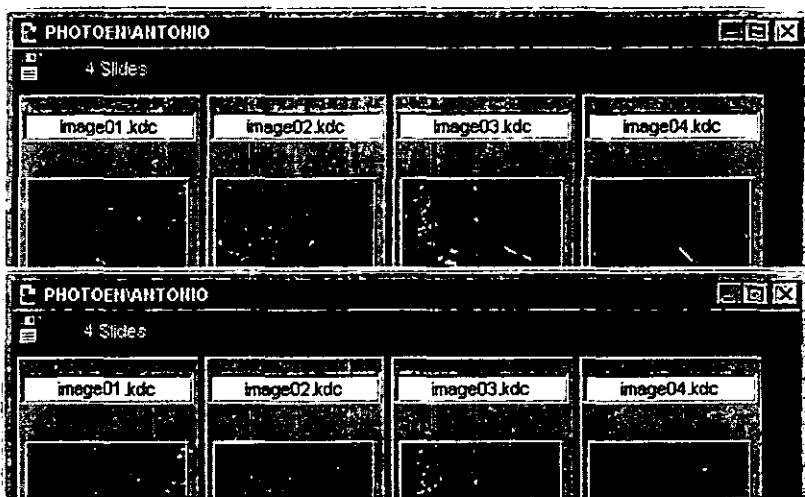


Figura 1.5.16. Tabla de diapositivas.

SmartPix

Utilice SmartPix para realizar fotos tomadas en determinadas condiciones de luz. Seleccione el SmartPix que describa la luminosidad de su foto, por ejemplo, Brillo de luz diurna. También puede aplicar SmartPix a muchas fotos al mismo tiempo desde una tabla de diapositivas. Mantenga presionada la tecla Shift Mayúsculas, haga clic en las diapositivas que desee realzar, luego seleccione un SmartPix. Ahora se abrirá el cuadro de diálogo "Guardar como" a fin de que pueda almacenar sus fotos recién

realzados. Elija una carpeta o directorio diferente o la misma carpeta y escriba encima de las fotos originales del disco.

Edición en ventanas

Haga doble clic en las fotos de la tabla de diapositivas, o bien elija Abrir del menú Archivo para abrir las fotos en una ventana de edición con una resolución total. Desde la ventana de edición puede utilizar las herramientas de selección, las brochas de realzar, hacer correcciones a las fotos y verlas con toda su resolución. La foto tiene que estar en una ventana de edición a fin obtener acceso a las correcciones Por Ejemplo del menú Realzar.

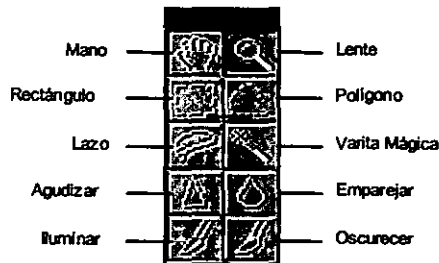


Figura 1.5.17. Barra de herramientas

Barra de herramientas

La barra de herramientas contiene las herramientas para hacer selecciones, cambiar la visualización y crear zonas de realce (fig. 1.5.17). Cuando utilice las brochas, Agudizar, Emparejar, Aclarar y Oscurecer, seleccione el tamaño de brocha deseado en el cuadro de Brochas. Cree una selección primero para aislar las correcciones hechas con brochas o comandos en el área cubierta y proteger las áreas de alrededor de modo que no cambien.

Otros realces

Desde los menús Imagen o Realzar, PhotoEnhancer ofrece comandos que sirven para modificar los valores de las fotos y voltear o girar las mismas. Al igual que con SmartPix, usted puede aplicar realces a una sola foto en una ventana de edición o a múltiples imágenes en una tabla de diapositivas. Mantenga presionada la tecla Mayúsculas, haga clic en las diapositivas que desee realzar, luego elija un realce. En el menú, ilumine el nombre del filtro que desea utilizar para iniciar el cambio. El cuadro de diálogo Guardar como se abrirá para que pueda guardar sus fotos recién realzadas. Elija una carpeta o unidad diferente o la misma carpeta y escriba encima de las fotos originales que se encuentran en el disco.

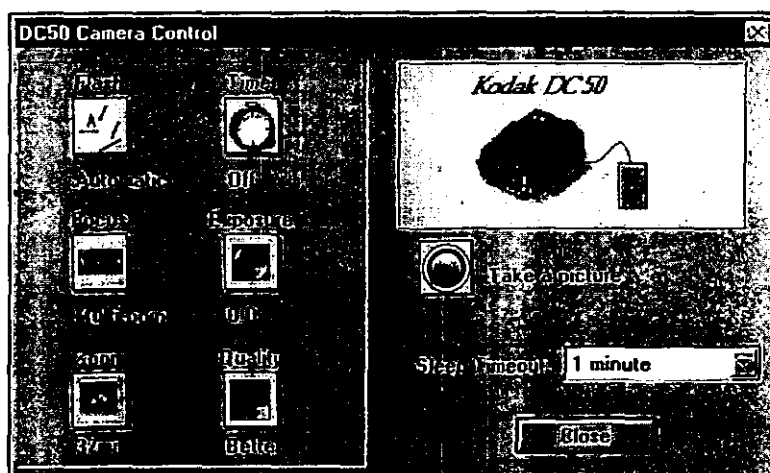


Figura 1.15.18. Controles de la cámara

Controles de cámara

PhotoEnhancer ofrece acceso a los controles para la cámara digital conectada a la computadora (fig. 1.5.18). Primero, debe conectarse la cámara. Utilice los controles del menú "Camera" para configurar el flash, el reloj de la cámara, el enfoque, la exposición, el acercamiento o alejamiento de la cámara, la resolución, así como para tomar fotos. Las imágenes tomadas aparecen en una mesa de diapositivas PhotoEnhancer. Para cambiar las configuraciones de la cámara, haga clic y mantenga oprimido el botón del ratón en el icono gráfico y arrástrelo por las opciones. Elija "Preference" en el menú "Edit" para configurar el puerto de acceso al cual se conecta la cámara. Elija "View Slides in Camera" para crear una tabla de diapositivas de las imágenes que están en la cámara actualmente o en la tarjeta PCMCIA.

By Example (por ejemplo)

By Example es un modo de edición visual para corregir de manera interactiva el enfoque, la exposición, el brillo y el contraste y la armonía de colores de imágenes. Elija "By Example" desde el menú "Enhance". Se aplican configuraciones distintas a nueve ejemplos de las imágenes y usted elige la mejor. El ejemplo del centro siempre muestra el estado actual de la imagen. Haga clic en el botón "Undo" para borrar el último cambio realizado, en "Revert" para regresar la imagen al estado original según se guardó en el disco, en "Done" para salir del diálogo y regresar a la ventana de edición y en "Apply" para actualizar la imagen de vista previa. Cuando se realiza una corrección, se generan nueve ejemplos nuevos.

Color

Los comandos Cambiar profundidad del color le permiten cambiar la profundidad del bit, es decir, el número de colores, de la foto de la ventana de edición. Usted puede reducir la información del color de la foto de Millones de colores a Miles ó 256 matices de gris, reduciendo así la cantidad de datos de foto y el tamaño del archivo en el disco. Usted puede cambiar la profundidad del color de su imagen al guardar desde el cuadro de diálogo "Guardar", o del menú Imagen mientras esté trabajando en la ventana de edición.

Guardar

Elija Guardar en el menú "Archivo" para escribir encima de la foto original con los cambios hechos o elija "Guardar como" para modificar el nombre del archivo, formato o unidad. Elija "Guardar todos" para guardar a la vez muchas fotos de una tabla de diapositivas. Mantenga presionada la tecla "Mayúsculas" y haga clic en las diapositivas que desee guardar. Guarde las fotos como archivos TIFF o PICT en Macintosh y como TIFF o BMP en Windows.

Impresión

Elija imprimir en el menú "Archivo" con una ventana de edición activa para imprimir una sola foto en su tamaño completo o desde una tabla de diapositivas para imprimir una serie de fotos en una página. Cuando imprima de una tabla de diapositivas, las fotos se imprimirán igual que como se ven en la pantalla, con muchas miniventanas en una sola página.

Cámara Kodak DC50

La cámara DC50 de Kodak Digital Science (fig. 1.5.19.), permite tomar fotografías y transferirlas al ordenador, donde pueden guardarse, manipularse usando un programa de edición de imágenes, o imprimirse en la impresora. La cámara DC50, ofrece las siguientes características:

- Enfoque automático.
- Objetivo con zoom.
- Selección de flash.
- Selección de temporizador.
- Selección de exposición.
- Tarjeta de memoria extraíble.
- Modo de ahorro de energía.
- Distintas posibilidades de suministro de alimentación: con baterías o con adaptador de corriente alterna.
- Indicador del estado de las baterías.
- Acoplamiento al tripode estándar.

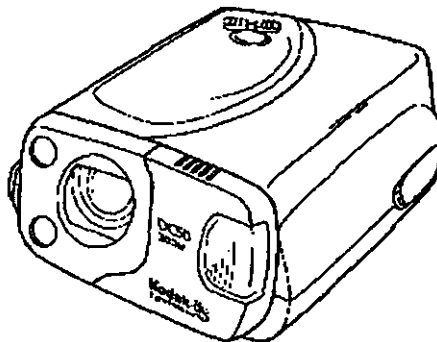


Figura 1.5.19. Cámara DC50 Kodak.

Ventajas:

- Permite la translación de imágenes tomadas en cámara digital a la computadora.
- Contiene herramientas para la edición de imagen.
- Corrige defectos que tenga la foto.
- Tiene elementos para observar la imagen en diferentes aumentos.
- Por ser un software realizado en ambiente Windows, esto le permite trasladar las imágenes a otras aplicaciones, con cuatro tipos de formato (.TIFF, .BMP, .JPG, y .KDC).
- Tiene un control para manipular la imagen en blanco y negro, hasta un millón de colores.

Desventajas:

- Se encuentra limitado para leer archivos de diferentes formatos.
- Se encuentra limitado para guardar archivos en diferentes formatos.
- Utiliza muy pocas herramientas.
- No contiene ayuda.
- Se utiliza en sistemas de procesadores mayores del 80386, memoria mínima de 4 Mb. en RAM, con un espacio libre en disco duro de 5 Mb.

Guía Roji

Los tradicionales y antiguos mapas y la novedosa tecnología en computadoras permite un solo concepto: InfoGuía Roji, un revolucionario sistema de cartografía en una nueva dimensión en el manejo de los mapas.

Con InfoGuía Roji la superficie del mapa pasa a ser de una superficie de consulta a un área de trabajo sobre el cual se pueden trazar áreas para seccionarlo, marcar puntos de interés, trazar rutas conectando dichos puntos, trazar líneas representando caminos, cableado, etc., y colocar textos para dar mayor claridad a la información allí vertida.

InfoGuía Roji mantiene la información en una superficie distinta a la del mapa. Los objetos colocados sobre el mapa no alteran físicamente su superficie, y pueden ser movidos, alterados o borrados cuando se desee, sin alterar la calidad visual del mismo. Esto permite, además, que se obtenga la actualización del mapa con el que está trabajando sin temor a perder la información en él marcada. Al actualizar el mapa, la superficie que contiene la información no cambia, solamente cambia el dibujo del mapa.

Además de aceptar la colocación de objetos sobre la superficie del mapa, InfoGuía Roji permite el acceso instantáneo a ellos a través de índices de área, línea, ruta y marcador (los textos no forman parte de ningún índice). Estos índices, inicialmente vacíos, crecen y se actualizan cuando se coloca, mueve y borra objetos de la superficie del mapa. InfoGuía Roji está diseñado para manejar miles de objetos con degradación imperceptible en su tiempo de respuesta, haciendo posible localizar en segundos cualquiera de ellos.

Una de las características más importantes de InfoGuía es que permite la asociación de información con las áreas, líneas, marcadores o rutas colocadas sobre el mapa. El tipo de información que se desee dependerá de la aplicación que se esté manejando.

A diferencia de otros sistemas de carácter cartográfico, InfoGuía Roji está diseñado para proporcionar la más alta calidad visual posible, combinada con gran facilidad de uso. El ambiente Windows, bajo el cual corre el programa, representa una de las interfaces de usuario más fáciles de manejar y aprender. La calidad de las imágenes manejadas por el sistema es equivalente a la presentada en la Guía Roji convencional,

mientras que otros sistemas representan el mapa por medio de líneas y otros objetos geométricos cuya legibilidad es casi nula y causan pronta fatiga al operador.

La figura 1.5.23. muestra la ventana principal de InfoGuía Roji, proporciona cinco herramientas para la colocación de objetos, una para la selección de objetos y dos para cambiar el nivel de acercamiento. En cualquier momento solamente una de las herramientas puede estar activa.

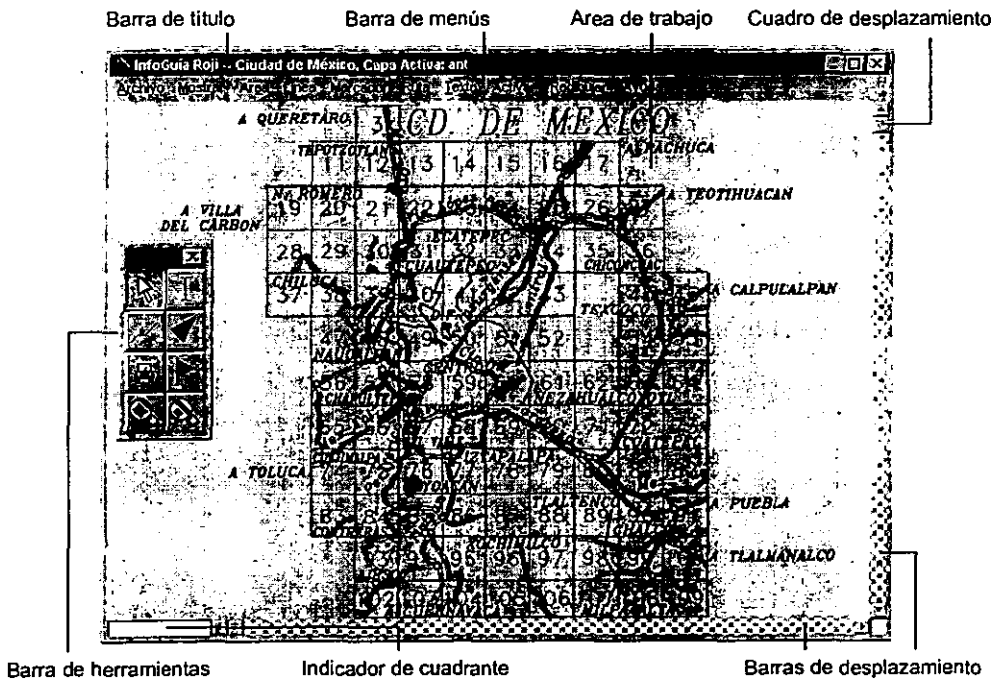


Figura 1.5.23. Ventana principal de Guía Roji.

Menú Archivo

Crear Capa. Permite crear una capa. Una vez creada, la capa estará vacía (no contendrá información), pero no necesariamente será la capa activa.

Abrir Capa. Permite abrir una capa. Abrir una capa equivale a "comenzar de trabajar" con ella. Puede mantener abiertas hasta 5 capas simultáneamente.

Cerrar Capa. Permite cerrar una capa.

Activar Capa. Selecciona una capa para volverla la capa activa. Use también este comando para averiguar cuál es la capa activa.

Importar. Importar información a la capa activa.

Exportar. Exportar información de la capa activa.

Editar Campos. Alterar el tipo de información asociada con los objetos contenidos en la capa activa. Esta operación se puede realizar independientemente para áreas, líneas, marcadores y rutas.

Compactar. Eliminar la fragmentación existente en los archivos correspondientes a la capa activa.

Utilización Archivos. Presenta una tabla indicando el número de objetos de cada tipo contenidos en la capa activa, e indica el porcentaje de desperdicio en los archivos que los contienen.

Impresora. Presenta una lista de las impresoras instaladas en el sistema, permite seleccionar la que se desea utilizar para imprimir y permite activar al manejador de la impresora para configurarla.

Formatos de Impresión. Define el formato de los reportes generados por el programa para los objetos contenidos en la capa activa.

Imprimir. Imprime los objetos contenidos de la capa activa.

Información. Presenta una ventana con información sobre autores, derechos reservados, etc.

Terminar. Suspense la ejecución del programa, cerrando automáticamente todas las capas abiertas.

Menú Mostrar

Mostrar, Ciudad. Muestra una vista general de la ciudad y dibuja sobre ella los objetos visibles correspondientes a todas las capas abiertas.

Mostrar, Plano. El plano seleccionado por el usuario y mostrar sobre él los objetos visibles correspondientes a todas las capas abiertas.

Mostrar, Cuadrante. El cuadrante seleccionado por el usuario y mostrar sobre él los objetos visibles correspondientes a todas las capas abiertas.

Mostrar, Calle. Accesa al índice de calles y mostrar el cuadrante con el que esté asociado la calle que se está buscando.

Mostrar, Colonia. Accesa al índice de colonias y mostrar el cuadrante con el que esté asociado la colonia que se está buscando.

Mostrar, CP. Accesa al índice de códigos postales y mostrar el cuadrante con el que esté asociado el código postal que se está buscando.

Mostrar, Área. Accesa al índice de áreas y mostrar el área seleccionada por el usuario.

Mostrar, Línea. Accesa al índice de líneas y mostrar la línea seleccionada por el usuario.

Mostrar, Marcador. Accesa al índice de marcadores y mostrar el marcador seleccionada por el usuario.

Mostrar, Ruta. Accesa al índice de rutas y mostrar la ruta seleccionada por el usuario.

Mostrar, Vehículo. Presenta un listado de los vehículos asociados con rutas despachadas.

Menú Area

Area, Editar. Permite consultar y modificar la información asociada con un área.

Area, Borrar Area. Este comando le permite borrar el área seleccionada.

Area, Borrar Vértice. Este comando le permite borrar el vértice seleccionado del área seleccionada.

Area, Pegar. "Pegar" dos áreas contiguas consiste en juntar los vértices de una y otra que se encuentren suficientemente cerca.

Area, Visibilidad. Por medio de este comando, el usuario puede definir una tabla de colores que el sistema consulta al mostrar áreas.

Area, Proteger Una. Permite proteger individualmente el área seleccionada y asignarle un pasaporte que restringe la utilización subsecuente de este comando sobre dicha área.

Area, Proteger Todas. Equivalente a seleccionar una por una las áreas de la capa activa y aplicar sobre ellas el comando Area, Proteger Una.

Menú Línea

Línea, Editar. Permite consultar y modificar la información asociada con una línea. Este comando opera sobre la línea seleccionada.

Línea, Borrar Línea. Este comando le permite borrar la línea seleccionada.

Línea, Borrar Vértice. Este comando le permite borrar el vértice seleccionado de la línea seleccionada.

Línea, Visibilidad. Por medio de este comando, el usuario puede definir una tabla de colores que el sistema consulta al mostrar líneas.

Línea, Proteger Una. Permite proteger individualmente la línea seleccionada y asignarle un pasaporte que restringe la utilización subsecuente de este comando sobre dicha línea.

Línea, Proteger Todas. Equivalente a seleccionar una por una las líneas de la capa activa y aplicar sobre ellas el comando Línea, Proteger Una.

Menú Marcador

Marcador, Editar. Permite consultar y modificar la información asociada con un marcador.

Marcador, Borrar. Este comando le permite borrar el marcador seleccionado.

Marcador, Cambiar Símbolo. Permite cambiar el símbolo con el cual se representa el marcador seleccionado.

Marcador, Seleccionar Símbolo. Permite seleccionar el símbolo con el cual se representarán los marcadores colocados posteriormente.

Marcador, Editar Símbolo. Permite seleccionar un símbolo de la capa activa y editarlo.

Marcador, Símbolos Nivel Ciudad. El tamaño de los iconos utilizados para representar marcadores no cambia con el nivel de acercamiento.

Marcador, Símbolos Nivel Plano. El tamaño de los iconos utilizados para representar marcadores no cambia con el nivel de acercamiento.

Marcador, Visibilidad. Por medio de este comando, el usuario puede definir una tabla de símbolos que el sistema consulta al mostrar marcadores.

Marcador, Proteger Uno. Permite proteger individualmente el marcador seleccionado y asignarle un pasaporte que restringe la utilización subsecuente de este comando sobre dicho marcador.

Marcador, Proteger Todos. Equivalente a seleccionar uno por uno los marcadores de la capa activa y aplicar sobre ellos el comando Marcador, Proteger Uno.

Marcador, Asignar Número de Área. Este comando permite asignar a cada marcador un número que identifica el área a la que pertenece.

Menú Ruta

Ruta, Editar Ruta. Consulta y modifica la información asociada con un ruta.

Ruta, Editar Horarios. Consulta y modifica los horarios correspondientes a un ruta.

Ruta, Crear sin Trazar. Permite crear una ruta por medio de la especificación de una serie de paradas, sin necesidad de trazar la ruta manualmente sobre el mapa.

Ruta, Borrar Ruta. Este comando le permite borrar la ruta seleccionada.

Ruta, Borrar Vértice. Este comando le permite borrar el vértice seleccionado de la ruta seleccionada.

Ruta, Despachar. Permite despachar la ruta seleccionada.

Ruta, Suspende. Permite suspender la ruta seleccionada.

Ruta, Ruta Planeada. Muestra la ruta planeada correspondiente a la ruta seleccionada.

Ruta, Ruta Recorrida. Muestra la ruta recorrida correspondiente a la ruta seleccionada.

Ruta, Editar Símbolo Vehículo. Permite editar el símbolo que recorre rutas despachadas en la capa activa.

Ruta, Visibilidad. Por medio de este comando, el usuario puede definir una tabla de colores que el sistema consulta al mostrar rutas.

Ruta, Proteger Una. Permite proteger individualmente la ruta seleccionada y asignarle un pasaporte que restringe la utilización subsecuente de este comando sobre dicha ruta.

Ruta, Proteger Todas. Equivalente a seleccionar una por una las rutas de la capa activa y aplicar sobre ellas el comando Ruta, Proteger Una.

Menú Texto

Texto, Editar. Edita el mensaje y define los efectos asociados con un texto.

Texto, Borrar. Este comando le permite borrar el texto seleccionado.

Texto, Visibilidad. El usuario puede definir una tabla de colores que el sistema consulta al mostrar textos.

Texto, Proteger Uno. Permite proteger individualmente el texto seleccionado y asignarle un pasaporte que restringe la utilización subsecuente de este comando sobre dicho texto.

Texto, Proteger Todos. Equivalente a seleccionar uno por uno los textos de la capa activa y aplicar sobre ellos el comando Texto, Proteger Uno.

Menú Activar

Activar, Herramientas. Activa o desactiva la ventana de herramientas.

Activar, Colores. Activa o desactiva la ventana de colores.

Activar, Anchos. Activa o desactiva la ventana de anchos.

Activar, Flechas. Activa o desactiva la aparición de cabezas de flecha a lo largo de líneas y rutas para indicar el sentido en que fueron creadas.

Activar, Letreros. Activa o desactiva la aparición de un letrero debajo de cada marcador conteniendo el campo índice de su información asociada.

Activar, Selección Areas Invisibles. Permite la selección de áreas invisibles.

Activar, Verificación Nombres Paradas. Al estar activada se determinará el nombre de las paradas de una ruta, al momento en que el usuario seleccione la opción Ruta, Editar Horarios, en base a la cercanía de los vértices de la misma con los marcadores que se encuentran a su alrededor.

Redibujar

Ocasiona que se re-dibuje por completo los contenidos de la ventana principal.

Ayuda

Presenta la ayuda del paquete.

Barra de herramientas

Se puede seleccionar la herramienta que se desea de la ventana que se ilustra en la figura 1.5.24.

- La flecha, que se utiliza para seleccionar objetos sobre el mapa
- Las áreas, que se utiliza para colocar áreas sobre el mapa
- Las líneas, que se utiliza para colocar líneas sobre el mapa
- Los marcadores, que se utiliza para colocar marcadores sobre el mapa
- Las rutas, que se utiliza para colocar rutas sobre el mapa
- El textos, que se utiliza para colocar textos sobre el mapa
- La lupa, que se utiliza para acercarse al mapa
- La lupa negativa, que se utiliza para alejarse del mapa

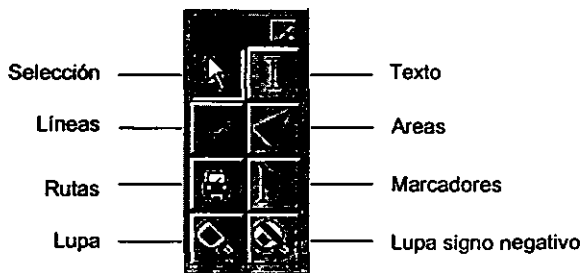


Figura 1.5.24. Barra de herramientas de Guía Roji.

La ventana de herramientas aparece solamente si existe alguna capa activa, y puede ocultarse y mostrarse de nuevo por medio del comando Activar, Herramientas del menú.

Ventajas:

- Este software en CD-ROM contiene la cartografía e índices de calles y colonias de una ciudad en particular (la Cd. de México) .
- InfoGuía puede trabajar con cualquier ComputerMap proporcionado por Guía Roji, S.A.
- Localización rápida y eficiente de puntos de interés.
- Fácil acceso a menús de aplicación, debido a que se maneja en ambiente Windows
- Por ser un software desarrollado en ambiente windows permite la exportación de imágenes a otras aplicaciones.
- Por ser una grabación en CD-ROM su tiempo de acceso es muy rápida.
- Facilidad en el manejo de impresión, de número indefinido y características.
- Permite colocar símbolos gráficos en el mapa, para ser identificado con mayor facilidad.
- InfoGuía puede exportar e importar información en archivos con registros de longitud fija (base de datos) y registros delimitados por tabuladores (hojas de cálculo).

Desventajas:

- No permite su manejo en ambiente de redes.
- No permite ser manejado en monitores de más de 256 colores (Ultra VGA y Super VGA).

- Sólo puede ser utilizado en equipos de capacidades mayores de procesadores 386 y 2 Mb en memoria RAM, debido que ocupa 600 Kb simplemente para leer los bloques.
- No puede ser grabado en disco duro por ser de búsqueda aleatoria.

1.6 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE VISUAL BASIC 4.0

1.6.1 Introducción a Visual Basic

Visual Basic es un poderoso sistema de programación gráfica que le permite crear aplicaciones Windows reales con código BASIC. Este sistema es un extraordinario avance para desarrollo de programas, porque combina la sintaxis simplificada de BASIC y GWBASIC con la estructura de la programación QBASIC y QuickBasic.

El sistema de programación Visual Basic le permite crear objetos, establecer y cambiar sus propiedades para posteriormente, asignarles un código BASIC funcional.

La filosofía de la programación Visual Basic consiste, primero en crear objetos como ventanas, íconos y menús y después elaborar procedimientos que sean llamados por cada uno de estos objetos. Esto es diferente del método tradicional de elaboración de un programa el cual exige estructuras para controlar el flujo del programa de un procedimiento a otro de manera lógica hasta que el programa termina. La programación por eventos es un método flexible y conveniente para escribir programas en Windows.

Por ejemplo, se puede elaborar el código para un objeto que se ha creado y después hacer copias del mismo con el código asignado; no se tiene que escribir el código otra vez.

Ahora bien, definir Visual Basic como un *sistema de programación orientada a objetos* sería erróneo, porque se crean objetos, llamados *formas* y *controles*, que hacen que la aplicación funcione. Sin profundizar en el tema, estos objetos carecen de las propiedades herencia y polimorfismo, las cuales tienen que estar presentes en un verdadero ambiente orientado a objetos. Además, se están creando aplicaciones que correrán en Windows, el cual es un ambiente orientado

a *lo visual* que proporciona librerías de herramientas y objetos para aplicaciones de programación. La mejor ventaja de utilizar Windows es que proporciona una interfaz consistente y manejable para desplegar información en muchas aplicaciones diferentes que comparten controles similares.

La filosofía de la interfaz Windows se basa en la utilización de metáforas visuales para realizar acciones y tareas.

El paradigma es simple: después de que se ha utilizado una aplicación Windows, se aprende fácilmente a utilizar otra ya que se está familiarizado con la interfaz.

Al utilizar controles Windows como menús, iconos, barras de desplazamiento y cajas de diálogo, la información es presentada a los usuarios en un formato dinámico y visualmente interesante.

Visual Basic reduce drásticamente las dificultades y la cantidad de tiempo de la programación en Windows, se puede construir una aplicación Windows dinámica en pocos días. Ya que este lenguaje es *interpretado* más que compilado, las aplicaciones están listas para correr tan pronto como se termine de escribir el código. En contraste con un lenguaje compilado, un programa debe compilarse antes de que pueda ser ejecutado. Esto significa que un compilador convierte los programas al lenguaje de máquina. Esta conversión se realiza solamente una vez, cada vez que se corre el programa, el lenguaje de máquina ya está disponible y puede ser ejecutado de manera directa por la computadora.

1.6.1 Requerimientos de Hardware

Antes de instalar el programa Visual Basic, hay que asegurarse de que la computadora cumpla con los siguientes requerimientos mínimos :

- Computadora IBM o PC compatible con un procesador 80386 o más poderoso
- Al menos 50 MB de espacio disponible en disco duro para una instalación completa.
- Unidad de 3.5" , o unidad de CD-ROM
- Cualquier monitor VGA o de alta resolución soportado por Microsoft Windows 3.1 o posterior.
- Al menos 8 MB de memoria RAM, requeridos para Windows 95. (Esto varía dependiendo del tipo de librerías específicas o DLL's que se incluya en las aplicaciones).
- Ratón.
- Para versiones de Visual Basic de 32 bits, se deberá contar con Windows 95 o más reciente, o Windows NT 3.51 o más reciente.
- Para versiones de Visual Basic de 16 bits, se deberá contar con Windows 3.1 o más reciente en modo mejorado.

Para cualquier duda, consultar el archivo readme.doc que viene en el directorio raíz del disco de instalación.

1.6.3 Elaboración de Programas Orientados a Eventos

Cuando se inicia una aplicación, durante la corrida sólo las formas que se crearon y se programaron serán visibles. Una *forma* es una ventana que se crea y se adapta a las necesidades para cada aplicación. Esta es la base de cualquier aplicación Visual Basic que eventualmente correrá como un programa independiente de Windows. Un *control*, es el nombre preciso para cualquier objeto que se dibuje en una forma, pero también se refiere a la forma misma. Los botones de comando, menús, cajas de diálogo y barras de desplazamiento, son ejemplos de controles. Cuando un usuario activa un control, el código BASIC que se tiene asignado el control, es invocado y realiza una tarea específica. Un *módulo*

es una estructura para escribir el código que se le asigna a una forma y sus controles. Después de dibujar los controles que se desean que aparezcan en una forma, se escribe el código funcional para cada control que los usuarios pueden invocar. Visual Basic utiliza, la metáfora del "evento" para describir su paradigma de programación, siempre utilizará este enfoque orientado a los eventos cuando se cree la aplicación. *Orientado a eventos* significa que todos los controles que se dibujen en una forma especifican cómo se comportará la interfaz. Esto es, los controles de Visual Basic esperan que sucedan eventos particulares antes de que responda. Un *evento* es una acción que es reconocida por un control, es código escrito para cada control a fin de que algo suceda cuando los usuarios interactúan con la forma.

Mientras se está en Visual Basic y se corre una aplicación, cada línea de código es interpretada y ejecutada por separado cada vez que aparece. Es esta ejecución individual de cada línea la que puede hacer que un programa interpretado corra con mayor lentitud que un programa compilado. Por consiguiente, el archivo terminado también es más grande. Por supuesto, un archivo compilado es un archivo ejecutable, que ya no es interpretado.

1.6.4 Terminología de Visual Basic

La necesidad de familiarizarse con la terminología de Visual Basic tiene su singular enfoque orientado a eventos, para programar aplicaciones. Por consiguiente, es útil para describir los elementos de una aplicación Visual Basic:

- *Control*. Un término general utilizado para describir cualquier forma o elemento gráfico que se dibuje sobre la forma, incluyendo cajas de texto, cajas de listado, botones de comandos, cajas de imágenes, barras de desplazamiento e íconos. Una definición más precisa de un control, es la de datos acoplados con una serie de rutinas, conocida como métodos. Las únicas operaciones que

pueden desarrollarse sobre un control son aquellas definidas como métodos para él.

- *Evento*. Acción reconocida por un control Visual Basic.
- *Forma*. Una ventana que se crea y adapta a las necesidades de la aplicación.
- *Método*. Una palabra de código Visual Basic que es similar a una función o instrucción, pero que siempre actúa sobre un control particular. Para cada control, Visual Basic predefine una serie de métodos que se pueden utilizar.
- *Procedimiento*. Un término que se refiere tanto a procedimientos **Sub** (Sub procedimientos) y **Function** (Función). Un procedimiento es sencillamente una secuencia de instrucciones Visual Basic que son ejecutables en grupo durante la corrida. Existen dos tipos de procedimientos :

Procedimientos de eventos y procedimientos generales. Los *procedimientos de eventos* se limitan a las formas y los controles, mientras que los *procedimientos generales* se utilizan durante toda la aplicación y pueden ser requeridos por procedimientos de eventos.

- *Proyecto*. Conjunto de todos los archivos que forman la aplicación.
- *Propiedad*. Una característica o atributo de un control para cada tipo de control, Visual Basic define una serie de propiedades que se aplican solamente a ese control.
- *Definición*. El valor de una propiedad, se puede cambiar la definición de la mayoría de las propiedades mientras se está construyendo una aplicación. El código de una aplicación que se corre también puede cambiar las definiciones.

1.6.5 Acceso a la Barra de menús

El ambiente de programación Visual Basic está formado por cinco elementos a los que se puede acceder a fin de crear aplicaciones propias :

- Barra de menús
- Barra de propiedades
- Ventana Form
- Ventana Project
- Caja de herramientas

Estas ventanas son lo primero que se observa cada vez que se inicia Visual Basic como se muestra en la figura 1.6.5.1

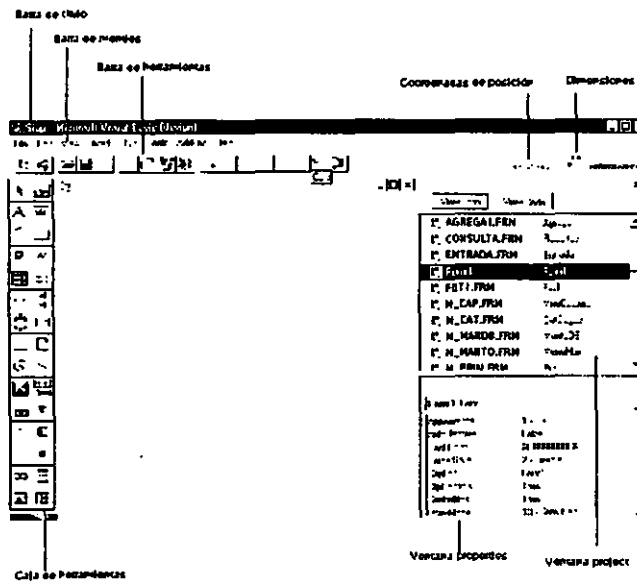


Figura 1.6.5.1 El ambiente de programación Visual Basic

Barra de menús

La barra de menús hace aparecer los nombres de los menús que están disponibles para utilizar en la ventana activa. Encima de menú de barra está la barra de título, los botones de maximización y minimización y la caja de control los cuales se muestran en la figura 1.6.5.2 . Para entrar al menú hay varias maneras : se puede hacer clic sobre un nombre del menú para abrirlo y hacer que aparezcan sus comandos o también puede oprimir una tecla aceleradora de nombre de menú, reconocidas porque la letra esta subrayada. Por ejemplo, la tecla aceleradora para el menú File (Archivo) es F.



Figura 1.6.5.2 Barra de menús y títulos de la ventana

Características

El sistema dinámico de ventanas, menús y herramientas permite crear formas y controles, manipular archivos de proyectos, manipular texto, correr programas y depurar el código. Entre las características más interesantes de Visual Basic se incluyen las siguientes :

Cuando se comienza un programa durante la corrida, se puede seleccionar comandos del menú File de la figura 1.6.5.2, para poner o quitar puntos de interrupción del código y para ejecutar un programa a fin de evaluar cómo se ejecutan las instrucciones.

- La ventana **Inmediate** (Inmediato) permite observar como se ejecuta una instrucción en particular mientras se corre un programa.

- La ventana **Menu Design** (Diseño de menú) está disponible para crear menús con teclas aceleradoras, la figura 1.6.5.3 muestra las características de esta ventana.
- Se puede crear un sistema de ayuda completo, un tutorial y un índice por medio del menú **Help** (Ayuda).
- La ventana de propiedades le ofrece una manera conveniente de escoger propiedades de control y cambiar definiciones durante el diseño, la figura 1.6.5.4 muestra los elementos de esta barra.

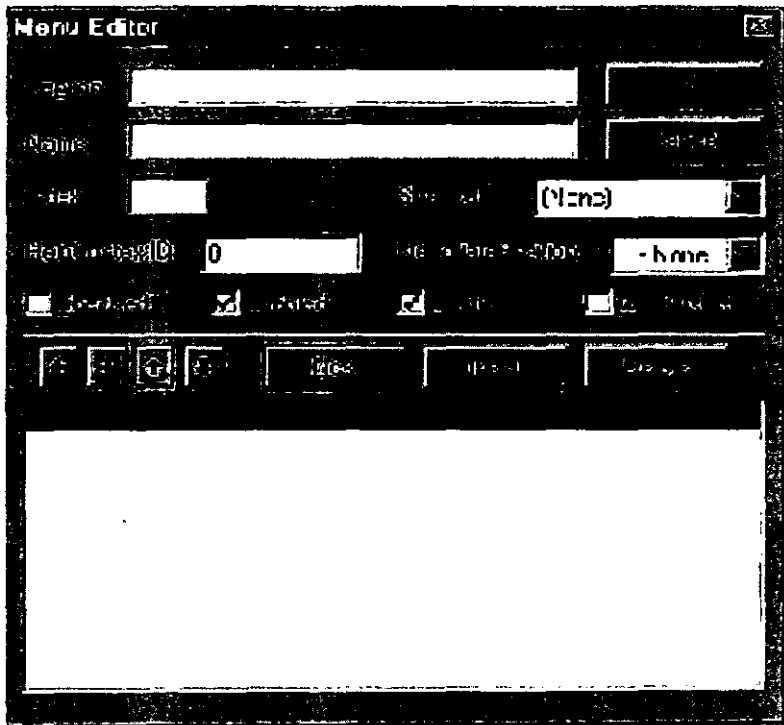


Figura 1.6.5.3 Ventana de Diseño de menú.

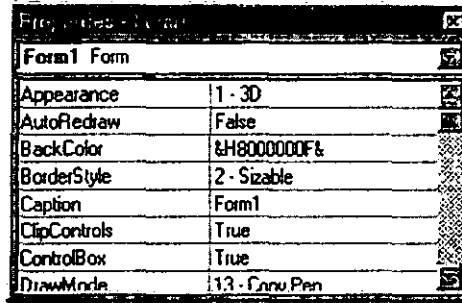


Figura 1.6.5.4 Ventana de Propiedades

- La caja de herramientas permite crear 15 tipos diferentes sobre una forma.
- Una ventana de forma, como muestra la figura 1.6.5.5, por omisión, aparecerá automáticamente cuando se inicie Visual Basic. Cuando se hace clic con el botón izquierdo del **mouse** (ratón, dispositivo de señalización o apuntador óptico) sobre una forma o control, aparece la ventana **Code** (Código) en la cual se escribe el código funcional para una aplicación.
- La ventana **Code**, figura 1.6.5.6, muestra las cajas del listado de las cuales se selecciona los nombres de controles y eventos, y los nombres de procedimientos generales.

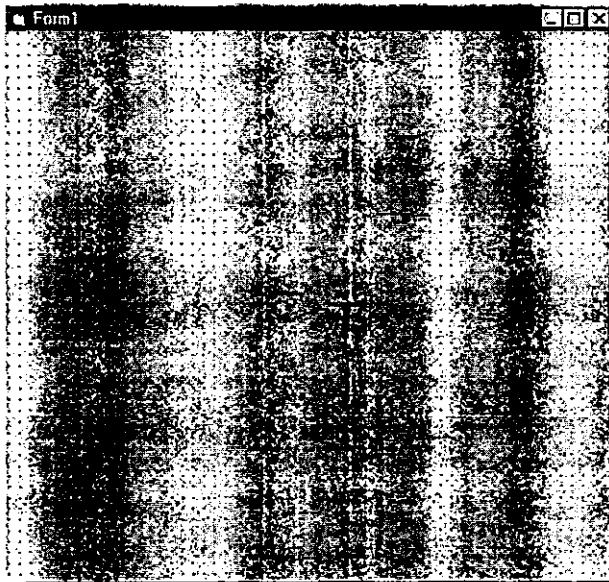


Figura 1.6.5.5 Ventana Forma de muestra

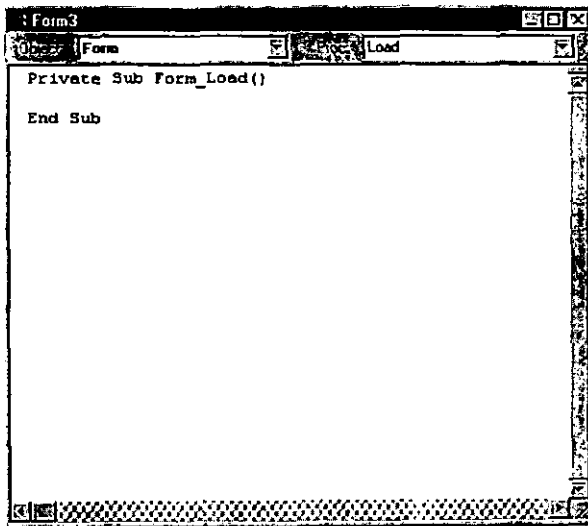


Figura 1.6.5.6 Ventana de código

- La ventana de **Project** (Proyecto) organiza y muestra los nombres de todas las formas y módulos que son parte de una aplicación particular. Se puede mostrar el código para una forma destacando el nombre del archivo en esta ventana y haciendo clic en el botón **View Code** (Vista del código); no tiene que mostrar primero la forma (o control) real para observar el código. La figura 1.6.5.7 muestra la ventana **Project**.

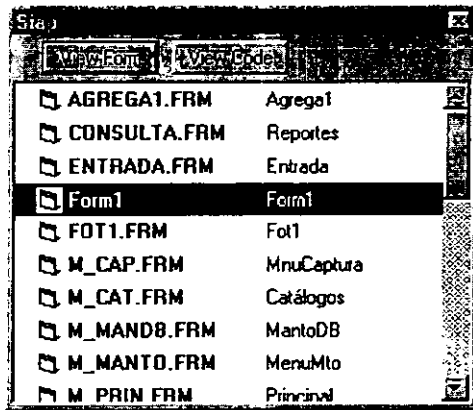


Figura 1.6.5.7 Ventana Project

1.6.6 Creación de Controles con la Caja de Herramientas

La caja de herramientas Visual Basic contiene un conjunto de 16 herramientas (las cuales se muestran en la figura 1.6.6.1) que puede utilizar para dibujar, mover o redimensionar los controles de su forma y a los cuales posteriormente se les asignará código BASIC funcional.

Creación de un control

Para crear un control, se siguen los pasos :

1. Hacer clic sobre la herramienta **Text Box** (Caja de Texto) para seleccionarla.
El apuntador del ratón se convierte en una cruz.
2. Colocar la cruz sobre la forma en la posición deseada.
3. Mantener apretado el botón izquierdo del ratón y arrastrar la cruz hacia abajo diagonalmente hacia la derecha hasta que el control tenga el tamaño deseado.
4. Soltar el botón del ratón. El encabezado **Text1** aparecerá sobre la caja de texto y *Selection Handles* aparece alrededor de la caja para indicar que está seleccionada.
5. Centrar la caja de texto sobre la forma, como se muestra en la figura 1.6.6.2 arrastrándola.
6. También se puede arrastrar las manijas de selección para aumentar o disminuir el tamaño de la caja de texto.

Modificación de las Propiedades de un Control

Ahora que se ha creado un control de caja de texto, se puede establecer las propiedades tanto para la forma como para la caja de texto para que aparezcan como se desee.

Cambiar **Caption** (Título) que aparece en la barra de título de la forma :

Seleccionar la forma haciendo clic sobre ella. El nombre **Form1** aparece en la caja de listado **Settings** (Valores) para la propiedad **Caption**. (La caja de listados **Settings** esta a la mitad de la barra de propiedades. debajo de la barra de menú de **Visual Basic**).

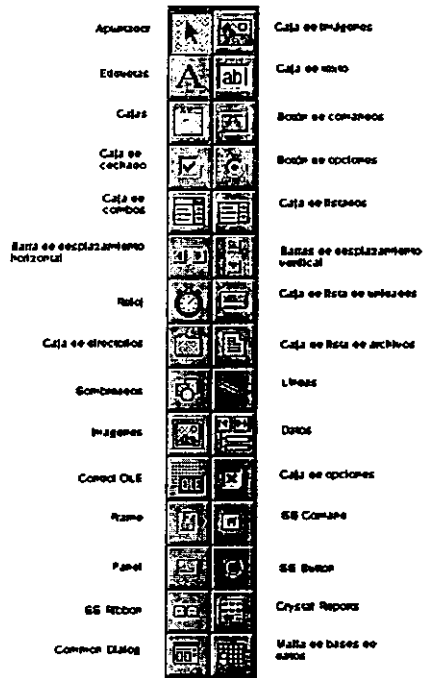


Figura 1.6.6.1 Caja de herramientas de Visual Basic

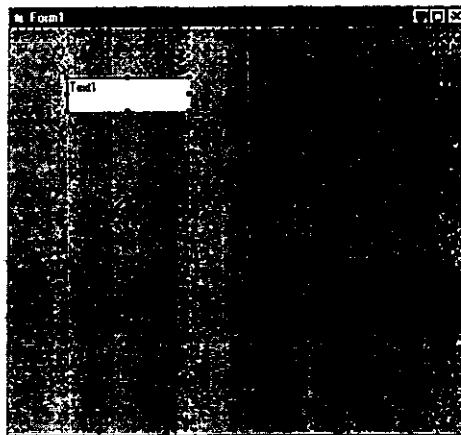


Figura 1.6.6.2 Caja de texto sobre una forma

2. Mover el apuntador del ratón a la caja de listados Settings. Se convertirá en un cursor en forma de "I" al que se le llama rayo "I".

3. Hacer resaltar el título haciendo doble clic sobre él colocando el rayo "I" al principio de la palabra y arrastrándolo a la derecha.

4. Oprima la tecla **Delete** (Borrar) para quitar Caption

5. Teclar **Bienvenido** y presionar **Enter** para confirmar la nueva definición, o hacer clic sobre el botón **Enter** (botón izquierdo del ratón para configuración de mano derecha).

Ahora se cambia la propiedad **Name** (Nombre) de la caja de texto para que se pueda hacer referencia a ella en un procedimiento. :

1. Con la caja de texto sobre la forma seleccionada, hacer clic sobre la caja de listado **Properties** (Propiedades), a la izquierda de la caja de listado **Settings**.
2. Seleccionar la propiedad **Name**, si no ha sido seleccionada. El nombre **Text1** aparece en la caja de listado **Settings**.
3. Teclar **Display** en la caja de listado y oprimir **Enter** para confirmar el cambio.

Ahora se cambia la propiedad **Text** del control :

1. Seleccionar la propiedad **Text** en la caja de listado **Properties**. El nombre de **Text1** aparece en la caja de listado **Settings**.

2. Hacer doble clic dentro de la caja de listado Settings para resaltar el texto.
3. Oprimir la tecla **Delete** para quitar el texto.
4. Oprimir **Enter** para confirmar el cambio.

Finalmente, cambiar la propiedad **BorderStyle** del control :

1. Hacer clic sobre la caja del listado de **Properties** que está abierta, mover consecutivamente la lista, y seleccionar la propiedad **BorderStyle**. La definición 1 aparece por omisión en la caja Settings, el cual es el valor por omisión de una sola línea compuesta.
2. Hacer clic sobre la flecha, del lado derecho de la caja de listado Settings (la flecha indica que existen más definiciones para la propiedad). Aparece una lista.
3. Seleccionar la definición Settings, la cual es el valor para no tener límites.
4. Hacer clic sobre el botón **Enter** para confirmar el cambio.

La caja de texto deberá aparecer como la que se muestra en la figura 1.6.6.3

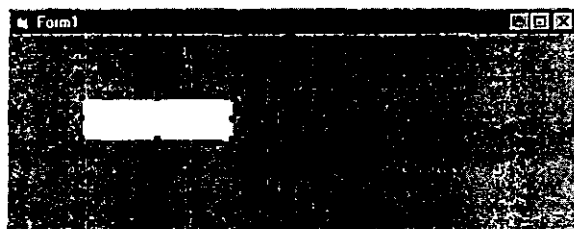


Figura 1.6.6.3 Apariencia del título modificada

1.6.7 Escritura del Código Funcional

Para escribir el código para un evento **Load** (Carga), se deben seguir los pasos :

1. Hacer doble clic en cualquier parte de la forma. La ventana de código aparecerá como se muestra en la figura 1.6.7.1. Una plantilla de código aparece automáticamente mostrando la primera y la última línea del procedimiento de evento. En este ejemplo, el procedimiento de evento es identificado con el nombre **Form_click()**.
2. Hacer clic sobre la flecha hacia abajo del lado derecho de la caja de listado **Procedures**.

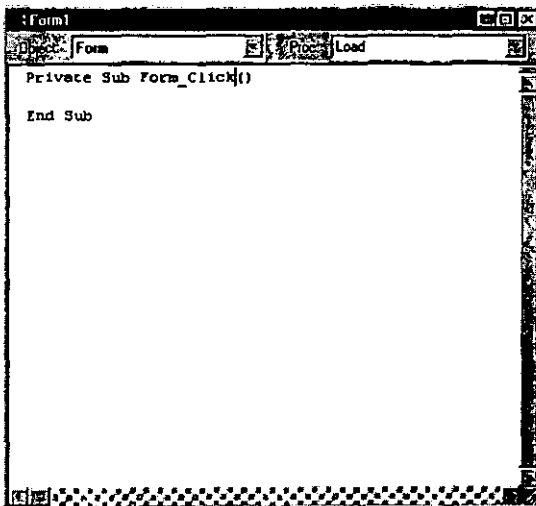


Figura 1.6.7.1 El código de la ventana con el procedimiento de evento clic mostrado en la forma

3. Seleccionar **Load**. El nuevo evento aparecerá en la plantilla de código automáticamente.
4. Colocar la barra destellante de inserción entre las instrucciones **Sub** y **End Sub**.
5. Presionar la tecla **Tab** para darle sangría a esta línea.
6. Teclar
`Display.Text="Bienvenido a Visual Basic"`
7. Hacer doble clic sobre la caja de control para cerrar la ventana de código.
8. Abrir el menú **Run** y seleccionar el comando **Start** para correr el programa. La aplicación hará aparecer el mensaje tecleado como se muestra en la figura 1.6.7.2.
9. Abrir el menú **Run** de nuevo y seleccionar el comando **End** para detener la ejecución del programa.

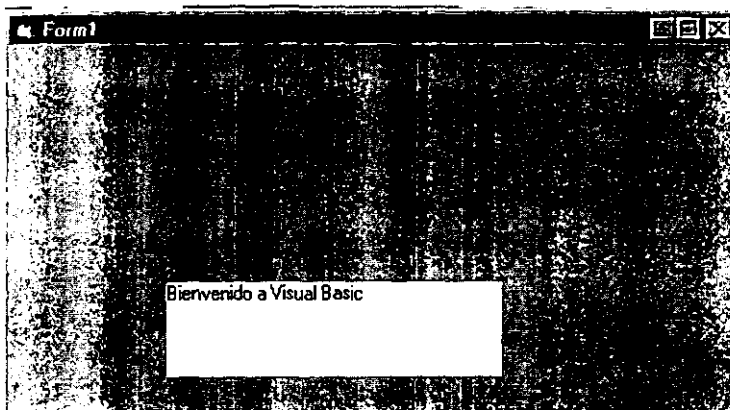


Figura 1.6.7.2 La forma con el mensaje desplegado al ejecutar el Proyecto.

Para cualquier control automáticamente, sólo hacer, doble clic sobre la herramienta en particular, aparecerá un control de tamaño por omisión en el centro de la forma. Para redimensionar un control, arrastrar sus manijas de selección. Para repositionar un control, sólo se debe arrastrar a otra posición dentro de la forma.

1.6.8 Utilización del Ambiente de Trabajo Visual Basic

¿ Qué es una interfaz ?

El término interfaz para el usuario es difícil de definir. Si se les pregunta a los programadores ¿como la definirían? , es probable que se obtengan respuestas muy diferentes porque no hay una interfaz para el usuario estándar para aplicaciones basadas en DOS (Disk Operating System). Aunque las aplicaciones Windows suelen reflejar especificaciones que los programadores pueden seguir para mantener una consistencia funcional entre las aplicaciones, la consistencia no es forzosa. Esto no quiere decir que las aplicaciones Windows no se apeguen a las convenciones generales de diseño. Pero se puede aplicar una cierta libertad, que con frecuencia resulta en aplicaciones Windows tan diferentes como los programadores que las crearon.

El área de las interfaces gráficas para el usuario abarca todo desde los aspectos filosóficos hasta psicológicos, y se ha convertido en una importante disciplina, captando la atención de programadores e investigadores en las áreas de la computación, ingeniería y psicología cognoscitiva. Lo que ha surgido es una visión del usuario que utiliza una computadora para realizar muchas tareas y cumplir con su trabajo.

Generalmente, la interfaz entre un usuario y la computadora consiste en una pantalla, un teclado y un ratón. Eso es lo que presenta información al usuario y acepta información del mismo.

El usuario interactúa con su computadora respondiendo con lo que ve en la pantalla, escribiendo comandos en el teclado o apuntando y haciendo clic en el ratón.

Mediante una aplicación, el usuario se comunica o más precisamente, tiene una *interfaz* con su computadora. De hecho, una aplicación Visual Basic como todas las aplicaciones Windows, se accesa mediante una interfaz gráfica con el usuario. Una interfaz es más que una visión de objetos, textos, gráficas y patrones. Una interfaz gráfica para el usuario (o interfaz computadora-humano), tal como una aplicación, es el "diálogo", que ocurre entre una computadora y sus usuarios. En este caso, el usuario interacciona con la computadora dándole comandos; la computadora, a su vez, le responde al usuario realizando una tarea. En otras palabras, la interfaz es aquel componente de una aplicación que traduce una acción del usuario con las consecuencias de sus acciones.

Visual Basic permite diseñar la interfaz de una aplicación utilizando la interfaz Windows para dar a los usuarios un método consistente de interacción con la computadora. Probablemente la consistencia sea la mayor ventaja de una interfaz. Todas las aplicaciones utilizan comandos y controles cuando se usa una aplicación Visual Basic. Una manera como Visual Basic mantiene la consistencia de una aplicación con otra, es mediante el uso de objetos y metáforas de la vida real. La metáfora de las ventanas es utilizada porque representa una experiencia de la vida real.

La creación de una aplicación efectiva comienza con un buen diseño y una visión amplia del contenido y propósito de la aplicación. La manera en que el usuario

organiza una aplicación depende de la información, que ésta contiene y de cómo se espera que el usuario la utilicen y no sencillamente para que la observen.

Finalmente, una persona correrá la aplicación que el programador ha creado *fuera* de *Visual Basic*, simplemente como cualquier otra aplicación Windows. Por tanto, es importante que el programador se detenga y se haga varias preguntas fundamentales que afectarán el diseño :

- ¿Cuál es el propósito de la aplicación ?
- ¿ Qué razón (o razones) tiene el usuario para utilizarla ?
- ¿ Qué información se presenta en la aplicación ?
- ¿ Cómo se da a entender esta información en la aplicación ?
- ¿ Cómo encuentra el usuario la información en la aplicación ?
- ¿ Qué indicaciones le da la aplicación al usuario ?

Si se toma en consideración el diseño de una aplicación antes de comenzar, se verán mejores resultados y se ahorrará mucho tiempo.

Mediante el proceso de hacer claras las necesidades de los usuarios de la aplicación, se creará para que la utilicen todas las personas. Los usuarios serán capaces de transitar fácilmente por ella, siguiendo una trayectoria lógica, intuitiva y no lineal, marcada con condiciones claras. Una de las maneras de dar instrucciones es representarlas mediante controles. Por tanto una aplicación bien diseñada debe guiar a los usuarios de un punto a otro, permitiéndole salir en cualquier momento, convirtiéndose, los controles que se coloquen en la aplicación en señales de camino que los usuarios seguirán, para no perderse.

1.6.9 Alcance de las Variables Locales y Globales

Todos los lenguajes de programación de alto nivel funcionan sobre el principio de establecer un alcance de variables locales y globales. El *alcance de variables* se refiere a cómo una aplicación reconoce cada una de las variables que se han declarado en la aplicación. En Visual Basic, el alcance está determinado sobre la base de dónde se declararán las variables: en el módulo global, o en un módulo en la forma misma.

Las variables y constantes son con frecuencia declaradas en el módulo global de la aplicación. El colocar las declaraciones aquí, brinda a las variables su más amplio alcance, por eso el nombre de "global". Su colocación también las pone en el nivel más lejano de alcance en una aplicación. La ventaja de declarar variables en el nivel global, es que siempre mantienen su valor en toda la corrida de la aplicación. En otras palabras, las variables son persistentes.

Las variables a nivel modular se incluyen en la selección de declaraciones en un módulo separado, no en la sección de una forma. La ventaja de declarar las variables en un módulo, es que todos los procedimientos en la aplicación pueden reconocer y compartir las variables. Esto es importante cuando se tiene más de una forma en una aplicación refiriéndose a los mismos procedimientos generales y funciones.

Por ejemplo, si se tiene que declarar la variable una vez en el procedimiento general de un módulo. Cualquier procedimiento de evento dentro de toda la aplicación se puede entonces llamar al procedimiento general.

Las variables a nivel forma se incluyen en la sección de declaraciones de un procedimiento general de forma. Si se declaran las variables a nivel de forma cuando se desea que varios procedimientos dentro de la forma compartan datos;

una variable a nivel de forma es reconocida por todos los procedimientos asignados a la forma. Tanto los niveles de forma como las de nivel modular se consideran también persistentes; mantienen sus valores después de que un procedimiento ha sido ejecutado, pero sólo dentro de la forma.

Las variables locales se reconocen solamente por el procedimiento en el cual aparecen. También tiene el nivel más cercano de alcance. Utilizar una variable local es una buena manera de declarar cualquier variable que es temporal, porque el valor de la variable puede con frecuencia cambiar en un procedimiento. Además cada procedimiento puede declarar una variable con el mismo nombre; se puede nombrar muchas variables locales con el mismo nombre, en tanto se coloquen en procedimientos separados. Si se cambia un nombre de variable, las otras variables con el mismo nombre no serán afectadas.

Es necesario tener en mente que pueden ocurrir conflictos cuando se declara una variable local con el mismo nombre que se le ha dado a una variable en otro nivel.

Como una variable local tiene el más cercano nivel de alcance, cualquier referencia a la variable será accesada primero. Si una variable con el mismo nombre existe a un nivel más alto, la instrucción en la cual se encuentra la variable podría no ejecutarse adecuadamente, dando como resultado un error. Cuando se depura un programa, no hay nada más frecuente que descubrir que el culpable del conflicto es un nombre de variable. Cuando sea posible se deben declarar las variables en un solo lugar.

CAPITULO 2
ANÁLISIS DE LA PROBLEMÁTICA

2. ANÁLISIS DE LA PROBLEMÁTICA.

2.1 Problemática Actual.

El Sistema de transporte Colectivo Metro en vinculación con la Dirección general del Patrimonio Inmobiliario la cual tiene la encomienda de integrar, actualizar y consolidar el patrimonio inmobiliario del Distrito Federal, unen esfuerzos para permitir el resguardo, cuidado, protección legal, recuperación administrativa del Patrimonio del Sistema de Transporte Colectivo Metro.

Esta vinculación del Sistema de Transporte Colectivo Metro y la Dirección General del Patrimonio Inmobiliario lleva el nombre de Programa de Regularización de los Bienes Inmuebles del S.T.C. Metro, el cual tiene la tarea de actualización e integración del archivo físico del Sistema de Transporte Colectivo Metro y para la creación del archivo electrónico con información automatizada de cada uno de los Inmuebles del Sistema de Transporte Colectivo Metro.

Se procedió a la elaboración de un análisis y diagnóstico para determinar la calidad de la información disponible y la coordinación de los documentos del archivo físico del Sistema de Transporte Colectivo Metro y la Dirección General del Patrimonio Inmobiliario.

En un primer análisis, reflejó que el problema inicial era de la información más que de informática. Este análisis lo reflejamos en el siguiente diagrama de la fig. 2.1.a y 2.1.b.

El archivo del Sistema de Transporte Colectivo Metro, no contaba totalmente con toda la documentación referente a los predios expropiados para la construcción de la red del Metro en la Ciudad de México, a partir del comienzo de la primera línea del Metro, por

el año de 1967, algunos predios en la actualidad que son del Metro están ocupados por particulares o en este momento son usados como vialidad.

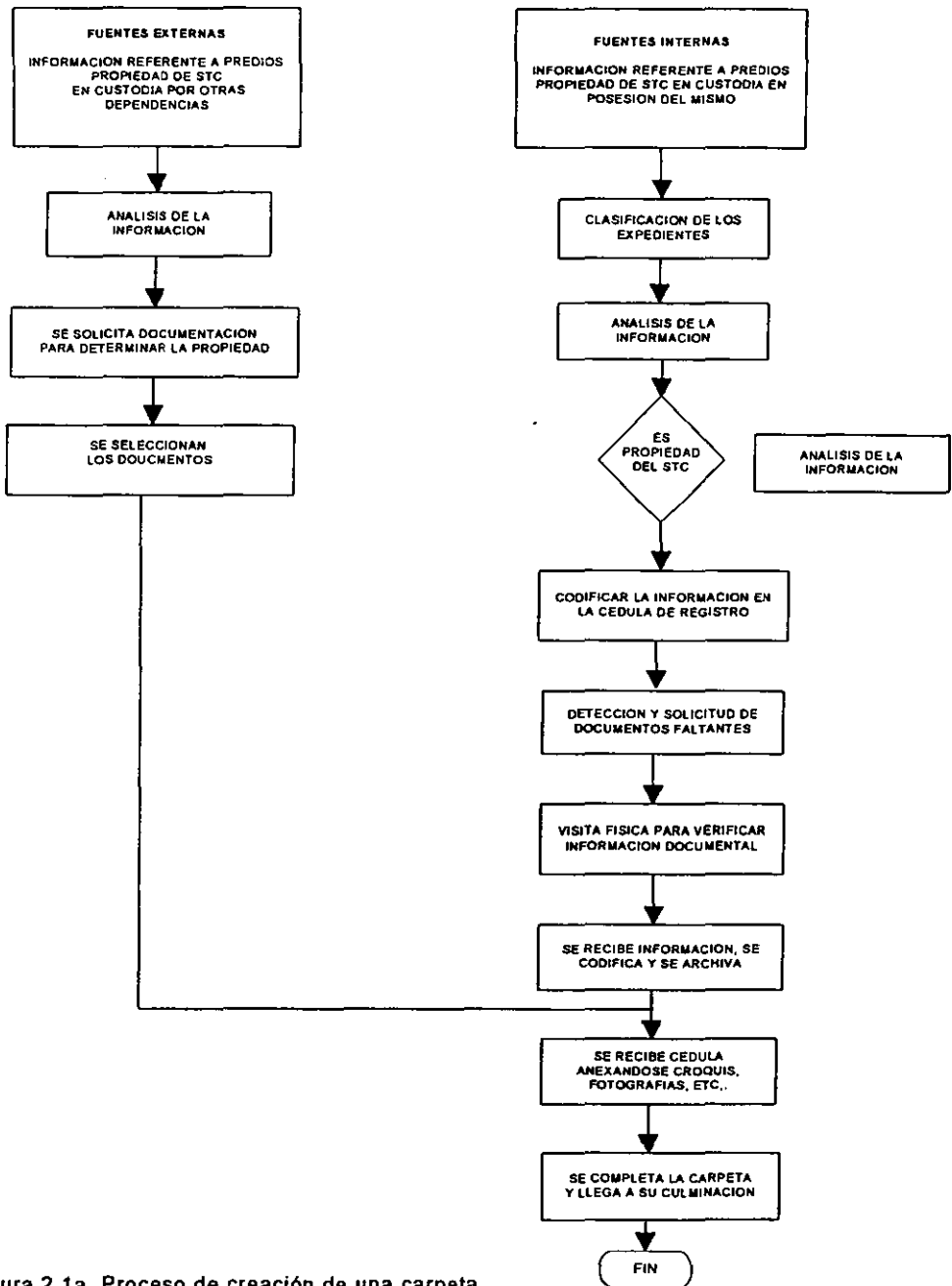


Figura 2.1a Proceso de creación de una carpeta

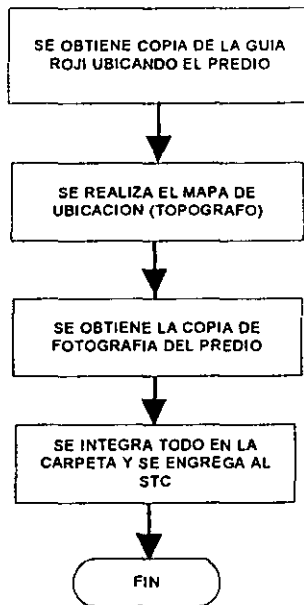


Figura 2.1b Proceso de creación de una carpeta al momento de recibir la información

Hoy en día, el Sistema de Transporte Colectivo Metro tiene en uno de sus puntos principales la obtención de toda la información referente a sus predios por donde pasan las líneas del Metro.

La problemática actual del Sistema de Transporte Colectivo Metro es la siguiente:

- Extravío de documentos originales, desde títulos de propiedad, documentos que amparan la propiedad, hasta expedientes completos.
- Falta de Seguimiento en los asuntos relativos a los Inmuebles del S.T.C.
- Inexistencia de control, orden y seguridad en el archivo y los expedientes que lo conforman.

- Cajas con documentación diversa en malas condiciones.
- El aumento de documentos, los cuales ocuparon un espacio para su administración.
- Con el transcurso del tiempo, la documentación se deteriora.
- El uso de los expedientes llega a deteriorar los documentos.

Es muy importante describir el estado de los expedientes para apreciar el proceso de depuración, pues si pudiera pensarse que cada expediente corresponde a un inmueble en particular, la realidad en el archivo es diferente; en un mismo expediente están integrados documentos relativos a varios inmuebles. El grado de desorganización es evidente, dentro de un mismo expediente con documentación de múltiples inmuebles, no existe una secuencia lógica, ya que los documentos de cada inmueble no están juntos ni ordenados. Esto conlleva a la separación de documentación de cada inmueble en expedientes individuales, para saber con que información se cuenta. Esta acción va a facilitar también la identificación de inmuebles del S.T.C. y a particulares que no forman parte del Patrimonio Inmobiliario.

- No se cuenta con el archivo en el mismo lugar donde se ubica la sede del programa.
- El equipo informático es limitado

El buen desempeño del área se ha visto afectado desde su creación, ya que ha tenido que enfrentar una serie de problemas o aspectos negativos, que a veces rebasan su capacidad de solución.

A continuación se describen algunos, enfocados a los puntos anteriormente mencionados:

- Extravío de documentos originales desde títulos de propiedad, documentos que amparan la propiedad, hasta expedientes completos.

Al momento de iniciar el programa en los archivos tanto de la Dirección General del Patrimonio Inmobiliario como del Sistema de Transporte Colectivo Metro, no se cuenta con documentos originales de propiedad, por lo que no puede completarse ya que se requiere que toda documentación original no sea separada de su expediente original, sino que se entregue una copia de estos documentos para evitar su pérdida.

- Falta de seguimiento en los asuntos relativos a los inmuebles del Sistema de Transporte Colectivo Metro.

En varias ocasiones no puede darse un seguimiento a un expediente, debido a que hace falta algún dato importante por no llevar el control de la documentación que se retira, por lo que si se entregara solo una copia, no se perdería el control de éste. Otro de los casos por lo que no se sigue el procedimiento es que se asigna a un abogado el expediente y debido a esto llega a traspapelarse y no se lleva el control adecuado de los expedientes.

- Con el transcurso del tiempo la documentación se deteriora.

Debido a que en este programa, una de las materias primas importantes para la realización de un expediente es el papel, este recurso al pasar el tiempo llega a deteriorarse por muchos aspectos, como son: factores ambientales, como la humedad hace que este material llegue a deteriorarse, así como los microorganismos que llegan a formarse con el papel llegando a dañar la información; factores físicos, este material

no es muy resistente y menos con el viaje de persona a persona porque poco a poco llega a gastarse y con el transcurso del tiempo su resistencia es casi nula, llegando a tener pérdidas irreparables con estos documentos.

- El aumento de documentos, los cuales ocuparán un espacio para su almacenamiento.

Debido al gran número de expedientes y de que cada expediente tiene un número no menor a 10 hojas por cada uno, llega un momento en que ya no es posible tener los documentos en un lugar pequeño, por lo que se requiere ya de un lugar especial para esta documentación para su control y resguardo, y dicho lugar especial deberá estar en condiciones óptimas para que no se deterioren los documentos, pues requiere de un clima templado, de un espacio de acuerdo al número de expedientes manejados y sobre todo una organización eficaz.

- Cajas de documentación diversas en malas condiciones.

Por motivo de que algunos expedientes llevan años resguardados en un archivo sin condiciones óptimas, ya no es posible manejar con facilidad algunos documentos debido a que por muchos años no se tuvo la precaución de resguardar la documentación en lugares donde no influyeran factores ajenos a los expedientes, por lo que en muchas ocasiones ya no sirve gran parte de esa documentación, porque no llega a entenderse ninguna idea por lo menos de lo que está escrito. Un ejemplo de esto es que hay expedientes de más de 100 años de existencia y debido a esto requiere de un cuidadoso manejo.

- El uso de los expedientes llega a deteriorar los documentos.

Simplemente el contacto del ser humano en el papel llega a maltratar poco a poco el papel debido a las reacciones químicas que produce el cuerpo humano, el estado físico del documento llega a deteriorarse.

- No se cuenta con el archivo en el mismo lugar donde se ubica la sede del programa.

Los expedientes son materia prima de nuestro proceso de trabajo y al no tenerlos en el mismo edificio nos hemos visto obligados a tener grupos de gavetas en la Dirección, por lo que se requiere de un mayor número de personas que se encarguen de llevar el control de la distribución de los expedientes, por lo que se esta dependiendo de las condiciones climatológicas y equipo de traslado de archivo a la sede.

- El equipo informático con el que se cuenta es limitado, considerando todo el proceso que lleva la elaboración de un expediente sin el uso de los medios informáticos. Para su elaboración se requiere lo siguiente:

- Guía Roji (en formato papel)
- Diario Oficial (en formato papel)
- Croquis (en dibujo manual)
- Mapa Catastral (en copia de papel)
- Fotografía (en rollos fotográficos de papel) y equipo topográfico.

2.1.1 Areas de interés

El Sistema de Transporte Colectivo Metro inició en diciembre de 1995, el Programa de Regularización de los Bienes Inmuebles propiedad o en posesión de este Organismo, hasta octubre de 1996. Únicamente había logrado regularizar 202 inmuebles de un universo de 2500 predios aproximadamente.

Por lo anterior, se solicitó la participación de la D.G.P.I. Dirección General del Patrimonio Inmobiliario, para dar apoyo al Metro en la regularización de los predios, sin embargo, no se continuo con este proyecto en ese momento debido a que el Metro no podía apartar suficientes recursos, a principio del mes de julio se reinicio esta participación de ambas dependencias, empezando a canalizar un grupo de personas a buscar antecedentes de propiedad lográndose un considerable avance en el programa hasta el momento.

Las dependencias que de igual manera han colaborado con este programa son:

- El Sistema de Transporte Colectivo, el cual al principio de este programa proporcionó aproximadamente 1000 predios, de los que se solicitarán antecedentes registrales.
- Se recibió de COVITUR una carpeta con relación de predios comprados para la construcción de las líneas 1 a 7, 9 y A.
- De la primera búsqueda en el archivo de la Dirección General del Patrimonio Inmobiliario, se identifico un tanto más de Inmuebles con antecedentes de propiedad, tales como contrato de compraventa y de escrituras.
- También se ha obtenido información del Diario Oficial de la Federación.

- Se acordó con la Dirección General Jurídica y de Estudios Legislativos el acceso a sus archivos, obteniéndose un listado de inmuebles comprados para la construcción de las líneas " A y 8" del Metro o para obras complementarias. Asimismo se fotocopia la documentación más importante de un porcentaje de los predios encontrados.
- Se solicitó al Archivo General de Notarias el envío de testimonios de escrituras.
- El Sistema de Transporte Colectivo, remitió a esta Dirección antecedentes registrales de predios en posesión del mismo y por parte del Registro Público se reciben asientos registrales.
- Se solicitaron a la Tesorería 50 planos. La información obtenida de estos facilitó la identificación de Inmuebles afectados por la construcción del Metro.
- Se solicitó a SERVIMET la información relacionada con el Programa de Regularización de Bienes Inmuebles del Metro, es importante resaltar que a fin de este programa es tener plenamente identificados los inmuebles de forma individual y contando con información sistematizada que permitiera conocerlos desde la computadora por medio de imágenes digitalizadas.
- Como se observo en este Programa, entran alrededor de 10 dependencias en la obtención de información para crear una carpeta - expediente para la identificación de un inmueble, con más de 10 hojas cada expediente (Figura 2.1.1).

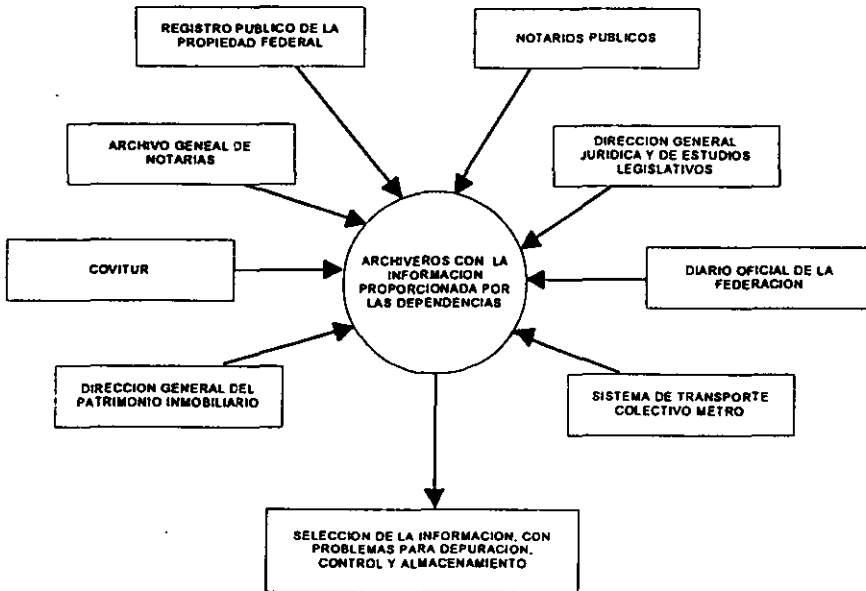


Figura 2.1.1 Areas de Interés

2.1.2 Consecuencias del problema

Como el producto final de este conjunto de acción es la carpeta con la información de investigación integrada, y los problemas que anteceden son los siguientes:

- Desorganización de la información
- No existe un lugar amplio y seguro para la documentación obtenida de las dependencias y almacenamiento de las carpetas terminadas.
- No se tiene una presentación de alta calidad.
- Retraso en la integración de las carpetas.

- Solamente se cuenta con una carpeta original, y debido a esto se tiene dos opciones para tomar la información:

Obtener de la original o sacarle en cada ocasión una copia del original, logrando que se exponga a condiciones desfavorables a los documentos.

- Mayor costo en lo referente a los expedientes.

A continuación detallaremos dichos problemas expuestos anteriormente:

- *Desorganización de la información.* En este caso se puede tener mucha información de algún predio, pero si no esta organizada como debe estar no se puede tomar decisión sobre ese predio, ya que se tendría un conjunto de papeles y no un conjunto de información
- *No existe un lugar amplio y seguro para la documentación obtenida de las dependencias y almacenamiento de las carpetas terminadas.* Por el gran número de información con que cuenta, la forma más segura de proveer una información se toma en cuenta el papel, ya que en el se puede plasmar los hechos de cada predio, por lo que se necesita un lugar grande para poder almacenar toda la información soportada en este material, además que debido a grandes volúmenes también se relacionan condiciones climatológicas los cuales provoca que se deteriore el papel y por consiguiente la información recopilada en algún predio.
- *No se cuenta con una presentación de alta calidad.* Como se cuenta con información que es recabada de varias fuentes de información, se tiene un conjunto de papeles que juntándolos todos no llegan a homogeneizarse su presentación

- debido a las diferentes presentaciones de la información que conforma una carpeta. Y debido a que esta información se requiere de un momento a otro se va perdiendo la visibilidad de la información.
- *Retraso en la integración de las carpetas*, esto se debe a que la integración se lleva de la siguiente forma:
 - Se cuenta con una cédula donde se integran los datos importantes del predio en cuestión, por lo regular se transcribe a mano y no todo el personal tiene legible su expresión escrita.
 - Los croquis que indican la ubicación del predio, es hecho por el arquitecto el cual no siempre los harán con gran calidad por el tiempo que se tiene para realizar una visita física.
 - Las fotografías que se toman de los predios, llevarán tiempo o factor económico para anexarlas a carpeta.
 - La búsqueda de la información dentro de los archiveros donde se localiza la información
 - Y después conjuntar toda la información de un mismo predio en una carpeta.

Pero esto no queda aquí, en dado caso que se requiera de la misma información para cualquier motivo y no se tenga copia o el original, se tendrá que realizar una vez más todo este proceso.

- Debido a que se cuenta con un solo original en casi todas las ocasiones, si uno le saca copia a esta carpeta cada vez se deteriorará más y en la copia se reflejará esta acción debido a que la información va desapareciendo con el transcurso del tiempo deteriorándose y de las copias que se obtienen del original.
- Se tiene que tomar en cuenta que para obtener una copia de la carpeta se tendrá que tener una fotocopidora, papel, así como la compra de rollos fotográficos, cada que se saque una copia más de la carpeta, gasolina, hacer nombre en cada proceso de la carpeta y el alquiler de una bodega cada vez más grande, personal para llevar el control del archivo, teniendo la exposición de la pérdida o cambio de información del personal.

2.2 Recopilación de la Información

Durante este proceso, las personas encargadas de recabar la información referente a los inmuebles, nos proporcionaron un reglamento para la recolección de dicha información a través de formatos que conforman la cédula del inmueble, para posteriormente capturarla y alimentar al sistema.

Dicho documento se detalla a continuación:

INSTRUCTIVO DE LLENADO DE CEDULAS

1. Consideraciones de unidades de medida.

Con la finalidad de homogeneizar los criterios para las unidades de medida que se utilizarán en el Proyecto de Regularización de Bienes del Metro, se emiten las siguientes consideraciones:

La Unidad de Básica de medición será denominada **"Predio"**, que significará el terreno o fracción adquirido de un particular u ocupado por instalaciones del Metro, propiedad del Departamento del Distrito Federal o de un particular. Otra Unidad de medida utilizada será denominada **"Inmueble"**, definida como el conjunto de predios que conforman una unidad física y que prestan un servicio de manera conjunta. **"Inspección Física"**, se referirá a la visita material de los predios que conforman un inmueble, con el fin de obtener información topográfica, planimétrica y fotográfica.

2. Cédulas de Registro.

Las Cédulas de Registro contendrán los datos relativos al régimen de propiedad de cada uno de los predios que actualmente está utilizando el Metro, ya sean

propiedad de éste, del Departamento del Distrito Federal o de algún particular. Se compondrá de los siguientes apartados:

Apartado No 1: [Contendrá el número de la Cédula,; formado por una "L" y el número o letra de la línea que corresponda, guión y el número progresivo de la cédula; ejemplo: L1-01, L1-02, ..., etc.] Fecha de captura: la que corresponda al momento de su elaboración.

Apartado No 2: Señalará la ubicación del predio y se deberán llenar los espacios con letra mayúscula, de molde y sin abreviaturas.

Apartado No 3: Indicará los datos del predio, con la cuenta catastral anterior (cuando esto sea posible) y la actual; la Superficie del Terreno expresado en metros cuadrados; la superficie construida en la misma unidad de medida; el tipo de inmueble indicándose si se trata de edificación lote baldío o mixto; el número de Niveles si los tiene; los Servicios que tiene señalándose "SI" o "NO"; nombre del inmueble, anotándose sin abreviaturas, en los casos en que se tenga la información, aclarando que se trata del nombre por el que es conocido por los vecinos del lugar; Línea a la que pertenece con letra; el uso autorizado será el destino original que se le iba a dar cuando fue adquirido el predio; uso real, se referirá a la utilización del mismo en la fecha de la elaboración de la cédula; usuario, se anotará quien lo detenta al momento del levantamiento; y en el campo de propietario, a quien se le adjudica el Dominio de mismo.

Apartado No 4: Señalará las medidas y colindancias contenidas en el documento del cual se extrajo la información.

Apartado No 5: Indicará los Antecedentes de Propiedad. En el campo de *convenio* se citará a qué tipo se refiere (Novación, Fideicomiso, etc No se debe olvidar que los convenios en estricto sentido, son los que modifican y extinguen

una relación jurídica); *fecha*, la que corresponda; *contrato*, aquel de que se trate (compraventa, donación, permuta, como dato, arrendamiento, etc); *fecha*, la que corresponda; *decreto*, indicar si es de expropiación o de cualquier otro tipo, fecha , la de su elaboración; fecha de publicación, la que corresponda en el Diario Oficial de la Federación o Gaceta del Departamento del Distrito Federal, indicándose con las siglas DOF o GDDF; Ley Orgánica del DDF, indicar los artículos que sirvan de fundamento legal, *escritura pública*, citar el número; fecha, la que corresponda al instrumento; *Notario Nombre y No*, se anotará el nombre completo y el número de la Notaría.

Apartado No 6: Contendrá la información referente a los antecedentes registrales del predio en cuestión ante la Dirección General del Registro Público de la Propiedad y de Comercio del Departamento del Distrito Federal. Se anotará si está inscrito con un "SI" o un "NO", el número de folio real y su fecha de inscripción, o en su caso la sección, volumen, tomo, fojas y número correspondiente así como la fecha que le corresponda.

Apartado No 7: Similar al anterior, pero con datos de la Dirección General del Registro Público de la Propiedad Federal, dependiente de la Secretaría de la Contraloría y Desarrollo Administrativo.

Apartado No 8: Señala cual es el soporte documental de la Cédula de Registro. Indica si se cuenta con planos, croquis, fotos o algún otro que nos permita identificar el inmueble.

Apartado No 9: Indica si el predio en cuestión ha pasado por alguna sesión del Comité del Patrimonio Inmobiliario, señalando el número de la sesión, la fecha de celebración, el responsable de presentación, de la propuesta hecha, el acuerdo al que se llegó, el tipo de permiso otorgado en caso de que se asistiera, de que tipo es y la vigencia del mismo.

Apartado No 10: El nombre del funcionario que elaboró la cédula, y su fecha de elaboración, el nombre del funcionario que da el visto bueno en la fecha correspondiente.

Apartado No 11: Se utilizará para las observaciones necesarias o para la ampliación de la información de cualesquiera de los apartados mencionados.

3. Cédulas de Inspección Física.

Para el llenado de la cédula de Inspección Física se han determinado las siguientes consideraciones:

El Apartado No 1, corresponde a la información necesaria para la identificación del predio correspondiente. Se deberán llenar los campos como a continuación se indica:

Línea: Se anotará con letra, el número de la que corresponda.

Número de Cédula: Se anotará "L" el mismo número que la correspondiente a la cédula de registro.

Correspondencia: Se anotarán los nombres de las Estaciones Terminales de dicha Línea. Ejemplo: Observatorio-Pantitlán.

Estación: Se anotará el nombre de la que corresponda sin abreviaturas.

Este apartado aparecerá en cada una de las hojas de la cédula que corresponda a cada predio y deberán ser anotados todos, vigilando que los datos sean los mismos. El número de las hojas se numerarán: 1/3, 2/3, 3/3, ..., etc.

El Apartado No 2, denominado "Domicilio", corresponde a la ubicación del predio y los campos se deberán llenar con letras mayúsculas de molde y sin abreviaturas y se deberá vigilar que los datos coincidan con los de la cédula de registro.

El Apartado No 3, identificado como "Plano de Localización", corresponde a una porción del plano obtenido de Guía Roji, en el cual se señalará la manzana en la que se encuentra el predio, indicándose el número de accesos al mismo y su orientación geográfica.

En el Apartado No 4, denominado "Tipo de Inmueble", se anotarán los datos relativos al tipo de estación, marcando "SI" o "NO", a qué clase pertenece la misma, de acuerdo con los siguientes criterios:

Edificación: Si sobre la Estación a nivel de la calle se encuentra una construcción de uno o varios niveles.

Elevada: Estación por encima del nivel de la calle, como es el caso de algunas estaciones de las diferentes líneas.

Subterránea: Estación por debajo del nivel de la calle, como es el caso de algunas estaciones de las diferentes líneas.

Superficial: Tanto Estación como accesos, están a nivel de la calle. Tal es el caso de algunas estaciones de las diferentes Líneas.

Mixta: Cuenta con las características de dos o más inmuebles.

En el Apartado No 5, denominado "Tipo de Estación", se marcará con una "X" la que corresponda; las de paso solo pertenecen a una línea; las de transbordo

a 2 ó más y las terminal aquellas en las que concluya una línea. Los casos particulares de las Estaciones Pantitlán, El Rosario, Santa Anita, Garibaldi y Martín Carrera, se manejarán como estaciones de transbordo.

En el Apartado No 6, denominado "Información del Inmueble", se indicará la superficie del terreno en metros cuadrados, superficie construida en la misma medida, número de niveles (se tomará en cuenta únicamente los niveles que sobresalgan del nivel de la superficie, esto es, de la calle) así como los servicios públicos con los que cuenta. También contendrá la información referente al uso inicial, que será aquel para el que fue adquirido originalmente y el uso actual, que se referirá a la utilización del mismo en la fecha de la visita respectiva. El nombre del inmueble se anotará sin abreviaturas, en los casos en que se tenga la información, aclarando que se trata del nombre por el que el terreno es conocido por los vecinos del lugar, se deberá vigilar que los datos coincidan con los de la Cédula de Registro.

En el Apartado No 7, se utilizará para vaciar la información referente a las medidas y colindancias, dividida en dos aspectos, las derivadas de la documentación que acredite la propiedad y las que resulten de la propia visita física o reales.

El Apartado No 8, se usará para indicar el estado físico del inmueble, debiéndose indicar con precisión, en el campo de observaciones anexo, porque se escogió la elección efectuada.

El Apartado No 9, será el espacio destinado a la fotografía digital del inmueble, debiéndose indicar en que orientación se tomó la fotografía. Estas deberán tomarse, en la medida de lo posible, siempre de frente al inmueble, y en los casos en que el mismo sea demasiado grande para ello, con una inclinación de hasta 45° respecto a la vertical de dicho inmueble, o más si es preciso. En los

casos en que se requiera más de una fotografía por inmueble, no deberá olvidarse el anotar la orientación en que se tomó cada una.

El Apartado No 10, se utilizará para las observaciones necesarias o para la ampliación de la información de cualquiera de los apartados mencionados, recordando que siempre se deberá escribir en mayúsculas.

El Apartado No 11, contendrá el croquis de localización, que será el levantamiento topográfico y planimétrico del inmueble de referencia, el cual contendrá la escala a la que se encuentra elaborado, siendo las cotas que se utilizarán en metros, salvo cuando se utilice alguna otra, la cual deberá especificarse de manera precisa. En este apartado como en el No 3, el norte siempre se localizará hacia la parte superior de la cédula.

El Apartado No 12, indicará la fecha de solicitud de visita, la fecha de realización de la misma y la fecha de confirmación, únicamente en aquellos casos en que se requiera ya sea por inconsistencias en la visita o en relación a la documentación del inmueble. También señalará el nombre de la persona responsable del levantamiento y el que otorgue el visto bueno de autorización para su incorporación al inventario computarizado del Area de Informática.

2.3 Clasificación de la Información

Una vez reunida la información necesaria para poder investigar la situación de un predio, veremos que se clasifica en información jurídica y topográfica. Ver figura 2.3.1

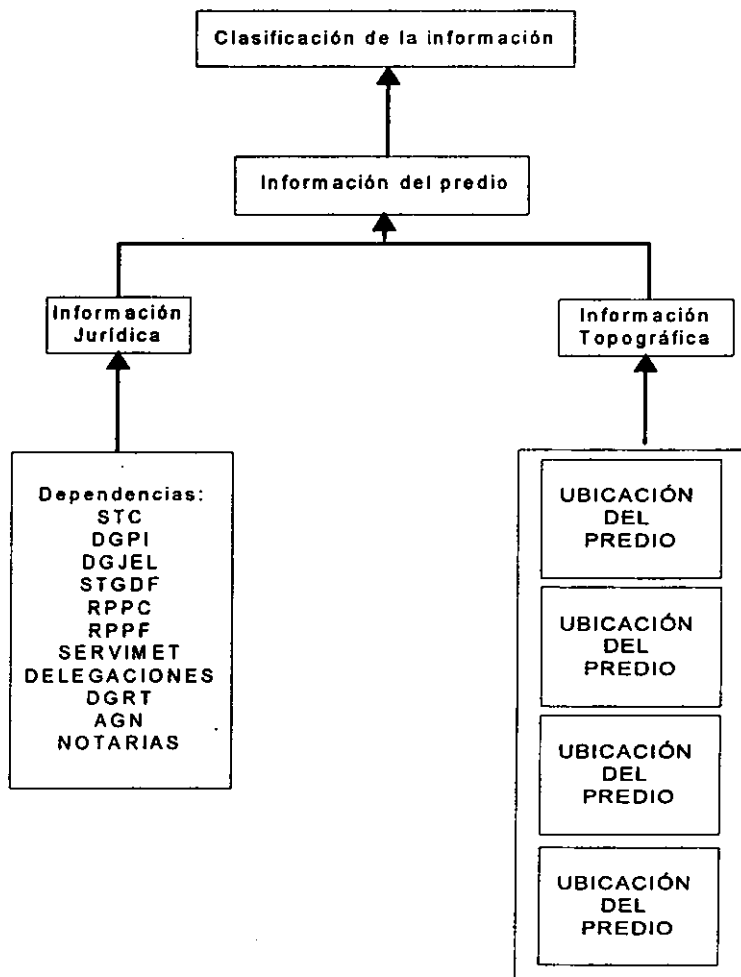


Figura 2.3.1 Clasificación de la información

Información Jurídica

La información jurídica es aquella información que proviene de documentos proporcionados por las dependencias externas a ésta dirección. La colección de escritos o documentos que acreditan la propiedad en favor del Sistema de Transporte Colectivo lleva involucrados a otras dependencias las cuales colaboran en la investigación de cada predio, el cual es la unidad básica de medición en este sistema.

Estas dependencias se describen a continuación:

1. Del Sistema de Transporte Colectivo (STC), antes COVITUR:

Dependiente de la Secretaría de Obras Públicas, cuenta con documentación referente a las ampliaciones y construcción de varias líneas que sirven como antecedente para la acreditación de los predios que ocupa el metro.

- Elabora los proyectos ejecutivos y de detalle de la construcción de las obras para la ampliación de sistema de transporte colectivo, incluyendo los correspondientes a las vialidades coincidentes y complementarias.
- Construir las obras de ampliación del Sistema de Transporte Colectivo (STC), así como sus vialidades y obras inducidas complementarias.
- Adquirir los materiales y equipos fijos para el funcionamiento del Sistema de Transporte Colectivo (STC).

- Verificar que las obras civil y electromecánica del Sistema de Transporte Colectivo (STC), se realicen de acuerdo con el programa autorizado y de conformidad con los proyectos ejecutivos y de detalle.
- Entregar las obras de ampliación del Sistema de Transporte Colectivo (STC) a las instituciones operadoras.
- Efectuar los estudios para optimizar y actualizar el proyecto y la construcción de la ampliación del Sistema de Transporte Colectivo (STC) y su vialidad.

2. De la Dirección General de Patrimonio Inmobiliario:

- Integrar el programa anual de necesidades inmobiliarias y arrendamientos de las dependencias, unidades administrativas y órganos desconcentrados del Distrito Federal.
- Llevar el registro, control y actualización del patrimonio inmobiliario del Distrito Federal, concentrando y resguardando los títulos, contratos y demás instrumentos que acrediten los derechos sobre inmuebles de su propiedad o posesión, así como proporcionar información respecto del mismo, a las autoridades competentes.
- Establecer las normas, criterios y políticas de administración, aprovechamiento y explotación de los inmuebles propiedad del Distrito Federal.
- Requerir de las dependencias, unidades administrativas, órganos desconcentrados y entidades paraestatales del Distrito Federal, la información documental de los inmuebles propiedad del mismo.

- Proponer el programa anual de administración, racionalización, uso y aprovechamiento del patrimonio inmobiliario del Distrito Federal.
- Efectuar el registro y control de los contratos de arrendamiento de bienes inmuebles que con el carácter de arrendatario celebre el Distrito Federal (D.F.).
- Controlar los patrones de concesionarios, permisionarios, usuarios y asignatarios de los inmuebles propiedad del Distrito Federal (D.F.).
- Coadyuvar en la realización de los levantamientos topográficos, así como de los trabajos técnicos necesarios para el apeo y deslinde de los inmuebles propiedad del Distrito Federal.
- Desarrollar programas de inspección física de inmuebles propiedad del Distrito Federal (D.F.).
- Realizar los trabajos de identificación y señalización de los inmuebles propiedad del Distrito Federal (D.F.), así como promover su debida custodia.
- Gestionar y promover en el ámbito de su competencia, las recuperaciones administrativas y judiciales, así como coadyuvar en los procedimientos de inmatriculaciones de inmuebles propiedad del Distrito Federal (D.F.).
- Opinar sobre el uso, aprovechamiento y destino de los inmuebles propiedad del Distrito Federal (D.F.).
- Promover ante el Registro Público de la Propiedad y de Comercio (RPPC) y, en su caso ante el Registro Público del Patrimonio Inmueble Federal (RPPF), la inscripción de los documentos en que consten actos jurídicos, en virtud de los

- cuales, se adquiriera la propiedad o derechos posesionarios de inmuebles en favor del Distrito Federal (D.F.).
- Proponer las políticas para la expedición de autorizaciones, permisos y licencias relativas a la explotación de yacimientos de arena, cantera, tepetate, piedra y arcilla en inmuebles propiedad del Distrito Federal (D.F.).
- Establecer los lineamientos para aceptar donaciones de inmuebles en favor del distrito Federal (D.F.).
- Intervenir en la entrega-recepción de los inmuebles que incidan en el patrimonio inmobiliario del Distrito Federal (D.F.).

3. De la Dirección General Jurídica y de Estudios Legislativos:

- Formular y revisar los anteproyectos de iniciativa de leyes, reglamentos, decretos, acuerdos, órdenes, circulares y demás ordenamientos jurídicos, competencia del distrito Federal (D.F.), con excepción de los de materia fiscal.
- Publicar la Gaceta Oficial del Distrito Federal (D.F.).
- Gestionar la publicación en el Diario Oficial de la Federación, de las leyes, reglamentos, decretos, acuerdos y demás disposiciones jurídicas que deban regir en el Distrito Federal (D.F.), así como publicarlos en la Gaceta Oficial del Distrito Federal (D.F.).
- Revisar y sancionar los contratos, convenios, concesiones, permisos y autorizaciones, de los que se deriven derechos y obligaciones para el Distrito Federal (D.F.), con excepción de los de materia fiscal.

- Conservar, administrar y vigilar el funcionamiento del Archivo General de Notarías.
- Supervisar la elaboración, trámite y seguimiento de las solicitudes de expropiación de bienes ejidales o comunales en favor del Distrito Federal (D.F.).

4. De Archivo General de Notarías:

Por ser la responsable directa de la custodia, conservación y expedición de documentación auténtica, esta unidad administrativa proporcionará las escrituras públicas necesarias para el acreditamiento de la propiedad.

5. De la Tesorería del Departamento del Distrito Federal:

Se le solicitan las cuentas catastrales de los predios, así como los planos para su ubicación y los antecedentes documentales en el archivo de esa unidad Administrativa.

- Realizar los estudios de mercado inmobiliario, a efecto de determinar los valores catastrales, así como de aquéllos orientados a definir y establecer la política tributaria, para determinar tasas y tarifas impositivas en materia de contribuciones relacionadas con bienes inmuebles.
- Representar los intereses de las dependencias, unidades administrativas y órganos desconcentrados, en la compensación de adeudos con dependencias y entidades del Gobierno Federal.

6. Dirección General del Registro Público de la Propiedad y de Comercio (RPPC):

Los antecedentes registrales que nos proporciona, constituye uno de los documentos con los que se puede acreditar la propiedad de un predio.

7. Servimet, S.A. de C.V. (Servicios Metropolitanos):

Originalmente agente inmobiliario del Gobierno del Distrito Federal (D.F.), conserva en sus archivos parte sustancial de la documentación necesaria para acreditar la propiedad de los predios utilizados en la primera etapa de creación del Metro.

8. Dirección General Jurídica y Estudios Legislativos:

Esta unidad Administrativa es la encargada de la revisión y aprobación jurídica de los documentos con los que se acredita la adquisición o enajenación de los diversos predios que conforman el Sistema, por lo que en su archivo se encuentran documentos e indicios muy importantes para el éxito del Proyecto.

9. Delegaciones Políticas en el Distrito Federal (D.F.):

Muchas de las adquisiciones realizadas para la primera etapa del metro, se realizaron mediante las Delegaciones Políticas, por lo que sus archivos contienen muchos de los documentos necesarios para el acreditamiento de la propiedad, o al menos cuentan con indicios para su obtención.

10. Secretaría de Transporte y Vialidad:

Derivado del inicio heterogéneo en la adquisición de los inmuebles y a la

dispersión de la información, esta Unidad Administrativa también cuenta con importante documentación que sirve de indicio para la obtención de la documentación necesaria para el acreditamiento de propiedad de los inmuebles del metro.

11. Dirección General de Servicios Legales:

Muchos inmuebles, obtenidos de la ejecución de actos de autoridad como expropiaciones, se encuentran en litigio por parte de sus antiguos propietarios, la unidad Administrativa que conserva la documentación necesaria para determinar la situación jurídica de tales predios en esta área.

12. Dirección General de Regularización Territorial:

Algunos predios, propiedad del Distrito Federal (D.F.), remanentes de procesos de regularización de la tenencia de la tierra, se utilizaron en obras del metro, por lo que se constituye también en una opción para el rescate de información.

13. Archivo General de la Nación: Sobre algunos inmuebles, la propiedad del Distrito Federal (D.F.) se acredita mediante documentación muy antigua, cuya custodia la tiene esta Unidad Administrativa del Gobierno Federal.

De cada una de estas dependencias se obtiene lo que se refiere a escritos del predio en investigación, no necesariamente nos dan todas estas dependencias información de un solo predio. Esta documentación se clasifica de la siguiente manera:

- Planos.
- Convenios
- Contratos.
- Acuerdos.
- Escrituras
- Decretos.
- Antecedente registral.
- Documentos administrativos.
- Otros.

Ahora se explicarán cada uno de estos documentos:

- Planos.- Dibujo que representa las diferentes partes de un predio como son medidas colindancias del predio en cuestión para una mejor ubicación.

Estos planos son proporcionados principalmente por la secretaría de Finanzas, los cuales cuentan con planos de la Cd. de México. La forma en que se clasifican los planos en esta dependencia es la siguiente, cuentan con un número el cual representa de la siguiente manera: rrr-mmm-III, la letra r significa región, en la cual está dividida la Cd. de México, la m es la manzana y esta se refiere al conjunto de predios dentro de un límite sin división de vía pública, y la I es el lote, el cual es un predio individual y tiene colindancias con otro predio.

Pero no solamente esta dependencia nos proporciona planos, en algunas ocasiones las otras dependencias cuentan con planos de los cuales nos proporcionan una copia del mismo.

- Convenios.- Son los que modifican y extinguen una relación jurídica.

En este tipo de documentación se citará al tipo de convenio que se refiere, ya sea novación, fideicomiso, etc...Es decir sustitución de una obligación a otra anterior quedando ésta anulada, donación testamentaria hecha a una persona encargada de restituirla a otra o ejecutar alguna voluntad del testador, todo esto referente a predios, etc...

- Contratos.- Pacto entre 2 o más personas físicas o morales, el cual acredita la propiedad de un predio.

Los contratos pueden ser de compra-ventas, donación, permuta, comodato, arrendamiento, etc., En ellos se acredita la propiedad de un predio a una persona cediéndole derechos y obligaciones sobre este.

Los contratos los clasificaremos como antecedentes de propiedad y se expide ante un notario, y este a su vez al Archivo General de Notarias.

- Acuerdos.- Resolución tomada por una o varias personas de un predio en cuestión. Considerando que es en estricto sentido de un predio en particular.

En estos documentos se toma una decisión sobre algún predio, esto elaborándolo un notario a la persona interesada, esto lo proporciona tanto el archivo General de la Notarias así como puede proporcionarlo a otras dependencias cediendo una copia de este documento.

- Escrituras.- Instrumento público en el cual da fe el notario para asegurar un contrato mediante el cual se acredita la propiedad de un predio.

En las escrituras se da la acreditación de que la propiedad *pertenece a*", este tipo de documentación lo proporciona la dependencia a la cual esta acreditado

el predio.

- Decretos.- Disposición de carácter legislativo promulgada por el poder ejecutivo sobre predios que tienen en custodia una dependencia gubernativa.

En este rubro se indica si es de expropiación o de cualquier otro tipo, y estos datos los proporciona el D.O.F. (Diario Oficial de la Federación) ó Gaceta del Departamento del Distrito Federal. Estos documentos se toman como antecedentes de la propiedad.

- Antecedentes Registral. Contiene la información referente a los antecedentes del predio en cuestión ante la Dirección General del Registro Público de la Propiedad y de Comercio del Distrito Federal (D.F.).

En estos antecedentes se da la información de la ubicación de los documentos en el Registro Público del Patrimonio Inmueble Federal (RPPF) como son: Folio real, sección, Volumen, Tomo, Fojas, Números de los documentos que acredita la propiedad ante el Registro Público del Patrimonio Inmueble Federal (RPPF).

- Documentos Administrativos.- En estos documentos entran los datos de la Dirección General del Registro Público del Patrimonio Inmueble Federal (RPPF) que dependen de la Secretaría de Contraloría y Desarrollo

Administrativo, y también entra aquí lo que se refiere a la cesión sobre algún predio en el Comité del Patrimonio Inmobiliario, en el cual se da derechos de algún predio propiedad del Distrito Federal (D.F.) a alguna persona o institución.

Esta recopilación se da con el apoyo de las dependencias antes mencionadas y tomando lo esencial de cada uno de los documentos proporcionados por estas, así como una copia de los documentos.

- Otros: Esto se refiere a los documentos que provienen de otras dependencias que no son tan comunes o inusuales.

Información Topográfica

Esta información es proporcionada por el área de topografía, la cual se encarga de localizar físicamente al predio en cuestión y tomar los datos reales del predio, los datos que se toman son los siguientes:

- ubicación del predio,
- medidas y colindancias,
- características del predio,
- fotografías y croquis de localización.

Ubicación del predio

En esta parte de la investigación topográfica se ubica el predio para definir en que tipo de predio se clasificará, ya sea edificación, subterránea, superficial, así como que tipo de estación es de acuerdo si es de paso, de transbordo o terminal

Medidas y colindancias

En este rubro se vaciará la información referente a las medidas reales del predio así como sus colindancias, es decir datos sobre los predios contiguos a este, además la superficie del predio.

Características del predio

En este apartado se expondrá la situación física del predio, así mismo si cuenta con servicios públicos como son agua, luz y drenaje,

Fotografías y croquis de localización

En lo referente a la ubicación gráfica del predio se tendrá que hacer por parte del área topográfica un croquis de ubicación mostrando toda la información que describió en la ubicación del predio, medidas y colindancias , para poder corroborar dicha información y además de sacar fotografías digitales para agregar al expediente dentro de la base de datos.

2.4. Requerimientos de usuario.

Son características que se deben incluir en un nuevo sistema, esto puede ser la inclusión de determinada forma para capturar o procesar datos, producir información, controlar una actividad o brindar soporte. El resultado que se obtenga, ya sea eficiente o deficiente dependerá en gran medida de la calidad que se recopile y procese la información. Primero debe conocerse las funciones y sus necesidades en el manejo de información, para dar pie al diseño del proyecto.

Las funciones generales en la administración del patrimonio inmobiliario del Sistema de Transporte Colectivo son:

- Identificar decretos de expropiación que hacen referencia al Sistema de Transporte Colectivo.
- Identificar expedientes de la Dirección General del Patrimonio Inmobiliario que contengan información al respecto.
- Revisión de documentación existente que acredite el predio.
- Solicitud de cuentas catastrales: revisión en base de datos y de planos catastrales de la Ciudad de México, de contenido predial.
- Envío de solicitudes a las Dependencias para solventar información faltante.
- Codificación de pre-cédulas de registro.
- Requisición de verificación física (visita física) del predio para levantamiento topográfico.
- Revisión de planos de afectación en complemento del levantamiento topográfico.
- Conformación de información jurídica y topográfica para formación de carpeta.

Consideraciones que deben conformar el registro de información:

- Debe manejar una estructura que relacione información pertinente al domicilio, datos del inmueble y documentación contenida en el expediente.
- Debe manejar una estructura que contenga antecedentes de propiedad, Registro Público de la Propiedad y del Comercio, Registro Público de la Propiedad Federal, soporte documental, y antecedentes del Comité del Patrimonio Inmobiliario.
- Debe conformar una estructura de planos de localización, tipo de inmueble, tipo de estación, información del inmueble, medidas y colindancias documentales y reales, croquis de localización, estado físico, fotografía del inmueble, y observaciones generales.
- En cada estructura debe contener información que lo vincule.

Consideraciones que deben conformar el procesamiento de información:

- Debe tener una interfaz amigable en pantallas (manejo y entendimiento fácil).
- Debe conformarse de una serie de utilidades para obtener reportes y consultas.
- Debe de ser lo más automatizado, evitando confusión y mal manejo.
- Debe contener seguridad en el acceso del sistema; restringir a la información a ciertas personas.

2.5 Opciones de Solución

La Dirección General del Patrimonio Inmobiliario, Sistema de Transporte Colectivo Metro, tiene por objetivo diseñar un sistema que sea capaz de estandarizar y delimitar las colindancias de los predios, en las diferentes Delegaciones del Distrito Federal y Zonas conurbadas, que forman parte del patrimonio inmobiliario del Sistema de Transporte Colectivo (STC) Metro.

A continuación se presentaran tres posibles soluciones:

Solución No. 1

- 1.- La DGPI iniciara su proceso recopilando la información necesaria del predio para después clasificarla.
- 2.- La información será transferida al Departamento Jurídico en donde se realizará un análisis muy detallado.
- 3.- En caso de que la información recopilada no sea suficiente se procederá a realizar un reconocimiento físico del predio y se acudirá a diferentes dependencias para reunir la información necesaria, una vez terminado este proceso se remitirá toda la información nuevamente al Departamento Jurídico.
- 4.- Si la información del predio es suficiente, entonces se procederá a la captura de la información.
- 5.- Se tomaran fotografías de los predios y también se realizarán croquis, planos y copias de los documentos.

6.- Una vez que se haya recopilado toda la información del predio se procede a presentar la carpeta completa del predio que se investigo.

7.- Cuando se haya terminado la investigación se procede a archivarla hasta que esta sea requerida para la obtención de algún dato sobre el predio en cuestión. Cuando sea este el caso se presentara solo la información solicitada sin necesidad de tener todos los documentos presentes.

8.- La información será almacenada en archiveros, ocupando demasiado espacio en el archivo de la DGPI. Esto representara el fin del proceso.

Las principales desventajas de esta situación es que se tiene un control manual sobre el procesamiento de captura, fotografías, croquis y copias realizadas a los documentos del predio, el espacio que ocupa el almacenamiento de las carpetas abarca gran espacio en el archivo y la presentación de cada carpeta se va deteriorando cada vez que se ocupa la carpeta, no perdiendo de vista que al ser tantos documento estos pueden ser extraviados o intercalados en folders que contengan otro tipo de información.

Solución No. 2

1.- La DGPI iniciara su proceso recopilando la información necesaria del predio para después clasificarla.

2.- La información será transferida al Departamento Jurídico en donde se realizará un análisis muy detallado.

3.- En caso de que la información recopilada no sea suficiente se procederá a realizar un reconocimiento físico del predio y se acudirá a diferentes

dependencias para reunir la información necesaria, una vez terminado este proceso se remitirá toda la información nuevamente al Departamento Jurídico.

4.- Si la información del predio es suficiente será capturada o en su defecto será digitalizada para poder introducirla a una base de datos, la cual estará desarrollada en Microsoft Access 2.0, utilizando como Front-End las mismas herramientas que posee esta base de datos relacional.

5.- Los operadores se podrán ayudar con herramientas adicionales para el mejor desempeño y presentación de los trabajos, estas herramientas pueden ser: AutoCad V.13, Photo Styler, Photo-End, Microsoft Power Point 4.0, algunas rutinas de Microsoft Visual Basic e información recopilada por la Guía Roji.

En esta base de datos podremos encontrar toda la información necesaria, como son: datos de predio (colindancias, dueños, ubicación, etc), croquis, fotografías y planos del predio totalmente digitalizados para que finalmente se presente completa "la carpeta" relacionada al predio que se investigo con el fin de poder hacer un análisis total de esta.

La información requerida de uno o mas predios podrá ser proporcionada en el instante en que se solicita, de forma podremos obtener todos los documentos o el documento que se requiera sin necesidad de tener toda la carpeta con la información recopilada.

La información será almacenada en discos duros o CD-ROM, con la facilidad de que no ocupe demasiado espacio como en el archivero de la DGPI, esto representará el fin del proceso.

Las principales ventajas de esta propuesta es que la captura se realiza en cada PC, de acuerdo al número de capturistas con que se cuente.

La digitalización de los documentos se realiza y relaciona a la cédula de la Base de Datos, los croquis y planos se anexan a esta misma.

Se pueden imprimir los registros las veces que se requieran, sin necesidad de que se deteriore la información.

La consulta puede hacerse de cualquier manera y se agiliza su consulta sin necesidad de ir físicamente a las carpetas por lo consiguiente es menos esfuerzo físico.

Su almacenamiento es en un espacio pequeño y es fácil de acceder, además de que las presentaciones son de gran calidad y se puede acceder a ellas las veces que uno desee.

Esta solución presenta una desventaja, la información que es dividida para su captura se tiene que estar transportando en disquete debido a que tenemos varias PC's con sus respectivos operadores en donde cada uno desarrolla y almacena la información en el Disco Duro de la PC en donde esta trabajando, esto genera una descentralización de la información, por ejemplo: el operador de la pc1 esta almacenando la información del predio x, mientras que el operador de la pc2 y pc3 esta capturando la información del predio w y z respectivamente ¿Que pasaría si el operador de la pc3 tiene que consultar información de la máquina del operador de la pc1? El operador de la pc3 tendrá que llevar su disquete y copiar la información que necesite a este, lo cual representa una gran desventaja para esta solución ya que esto representa perdida de tiempo y de acceso a la información cuando es requerida.

Solución No. 3

- 1.- La DGPI iniciara su proceso recopilando la información necesaria del predio para después clasificarla.
- 2.- La información será transferida al Departamento Jurídico en donde se realizará un análisis muy detallado.
- 3.- En caso de que la información recopilada no sea suficiente se procederá a realizar un reconocimiento físico del predio y se acudirá a diferentes dependencias para reunir la información necesaria, una vez terminado este proceso se remitirá toda la información nuevamente al Departamento Jurídico.
- 4.- Si la información es totalmente recopilada se procede a la captura y digitalización de esta misma.
- 5.- Si la información del predio es suficiente será capturada o en su defecto será digitalizada para poder introducirla a una base de datos, la cual estará desarrollada en Microsoft Access 2.0, utilizando como Front-End las mismas herramientas que posee esta base de datos relacional.
- 6.- Los operadores se podrán ayudar con herramientas adicionales para el mejor desempeño y presentación de los trabajos, estas herramientas pueden ser: AutoCad V.13, Photo Styler, Photo-End, Microsoft Power Point 4.0, algunas rutinas de Microsoft Visual Basic e información recopilada por la Guía Roji.
- 7.- Para poder realizar las actividades del punto 4 se necesitaran por lo menos 3 capturistas, esto es por el volumen de información que se maneja en cada predio.

8.- Estos podrán manipular y distribuir la información a través de una Red LAN, es decir mientras el capturista 1 digitaliza los planos, el capturista 2 digitaliza las fotografías y el capturista 3 digitaliza la información, la unión de toda la información podrá hacerse a través de la red sin necesidad de que el 1, 2 y 3 estén llevando la información de un lado para otro en disquetes corriendo el riesgo de que estos se contaminen o simplemente de que este medio de almacenamiento sea insuficiente para compartir o unir la información del predio en investigación.

9.- Para poder llevar a cabo todas las actividades del punto 6 se requieren al menos tres adaptadores de red (depende del número de capturistas con que se cuente) instaladas en Pc's 80486 o superior con un mínimo de 12 en RAM por la magnitud del sistema, monitores VGA, un servidor o una pc con características superiores a las antes mencionadas ya que en este se estará almacenando toda la información capturada en el sistema.

10.- Una vez que se haya terminado todo el proceso de la información relacionada al predio en investigación, será presentada al Director General del STC Metro.

11.- La información requerida podrá ser consultada sin necesidad de tener todos los documentos físicamente.

12.- El almacenamiento de la información podrá ser en Discos Duros o en CD-ROM del servidor o pc que se este utilizando para almacenar la información, con la facilidad de que no ocupa demasiado espacio como en el archivo de la DGPI.

Las principales ventajas de esta propuesta es que la captura se realiza en cada Estación de Trabajo.

La digitalización de documentos se realiza y relaciona a la cédula de la Base de Datos. Sin perder de vista que los planos y croquis serán agregados a la Base de Datos.

Las fotografías podrán ser accesadas a través de la base de datos.

Se imprime el registro las veces que se requiera, sin necesidad de que se deteriore la información.

La consulta puede hacerse de cualquier manera y se agiliza su consulta sin necesidad de ir físicamente a las carpetas por lo consiguiente menos esfuerzo físico.

Su almacenamiento es en un espacio pequeño y es fácil de acceder. El usuario puede hacer las presentaciones tantas veces el lo desee con una mejor calidad.

2.6 Requerimientos Mínimos Necesarios Para Arrancar La Aplicación

A continuación plantearemos la configuración de hardware y software mínimos para ejecutar nuestra aplicación.

Empezando por las características del hardware :

CPU, este debe incluir un procesador Pentium de 133 mhz o superior, ya que en la actualidad es la base mínima que se maneja en el mercado, 16 MB de memoria RAM que son suficientes para poder correr la plataforma de Windows 95, pero que seguramente tendrá que ser incrementada con nuevas versiones de esta plataforma, así también la capacidad del disco duro deberá de ser aproximadamente entre 1.7 y 2.0 GB ya que además de ser las capacidades mínimas en el mercado son las necesarias para almacenar la información que se pretende hasta el momento, además se considera que la información será almacenada en CD-ROM una vez que haya concluido el proceso de depuración, una unidad de CD-ROM de al menos 12x que nos servirá para acceder información ya sea de programas de utilerías (Guía-Roji), información de consultas de nuestra base de datos o programas en general.

Monitor SuperVga a color con resolución mínima de 800 x 600 dpi, con el objeto de que se puedan apreciar con buena calidad todas las imágenes del sistema.

Mouse PS/2 o serial compatible, debido a que el sistema está desarrollado para operar bajo ambiente Windows es esencial la presencia de éste dispositivo.

Unidad de disco removible "JAZZ" de 1 GB (paralelo), se requiere para ir realizando respaldos periódicos mientras no se tenga recopilada toda la información y se haya grabado en CD-ROM.

Unidad CD-WRITER, se requiere para grabar toda la información una vez que esté totalmente aprobada, ya que no será susceptible de modificaciones.

Una cámara digital, se requiere para convertir las fotografías tomadas en una imagen digital y poder integrarlas a la base de datos de nuestro sistema.

Scanner de cama plana tamaño oficio, se requiere para capturar los documentos formales (escrituras, convenios, acuerdos, etc.) y tenerlos como imagen digital y también poder integrarlos a la base de datos de nuestro sistema.

Impresora láser, que nos servirá para imprimir los reportes generados por el sistema.

UPS , nos servirá para evitar pérdida de información en el momento en que se está capturando la misma.

En lo que toca al software necesario, se requiere lo siguiente :

Windows 95, es la plataforma bajo la que correrá nuestro sistema.

Access, es el manejador de base de datos requerido por el usuario para desarrollar el sistema.

Visual Basic, se requiere para afinar algunas rutinas incluidas en los módulos desarrollados en Access.

Auto CAD, se requiere para elaborar los croquis de colindancias y dimensiones de los predios.

Guía-Roji, se requiere para extraer los planos de la localización de los predios de manera fidedigna y gráfica.

PhotoStyler, Photo Enhanced y PowerPoint, se requieren para retocar las imágenes de las fotografías y planos.

A manera de resumen se listan todas las características antes mencionadas:

HARDWARE

- Computadora Personal compatible
 - Procesador Pentium a 100 MHz o superior:
 - 16 MB memoria RAM.
 - 1.7 GBytes Disco Duro. (20 MB Instalación)
 - Unidad de Disco Flexible 3½" 1.44 MB
 - CD-ROM 12x
- Monitor. Super VGA color 800 x 600 dpi
- Mouse. Compatible MicroSoft
- Teclado Español
- Unidad de disco removible "JAZZ" de 1 Gbyte (paralelo)
- Unidad CD-WRITER (grabador de CD-ROM)
- Cámara digital
- Scanner de cama plana tamaño oficio
- Impresora Laser
- UPS de 600 VA

SOTFWARE

- Windows 95
- Access
- Visual Basic

- Auto CAD r13
- Guía Roji
- PhotoStyler, PhotoEnhancer
- PowerPoint

CAPITULO 3
DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

3. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

3.1 Acotamiento Del Problema

Para poder acotar el problema es importante definir que el objetivo es diseñar y construir un sistema que sea capaz de estandarizar y delimitar las colindancias de los predios, en las diferentes delegaciones del Distrito Federal y sus zonas conurbadas, que forman parte del patrimonio inmobiliario del Sistema de Transporte Colectivo (STC) METRO.

El sistema será desarrollado a petición expresa en ACCESS utilizando como front-end las mismas herramientas que posee esta base de datos relacional. Además se apoyara de herramientas adicionales como: AutoCad r13, PhotoStyler, PhotoEnd, Power Point, algunas rutinas de Visual Basic e información recopilada de Guía Roji, licencias con las que cuenta la dependencia.

El sistema recabara toda la información que posea en papel de los predios que serán o fueron afectados por la construcción del METRO, después se hará un análisis de la información por parte del departamento jurídico, el cual determinara si esta es suficiente o bien habrá que investigarse a quien corresponde la propiedad; en caso de ser insuficiente la información, se procederá a complementarla hasta que satisfaga los requisitos del departamento jurídico.

Una vez cubiertos los pasos anteriores se procederá a la captura de la información, la cual consistirá en fotografías, croquis, planos y copias de los documentos que deberán ser compiladas en un carpeta que posea toda la información, ya que una vez autorizada, deberá ser digitalizada y capturada en la base de datos ACCESS para posteriormente ser explotada por el personal que así lo requiera.

La información será almacenada inicialmente en discos duros de gran capacidad para poder ser consultada en línea en todo momento. Posteriormente, ya una vez eliminados todos los posibles "bugs" que pudiera contener, se procederá a la grabación en CD ROM, previa autorización de un comité de evaluación de esta.

El sistema deberá poseer el concepto amigable, es decir, que cualquier usuario con poca experiencia en programación pueda adentrarse rápidamente en el manejo del sistema a través del uso de menús gráficos que serán guiados por medio de una línea de navegación que estará en la parte inferior de la pantalla. Además, deberá contar con una barra de herramientas que nos permita acceder a los interiores, ya sea por medio del icono correspondiente, o bien de la letra subrayada de estos.

Se pretende que el sistema funcione en un ambiente de red local donde se tengan privilegios para que los diferentes usuarios puedan tener acceso a la información.

3.2 Diseño Y Construcción De La Base De Datos

Los registros de una base de datos están situados generalmente en el archivo por el orden en que han sido introducidos, aunque podemos cambiarlos y ordenarlos como precisemos, por el campo que deseemos: alfabéticamente, numéricamente, etc.

El contenido de una base de datos se obtiene combinando datos de todas las diferentes fuentes en una organización de tal manera que los datos estén disponibles para todos los usuarios y los datos redundantes puedan eliminarse o al menos minimizarse. El usuario podrá recobrar datos de cualquier parte de la base de datos ya que los archivos ahí almacenados están conectados directa o indirectamente.

El diseño de una base de datos se hace a partir del diagrama E-R e involucra los tres procesos siguientes :

1. Agrupamiento de los datos globales de una base de datos en uno o más registros de tipo conceptual (entidades).
2. Definición de las relaciones entre las entidades del modelo conceptual de la base de datos.
3. Transformación del modelo conceptual de la base de datos en un esquema mejorado que se puede describir mediante un DBMS específico para la implantación de la base de datos.

Estos puntos se verán aplicados más adelante en el apartado de Normalización de la base de datos.

3.2.1 Diagrama De Flujo De Datos

La notación de los diagramas de flujo de datos se deriva del trabajo de Constantine y Yourdon (1979), Yourdon (1975) y Myers (1975). Estos diagramas documentan cómo los datos de entrada se transforman en datos de salida, donde cada etapa del diagrama representa una transformación diferente.

Los diagramas de flujo de datos constan de tres componentes:

1. Flechas de anotaciones
2. Burbujas con anotaciones
3. Los operadores "AND" y "OR"
4. Los rombos de decisión

Las burbujas con anotaciones representan centros de transformación en los que la anotación especifica la transformación. Las flechas representan el flujo de datos hacia adentro y afuera de los centros de transformación, donde las anotaciones dan nombre al flujo de datos. Los diagramas de flujo de datos describen cómo una entrada se transforma en salida. No debe incluir la información de control o sucesión de la información.

Cada burbuja se puede considerar una caja negra independiente que transforma sus entradas en salidas. Los operadores "AND" y "OR" se utilizan para unir flechas, "AND" significa AND ("y" lógico) y "OR" significa EXCLUSIVE OR ("o" exclusivo lógico).

Los diagramas de flujo de datos o burbujas son gráficas dirigidas en donde los nodos especifican las actividades de proceso y los arcos de transferencia de datos entre nodos de proceso. Como cualquier otro diagrama de flujo, estos pueden ser utilizados en cualquier nivel de abstracción. Un diagrama de datos puede representar al flujo de datos entre estatutos individuales o entre bloques de

estatutos dentro de una rutina, flujo de datos entre rutinas secuenciales, flujo de datos entre procesos concurrentes o flujo de datos entre sistemas de cómputo distribuidos, donde cada nodo representa una unidad de proceso geográficamente separada. Distinto a otros diagramas de flujo, las burbujas no indican la lógica de decisión o las condiciones bajo las cuales varios nodos de proceso se activen.

Los diagramas de datos pueden expresarse utilizando una notación informal, como se ilustra en la figura, o por medio de símbolos especiales para denotar a los nodos de proceso, a los nodos de entrada, a los nodos de almacenamiento y a los nodos de salida.

Las burbujas son un mecanismo excelente para la comunicación con el cliente durante el análisis de requerimientos, así mismo, también son utilizados para representar las especificaciones del diseño externo y del diseño interno en un alto nivel, en este último caso, las burbujas son ciertamente valiosas para el establecimiento de las diversas convenciones de nombres para el sistema, los subsistemas, los archivos y las ligas de datos.

Los rombos determinan el flujo del diagrama, es decir, definen a través de una decisión el flujo que se debe de seguir en el resto del diagrama, son sumamente útiles, para indicar que en base a una decisión se puede realizar o dejar de realizar algún proceso o burbuja.

En un diagrama de flujo, la función u objetivo es permitir observar para cualquier persona los diferentes procesos o acciones que fluyen en una organización o área, sus interrelaciones y los niveles o tipos de decisión que debe acompañar al proceso.

Comúnmente son utilizados en las áreas de métodos y procedimientos , para definir el flujograma de alguna área de la empresa.

El diagrama de flujo de datos que podemos observar en la figura 3.2.1a nos da una visión general del flujo de datos que se pretende obtener, está en un nivel 0. La descripción de cada proceso nos lleva a la elaboración de diagramas de flujo por niveles, es decir, conforme se avanza en el número de nivel, se tendrá más detalle tanto en procesos como en el flujo de datos. Así, tenemos que las figuras 3.2.1 b y 3.2.1 c corresponden a los niveles 1 y 2 .

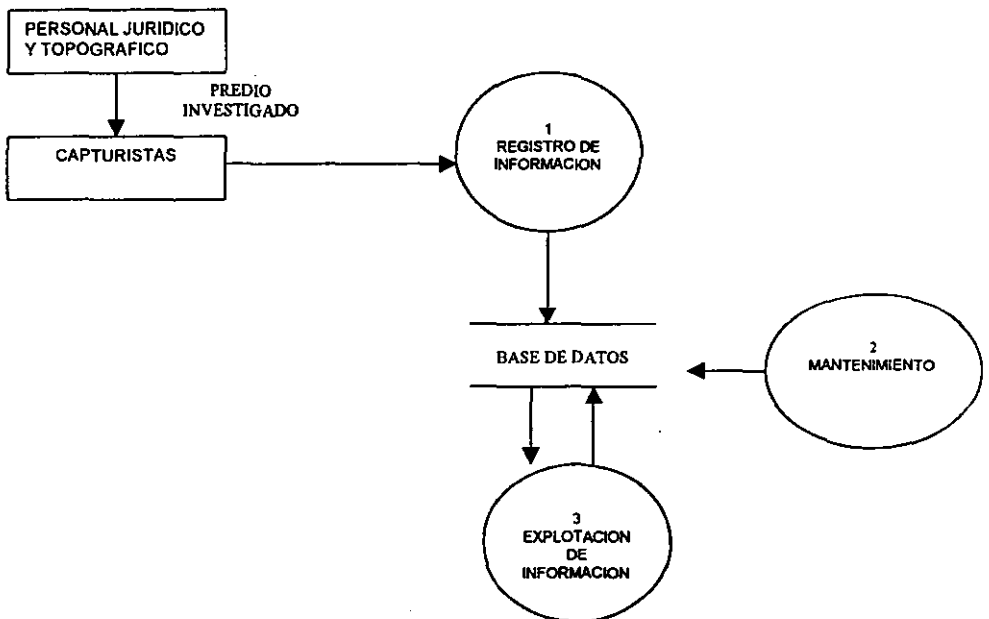


Figura 3.2.1 a Diagrama de Flujo de Datos de Nivel 0

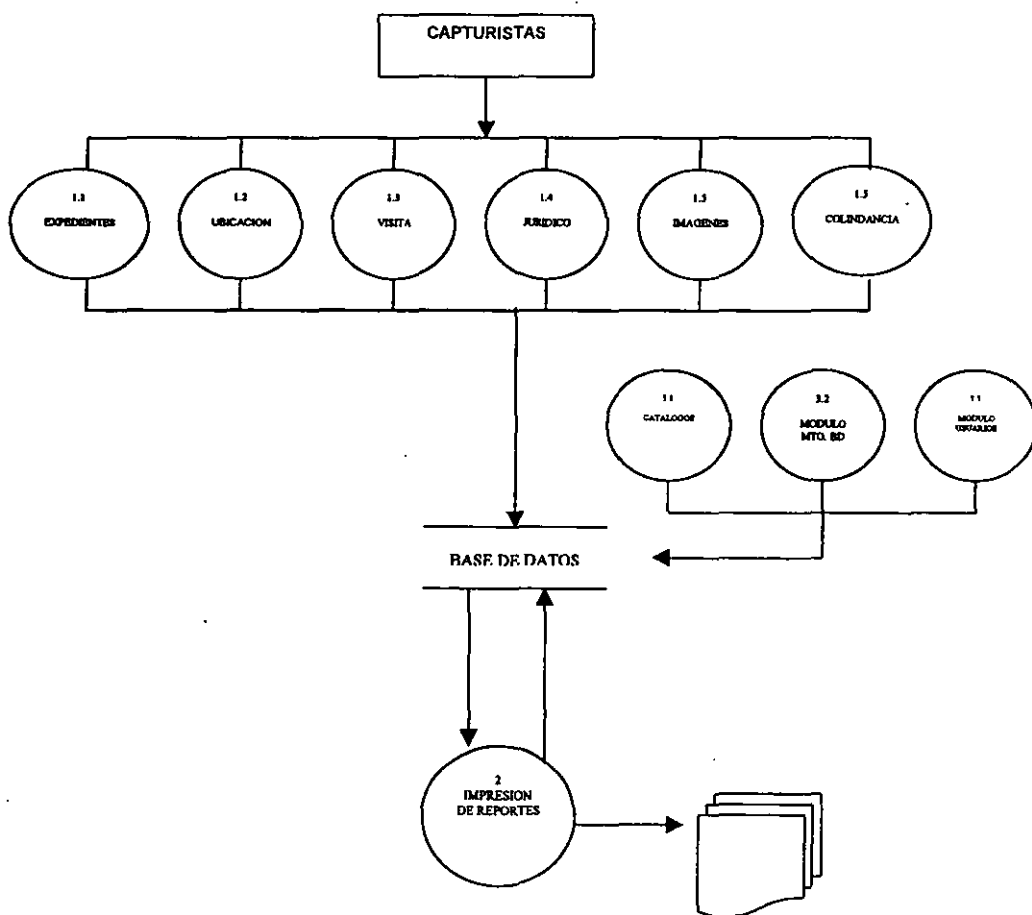


Figura 3.2.1 b Diagrama de Flujo de Datos de Nivel 1

En el diagrama de Nivel 2, todos los módulos de Registro tienen la misma funcionalidad, por lo que solo se desglosa el módulo de Expedientes.

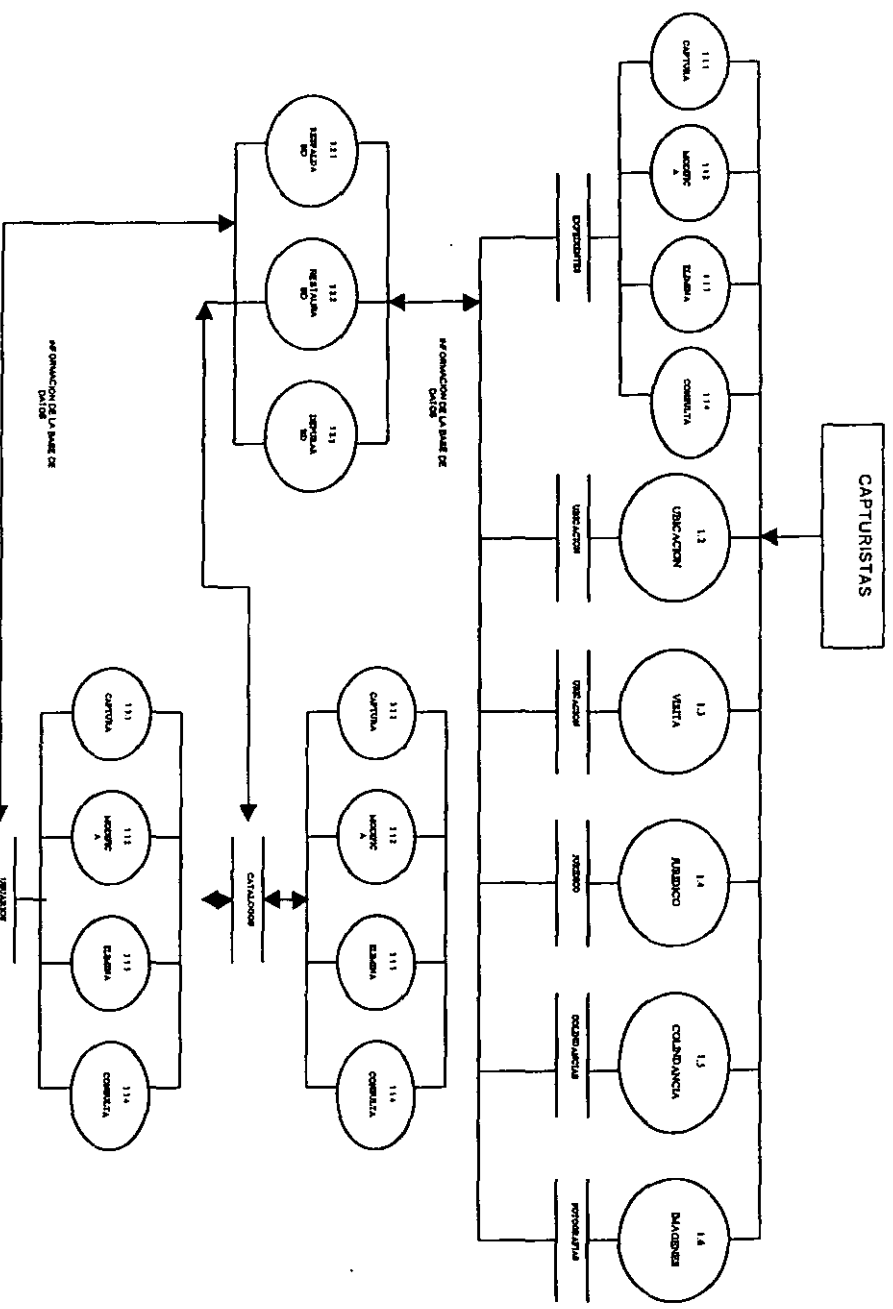


Figura 3.2.1 c Diagrama de Flujo de Datos de Nivel 2

3.2.2 Normalización de la Información

Para llevar a cabo la normalización de la información se tomaron en cuenta algunos puntos para nuestro diseño en particular, que se describen a continuación.

Se advierte que existe gran cantidad de información identificada por cada Cédula. Debido a esto se tendrá una entidad principal que posea demasiados atributos, o tal vez se opte porque haya algunas entidades cuya relación sea de uno a uno aunque se pretenderá que estas sean las menos posibles.

Además se debe tener en cuenta para la normalización, el tamaño de algunos atributos de las entidades donde se requerirá almacenar imágenes.

Se identifican las siguientes probables entidades, sus nombres por convención iniciarán con "s_" y "sc_" para catálogos a manera de identificar el nombre del sistema :

Entidad principal

- **s_cedula**. Contendrá toda la información referente a la cédula, desde la información teórica que se tenga sobre los predios hasta la que tenga que ver con las visitas físicas efectuadas a estos, incluyendo imágenes.
- **s_jurídico**. Toda la información de carácter jurídico
- **s_docdigi**. Información sobre los documentos que se deben digitalizar para cada predio.

Catálogos

- **sc_tipo_inmueble**. Catálogo para tipos de inmueble.
- **sc_tipo_estacion**. Catálogo de tipos de estaciones.
- **sc_estaciones** Catálogo de estaciones.
- **sc_lineas**. Catálogo de Líneas

- **sc_estado_fisico**. Catálogo de estados físicos
- **sc_correspondencia**. Catálogo de correspondencia.
- **sc_delegaciones** Catálogo de delegaciones.

Al llevar a cabo este diseño se observó que la entidad Cédulas presentaba varias redundancias, pues quedaba de la siguiente manera :

s_cedula

Contador	cedula	linea	fecha	consecutivo	Correspondencia	estacion	calle
----------	--------	-------	-------	-------------	-----------------	----------	-------

Numero	no_interior	colonia	entre_calles	cp	Delegacion	cuenta_ant	cuenta_act
--------	-------------	---------	--------------	----	------------	------------	------------

fecha_cap	superficie	superficie_cons	tipo_inmueble	tipo_est	niveles	servicios	
-----------	------------	-----------------	---------------	----------	---------	-----------	--

nom_inmueble	uso_inic	uso_actual	usuario	propiet	norte_r	colind_1_r	
--------------	----------	------------	---------	---------	---------	------------	--

superf_1_r	...superf_6_r	fecha_conf	fecha_perf	fecha_conclus	fecha_realiz		
------------	---------------	------------	------------	---------------	--------------	--	--

elaboro_visita_fisic	fecha_elabora	VoBo	Fecha_Vobo	observac	agua	luz	
----------------------	---------------	------	------------	----------	------	-----	--

drenaje	estado_fisico	obser_edo_fisico	contratos1...10		fotografias1...10		
---------	---------------	------------------	-----------------	--	-------------------	--	--

escrituras1...10	convenios1...10	decretos1...10	documentos1...10	acuerdo1...10			
------------------	-----------------	----------------	------------------	---------------	--	--	--

s_juridico

cedula	fecha_convenio	contrato	fecha_contrato	convenio	decreto	fecha_decreto	
--------	----------------	----------	----------------	----------	---------	---------------	--

fecha_publicacion	ley_organica	escritura_publica	fecha_escritura	notario_nombre			
-------------------	--------------	-------------------	-----------------	----------------	--	--	--

no_notaria	seccion_cpi	fecha_cpi	responsable_cpi	propuesta_cpi			
------------	-------------	-----------	-----------------	---------------	--	--	--

colind_1_d	superf_1_d	...superf_6_d	observ	elabora	fecha_elaboracion		
------------	------------	---------------	--------	---------	-------------------	--	--

autorizacion	fecha_autorizacion
--------------	--------------------

s_expediente

cedula	expediente	archivero	gaveta	total_fojas	plano_exp_fojas
--------	------------	-----------	--------	-------------	-----------------

plano_fojas	convenio_exp_fojas	convenio_fojas	acuerdo_exp_fojas	acuerdo_fojas
-------------	--------------------	----------------	-------------------	---------------

croquis_exp_fojas	croquis_fojas	contrato_exp_fojas	doc_exp_fojas	doc_fojas
-------------------	---------------	--------------------	---------------	-----------

escrit_exp_fojas	escrit_fojas	decret_exp_fojas	decret_fojas	visita_exp_fojas
------------------	--------------	------------------	--------------	------------------

visita_fojas	exp_nuevo	exp_ddf
--------------	-----------	---------

exp_particular	exp_federal	exp_desincorporado	exp_stc
----------------	-------------	--------------------	---------

Aplicando Primera Forma Normal

Antes de aplicar la primera forma normal, dado que el tamaño de la entidad **s_cedula** es muy grande se observa que los datos se pueden separar en dos tipos de datos, datos fijos del predio como son los que tienen que ver con su **s_ubicacion** y los involucrados en una **s_visita**.

Por lo que se obtienen dos entidades al separar esta.

s_ubicación.

contador	cedula	Linea	fecha	consecutivo	correspondencia	estacion	calle
----------	--------	-------	-------	-------------	-----------------	----------	-------

numero	no_interior	Colonia	entre_calles	cp	delegacion	cuenta_ant	cuenta_act
--------	-------------	---------	--------------	----	------------	------------	------------

fecha_cap	contratos1...10	fotografias1...10	escrituras1...10	convenios1...10
-----------	-----------------	-------------------	------------------	-----------------

decretos1...10	Documentos1...10	acuerdos1...10	contratos1...10
----------------	------------------	----------------	-----------------

s_visita

cedula	superficie	superficie_cons	tipo_inmueble	tipo_est	niveles	servicios
nom_inmueble	uso_inic	uso_actual	usuario	propiet	norte_r	colind_1_r
superf_1_r	...superf_6_r	fecha_conf	fecha_perf	fecha_conclus	fecha_realiz	
elaboro_visita_fisic	fecha_elabora	VoBo	Fecha_Vobo	observac	agua	luz
drenaje	estado_fisico	obser_edo_fisico				

Aplicando la 1ª Forma Normal a las cuatro entidades para eliminar ocurrencias repetidas y transformarlas en entidades bidimensionales. Se observa que en *s_visita* y *s_juridico* aparecen las mismas colindancias excepto que en una de tipo real y otra de tipo documental, por lo que se puede crear una entidad *s_colindancias* con identificador de tipo real y documental.

De la entidad ubicación y *s_docdigi* se pueden derivar las entidades:

s_ubicación con información que de describió arriba a excepción de las imágenes que se almacenaran en la siguiente entidad.

s_docdigi con todos los documentos digitalizados (escrituras, documentos, contratos, convenios y decretos) y con un campo para identificar el tipo de documento

sc_tipo_docto con los tipos de documentos y su descripción.

s_fotografias con las imágenes de las fotografías de la cédula

s_planos con la imagen del croquis y del plano de la guía roji.

Y por último de la entidad *s_expediente* se crearán las entidades :

s_expediente y **sc_tipo_expediente**.

Finalmente las entidades quedan de la siguiente manera:

s_ubicacion

contador	cedula	linea	fecha	consecutivo	correspondencia	estacion	calle
----------	--------	-------	-------	-------------	-----------------	----------	-------

numero	no_interior	colonia	entre_calles	cp	delegacion	cuenta_ant	cuenta_act
--------	-------------	---------	--------------	----	------------	------------	------------

fecha_cap

s_visita

cedula	superficie	superficie_cons	tipo_inmueble	tipo_est	niveles	servicios
--------	------------	-----------------	---------------	----------	---------	-----------

nom_inmueble	uso_inic	uso_actual	usuario	propiet	fecha_conf	fecha_pet
--------------	----------	------------	---------	---------	------------	-----------

fecha_conclus	fecha_realiz	elaboro_visita_fisic	fecha_elabora	VoBo	fecha_VoBo
---------------	--------------	----------------------	---------------	------	------------

observac	agua	luz	drenaje	estado_fisico
----------	------	-----	---------	---------------

s_planos

cedula	croquis	plano
--------	---------	-------

sc_tipo_docto

cve_tipo_doc	descripcion
--------------	-------------

s_juridico

cedula	fecha_convenio	contrato	fecha_contrato	convenio	decreto	fecha_decreto
--------	----------------	----------	----------------	----------	---------	---------------

fecha_publicacion	ley_organica	escritura_publica	fecha_escritura	notario_nombre
-------------------	--------------	-------------------	-----------------	----------------

no_notaria	seccion_cpi	fecha_cpi	responsable_cpi	propuesta_cpi
------------	-------------	-----------	-----------------	---------------

elabora	fecha_elaboracion	autorizacion	fecha_autorizacion	observac
---------	-------------------	--------------	--------------------	----------

s_colindancias

pcardinal	colindancia	superficie	tipo_colindancia
-----------	-------------	------------	------------------

sc_tipo_colindancia

cve_tipo_colin	descripcion
----------------	-------------

s_expediente

cedula	expediente	archivero	gaveta	total_fojas	exp_nuevo	exp_ddf
--------	------------	-----------	--------	-------------	-----------	---------

exp_particular	exp_federal	exp_desincorporado	exp_stc
----------------	-------------	--------------------	---------

sc_tipo_expediente

cedula	documentos	fojas	cve_tipo_doc
--------	------------	-------	--------------

Aplicando Segunda Forma Normal

En la segunda forma normal, se busca establecer que los campos no llave estén relacionados con alguna parte de la llave. Si los campos de datos sólo dependen de una parte de la llave, la llave y los campos conectados a la llave son susceptibles de separarse en registros independientes. La división de la primera tabla normalizada, en una serie de tablas en las que cada campo sólo depende de la llave completa se llama segunda tabla normalizada.

Observando nuestras entidades, se advierte que en todas las entidades excepto en los catalogos se tiene únicamente a el campo cedula como llave por lo que las entidades se encuentran también en segunda forma normal.

Aplicando Tercera Forma Normal

Se dice que una entidad se encuentra en tercera forma normal si el valor de cada campo depende de toda la llave y no de cualquier otro que no lo sea. Para nuestro caso como se observó anteriormente no existe este tipo de dependencia, por lo que también cumple con la tercera forma normal.

3.2.3 Diagrama Entidad-Relación

Apoyándonos en el punto anterior, haciendo un análisis de la normalización efectua buscando una óptima relación entre las entidades, se obtuvieron como resultado las siguientes relaciones:

La entidad s_ubicación tiene una relación de **uno a uno** con las entidades s_visita, s_juridico y s_planos cada una de ellas maneja un tipo de información en particular de los predios y se relacionan por la llave cedula.

La entidad s_ubicación tiene una relación **uno a muchos** con las tabla s_docdigi y s_foto y se relacionan a través de cedula, esto ya que para cada cedula puede haber uno o varios documentos así como fotos. Tiene relación **muchos a uno** con las entidades s_correspondencias, s_delegaciones y s_estaciones que son catalogos a través de los campos de nombre cve y el del nombre de la entidad.

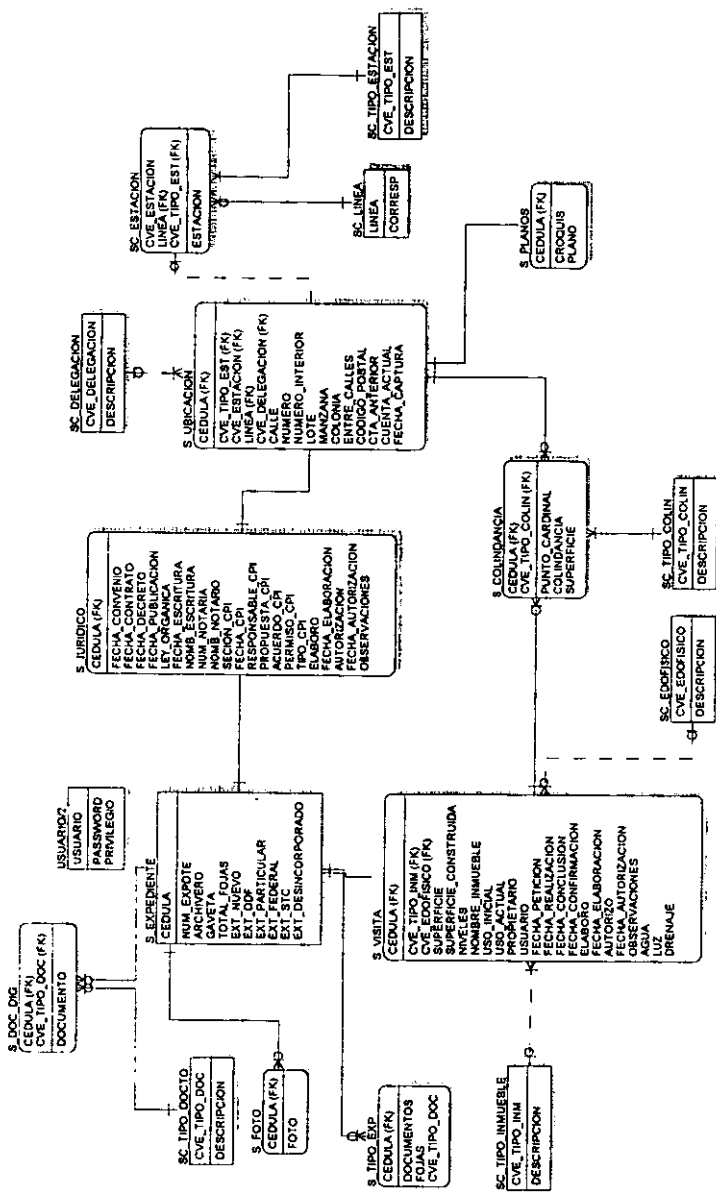
Las entidades s_visita y s_juridico tienen relación **muchos a uno** con la entidad s_colindancias cada una por separado, ya que ambas presentan las mismas colindancias excepto que una de tipo real y otra de tipo documental, se relacionan con s_colindancias a través del campo cedula

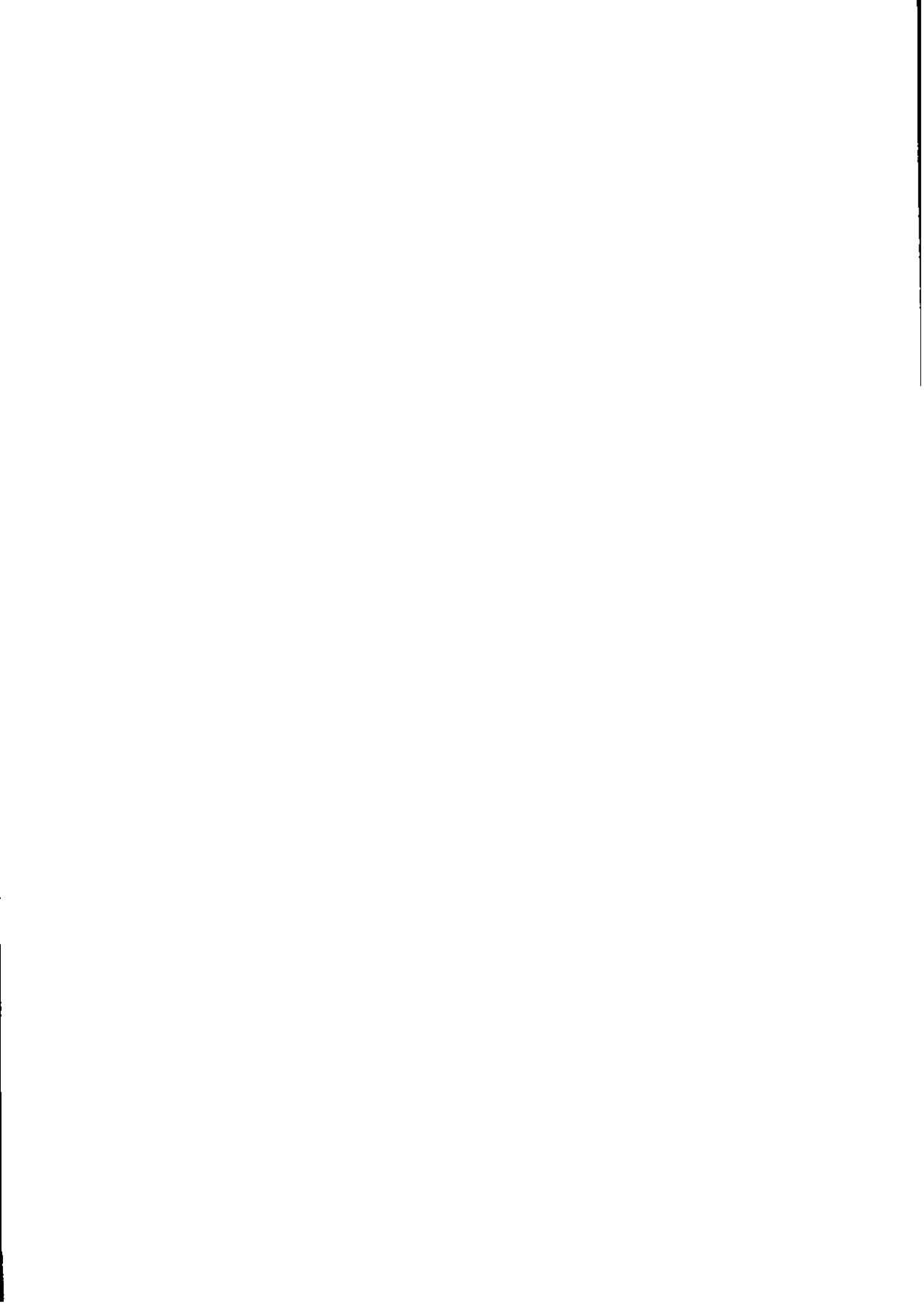
Además la entidad s_visita tiene relación muchos a uno con las entidades sc_tipo_estacion, sc_tipo_inmueble que son catálogos del sistema a través de los campos de nombre cve.... y el nombre de la entidad.

La entidad s_colindancias tiene relación muchos a uno con sc_tipo_colin a través del campo cve_tipo_colin, sc_tipo_colin es un catálogo que describe si la colindancia es real o documental.

La entidad sc_estaciones tiene relación muchos a uno con la entidad sc_linea a través del campo cve_linea , ya que la entidad sc_linea es un catálogo y cada línea tiene muchas estaciones.

De este análisis se obtiene el diagrama entidad-relación de la figura 3.2.3





3.2.4 Diccionario De Datos

A través del Diccionario de Datos nos es posible obtener una completa independencia de la estructura del modelo de datos completamente. Se almacenan los que en el medio se le conoce como Metadatos (datos definidos por datos) para todos los objetos que residen en nuestra base de datos. Este diccionario se almacena en una ubicación especial del resto de las tablas de trabajo o datos y es utilizada exclusivamente por el kernel del manejador de base de datos que utilizamos, es decir, Access.

Este diccionario de datos también nos permite validar la existencia de los objetos definidos en la base de datos, proporcionando su acceso y mapas de la ubicación física del almacenamiento

Tuvimos especial cuidado en la creación del diccionario de datos puesto que determina una ruta de acceso óptimo para guardar o extraer los datos. Normalmente el proveedor del manejador de base de datos proporciona un algoritmo sofisticado para permitir la extracción de los datos proporcionando la mejor respuesta para el primer conjunto de renglones presentados o para todos los registros que son extraídos. (Ver Apéndice C)

3.3 Implementación del Back-End

El sistema "nombre" ha sido desarrollado mediante el manejador de bases de datos relacionales Microsoft Access 2.0 y Visual Basic 4.0 que es un entorno de interfaz gráfica para el usuario, entre ambos se permite la integración de objetos como tablas, consultas, formularios, reportes, módulos y código en el lenguaje de programación Access y Visual Basic, para la creación de aplicaciones interactivas dirigidas a usuarios finales.

Los requerimientos mínimos de hardware y software para la ejecución del sistema en una PC cliente son:

- Procesador 80486 o superior
- 16 MB de memoria RAM o superior
- Monitor VGA
- Mouse Windows versión 3.1 o superior
- Microsoft Access 2.0 y Visual Basic 4.0 (sólo en caso de utilizar el sistema en modo stand-alone)
- 2.0 Gbytes de espacio en disco duro (sólo en caso de utilizar el sistema en modo stand-alone)

Si tanto el sistema como Access se instalan en un servidor, el usuario podrá ejecutar el sistema. Para ello, es importante señalar que todos los objetos mencionados anteriormente (tablas, consultas, formularios, etc) están incluidos en un sólo archivo, que en este caso es la base de datos siap.mdb. Por lo tanto, para ejecutar la aplicación, sólo es necesario crear un icono en cada PC cliente, que incluya la ruta que inicia Access, seguida del nombre de la base de datos que se quiere abrir. Al abrirse la base de datos, se ejecuta una macro llamada autoexec, la cual muestra la pantalla principal del sistema.

A continuación se describen brevemente las distintas clases de objetos del sistema:

- **Tablas:** son arreglos de renglones y columnas, que contienen todos los datos necesarios para utilizar el sistema. Más adelante se explicará la función de cada una de ellas.
- **Consultas:** son especificaciones de los datos que se desea extraer de una o varias tablas. Las consultas sirven de base para la definición de formularios, reportes y diversos controles, como las listas desplegables.
- **Formularios:** Son definiciones de presentación de los datos en un medio impreso.
- **Módulos:** es un archivo que contiene objetos gráficos llamados controles más el código.

II TABLAS

A continuación se enlistan las tablas utilizadas en la aplicación del sistema y en las páginas subsecuentes se muestra la forma en que fueron estructuradas.

1. **S_plano:** Se utiliza para almacenar la ubicación gráfica del predio.
2. **S_Docdig:** Se utiliza para almacenar todos los documentos digitalizados durante la captura de la información.
3. **S_foto:** Aquí se guardarán las imágenes del predio tomadas por la cámara digital.

4. **S_Colindancia:** Aquí se almacenará referente a las colindancias del predio a procesar dentro del sistema.
5. **S_doc_dig:** Con todos los documentos digitalizados (escrituras, documentos, contratos, convenios y decretos).
6. **S_Expediente:** Se almacenarán los datos del predio que conformaran al expediente.
7. **S_Foto:** Aquí se ubicarán las imágenes a digitalizar de los documentos.
8. **S_Jurídico:** Es la información jurídica del predio.
9. **S_Planos:** Contendrá las imagenes del croquis y los planos de la guía roji.
10. **S_tipo_exp:** Contendrá el tipo de documento, el No. de fojas y cuantos documentos contiene el expediente.
11. **S_Ubicación:** Contendrá todos los datos necesarios para la ubicación del predio.
12. **S_Visita:** Es la información recopilada de la visita física al predio, por parte del área topográfica.
13. **Sc_Delegacion:** Esta tabla contiene toda la información de los catálogos para las delegaciones.
14. **Sc_EdoFísico:** Aquí se almacenará la información de los catálogos del estado físico del inmueble.

15. **Sc_Estacion:** Se concentrará en esta table el catálogo que contiene todas las estaciones del Sistema de Transporte Colectivo, Metro.
16. **Sc_Linea:** Es el catálogo de las todas las líneas que conforman al Sistema de Transporte Colectivo, Metro.
17. **Sc_tipo_colin:** Este catálogo contendrá información real del tipo de colindancia del predio, es decir si esta es real, documental o no existe.
18. **Sc_tipo_docto:** Aquí se almacenará el catálogo que contendrá todos los tipos de documentos que pudieran existir para registrar el predio, por ejemplo: Acuerdos, Contratos, Convenios, Decretos, etc.
19. **Sc_tipo_estacion:** Contendrá que tipo de estación comprende el predio, es decir si esta es una terminal de paso, de transbordo, terminal y transbordo, etc.
20. **Sc_tipo_inmueble:** Se especificará el tipo de inmueble que es el predio, por ejemplo: si es casa particular, taller general, edificio de oficinas, etc.
21. **Usuario:** Contendrá a todos los usuarios que serán dados de alta para el uso y manipulación del sistema.

Para crear cada una de las tablas anteriormente definidas, se tuvo que seguir el siguiente procedimiento:

- 1.- Localizar y dar doble clic al icono de Microsoft Access 2.0
- 2.- Después, seleccionar el menú "File", y posteriormente la opción de "New Database"

3.- En el cuadro de dialogo "File Name" escribiremos el nombre de la base de datos a crear, por ejemplo siap.mdb, ver figura 3.3.1.

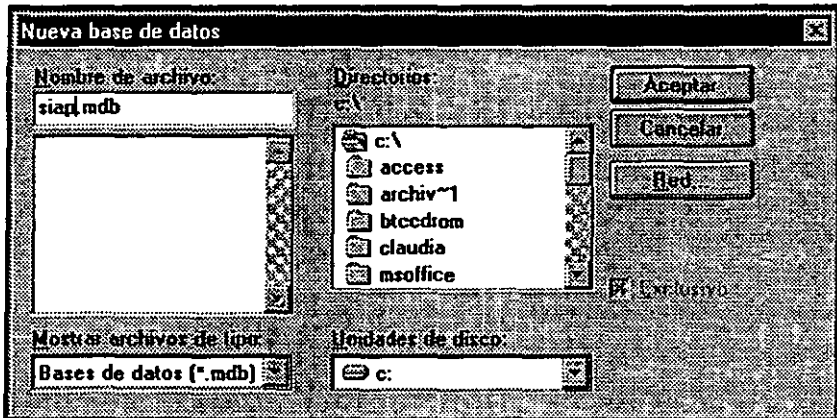


Figura 3.3.2.1 Creación de una nueva Base de Datos en Access

4.- En seguida aparecerá otra ventana en donde nosotros del cuadro de dialogo seleccionaremos la opción de "New", ver figura 3.3.2.

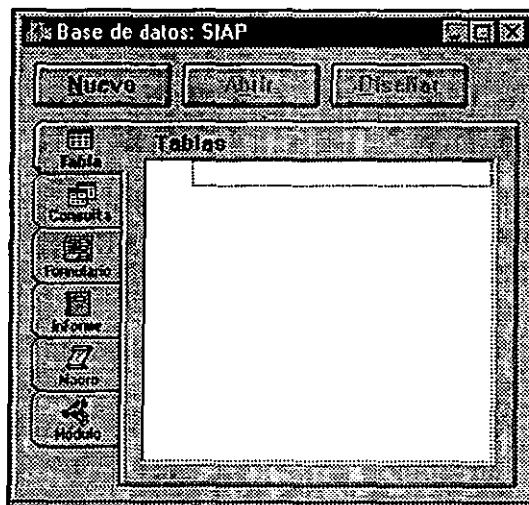


Figura 3.3.2.2 Elaboración de una tabla en Access

5.- Después aparecerá otra ventana con dos opciones:

- **Table Wizards:** este es un asistente para la creación de tablas.
- **New Table:** esta opción nos sirve cuando ya no requerimos del ayudante de Microsoft Access para generar las tablas o bien cuando tenemos que generar un tabla muy distinta a las proporcionadas por el ayudante.

En nuestro caso seleccionamos "New Table" con esta opción fueron generadas todas las bases de datos que se diseñaron para el desarrollo de este sistemas, ver figura 3.3.3.

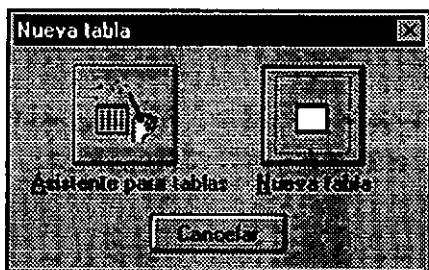


Figura 3.3.2.3 Utilización del asistente de Access para crear una tabla

6.- Con la opción de "New Table" se generaron las tablas de una forma más específica. A continuación presentaremos la forma en que fueron generadas algunas de las tablas que anteriormente se listaron:

a. Croquis

FIELD NAME	DATA TYPE	DESCRIPTION
Cédula	Text	Llave primaria

Ruta_uno	Text	Ruta donde se encuentra el croquis
----------	------	------------------------------------

b. Doc_dig

FIELD NAME	DATA TYPE	DESCRIPTION
Cédula	Text	Llave primaria
Ruta	Text	Ruta donde esta la fotografía
Cve_Tipo_doc	Numeric	Para extraer la descripción de tipos de doc.

c. S_Expediente

FIELD NAME	DATA TYPE	DESCRIPTION
Cédula	Text	Llave primaria
Num_Expdte	Text	Clave o número del expediente
Archivero	Text	Clave del archivero
Gaveta	Text	Clave de la gaveta
Total_fojas	Numeric	Total de hojas
Exp_Nuevo	Si/No	Expediente nuevo
Exp_DDF	Si/No	Expediente del D.D.F
Exp_Particular	Si/No	Expediente particular
Exp_Federal	Si/No	Expediente Federal
Exp_STC	Si/No	Expediente del STC
Exp_Desincorporado	Si/No	Expediente desincorporado

Para consultar el resto de las tablas ir al apéndice A.

7.- Una vez que terminamos de definir los campos de las tablas solo tenemos que seleccionar de la barra de menú "File" y posteriormente "Save As" para darle un nombre a nuestra tabla, ver figura 3.3.4.

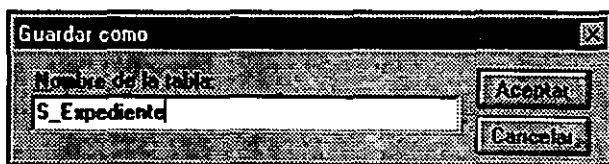


Figura 3.3.2.4 Como se guarda una Base de Datos en Access

8.- Para definir nuestra llave primaria solo tenemos que colocarnos en el campo de la tabla que deseamos establecer como nuestra llave maestra, después ir a la barra de menú y seleccionar "Edit" y finalmente "Set Primary Key".

Se especifico un identificador exclusivo de registros, mismo que se denomina llave primaria, esta puede consistir en uno o más campos que identifican a cada registro almacenado en la tabla.

En nuestro caso la llave primaria será el campo "cédula" que es el que establecerá la relación entre la mayoría de las tablas excepto los catálogos que tienen su propio identificador.

III CONSULTAS

Las consultas son especificaciones de los datos que se desea extraer de una o varias tablas, las consultas sirven de base para la definición de formularios, reportes y diversos controles, como las listas desplegables.

Para crear una consulta se tuvo que seguir una serie de pasos, que a continuación se describen.

1.- Del cuadro de dialogo se tiene que seleccionar la opción de "Query" (consulta) y posteriormente seleccionaremos la opción de "New", ver figura 3.3.3.1.

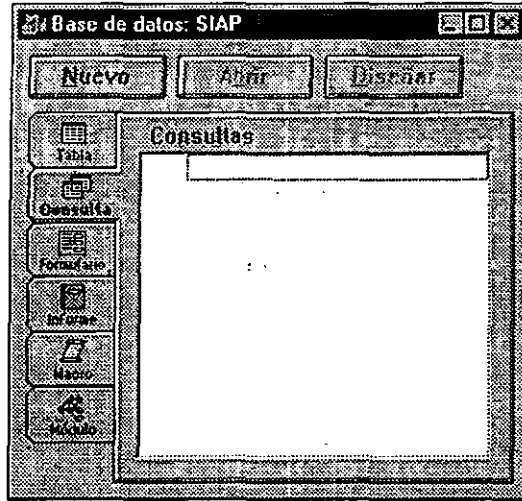


Figura 3.3.3.1 Creación de una nueva Consulta en Access

2.- Después aparecerá otra ventana con dos opciones:

- Query Wizards: este es un asistente para la creación de consultas que nos ayuda a crear consultas complejas para tareas frecuentes de administración de datos.
- New Query: esta opción nos servirá cuando ya no se requiera del ayudante de Microsoft Access para generar las consultas o bien cuando se tenga que generar un consulta muy distinta a las proporcionadas por el ayudante, ver figura 3.3.3.2.

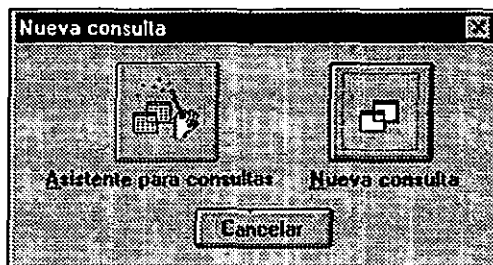


Figura 3.3.3.2 Utilización del asistente de Access

En nuestro caso seleccionamos "New Query" con esta opción fueron generadas todos los queries que se diseñaron para el desarrollo de este sistema.

3.- Una vez que seleccionamos "New Query" aparecerá una ventana en donde se tiene que especificar que tablas se quieren agregar para realizar una consulta. La forma en que se relacionan estas tablas unas con otras es a través de la llave primaria, la cual fue generada cuando se diseñaron las tablas, en nuestro caso la llave primaria es "CÉDULA", ver figura 3.3.3.3.



Figura 3.3.3.3 Agregando tablas a las consultas

En este punto se definieron características esenciales para realizar las consultas tomando en cuenta los campos de cada una de las tablas.

4.- La forma en que se relacionan estas tablas unas con otras es a través de la llave primaria, la cual fue generada cuando se diseñaron las tablas, en nuestro caso la llave primaria es "Cédula", ver figura 3.3.3.4.

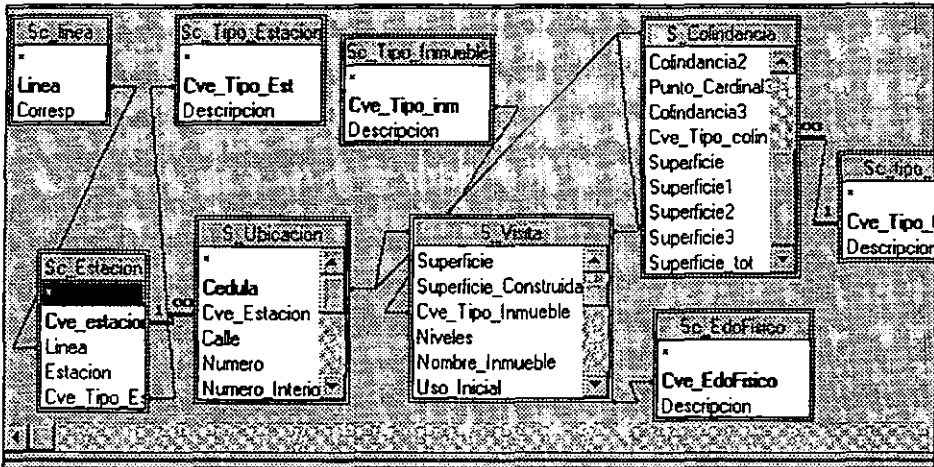


Figura 3.3.3.4 Relación de las tablas que conforman el sistema

5.- Para introducir todos los campos a la cuadrícula QBE (es la cuadrícula que aparece en la parte inferior de la pantalla), solo se posiciono el cursor del mouse al principio de cada tabla en donde me muestra un "*", con el botón izquierdo del mouse se selecciona y se suelta hasta que se haya llevado a una de las celdas de QBE, el asterisco me servirá para llevar todos los campos de la tabla al QBE, ver figura 3.3.3.5.

Campo	Linea	Cedula	Corresp	Estacion	Descrip
Orden		Cedula			
Mostrar	<input checked="" type="checkbox"/>	Cve_Estacion	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Criterios		Calle			
		Numero			
		Numero_Interior			
		Lote			
		Manzana			
		Colonia			

Figura 3.3.3.5 Ejemplo de una QBE

Con el punto anterior se esta definiendo el campo, el orden, que los campos sean mostrados en consulta o no y los criterios que se deseen asignar.

IV FORMULARIOS

Los formularios y reportes se elaboraron con la herramienta de desarrollo Visual Basic 4.0.

Visual Basic contiene herramientas que fueron utilizadas para crear la interfaz gráfica del sistema, por ejemplo: formularios y controles, módulos, lenguaje VB, barra de menús, paleta de colores.

A lo largo de este tema iremos explicando la forma en que se diseñaron los formularios y reportes.

Primero se diseñó la interfaz gráfica de usuario que es la que nos va a servir de fondo para los controles y para los gráficos situados sobre la misma, el procedimiento fue el siguiente:

1.- Seleccionamos de la barra de menú "File" y después la orden a ejecutarse "New Project", ver figura 3.3.4.1.

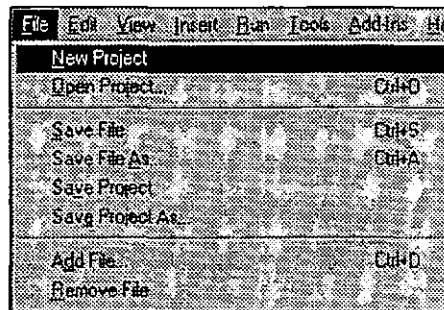


Figura 3.3.4.1 Creación de un proyecto en Visual Basic

2.- Se desplegará una nueva ventana la cual es denominada como "Form1", en ella colocamos todos los controles (objetos) que formaron parte de los formularios diseñados para SIAP, ver figura 3.3.4.2.

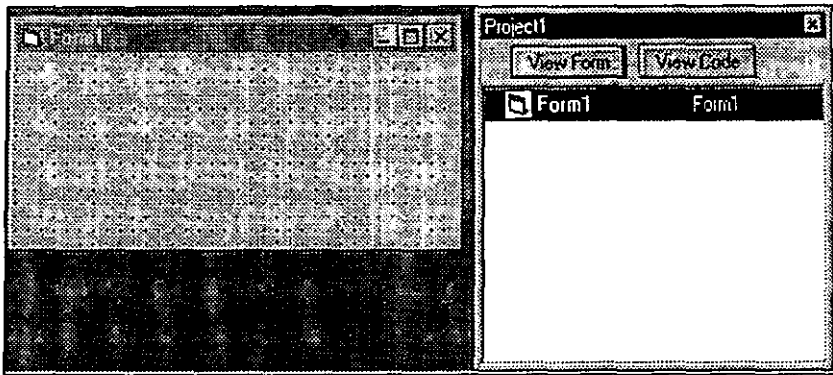


Figura 3.3.4.2 Forma de Visual Basic

Al mismo tiempo se desplegará una ventana de proyectos llamada "Project 1", la cual contendrá todos los ficheros como son formularios, clases, recursos y módulos, que forman una aplicación denominada "aplicación" o "proyecto" creados para SIAP.

Con el botón "View form" de "Project" podremos visualizar los formularios, es decir todos los controles específicos para cada una de los formularios de SIAP, ver figura 3.3.4.3.

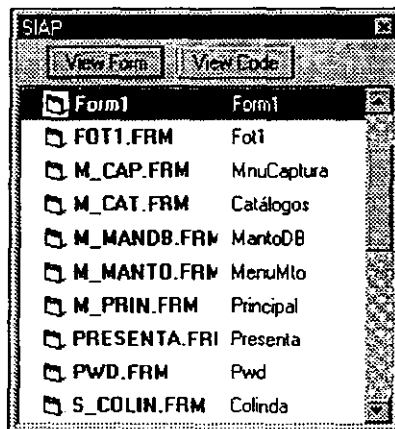


Figura 3.3.4.3 Visualización de todas las formas que conforman un proyecto

En esta misma pantalla seleccionando el botón "View Code" se podrá visualizar el código de cada uno de los formularios, módulos o clases que conforman la interfaz gráfica de SIAP, ver figura 3.3.4.4.

```

" El formato de fecha es: dd/m
Data1.Refresh
Screen.MousePointer = vbDefault
On Error Resume Next
End If
End Sub

Private Sub Command1_MouseMove(Button
SSPanel6.Caption = "Al dar clic. o E
End Sub

Private Sub Command2_Click()
Ejecuta = 2
    
```

Figura 3.3.4.4 Visualización del Código de un formulario

3.- Se fueron colocando los controles que son objetos sobre el formulario, tales como etiquetas, cajas de texto, marcos, casillas de verificación, botones de opción, botones de pulsación, etc., con la finalidad de aceptar, ejecutar o visualizar datos, ver figura 3.3.4.5.

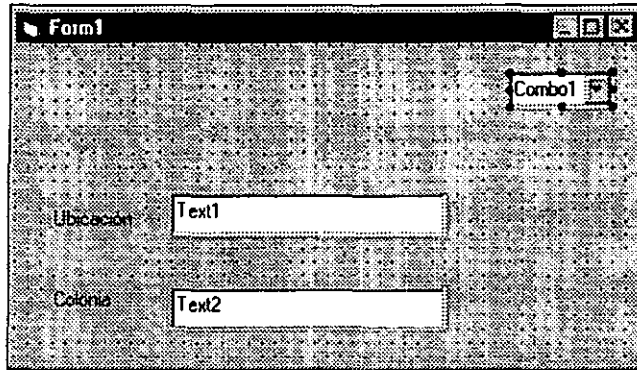


Figura 3.3.4.5 Creación de una nueva forma en Visual Basic

4.- Una vez seleccionados y colocados los objetos necesarios para nuestros formularios se configuraron las propiedades de cada uno de ellos en la ventana de "Properties", cada objeto lleva asociado un conjunto de propiedades. En esta ventana se pueden configurar características específicas como son colores, tamaños de los objetos, tipo de letra para las etiquetas, nombres, posiciones, etc; ver figura 3.3.4.6.

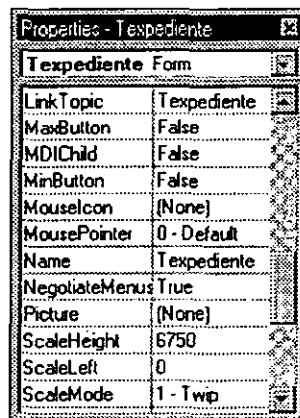


Figura 3.3.4.6 Características de un objeto en Visual Basic

5.- Cuando se finalizo la definición de las propiedades de los controles, se escribio el código, el que se uniría a un objeto denominado también como procedimiento conducido por un susceso o simplemente procedimiento, ver figura 3.3.4.7.

```
El formato de fecha es: dd/mm
Data1.Refresh
Screen.MousePointer = vbDefault
On Error Resume Next
End If
End Sub

Private Sub Command1_MouseMove(Button
SSPanel6.Caption = "Al dar clic. o E
End Sub

Private Sub Command2_Click()
Ejecuta = 2
```

Figura 3.3.4.7 Ejemplo del código utilizado por Visual Basic

En el cuadro de "Procedure" aparecerá una lista de eventos para cada uno de los objetos.

Para visualizar la ventana de código de un determinado formulario, hay que hacer un doble clic dentro del área correspondiente a un control y entonces se visualizará la ventana de código para ese control.

De esta forma podemos decir que se construyeron todos los formularios de SIAP utilizando Visual Basic 4.0.

A continuación se muestra una de las pantallas que conforman a SIAP totalmente terminadas, ver figura 3.3.4.8.

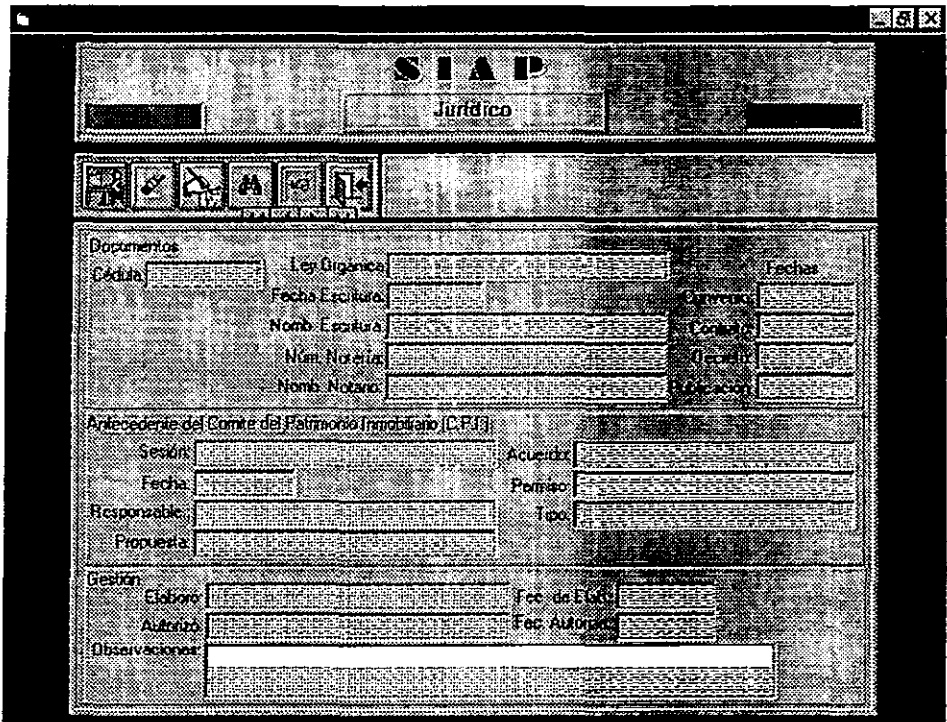


Figura 3.3.4.8 Ejemplo de una pantalla de SIAP elaborada en Visual Basic

6.- Una vez finalizada la forma de nuestro sistema tuvimos que salvar la aplicación de la siguiente forma:

Seleccionamos de la barra de menú "File" después "Save Project" y por último asignamos un nombre a nuestro proyecto.

V. MÓDULOS

Un módulo correspondiente a un formulario es un archivo que contiene objetos gráficos, llamados controles, más código; mientras que un módulo estándar o un módulo correspondiente a una clase sólo contiene código.

Para crear un módulo se tienen que seguir los siguientes pasos:

- 1.- En la barra de menú seleccione "Insertar" y posteriormente la orden de "Módulo".
- 2.- Aparecerá una ventana de código en la cual escribiremos el procedimiento, incluyendo las líneas Sub y End Sub del mismo, ver figura 3.3.5.1.

```

End Function

Public Function Llena_lista(base As Database, reg As ...
    Dim i As Integer
    Dim j As Integer
    Dim texto As String
    On Error Resume Next
    reg.MoveFirst
    j = 0
    While Not reg.EOF
        For i = 0 To reg.Fields.Count - 1
            texto = ""
            texto = reg.Fields(i)
        
```

Figura 3.3.5.1 Elaboración de un Módulo del sistema en Visual Basic

VI REPORTES

Los reportes fueron creados con un generador de informes (Crystal Reports) para la creación de informes utilizando datos de la base de datos SIAP.

Para la elaboración de reportes con esta herramienta se realizó el siguiente procedimiento:

1.- De la barra de menú selecciónese "File" y posteriormente la orden "Report", ver figura 3.3.6.1.

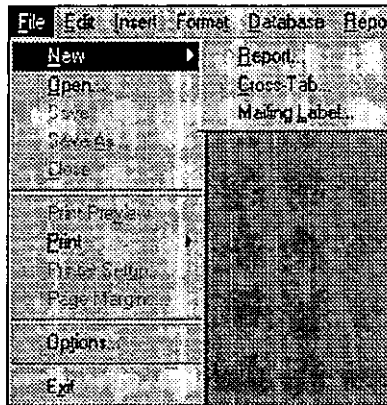


Figura 3.3.6.1 Seleccionando la orden de Reporte en la Barra de Menú

2.- Cuando aparece esta opción en Crystal Reports nosotros seleccionamos en "Create new" la opción de "Report" y en "From" seleccionamos "Data File", ver figura 3.3.6.2.

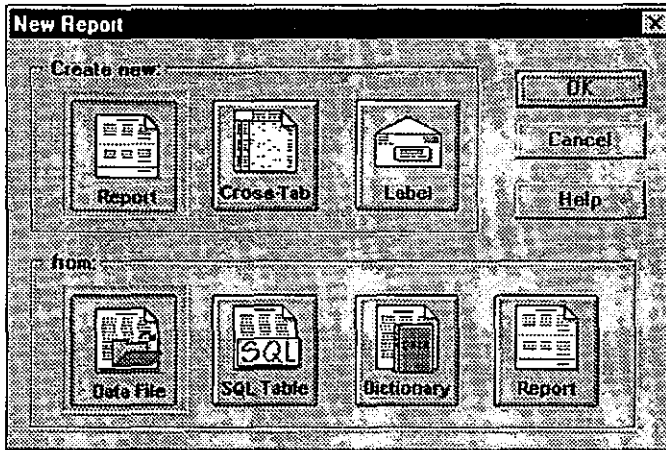


Figura 3.3.6.2 Selección del tipo de reporte a crear en Crystal Reports

3.- Cuando se selecciono el tipo de reporte que se queria generar y damos un clic en aceptar, automaticamente aparecerá una ventana nueva en donde nos pide proporcionemos la ruta de donde va a tomar la base de datos a utilizar para generar los reportes, ver figura 3.3.6.3.

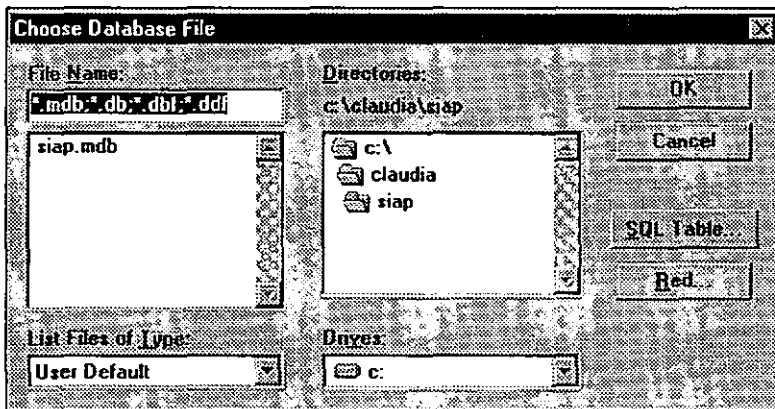


Figura 3.3.6.3 Proporcionando el nombre de la base de datos para generar el reporte

4.- Una vez que nosotros seleccionamos la base de datos SIAP nos aparecerá una pequeña ventana en donde nos pedirá que agreguemos los datos necesarios generar los reportes, ver figura 3.3.6.4.

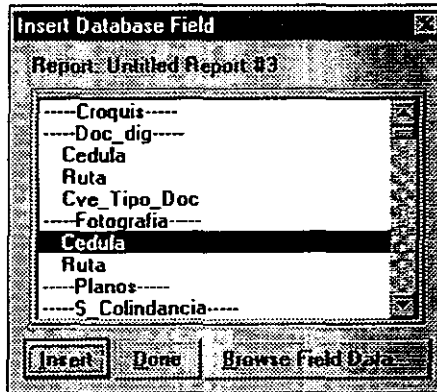


Figura 3.3.6.4 Selección de los campos que conformarán el reporte

Al mismo tiempo se abrirá una venta en donde nosotros iremos colocando los datos seleccionados en la ventana anterior solo hay que dar un clic sobre el registro que nos interesa insertar y finalmente un clic en la orden "Insert", ver figura 3.3.6.5.

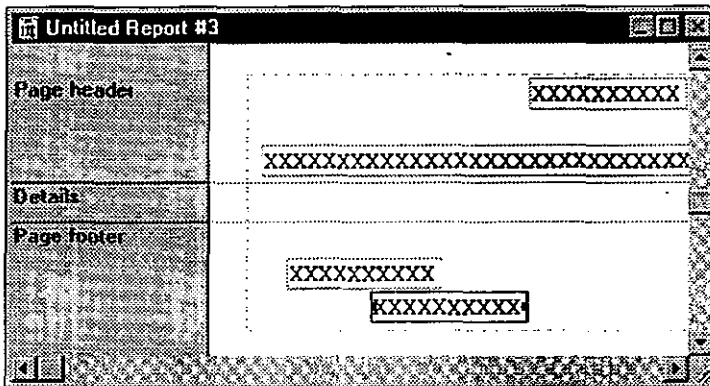


Figura 3.3.6.5 Colocación de los campos en el reporte

Una vez que nosotros terminamos de colocar todos los registros sobre el nuevo formato del reporte solo nos resta escribir el nombre de cada uno de los registros colocados en el, ver figura 3.3.6.6.

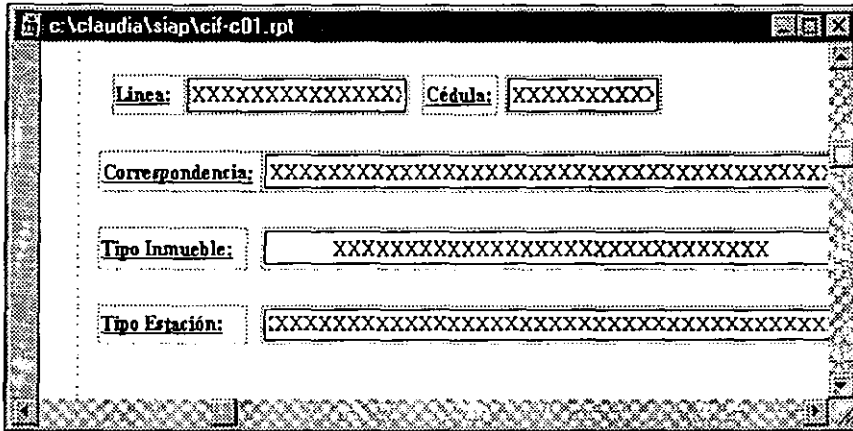


Figura 3.3.6.6 Escribiendo el nombre de cada uno de los registros colocados en el reporte

Para colocar el nombre de los campos los hicimos de la siguiente forma: de la barra de menu seleccionamos "Insert" y por ultimo "Text field", con estos comandos nosotros podremos escribir las etiquetas correspondientes a los campos que insertamos en el reporte.

VII ARCHIVO EJECUTABLE

Una vez que se ha salvado la aplicación y se ha asignado un nombre, solo nos resta hacer que esta aplicación sea un archivo ejecutable, es decir la aplicación se podrá desplegar en cualquier otro ambiente.

- 1.- Para crear este archivo hay que seleccionar de la barra de menú "File" y por último la opción de "Make EXE File", ver figura 3.3.7.1.

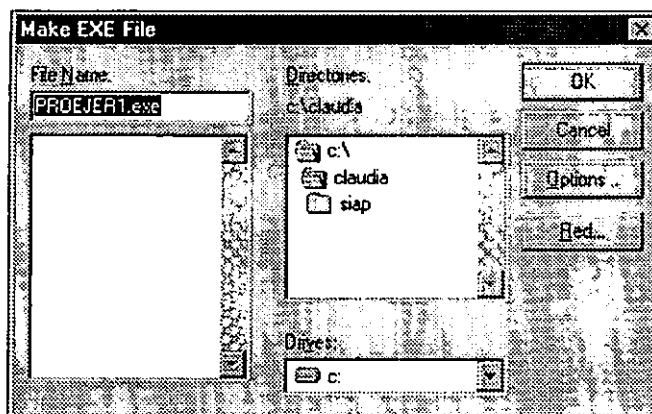


Figura 3.3.7.1 Creando un archivo ejecutable en Visual Basic

Cuando aparece esta ventana solo tenemos que proporcionar el nombre del archivo y la ruta donde deseamos colocarlo.

3.4 Desarrollo De Las Pantallas De Captura

Para este punto se tomará una Pantalla o Forma, la cual se descompondrá en los diferentes objetos que contiene, se describirán todos estos, el nombre que reciben, así como una breve descripción del propósito que se busca al incluirse en esta pantalla, también se tomará en cuenta que en algunas opciones esta pantalla llama a otras pantallas que a su vez son subpantallas de la primera, cabe mencionar que el objetivo es describir a detalle esta Forma ya que la construcción e implementación de las demás es similar y la información contenida varía dependiendo de la parte del sistema en la que se ubique para ilustrar esto se considerará el Módulo de Captura en la cual se captura las visitas efectuadas a los predios, se explicará la construcción de la pantalla principal y las dos subpantallas que a vez llama esta.

Para empezar se describirán los elementos principales de una ventana en cualquier aplicación Windows pero enfocada a la pantalla del sistema que se explicará.

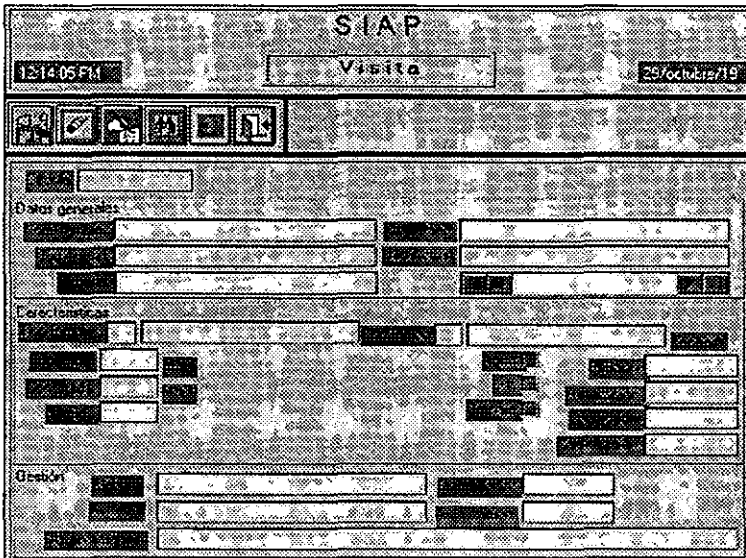


Figura 3.4.1 Pantalla de Módulo Visita

Como se mencionó en el capítulo 2 un objeto dentro de la forma tiene ciertas propiedades, las cuales hacen a los objetos diferentes y únicos entre sí.

En la construcción de la forma se mencionará el tipo de objetos que contendrá así como las propiedades que los forman, para hacer la explicación breve y poco redundante se analizarán las propiedades de un solo objeto.

Primero se da el Título a la forma en la propiedad Caption que aparece en la ventana de propiedades.

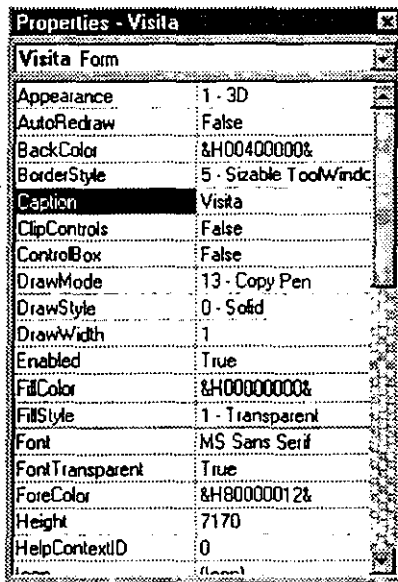


Figura 3.4.2 Ventana de Propiedades

Otras propiedades importantes para este objeto pueden ser el tipo de letra, el estilo, el color del fondo y de las fuentes, tamaño, etc.

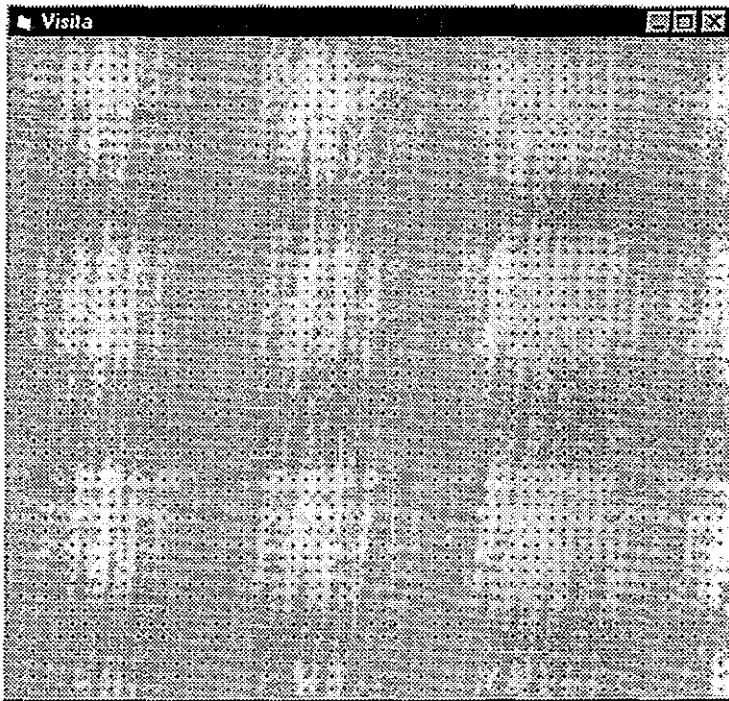


Figura 3.4.3 Inicio de Forma Visita

El siguiente paso es crear la barra de menús con las diferentes opciones, la cual se realizó con el botón que se encuentra en la barra de herramientas, como se indicó anteriormente.

Al igual que el objeto forma, el menú cuenta con propiedades similares, como Caption que siempre muestra el rótulo que el usuario observará, y además el nombre lógico que llevará el objeto para elaborar el código del Sistema. Estas son dos de las propiedades de mayor importancia, debido a que constituyen la manera de ser llamadas por el usuario (Caption) y por el desarrollador (Name).

A continuación crearemos la barra de herramientas que ayuda a realizar las operaciones más comúnmente usadas durante la operación de un sistema.



Figura 3.4.4 Barra de Herramientas

En estos elementos las propiedades que destacan y se manejan con más frecuencia son la alineación, el picture o bmp que se ligará con el botón, el estado del botón y su habilitación.

Las propiedades de los objetos iconos son las que se muestran a continuación :

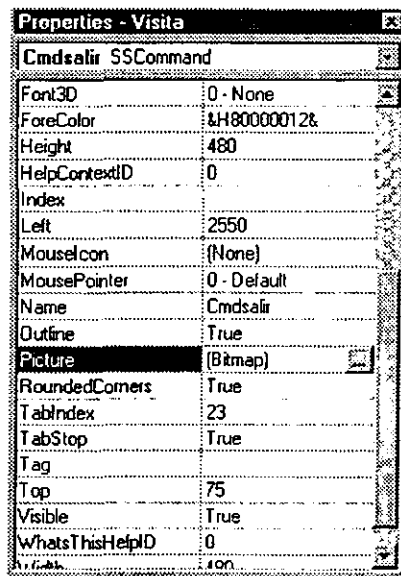


Figura 3.4.5 Propiedades de los botones de la barra de herramientas

El siguiente paso es colocar tres cajas de texto para informar el número de forma y el título, esto se maneja para facilitar la identificación de la forma al momento de la programación, y más adelante, una vez implementado el sistema como referencia pedirle al usuario el código de la forma en caso de nuevo requerimiento o mantenimiento al Sistema.

Las propiedades que componen al objeto Caja de texto son las siguientes:

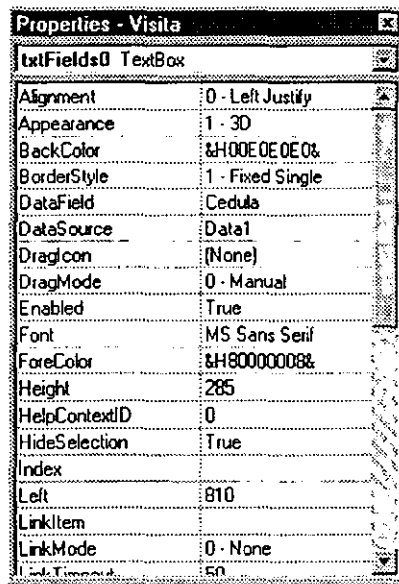


Figura 3.4.6 Propiedades de una caja de texto

Estas propiedades son similares en características para los tres cuadros de texto, en lo que difieren es en la alineación y el tipo de Fuente que se utilizó.

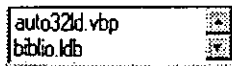
A continuación se mostrará la parte en la que el usuario comienza a seleccionar e introducir información pero antes se mencionarán las funciones de algunos elementos que se emplean con frecuencia ya que son de gran utilidad para simplificar la manera de construir la forma y estos son:

Cajas de selección :



o conocido en la barra de herramientas como Combo Box que permiten suavizar la forma de introducir los datos por el usuario, ya que despliega una lista de opciones válidas, o en un determinado momento capturar y memorizar la opción. Además permite ahorrar espacio al programador dentro de la forma.

Cajas de lista:



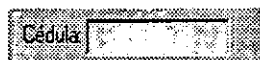
permiten desplegar elementos que le dan a escoger al usuario entre una lista de opciones disponibles. La caja muestra los items que están disponibles, y el usuario selecciona uno de ellos mediante un click. En la barra de herramientas se tiene el



siguiente icono . A diferencia de los anteriores, en esta lista no podemos introducir nuevos datos.

Una vez descritos estos elementos de la forma se llevará a cabo el armado de la misma.

Se incluirá una caja de texto :



En esta se coloca la cédula que identifica el predio en el que se realizó la visita, esta parte está acompañada de un objeto Label llamado "Cédula" y las propiedades para la caja son las siguientes:

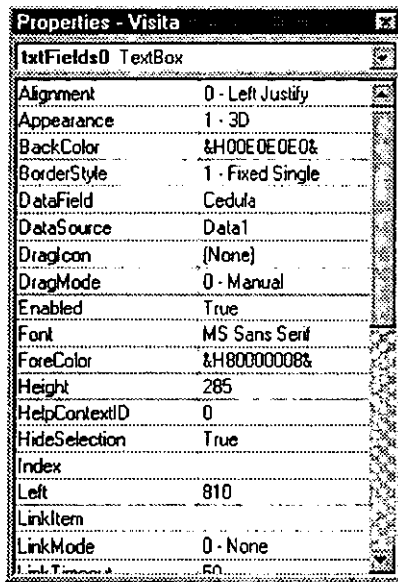


Figura 3.4.7 Propiedades de la caja de texto

La siguiente parte está encerrada en un panel, con la finalidad de agrupar los objetos que se refieren a una misma actividad, su ícono en la barra de herramientas es el siguiente :

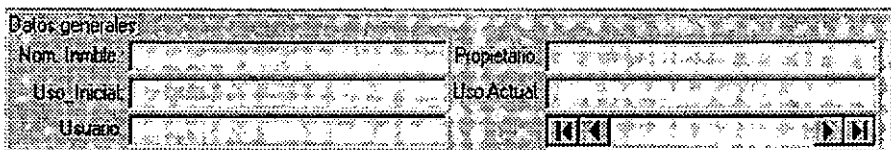


Figura 3.4.8 Un panel de la forma s_visita.

dentro de este frame se colocan cinco cajas de texto y una barra de datos.



Las propiedades para los paneles son las siguientes :

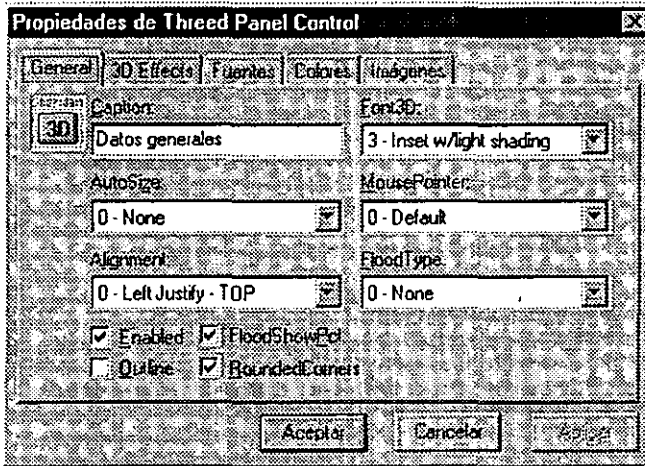


Figura 3.4.9 Propieades de los paneles.

Para las barras de datos sus propiedades se definen de la siguiente manera :

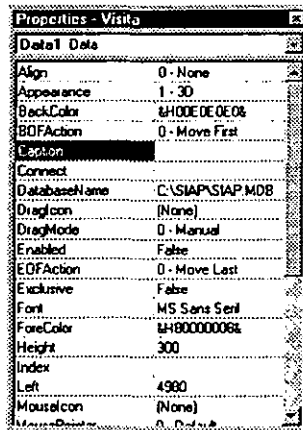
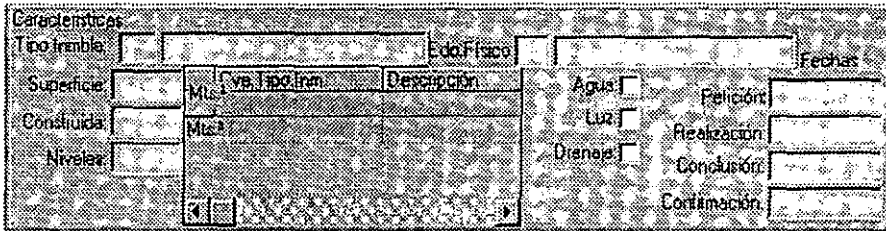


Figura 3.4.10 Propieades de las barras de datos.

La siguiente parte es otro panel que se encuentra bajo del primer panel.



Dentro de este se encuentran once cajas de texto con sus respectivos objetos Label para identificar los campos, también contienen tres cajas de selección, su icono en la barra de herramientas es el siguiente:



Las propiedades de una caja de selección (Check Box) son las siguientes :

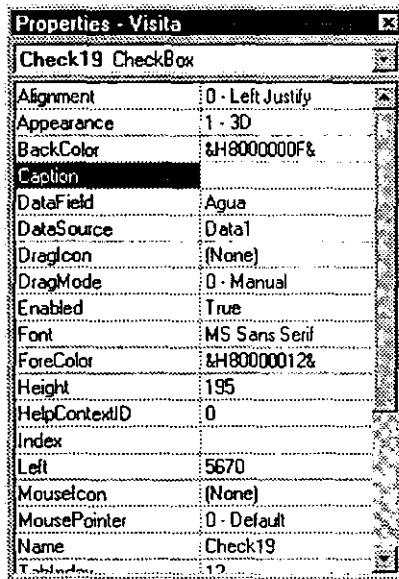


Figura 3.4.11 Propiedades de una caja de selección.

Además contiene dos objetos Apex Data Bound Grid Control, su ícono es el siguiente:



Este objetos solo aparecen cuando se va a introducir información en los campos clave tipo inmueble y clave tipo estación, aparece la información permitida en estos campos y solo basta seleccionar el valor adecuado, el cual es desplegado en estos objetos. Las propiedades de estos objetos son las siguientes :

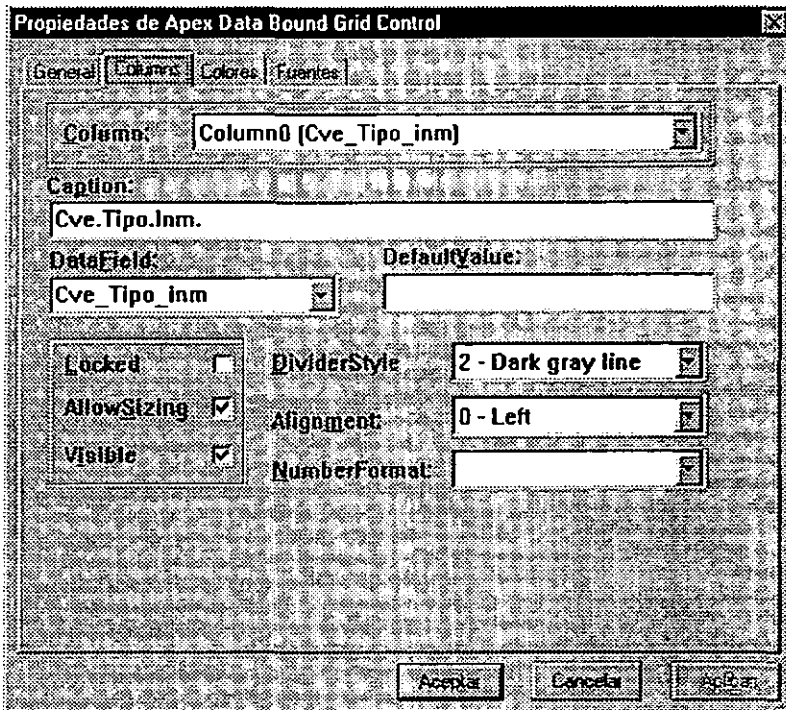


Figura 3.4.12 Propiedades de un objeto Apex Data Bound Grid Control

Nuevamente bajo el anterior panel se encuentra un tercer y último panel.

En este panel se encuentra cinco cajas de texto y un objeto Label a la izquierda de cada caja.

Los tres paneles están contenidos en un panel principal , además contiene un panel con el título de la pantalla. De esta forma la pantalla quedará de la siguiente manera :

Figura 3.4.13 Pantalla de Captura de Información de Visita

Todas y cada una de las formas y objetos del Sistema van ligadas a un código. A continuación se presenta el código de una de las pantallas que se presentan en el sistema.

```
Dim sql, sql1, sql2 As String
Dim rs, r1, r2 As Recordset
Dim db As Database
Dim Ejecuta As Integer
Dim respuesta As Variant
```

```
` Agrega
```

```
Private Sub cmdAdd_Click()
```

```
Ejecuta = 1
```

```
Command2.Visible = False
```

```
Cmdupdate.Visible = False
```

```
Cmdsalir.Visible = False
```

```
Comconfirm1.Visible = True
```

```
Cmdadd.Visible = False
```

```
Consulta.Visible = False
```

```
C_limpTot
```

```
C_habTot
```

```
txtFields0.SetFocus
```

```
End Sub
```

```
` Borrar
```

```
Private Sub Command2_Click()
```

```
Ejecuta = 2
```

```
Cmddelete.Visible = False
```

```
Cmdupdate.Visible = False
```

```
Cmdsalir.Visible = False
```

```
Cmddelete.Visible = True
```

```
Cmdadd.Visible = False
Command2.Visible = False
Consulta.Visible = False
C_habTot
txtFields0.SetFocus
End Sub
` Actualiza
Private Sub cmdUpdate_Click()
    Ejecuta = 3
    Command2.Visible = False
    Cmdupdate.Visible = False
    Cmdsalir.Visible = False
    Cmdadd.Visible = False
    Command1.Visible = True
    Consulta.Visible = False
    C_habTot
    txtFields0.SetFocus
    C_limpTot
End Sub
` consulta
Private Sub Consulta_Click()
    Ejecuta = 2
    Command2.Visible = False
    Cmdupdate.Visible = False
    Cmdsalir.Visible = False
    Consulta.Visible = True
    Cmdadd.Visible = False
    Consulta.Visible = False
    C_limpTot
    C_habTot
    txtFields0.SetFocus
End Sub
```

```
` Restaurar
Private Sub cmdRefresh_Click()
C_Restaura
SSCommand1.Visible = False
SSCommand2.Visible = False
Data2.Visible = False
Data3.Visible = False
DBGrid1.Visible = False
DBGrid2.Visible = False
End Sub

` Salir
Private Sub Cmdsalir_Click()
MnuCaptura.Show
Visita.Hide
End Sub

Private Sub txtFields0_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
On Error GoTo V_error
Select Case Ejecuta
Case 1
    If txtFields0 = "" Then
        Beep
        MsgBox "Favor de Introducir la Clave de la Cédula",
48, "siap"
        Exit Sub
    Else
        sql = "SELECT S_Ubicacion.Cedula FROM S_Ubicacion
WHERE S_Ubicacion.Cedula = '" & txtFields0 & "'"
        Set rs = db.OpenRecordset(sql, dbOpenDynaset)
        If rs.RecordCount = 0 Then
```

```
        MsgBox "Esta Clave de Cédula No Existe,
favor de verificar, Gracias.", 48, "siap"
        C_Restaura
        SSCommand1.Visible = False
        SSCommand2.Visible = False
        Exit Sub
    Else
        Data1.Refresh
        Data1.Recordset.AddNew
        Data1.Recordset.AddNew
        txtFields0 = rs(0)
        txtFields5.SetFocus
        Llena_Fecha
        SSCommand1.Visible = True
        SSCommand2.Visible = True
        Exit Sub
    End If
End If

Case 2

    sql = "SELECT * FROM S_Visita WHERE S_Visita.Cedula = '" &
txtFields0 & "'"
        Data1.RecordSource = sql
        Data1.Refresh
        Arregla_fechas
        If Data1.Recordset.RecordCount < 1 Then
            MsgBox "No existe esta Clave de
Visita,Revise.", 48, "siap"
            C_Restaura
            Exit Sub
        End If
```

Case 3

```

SSCommand1.Visible = True
SSCommand2.Visible = True
txtFields0.Enabled = False
sql = "SELECT * FROM S_Visita WHERE S_Visita.Cedula = '" &
txtFields0 & "'"
    Data1.RecordSource = sql
    Data1.Refresh
    Arregla_fechas
    If Data1.Recordset.RecordCount < 1 Then
        MsgBox "No existe esta Clave de
Visita,Revise.", 48, "siap"
        C_Restaura
        SSCommand1.Visible = False
        SSCommand2.Visible = False
        Exit Sub
    End If

```

Case 4

```

End Select
sql2 = "SELECT
Sc_Tipo_Inmueble.Descripcion,Sc_EdoFisico.Descripcion" & _
" FROM Sc_Tipo_Inmueble,Sc_EdoFisico" & _
" WHERE Sc_Tipo_Inmueble.Cve_Tipo_inm =" & txtFields3 & _
" And Sc_EdoFisico.Cve_EdoFisico =" & txtFields22
Set rs2 = db.OpenRecordset(sql2, dbOpenSnapshot)
Text1 = rs2(0)
Text2 = rs2(1)
Data1.Enabled = True
rs2.Close
End If

```

```
V_error:
If Err.Number > 1 Then
    MsgBox "Ha ocurrido un error,por favor oprima el botón de
Reiniciar.Identifique el problema y/o Consulte a su
Administrador....", 48
    Datal.Refresh
    On Error Resume Next
End If
End Sub

Private Sub txtFields3_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
txtFields22.SetFocus
End If
End Sub

Private Sub DBGrid1_DblClick()
txtFields3 = DBGrid1.Columns(0).Value
Text1 = DBGrid1.Columns(1).Value
DBGrid1.Visible = False
End Sub

Private Sub txtFields22_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
txtFields1.SetFocus
End If
End Sub

Private Sub txtFields10_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
txtFields11.SetFocus
End If
End Sub
```

```
Private Sub txtFields14_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
txtFields15.SetFocus
End If
End Sub
```

3.5 Desarrollo De Los Reportes Del Sistema

Para el desarrollo de los reportes se utilizó el programa complementario Crystal Reports el cual viene con Visual Basic en sus ediciones Profesional o Empresarial. Los reportes son diseñados por Crystal Reports y se imprimen desde Visual Basic colocando el control Crystal Reports dentro de los formularios y manejando sus propiedades en tiempo de ejecución.

Crystal Reports contiene dos elementos separados: el programa Crystal Reports y el control personalizado Crystal Reports. Si se usan estos dos elementos a la vez, dan la habilidad de generar reportes personalizados transparentemente por medio de la aplicación Visual Basic.

El programa Crystal Reports es un poderoso escritor de reportes de Windows que se usa para diseñar una variedad casi interminable de reportes personalizados. El programa viene con un juego completo de herramientas de diseño intuitivo, que permiten arrastrar y soltar los elementos del reporte, como campos y otras variables, agrupamientos y subtotales, formateo de tipo de letra sencillo en textos literales y variables.

El control personalizado Cristal Report permite acceder las capacidades de escritura de reportes de Crystal Reports rápida y fácilmente e incluir simplemente el control en el proyecto Visual Basic. El control personalizado es un conjunto de herramientas que facilita la construcción de conexiones entre la aplicación y la máquina de impresión. Mediante el uso de estas herramientas se puede hacer que la aplicación genere los reportes en un periodo muy corto. El control personalizado está visible en tiempo de diseño, pero invisible en la ejecución. Cuando se pone el control en un formulario y se establecen las propiedades, se añade una avanzada funcionalidad de reporte a la aplicación sin escribir mucho código adicional.

Agregar el Control personalizado a un proyecto Visual Basic

Se tienen que seguir los siguientes pasos :

1. Abrir el cuadro de diálogo **Custom Controls** seleccionando Custom Controls del menú *Tools* o usando la combinación de tecla de atajo, **Ctrl-T**.
2. En el cuadro de lista, se localiza Crystal Report Control y se marca.
3. Cerrar el cuadro de diálogo y el control se añadirá a el cuadro de herramientas.
4. Cuando se quiera añadir las capacidades Crystal Reports a un programa, se puede hacer doble click en el control o clic sencillo y colocarlo de la manera usual. El control se añadirá al formulario actual, pero estará invisible en tiempo de ejecución.

Uso del control personalizado Crystal Report

Una vez que se tiene el objeto de control personalizado Crystal Report en un formulario, se debe construir la conexión entre la aplicación y los reportes que se diseñaron, estableciendo las propiedades del objeto en la forma usual, esto es, mediante la lista *Properties* del control. Las propiedades que se pueden especificar incluyen :

ReportFileName	Es una expresión de cadena que indica el nombre del reporte que se quiere imprimir en respuesta a un evento de aplicación.
Destination	Es el destino del reporte : ventana = 0, archivo = 1, o impresora = 2.
Copies to printer	Si el reporte va a la impresora, es la cantidad de copias.
PrintFileName	El nombre del archivo, si se está imprimiendo a disco.

Se pueden también establecer muchas propiedades que se relacionan con la apariencia, tamaño y porción de la ventana de reporte, en caso de que se decida imprimir a la pantalla. Estas son las propiedades relacionadas.

WindowBorderStyle	Es una expresión entera entre 0 y 3, que corresponde a None, Fixed Single, Sizable y Fixed Double.
WindowControlBox	False si se desea que no aparezca el cuadro de control de la ventana. True si se desea lo contrario.
WindowMaxButton	False si se desea que desaparezca el botón minimizar de la ventana. True si se desea lo contrario.
WindowTitle	Es una expresión de cadena por desplegarse en la barra de título de la ventana del reporte.
WindowTop	Establece la posición Y de la ventana en pixeles.
WindowLeft	Establece la posición X de la ventana en pixeles.
WindowHeight	Establece la altura de la ventana del reporte en pixeles.
WindowWidth	Establece la anchura de la ventana del reporte en pixeles.

La salida impresa atractiva y legible es una parte indispensable de cualquier programa de base de datos. El diseñador y el control de reportes de Crystal

Reports son, por lo tanto, adiciones poderosas del IDE (ambiente de desarrollo integrado) de Visual Basic.

USO DE CRYSTAL REPORTS

Se arranca el diseñador de reportes Crystal Reports, mediante la selección de Report Designer desde el menú Add-Ins de Visual Basic.

El área de diseño de Crystal Reports incluye una barra de herramientas y una cinta para el formateo de texto, similar a las de muchos programas de procesamiento de palabras y hojas de cálculo. El lado derecho de la barra de herramientas muestra diez botones que se refieren al proceso del diseño del reporte. Estos se describen a continuación :

- | | |
|-------------------------|---|
| Insert a Database Field | Despliega un cuadro de diálogo que contiene campos de la tabla de base de datos asociada; se puede seleccionar un campo y colocarlo dentro de la plantilla del reporte. |
| Insert a Text Field | Abre un cuadro de diálogo dentro del cual se puede teclear el texto para colocarlo en el reporte. |
| Insert a Formula Field | Abre un cuadro de diálogo dentro del que se teclea el nombre del campo de fórmula. Al hacer click en Ok, abre otra ventana dentro de la cual se teclea la fórmula directamente o se construye mediante la selección de los elementos Field (campo), Function (función) y Operator (operadores). Después de marcar y hacer click |

en el botón Accept, se puede poner el campo de fórmula en cualquier parte del reporte.

Insert a Summary Field

Una vez que se coloca un campo de la base de datos en la sección de detalle de la plantilla del reporte, puede seleccionarse y oprimir este botón para crear un campo de resumen relacionado, que podría reportar el conteo mínimo o máximo. Si el campo de base de datos que se seleccionó es numérico, éste puede también totalizarse o promediarse.

Insert a Graphic Item

Abre un cuadro de diálogo desde el cual se puede seleccionar un archivo gráfico para incluirlo en el reporte.

Insert An OLE Object

Abre un cuadro de diálogo Insert Object desde el cual se puede seleccionar un objeto OLE para incluirlo en el reporte.

Draw a Line

El indicador del ratón se convierte en un "lápiz" con el que se puede trazar una línea en cualquier parte del reporte. Una vez que se traza, se puede hacer click en el botón derecho para ajustar su formato, e incluir el color, la anchura y el estilo.

Draw a Box

El indicador se convierte en un "lápiz" con el que se puede trazar un rectángulo en cualquier parte del reporte. Una vez que se traza, se puede hacer click con el botón derecho para ajustar el

formato, e incluir color de relleno, el del borde, la anchura y el estilo.

Sort The Report

Abre el cuadro de diálogo Record Sort Order desde el que se puede seleccionar el campo o campos para ordenar ascendente o descendentemente.

Sort Record

Abre el cuadro de edición Record Selection Formula, desde el cual se puede usar Selection Criteria para construir una fórmula de selección, como `Contacts.City = "Seattle"`.

CREACIÓN DE UN REPORTE

Se selecciona New (Nuevo) del menú File (Archivo), luego se selecciona Report (informe) (o haciendo click en el botón New Report de la barra de herramientas). Asegurándonos de crear un reporte nuevo a partir de un archivo Data, se hace click en OK. En la ventana Choose Database File, localizar SIAP.MDB, luego se hace click en Open (Abrir). Ahora está visible una nueva plantilla de diseño que consta de tres bandas, junto con el cuadro de diálogo Insert Database Field. Las tres bandas son Page header, Details y Page footer. Las bandas de la plantilla representan tres áreas del reporte. El encabezado de página, el registro del detalle y el pie de página. Cualquier cosa que se teclea dentro de las bandas del encabezado o del pie de página se repetirán en la parte superior e inferior de cada página en el reporte. Los encabezados se usan, por lo general, para desplegar descripciones de reportes, etiquetas de columnas y fechas de impresión. Los pies de página pueden usarse para desplegar números de página, por ejemplo, o grupos especialmente formateados y/o campos de resúmenes de reportes.

La banda de detalle contiene información de los campos que se quiere imprimir para cada registro de la tabla de base de datos asociada. Aquí es donde se colocarán los campos línea, cedula, correspondencia, estación, tipo inmueble, tipo estación, niveles, agua, luz, drenaje, uso inicial, uso actual, nombre del inmueble de la tablas s_ubicación, s_visita, sc_linea, sc_estacion. Para hacerlo, se seleccionan los campos en el cuadro de diálogo Insert Database Field y se hace click en Insert (o doble click en los nombres de los campos); esto permite arrastrar el campo a la posición que se desea. Figura 3.5.1

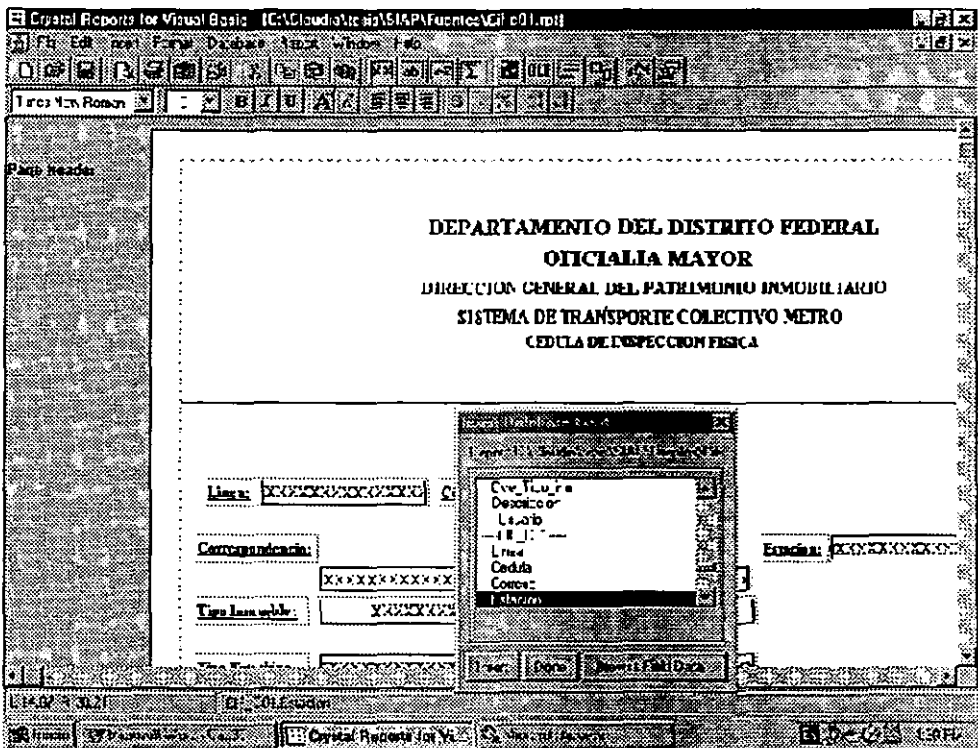


Figura 3.5.1 Modo Diseño de Crystal Reports

Mientras se colocan los campos lado a lado en la sección de detalles, CR coloca el nombre del campo correspondiente arriba de cada campo en el área del

encabezado de página. Una vez que los campos están en su lugar, se añade el título del reporte, se hacen algunos ajustes para mejorar la apariencia de los nombres de los campos, y se insertan algunos especiales para imprimir los números de página y la fecha actual.

Para colocar texto literal en el reporte simplemente se pone el cursor en donde se desea y se tecléa, o se puede usar el botón Insert Text Field como se describió anteriormente. El texto puede seleccionarse y formatearse mediante el uso de la cinta de formateo. Cuando se oprime Enter dentro de la banda del encabezado, se añade una nueva línea. Esta es una manera sencilla de aumentar el número de líneas disponibles en el encabezado. Se pueden eliminar las líneas que no se deseen seleccionándolas y oprimiendo Delete.

Crystal Reports despliega los encabezados de la columna en un típico formato subrayado. Se pueden seleccionar todos los encabezados a la vez y luego hacer clickk en el botón Underline de la cinta de formateo para eliminar el subrayado. Ahora arrastrando los encabezados de la columna una línea hacia arriba y dejando un espacio debajo de ellos. Mientras los encabezados están seleccionados, se hace click en el botón Bold y se aumenta el tamaño del tipo de letra a díganos 12, para darle más atractivo. El último ajuste a las etiquetas de la columna será trazar un subrayado grueso a través de la página por debajo de ellas. Haciendo click en el botón Draw Line y trazando la línea, luego haciendo click en el botón derecho para ajustar el grosor.

El último paso en el diseño del reporte será añadir campos especiales. Del menú Insert se selecciona Special Field y luego Print Date Field, y se pone el campo en la esquina superior derecha del encabezado de la página. Ahora, de manera similar, se añade un número de página a la banda de pie de página.

En cualquier momento se puede examinar cómo se ve el reporte durante la fase de diseño, imprimiéndolo en la pantalla o en el papel. Haciendo click en el botón Print Preview en la barra de herramientas se examina el reporte. Figura 3.5.2

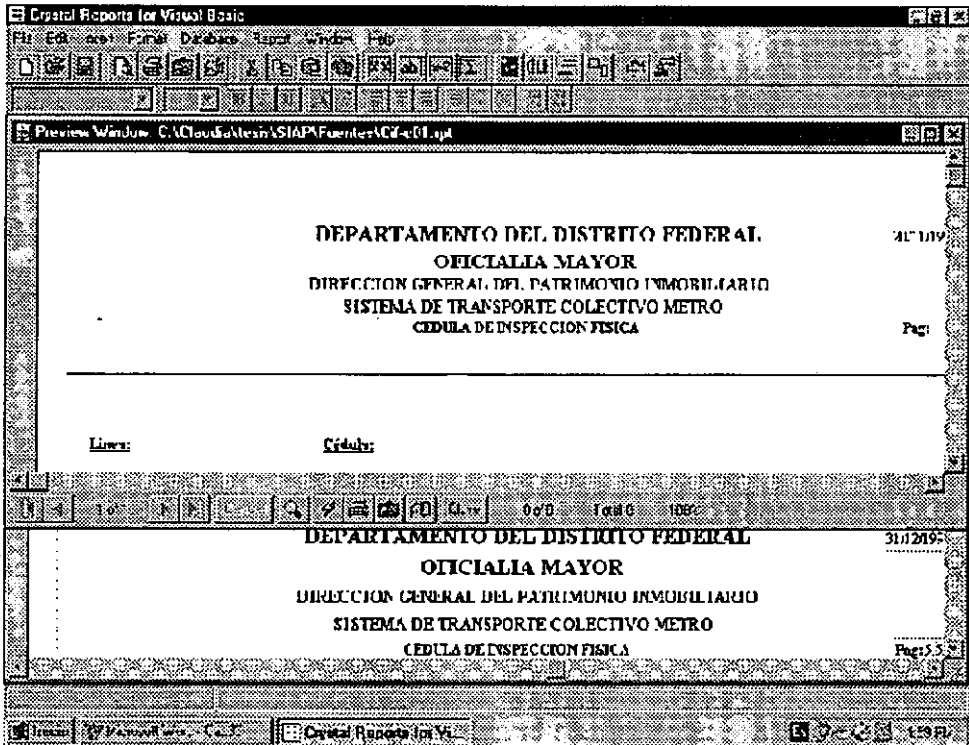


Figura 3.5.2 Impresión Previa del Reporte

Eso es todo en cuanto a la fase de diseño. Se guarda el reporte como CIF-C01, usando la extensión de archivos que se sugiere .RPT., y salimos de Report Designer. A continuación se creará el código para imprimir el reporte en una ventana de reporte personalizado, desde donde el usuario podrá enviarla a la impresora, según lo desee.

ADICION DEL CODIGO

Antes de añadir el código para acceder el reporte, se necesitará añadir el control personalizado Crystal Reports a el formulario **Reportes**. En este caso, aceptará el nombre por omisión del control, SiapReportes. Todas las demás propiedades se establecerán en tiempo de ejecución en el procedimiento Click del botón Report, que se lista y comenta a continuación.

```

Private Sub cmdReport_Click()
Dim V_INCH As Long, H_INCH As Long
H_INCH = 1440 / Screen.TwipsPerPixelX
V_INCH = 1440 / Screen.TwipsPerPixelY
SiapReportes.Destination = 0
SiapReportes.ReportFileName = "C:\SIAP\REPORTES\CIF-C01.RPT"
SiapReportes.WindowTitle = "Reportes"
SiapReportes.WindowTop = 2.25 * V_INCH
SiapReportes.WindowLeft = 0.1 * H_INCH
SiapReportes.WindowHeight = 2.4*V_INCH
SiapReportes.WindowWidth = 6.5*H_INCH
SiapReportes.Action = 1
End Sub

```

'Debido a que el control CR
'espera pixeles y no twips, se
'necesita hacer unas cuantas
'conversiones...

'Divide 1440 twips entre los
'twips por pixel de pantalla
'para obtener el valor de
'pixeles de una pulgada.

'Envía a la ventana de reporte
'Archivo de reporte
'Título de la ventana

'Coloca la ventana a 2.25
'pulgadas de la parte superior
'de la pantalla y aun décimo de
'pulgada de la orilla izquierda.

'Le da tamaño de 2.4 pulgadas de
'alto y 6.5 pulgadas de ancho...

'Pone Action a 1 para imprimir

En la aplicación haciendo click en el botón de la impresora se ejecutará el reporte y la pantalla aparecerá como la Figura 3.5.3

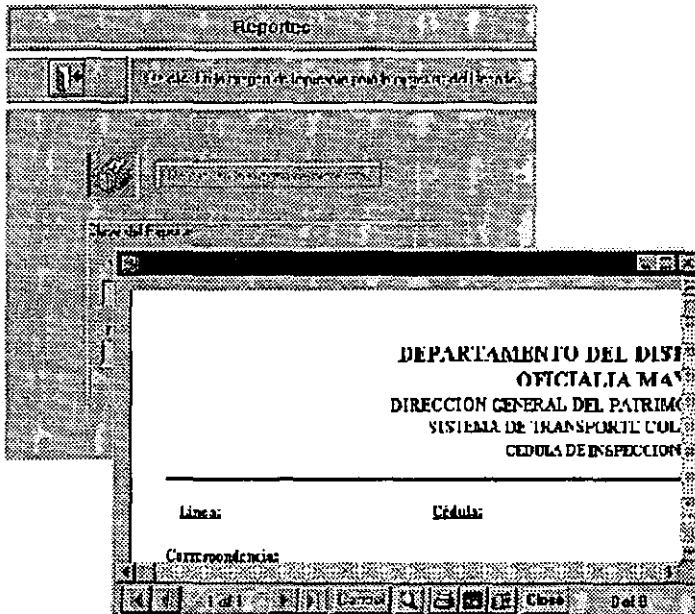


Figura 3.5.5 Módulo Reportes del Sistema SIAP

3.6 Desarrollo De Módulos De Administración Del Sistema.

La administración del sistema, en general, engloba todas aquellas actividades destinadas a mantener funcionando correctamente el sistema (mantenimiento correctivo, preventivo o perfecto); el responsable de dichas actividades es el "administrador del sistema".

Los siguiente puntos conforman la administración del sistema SIAP:

1. Respaldos del disco duro C: a la unidad JAZZ o a un disquette.
2. Restauración del Sistema
3. Control de Usuarios
4. Mantenimiento a catálogos
5. Depuración

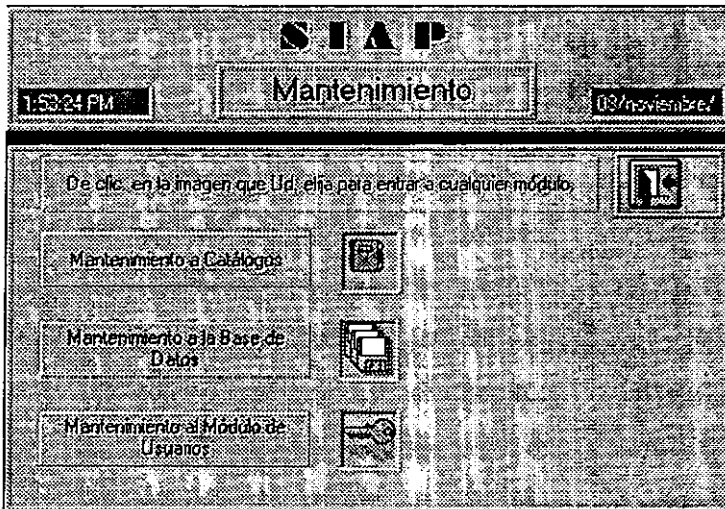


Figura 3.6.a Menú de Mantenimiento del Sistema

RESPALDO DEL SISTEMA

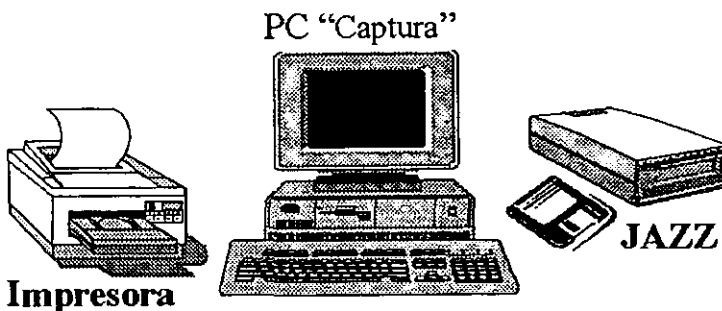
Para realizar el respaldo del sistema y bases de datos, se cuenta con cinco discos JAZZ de 1 GB rotulados de "LUNES" a "VIERNES", correspondientes a cada día (hábil); el respaldo será cíclico y será lanzado a las 10:00 p.m. por el operador de sistemas del centro de cómputo y serán almacenados, bajo llave, en la credenza destinada a contener los diskettes y CD originales del software institucional.

Se sugiere que el día viernes sean realizados dos respaldos adicionales para ser almacenados en ubicaciones diferentes fuera de las instalaciones; dichos respaldos incluirán tanto el sistema como las bases de datos y el archivo actualizado del manual del usuario. Esta medida con la finalidad de hacer frente a diferentes contingencias, como incendio o robo, que pudieran presentarse.

La computadora que se debe respaldar es la PC "Captura"; ésta debe contar con dos discos duros, C:\ y una unidad para discos JAZZ. **Figura 3.6.b**

El procedimiento de respaldo de bases de datos forma parte de nuestro sistema en el menú MANTENIMIENTO opción <respaldo>. El nombre del archivo generado por esta opción será SIAPBD.zip que será asignado automáticamente por el sistema. El disco deberá etiquetarse con la fecha de realización del respaldo, bajo el formato AA=año, MM=mes y DD=día, el prefijo RS=Respaldo de Sistema.

Figura 3.6.b



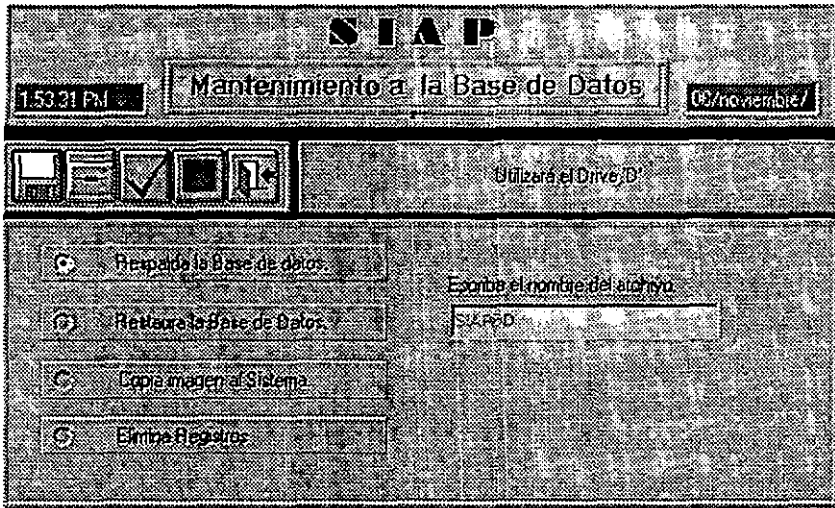


Figura 3.6.c Menú de Mantenimiento a la Base de Datos

RESTAURACIÓN DEL SISTEMA

El verdadero valor de un respaldo se reconoce al presentarse una "caída de sistema"; por ejemplo, corrupción de bases de datos, instalación de software en conflicto con nuestra aplicación, cambios en la configuración de la PC, presencia de virus, poco espacio o fallas en el disco duro y en última instancia daño físico de la computadora; por nombrar algunas situaciones.

"Levantar un sistema caído" es tarea del administrador del centro de cómputo y no del usuario final debido a las múltiples y diversas razones que pudieron causar la caída del sistema; por lo tanto, lo primero es identificar el por qué de la falla para luego restaurarlo. Así que si falla el sistema debemos notificarlo al administrador del centro de cómputo "inmediatamente", quien verificará que los diferentes elementos que conforman la aplicación operan correctamente:

1. La PC funciona: Arranca windows 95 y otras aplicaciones
2. La impresora funciona: imprime de otras aplicaciones.
3. Operación correcta de la unidad JAZZ y discos removibles
4. Existe espacio suficiente en el disco duro D: y no reporta fallas
5. no reporta virus
6. Arranca la paquetería de soporte: Power Point, Photo Enhancer, InfoGuía Roji, Photo Styler, AUTOCAD.
7. Inicia MS-Access correctamente
8. La Aplicación y bases de datos (METRO.MDB) operan normalmente.

En caso necesario habrán de reinstalarse todos y cada uno de los programas anteriores y restaurarse el respaldo inmediato anterior de la aplicación; en última instancia cambiar la PC.

3.6.1 Control De Usuarios

Una de las partes más importantes de cualquier sistema es sin duda el control de los usuarios, ya que de esta forma se garantiza la seguridad de la información, mediante el uso de niveles de seguridad dependiendo del usuario del que se trate.

El sistema **SIAP** para realizar esta función cuenta dentro del módulo de Mantenimiento con la opción Mantenimiento al Módulo de Usuarios (Figura 3.6.1.a)

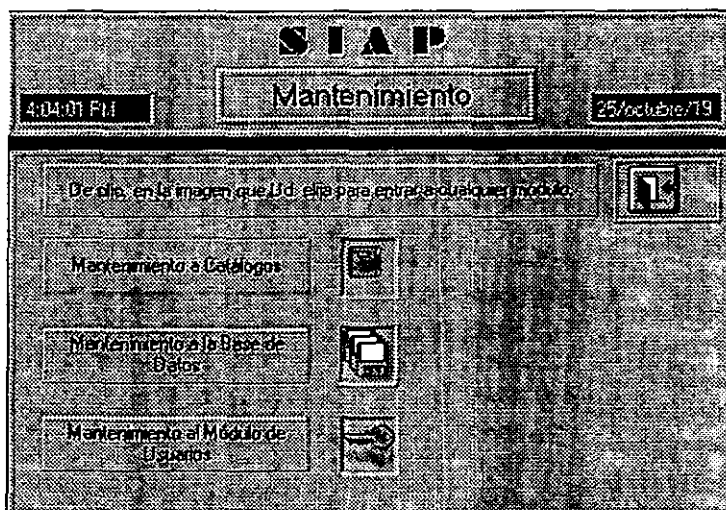


Figura 3.6.1.a Pantalla de Módulo de Mantenimiento

En el sistema se manejan dos niveles de usuarios, Nivel A para usuarios maestros y Nivel B para usuarios comunes. :

El **Nivel A** es para el usuario maestro el cual puede efectuar captura, modificación e incluso eliminación de información; además se encarga de la administración de los usuarios mediante altas y bajas de usuarios.

Los usuarios de Nivel B únicamente pueden realizar consultas al sistema y generar reportes, el resto de las opciones no le son habilitadas.

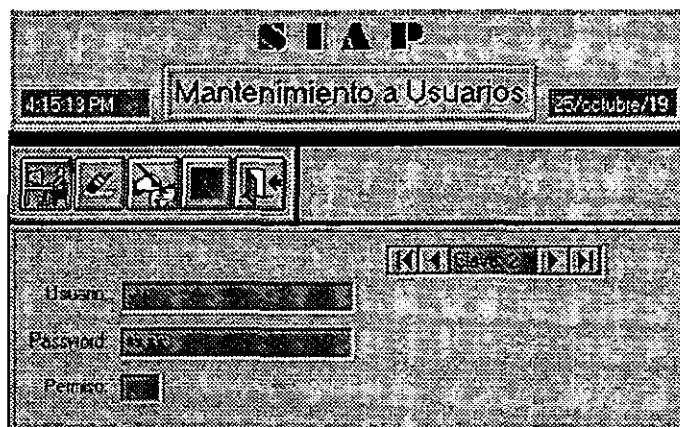
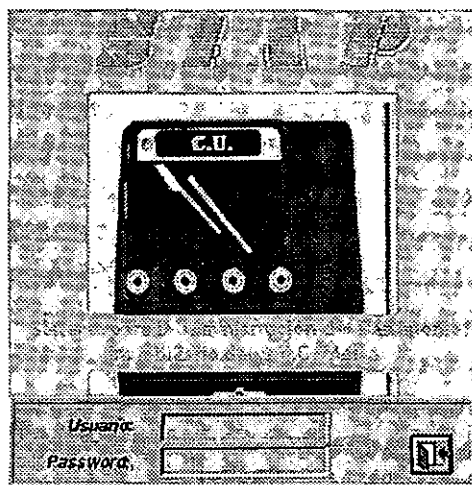


Figura 3.6.1 b Módulo de Mantenimiento a Usuarios

En el módulo de Manteminiento a Usuarios se tienen las operaciones de captura, eliminación y modificación de usuarios, este menú solo puede ser accesado por usuarios maestros o de nivel A (Figura 3.6.1 b). Solamente los usuarios autorizados en este módulo puede tener acceso al Sistema SIAP.



3.6.2 Mantenimiento A Catalogos

Todo sistema que se precie de serlo y que maneje información importante del organismo en el que se implemente, tendrá la tendencia al crecimiento. A pesar de que en nuestro caso el crecimiento de los catálogos será muy extraordinario, solo en el caso en que se creen nuevas líneas en el SCT Metro, aún así se deben contemplar procesos de mantenimiento y depuración para que su funcionamiento sea óptimo y con esto alcanzar los objetivos trazados al inicio de la planeación del sistema.

A continuación se explican los procesos de mantenimiento y depuración que tienen que ser llevados a cabo para que el sistema funcione dentro de los parámetros convenidos:

MANTENIMIENTO

A las tablas a las que se les aplicará un proceso de mantenimiento son las siguientes:

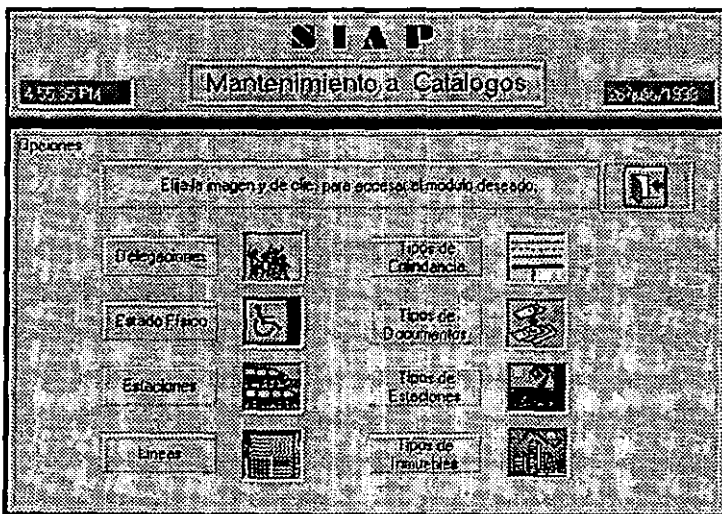


Figura 3.6.2.1 Módulo de Mantenimiento a Catálogos

A continuación se describirán algunas de estas opciones, ya que todas funcionan bajo el mismo esquema.

- LINEAS

A la tabla Línea se le aplicará un proceso de mantenimiento cada que a la red del metro se le agregue otra línea, o se cambie el nombre de alguna y de ser necesario borrar alguna.

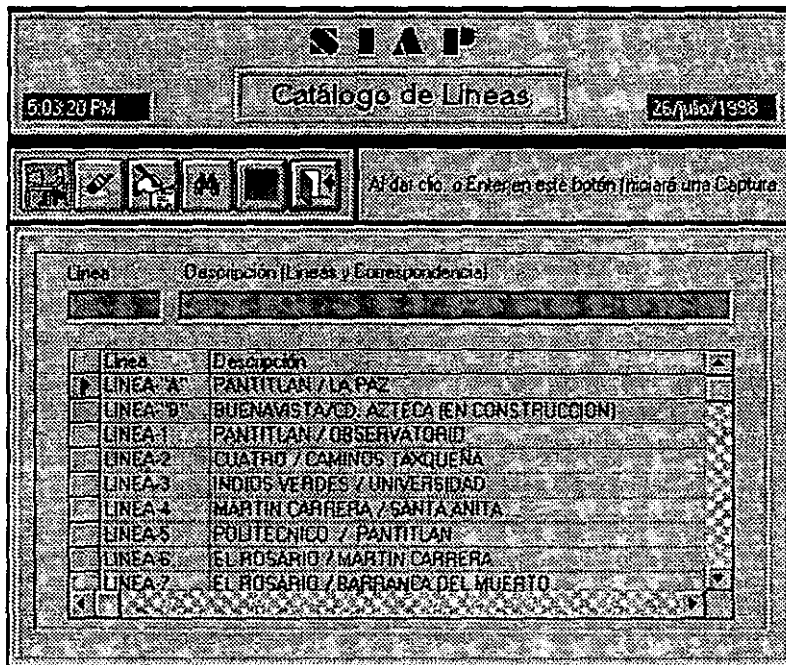


Figura 3.6.2.2 Pantalla de Catálogo de Líneas

Para acceder al catálogo basta con elegir la opción LINEA del menú , en la ventana desplegada basta elegir cualquiera de los catálogos, al dar click, se desplegarán las opciones que permitirán agregar, consultar, borrar y actualizar una LINEA, también desplegará el contenido de los registros que se encuentren al

inicio de la tabla, al introducir o modificar información se validará que no se introduzcan datos inválidos como duplicidad. Con las opciones habilitadas la ventana luce de la siguiente manera:

- ESTACIONES

A esta tabla se le agregarán registros cada vez que haya una nueva Línea del Metro, y se agregarán tantos registros como estaciones tenga la línea. Esta opción se activa desde el Menú de MANTENIMIENTO A CATALOGOS.

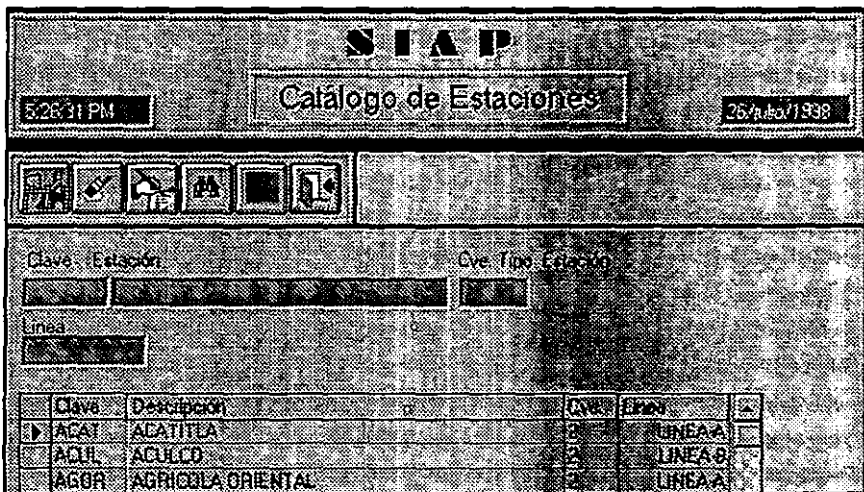


Figura 3.6.2.3 Pantalla de Catálogo de Estaciones

Una vez que se introduce la estación se valida que no este duplicada y además restringe la entrada de datos que no correspondan.

- TIPO DE ESTACION.

El mantenimiento a este catálogo es posible que ni siquiera se llegué a realizar, pues se refiere a adicionar un tipo de estación como de paso, transborde o

terminal, etcétera, sin embargo como se mencionó antes se deben contemplar todos los casos.

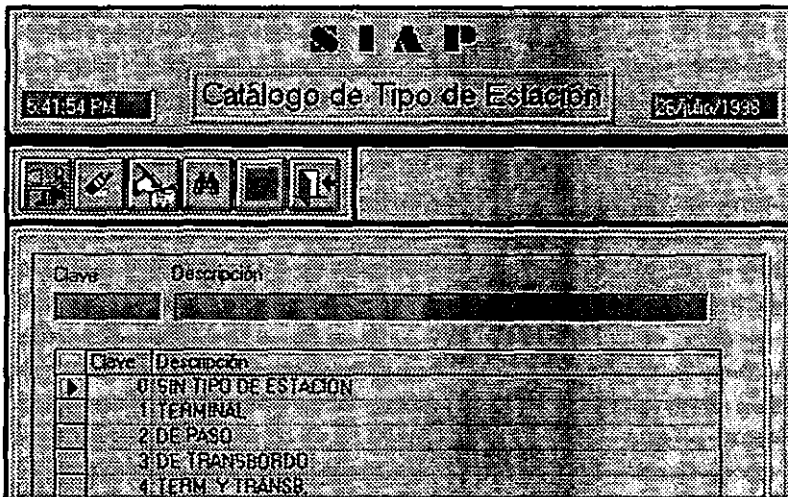


Figura 3.6.2.4 Pantalla de Catálogo de Tipo de Estación

TIPO DE INMUEBLE

El mantenimiento a esta tabla se deberá a que exista un tipo de inmueble con alguna característica muy en particular.

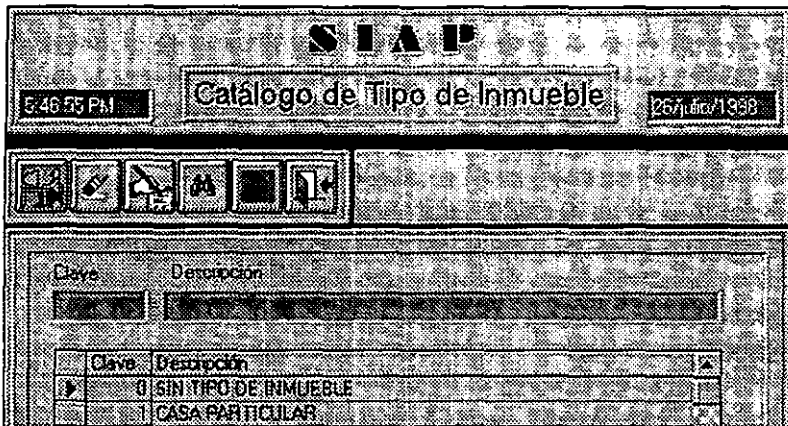


Figura 3.6.2.5 Catálogo de Tipo de Inmueble

De igual forma que en las opciones anteriores se validará que no se introduzcan datos duplicados, conservando así la integridad de los datos.

- ESTADO FISICO



Figura 3.6.2.6 Pantalla de Catálogo de Estado Físico

El mantenimiento de esta tabla se llevará a cabo por circunstancias iguales a las de la tabla anterior, tendría que ser para describir un estado físico particular de algún inmueble.

- DELEGACIONES

El mantenimiento a este catálogo será en un caso aún más extraordinario que los anteriores, sin embargo como todo catálogo cuenta con su utilería de mantenimiento.

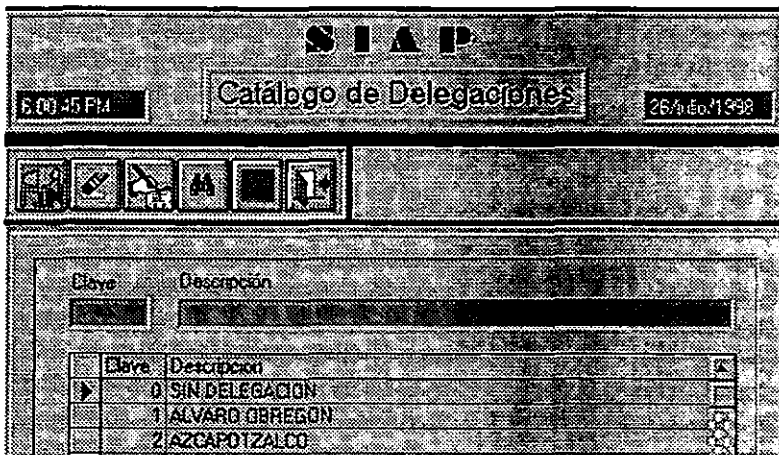


Figura 3.6.2.7 Pantalla de Catálogo de Delegaciones

3.6.3 Depuración Del Sistema

Todo sistema que se precie de serlo y que maneje información trascendental de la institución en la que se implemente, tendrá la tendencia al crecimiento, por lo que se tiene que sujetar a diversos procesos de mantenimiento y depuración para que su funcionamiento sea de niveles de excelencia y con esto alcanzar todos o la mayoría de los objetivos y metas que fueron trazadas cuando se comenzó la etapa de planeación del sistema.

A continuación se explican los procesos de mantenimiento y depuración que tienen que ser llevados a cabo para que el sistema funcione dentro de los parámetros convenidos:

Depuración de la Información

Captura

Al usuario se le presento un manejo de pantallas de captura en donde se describía visualmente cada parte del expediente que se integraría, es decir como primer icono se encuentra la definición del tipo de expediente así como la ubicación de este, el segundo icono es la ubicación del predio, el tercer icono era la cédula de inspección física, el cuarto icono es el plano de localización, el quinto icono es la presentación del croquis de localización y como sexta ícono es la fotografía del inmueble, como referencia principal todas las pantallas se manejan en forma independiente por petición del usuario

Se agregaron a estas pantallas botones para realizar búsquedas, altas, bajas y cambios de los registros, de impresión así como la visualización de cada uno de los informes que integran las cédulas.

Se agregaron reglas de validación a los campos para evitar que se repitieran datos por ejemplo el número de cédula.

Informes

Se tuvo que depurar constantemente el orden de los datos.

Se modifico la forma de impresión de los reportes ya que este era por ambos lados. El usuario pidió que la visualización de la información fuera por solo una cara.

En algunos campos el tamaño de la letra es reducida por la longitud de los datos que se captura.

Depuración del Archivo

Por el tipo de información que se maneja en el sistema no se pueden eliminar los registros de los predios por al menos un año, a petición del usuario; ya que este requiere realizar consultas en línea constantemente.

La información generada en la base de datos durante un año, se respaldará en disquetes o unidades de cinta.

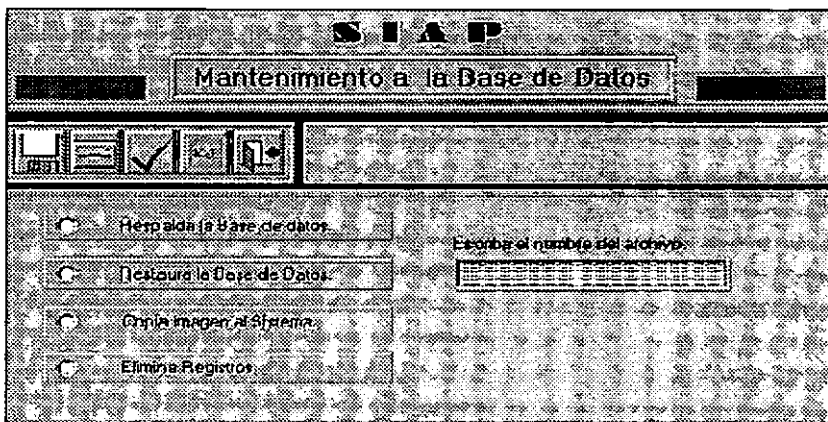


Figura 3.6.3.1 Pantalla de Mantenimiento a la Base de Datos

Respaldo y Restauración del Archivo

Una vez que ha finalizado el lapso de tiempo indicado por el usuario se podrán realizar respaldos en unidades de cintas o disquetes.

Cuando el usuario crea conveniente utilizar la información respaldada en caso de algun siniestro, este podra lograrlo mediante la opción "Restaurar Base de Datos".

3.7 Pruebas E Integracion

PRUEBAS PARA LOS MODULOS DEL SISTEMA DE ADMINISTRACION DEL PATRIMONIO INMOBILIARIO DEL METRO.

El sistema para la Administración del Patrinomio Inmobiliario del Metro está compuesto por los módulos.

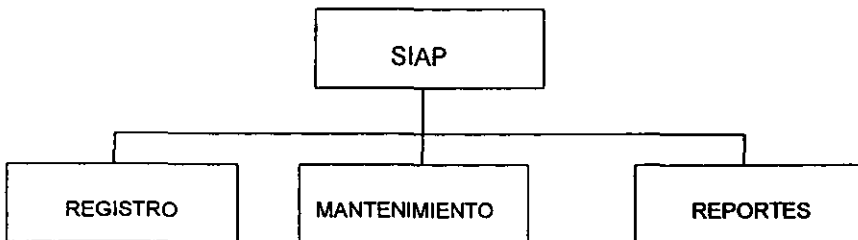


Figura 3.6.3 Modulos de SIAP

El módulo central o principal permite tener acceso al sistema mediante claves secretas de acceso (passwords) y una vez dentro, el usuario puede llamar a los distintos módulos que componen el sistema.

Se probó que cada uno de los menús de opciones de las pantallas de captura cumplieran con los siguientes puntos:

Ortografía Correcta

Cerciorándonos de que todos los mensajes y pantallas cumplen con una ortografía correcta.

Nombre del Módulo

Siempre aparece en la parte superior en el centro de todas las pantallas.

Opciones del Menú

Se muestran todas las opciones que comprenden al menú seleccionado.

Descripción del ícono seleccionado

Cuando el cursor esté ubicado un un ícono, aparecerá una descripción justo al lado del ícono que nos indica su función (Ayuda en Línea).

Nombre de la opción del Menú

Cuando se selecciona una opción del menú, se mantiene el nombre de esta en la parte superior de la pantalla y centrado, con el fin de saber en todo momento la opción en que nos encontramos.

Fecha del Sistema

En la parte superior derecha, de cada pantalla se despliega la fecha actual del sistema.

Uso del mouse

Se hace uso del mouse oprimiendo el botón izquierdo del mouse para activar un ícono, o para ubicarse en un campo en específico para introducir información.

PRUEBAS DEL MÓDULO DE CAPTURA

Este módulo está formado por los submódulos que se muestran en la Figura 3.6.3d

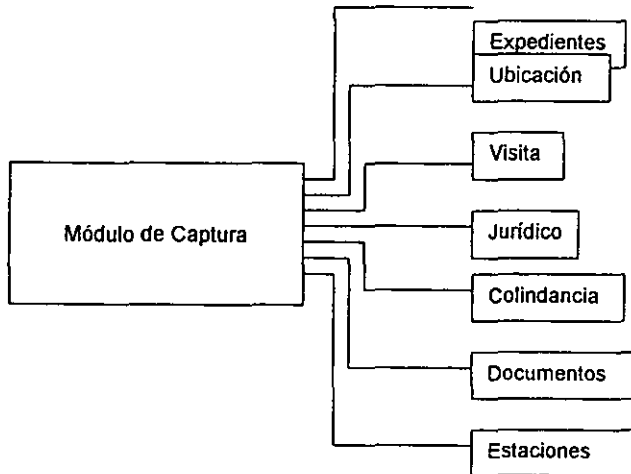


Figura 3.6.3e Pruebas del Módulo de Captura.

PRUEBAS DEL SUBMÓDULO CAPTURA DE EXPEDIENTES.

En este submódulo se podrán dar de alta el expediente de cada cédula y los documentos que puedan integrar ese expediente, como pueden ser acuerdos, decretos, contratos, etc. estos pueden ser agregados posteriormente. Los campos en los que se aplicaron las pruebas son los siguientes :

Cédula

Esta deberá componerse por letras mayúsculas, y se deberá validar su existencia en el caso en el que se deseen agregar documentos nuevos a los expedientes.

Así como se debe validar su no existencia al agregar una cédula nueva. Además el sistema asegura que la cédula no tenga un valor nulo en cualquier operación, ya que esta es la llave primaria en la mayoría de las tablas del sistema, este valor puede ser hasta de 10 caracteres.

Documento

Este campo sirve para insertar una detalle adicional al tipo de documento que se está agregando, como puede ser en el caso de que el tipo de documento sea un contrato este campo puede indicar el número, debido a lo variable de la información que se puede introducir el campo acepta todo tipo de caracteres hasta una longitud de 10.

Clave

Este campo sirve para elegir el tipo de documento de que se trate, los valores que se muestran son los contenidos en el catálogo de tipos de documentos, para garantizar que se introduce un documento válido, este clave es numérica.

Tipo de Documento

Este campo despliega la descripción que corresponda al tipo de documento que se haya elegido.

PRUEBAS DEL SUBMODULO UBICACION

En este módulo se captura toda la información que corresponda a la ubicación física del predio del que se trate. Los campos validados son :

Cédula

Al proporcionar el número de cédula el sistema valida si existe, en caso de que así sea desplegará los datos correspondientes, en caso contrario permitirá la captura de los datos para ese predio, el sistema asegura que este campo no se deje en blanco.

Clave de la Estación

Este campo desplegará las claves de estación válidas, las cuales son las que fueron dadas de alta en el catálogo de estaciones (sc_estaciones).

Calle

Se podrá capturar o modificar la calle del predio. Tiene longitud de 70.

Número

Se podrá capturar el número exterior de la calle del predio

No Interior, Lote y Manzana

En caso de que exista un número interior, lote o manzana para el predio se podrán capturar en estos campos.

Colonia

Se podrá capturar o modificar la colonia del predio seleccionado. Tiene como longitud máxima 70 caracteres.

Entre Calles

En caso de que se tenga información adicional de las calles en que se encuentra el predio podrá capturarse aquí. Tiene como longitud máxima 70 caracteres.

Cve. Delegación

Aquí se despliega una lista de las delegaciones o municipios válidos, y basta con seleccionarla.

Código Postal

Se podrá capturar o modificar el código postal al cual pertenece el predio seleccionado. Tiene longitud máxima de 5 dígitos.

Cta Anterior y Cuenta Actual

En caso de existir una cuenta catastral anterior y por ello una cuenta actual se podrá capturar en estos campos, en caso de ser la misma se debe capturar la misma. Tiene longitud máxima de 20 caracteres.

Fecha de Captura

Para esta clase de información es necesario contar con la fecha de captura, en este campo se valida que la fecha introducida sea una fecha válida.

PRUEBAS AL SOFTWARE SISTEMA SIAP

Las estrategias de pruebas de Software integran técnicas de diseño de casos de prueba en pasos bien planificados con la finalidad de construir correctamente algún software, estas estrategias de prueba deben ser planeadas para diseñar casos de prueba, recolectar y evaluar los resultados, todas estas tienen las siguientes características generales :

- La prueba comienza en el nivel del módulo y trabaja "hacia fuera" hacia la integración de todo el sistema.
- En diferentes etapas se utilizan distintas técnicas de prueba.
- La prueba la lleva a cabo el que desarrolla el software y a veces se hace necesario un grupo de prueba independiente (sobre todo en proyectos grandes)
- La prueba y la depuración son actividades diferentes, pero la depuración puede entrar en cualquier estrategia de prueba.

La gran mayoría de las veces la teoría referente a las pruebas de software es encontrada como verificación y validación.

- La verificación es el conjunto de actividades que aseguran que el software implementa correctamente una función específica.
- La validación es un conjunto de actividades que aseguran que el software construido se ajusta a los requerimientos del usuario(s).

La o las personas que desarrollan el software son responsables de probar las unidades individuales (módulos) del programa, asegurándonos de que cada una lleva a cabo la función para la que fue diseñada. En muchos casos también se encargará de la prueba integración, que lleva a la construcción y prueba de la estructura total del sistema.

Una vez que la arquitectura del software está completa entra en juego un grupo independiente de prueba. El papel del grupo independiente de prueba es eliminar los problemas inherentes asociados con el hecho de permitir al constructor que pruebe lo que ha construido. Una prueba independiente elimina el conflicto de intereses que de otro modo estará presente.

Las pruebas en el contexto de la ingeniería de Software consiste en cuatro pasos :

- Prueba de Unidad
- Prueba de Integración
- Prueba de validación
- Prueba del sistema

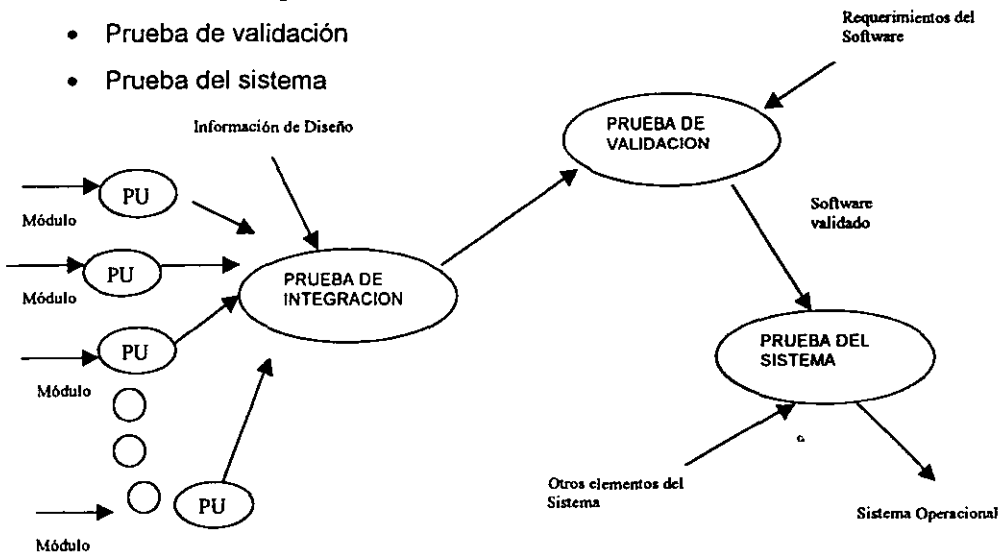


Figura 3.6.3.f Pruebas de Unidad

Prueba de Unidad

La prueba de unidad se centra en la menor unidad del diseño del software, es decir el módulo en las implementaciones en código fuente. Realiza un uso intensivo de ejercicios de caminos específicos de la estructura de control del módulo para asegurar un alcance completo y una detección máxima de errores del módulo.

Esta prueba se llevo a cabo en paralelo con otras pruebas del sistema, es decir mientras unos verificaban los módulos de registro y control , otros probaban los de mantenimiento.

Posteriormente se integraron los módulos para formar el sistema de Administración del Patrimonio Inmobiliario del STC Metro.

La Prueba de Integración

Para esta prueba las actividades se centraron en el diseño y construcción de la arquitectura del software. Se verifica y se construye el programa como un todo.

Durante la integración, las técnicas que más prevalecen son las de diseño de casos de prueba de la caja negra, aunque se pueden llevar a cabo algunas pruebas de la caja blanca con el fin de asegurar que se cubran los principales caminos de control.

Pruebas de caja negra

Es una prueba que asegura que todas las "piezas encajan" o sea, que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada, conociendo el funcionamiento del producto. Se llevan a cabo sobre la interface del software. Se pretende demostrar que las funciones del software son operativas, que la entrada se acepta en forma adecuada y que se

produce una salida correcta, así como la integridad de la información externa se mantiene. Se examinan algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software.

Prueba de la caja blanca

Demuestra que cada función es completamente operativa, conociendo la función específica para la que fue diseñada el producto. Se basa en el minucioso examen de los detalles procedurales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejercitan conjuntos específicos de condiciones y/o bucles.

Prueba de integración del Sistema para la Administración del Patrimonio Inmobiliario del STC Metro.

Se realizó esta técnica para construir la estructura del programa y para realizar pruebas para detectar errores asociados con la interacción entre los módulos. Se tomaron los módulos aprobados en unidad y nos aseguramos que la estructura del programa estuviera de acuerdo con lo que dictaba el diseño.

Se realizó una integración incremental en la cual el programa se construye y se prueban los módulos en los que los errores son más fáciles de aislar y de corregir con la finalidad de probar completamente todas las interfaces y aplicar una aproximación de prueba sistemática.

Existen dos estrategias de integración incremental:

- Integración descendente
- Integración ascendente

Empleamos la integración descendente de forma primero-en-profundidad se integraron los módulos moviéndolos por la jerarquía de control, comenzando con el módulo de control principal. Los módulos subordinados al módulo de control principal se van incorporando en la estructura de forma primero-en-profundidad.

Refiriéndonos a la siguiente figura, se eligió el camino a mano izquierda, se integraron los módulos M1, M2 y R1. A continuación será integrado M3. Acto seguido se construyen los caminos de control central y derecho.

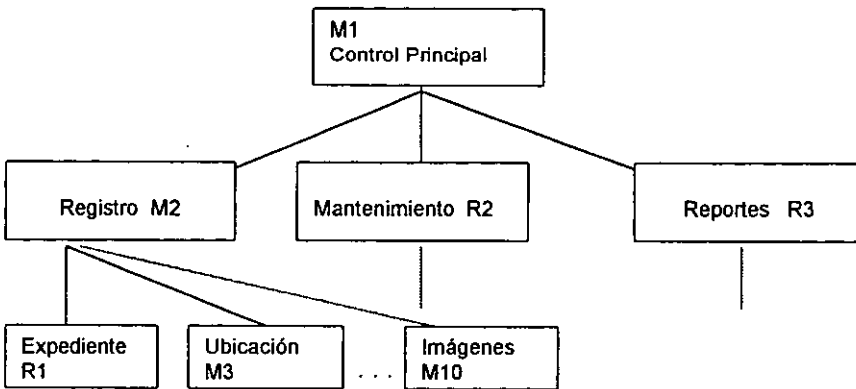


Figura 3.6.3g Prueba de Integración

Llevamos a cabo el proceso de integración en cinco pasos.

Se uso el módulo principal como conductor de la prueba disponiendo de los resguardos para todos los módulos directamente subordinados al módulo principal (los resguardos sirven para reemplazar módulos que están subordinados al módulo a ser probado un resguardo o "subprograma mudo" usa la interface del módulo subordinado, lleva a cabo la mínima manipulación de datos e imprime una verificación de la entrada y vuelve).

1. Se usa el módulo de control principal como conductor de la prueba, disponiendo resguardos para todos los módulos directamente subordinados al módulo de control principal.
2. Se fueron sustituyendo los resguardos subordinados uno a uno por los módulos reales.
3. Se llevaron a cabo pruebas de caja negra cada vez que integramos un nuevo módulo.
4. Después de las pruebas se reemplazo otro resguardo con el módulo real.
5. Realizamos la prueba de regresión , es decir, todas las pruebas anteriores para asegurarnos de que no se haya introducido nuevos errores.

El proceso continua desde el paso 2 hasta construir la estructura del programa entero.

En cada reemplazamiento se llevaron a cabo pruebas para verificar la interface. La principal desventaja de la aproximación descendente es la necesidad de resguardos y las dificultades de prueba que pueden estar asociadas con ellos.

ESPECIFICACION DE LA PRUEBA DE INTEGRACION

Alcance de la prueba

Se probaron características tales como:

- Fácil uso por parte del usuario
- Desplegado de ayuda
- El fácil acceso utilizando los dispositivos de entrada: el mouse y teclado
- Además que el número de módulos complieran con las especificaciones solicitadas.

Plan de Prueba

La estrategia general para la integración se dividió en dos fases y subfases, dirigidas a características específicas funcionales del software.

Interacción con el usuario

- selección de órdenes
- representación visual
- procesamiento y representación de errores

Manipulación y análisis de datos

Procesamiento y generación de información visual

- Reportes

Estructura y contenido de la base de datos

En cada fase se siguieron los criterios con sus correspondientes pruebas

- Integridad de la interface. Se probaron las interfaces internas y externas a medida que se incorporaron los módulos.
- Validez funcional. Se llevaron a cabo pruebas diseñadas para descubrir errores funcionales.
- Contención de información. Pruebas para descubrir errores asociados con las estructuras globales y locales.
- Rendimiento. Pruebas para verificar los límites de rendimiento establecidos durante el diseño del software.

Los recursos empleados fueron una PC Pentium MMX, con 2 GB en disco duro y 32 MB de Ram.

Procedimiento de prueba

Se describe el orden de integración y las pruebas correspondientes a cada fase de integración.

LA PRUEBA DE VALIDACION DEL SISTEMA SIAP

En esta prueba de validación se deben comprobar los criterios de validación establecidos durante la fase de definición del sistema. Proporciona una seguridad final de que el software satisface todos los requerimientos funcionales y de rendimiento. Se usan exclusivamente técnicas de prueba de la caja negra.

Después de encontrar y corregir los errores de interfaces, comenzamos la prueba de validación. Con el fin de descubrir errores cuando el usuario utiliza el sistema, se llevaron a cabo las pruebas alfa y beta.

- **PRUEBA ALFA**

En el lugar de desarrollo se invitó a los usuarios a probar el sistema, en presencia de los desarrolladores.

- **PRUEBA BETA**

La prueba se realizó con los usuarios en el lugar físico donde estaría funcionando el Sistema, ningún desarrollador estuvo presente.

Prueba alfa

Para llevar a cabo la prueba alfa se invito a un usuario en el lugar del desarrollo y en presencia del desarrollador, este registro errores y problemas de uso. Es decir, todo esto se llevó en un entorno controlado, al principio al utilizaar el sistema, el operador tenía cierto temor y no se presentaba seguro. Fueron modificados algunos de los mensajes que proporcionaba el sistema porque les faltaba claridad.

Las principales deficiencias en esta prueba fueron la falta de familiaridad en el uso de menús y el entender el procedimiento para manipular los datos.

Prueba Beta

En la prueba beta se llevó a cabo en el lugar donde estaría en funcionamiento el sistema, el encargado del desarrollo no estuvo presente.

Sobre todo se presentaron problemas tales como la necesidad de una explicación sencilla de los errores y mensajes y además dudas en el instructivo.

PRUEBAS DEL SISTEMA SIAP

Entre los tipos de prueba del sistema que llevamos a cabo fueron :

Prueba de recuperación

Con el fin de probar su tolerancia a fallos y el periodo de corrección de fallas

- Interrupciones de energía eléctrica
- Otras eventualidades

Por lo anterior forzamos el fallo del software mediante.

- Fallas de energía eléctrica

En esta la transacción en proceso era la que perdía y con la implementación de las utilerías diseñadas para estos casos, se mantenía la integridad del resto de la información. La recuperación requiere de la intervención humana se evaluaron los tiempos medios de recuperación que no excedieron de 1 minuto y sólo se perdía el registro que estaba siendo actualizado en ese momento determinándose que se encontraba dentro de los límites aceptables.

Pruebas de seguridad

El mecanismo que evita que una persona que no tenga acceso al sistema pueda modificar cierta información, se evita con el uso de passwords, inclusive de diferentes niveles de acceso.

Tienen acceso a todas las opciones del menú principal.

Al teclear alguno de los passwords, se desplegará el menú correspondiente a cada uno de estos y las opciones a las cuales el usuario tendrá acceso.

El campo de passwords permite la combinación de cualquier carácter ASCII lo que aumenta la combinación de nombres y por lo tanto la seguridad.

Pruebas de resistencia y de rendimiento

Entre las pruebas que se sometieron fueron las siguientes :

1. Se probó el tiempo de respuesta del sistema para los casos que requerían diferentes tamaños en la tabla de catálogos, desde un registro, pasando por 7000 registros y hasta 20000
2. Se probó casos que produjeran excesivas búsquedas de datos residentes en disco, por medio de localizar a un registro al mismo tiempo por 3 operadores. (Esto para el caso en el que en un futuro el usuario se decida a tener el sistema en red).

Operación del sistema

Después de todo lo anterior se dio por terminado el sistema para su operación en tiempo real, en la que se observaron que la inversión de los recursos humanos y económicos se vieron reflejados en la mejora del tiempo de respuesta en la obtención de resultados de consultas y reportes.

CONCLUSIONES

CONCLUSIONES

A través del proceso de desarrollo del sistema y en acorde al objetivo planteado podemos establecer lo siguiente:

- La implementación del sistema causará un buen impacto en los usuarios, ya que el objetivo del sistema finalmente es almacenar, recopilar y organizar de una manera rápida y eficiente toda la información relacionada con los predios.
 - El software requerido para el desarrollo del sistema fue suficiente para su implementación, cumpliendo con la expectativa de ser amigable y de facilitar su manejo.
 - Con el avance tecnológico que existe actualmente en el mercado dispositivos como el scanner, la cámara fotográfica digital y los grabadores de CD-ROM permiten el almacenamiento masivo de información en medios magnéticos dando como resultado la optimización en el manejo de la información, además de ahorrar espacios significativos.
 - Gracias a la creación del sistema se espera que los documentos originales no se deterioren con el manejo constante, ya que cualquier consulta podrá realizarse directamente en el sistema.
-

BIBLIOGRAFÍA

BIBLIOGRAFIA

- Acceso Rápido a Visual Basic
Stefan Dittrich
Computec Editores, S.A. de C.V.
1993.
 - Access Fácil
Paul McFedries
Prentice-Hall Hispanoamericana, S.A.
1995.
 - Introducción a la Programación de Visual Basic
Kenyon Brown
Grupo Noriega Editores.
 - Organización de las Bases de Datos
James Martin
Prentice -Hall Hispanoamericana, S.A.
 - Sistema de Administración de Bancos de Datos
Alfonso F. Cárdenas
Limusa.
 - Introducción a los sistemas operativos
Harvey M. Deitel
Addison-Wesley Iberoamericana.
 - Automatización de la oficina de Servicios Escolares de la
Facultad de Ingeniería por medio de PC's
Tesis
 - Diseño e Implementación de una Guía Gráfica de Ciudad Universitaria
Tesis
 - Diseño e Implementación de Administración para el Centro de Cálculo del
Instituto Tecnológico de Estudios Superiores de Monterrey.
Campus Ciudad de México.
Tesis
 - Implementación de un Sistema de Prestadores de Bienes y Servicios en
Ixtapa Zihuatanejo.
Tesis.
-

- AutoCad Avanzado V. 13 para Windows y MS-DOS
J. López / J. A. Tajadura
Editorial McGraw-Hill

 - AutoCad 13 (referencia rápida)
José Luis Cogollor
Editorial Computec RA-MA

 - Manual del Usuario de Power Point
Editorial Microsoft
-

APÉNDICE A
CÓDIGO DE PROGRAMACIÓN

Agregal - 1

```
Private Sub Cmdsalir_Click()
```

```
End Sub
```

```
Private Sub Combo1_Click()
```

```
    If Combo1.ListIndex = 0 Then
```

```
        File1.Pattern = "*.bmp"
```

```
    Else
```

```
        File1.Pattern = "*.dib"
```

```
    End If
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
    If (Right(Dir1.Path, 1) = "\") Then
```

```
        If Fot1.SSCheck1 = True Then
```

```
            Fot1.Picture1.Visible = True
```

```
            Fot1.Data1.Recordset.AddNew
```

```
            Fot1.Text2.Text = Dir1.Path & File1.filename
```

```
            Fot1.Command2.Visible = True
```

```
        ElseIf Fot1.SSCheck2 = True Then
```

```
            Fot1.Picture1.Visible = True
```

```
            Fot1.Data2.Recordset.AddNew
```

```
            Fot1.Text4.Text = Dir1.Path & File1.filename
```

```
            Fot1.Command2.Visible = True
```

```
        ElseIf Fot1.SSCheck3 = True Then
```

```
            Fot1.Picture1.Visible = True
```

```
            Fot1.Data3.Recordset.AddNew
```

```
            Fot1.Text6.Text = Dir1.Path & File1.filename
```

```
            Fot1.Command2.Visible = True
```

```
        Else
```

```
            Fot1.Picture1.Visible = True
```

```
            Fot1.Data4.Recordset.AddNew
```

```
            Fot1.Text10.Text = Dir1.Path & File1.filename
```

```
            Fot1.Command2.Visible = True
```

```
        End If
```

```
    Else
```

```
        If Fot1.SSCheck1 = True Then
```

```
            Fot1.Data1.Refresh
```

```
            Fot1.Picture1.Visible = True
```

```
            Fot1.Data1.Recordset.AddNew
```

```
            Fot1.Text2.Text = Dir1.Path & "\" & File1.filename
```

```
            Fot1.Command2.Visible = True
```

```
            Fot1.Picture1.Visible = True
```

```
            Fot1.SSPanel8.Caption = "Introduzca la clave de la Cédula."
```

```
            Unload Me
```

```
            Fot1.X = ""
```

```
            Fot1.Text1.SetFocus
```

```
        ElseIf Fot1.SSCheck2 = True Then
```

```
            Fot1.Data2.Refresh
```

```
            Fot1.Picture1.Visible = True
```

```
            Fot1.Data2.Recordset.AddNew
```

```
            Fot1.Text4.Text = Dir1.Path & "\" & File1.filename
```

```
            Fot1.Command2.Visible = True
```

```
            Fot1.Picture1.Visible = True
```

```
            Fot1.Text8.Visible = True
```

```
            Fot1.Text9.Visible = True
```

```
            Fot1.Combo1.Visible = True
```

```
            Fot1.Text1.Visible = False
```

```
            Fot1.Text3.Visible = True
```

```
            Fot1.Text5.Visible = False
```

```
            Fot1.Text7.Visible = False
```

```
            Fot1.Data1.Visible = False
```

```
            Fot1.Data2.Visible = True
```

```
            Fot1.Data3.Visible = False
```

```
            Fot1.Data4.Visible = False
```

Agregal - 2

```
        Fot1.SSPanel8.Caption = "Introduzca la clave de la Cédula."
        Unload Me
        Fot1.X = ""
        Fot1.Text3.SetFocus
    ElseIf Fot1.SSCheck3 = True Then
        Fot1.Data3.Refresh
        Fot1.Picture1.Visible = True
        Fot1.Data3.Recordset.AddNew
        Fot1.Text6.Text = Dir1.Path & "\" & File1.filename
        Fot1.Command2.Visible = True
        Picture1.Visible = True

        Fot1.Text1.Visible = False
        Fot1.Text3.Visible = False
        Fot1.Text5.Visible = True
        Fot1.Text7.Visible = False

        Fot1.Data1.Visible = False
        Fot1.Data2.Visible = False
        Fot1.Data3.Visible = True
        Fot1.Data4.Visible = False
        Fot1.SSPanel8.Caption = "Introduzca la clave de la Cédula."
        Unload Me
        Fot1.X = ""
        Fot1.Text5.SetFocus
    Else
        Fot1.Picture1.Visible = True
        Fot1.Data4.Refresh
        Fot1.Data4.Recordset.AddNew
        Fot1.Text10.Text = Dir1.Path & "\" & File1.filename
        Fot1.Command2.Visible = True
        Fot1.Picture1.Visible = True

        Fot1.Text1.Visible = False
        Fot1.Text3.Visible = False
        Fot1.Text5.Visible = False
        Fot1.Text7.Visible = True

        Fot1.Data1.Visible = False
        Fot1.Data2.Visible = False
        Fot1.Data3.Visible = False
        Fot1.Data4.Visible = True
        Fot1.SSPanel8.Caption = "Introduzca la clave de la Cédula."
        Unload Me
        Fot1.X = ""
        Fot1.Text7.SetFocus
    End If
End If

End Sub

Private Sub Command1_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
SSPanel2.Caption = "Al dar clic. Aceptará la imagen"
End Sub

Private Sub Command2_Click()
    Fot1.Reinicia
    Unload Me
End Sub

Private Sub Command2_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
SSPanel2.Caption = "Cancela la operación"
End Sub
```

Agregal - 3

```
Private Sub Dir1_Change()  
    File1.Path = Dir1.Path  
End Sub
```

```
Private Sub Dir1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)  
SSPanel2.Caption = "Elija el directorio."  
End Sub
```

```
Private Sub Drive1_Change()  
    Dir1.Path = Drive1.Drive  
End Sub
```

```
Private Sub File1_Click()  
    If (Right(Dir1.Path, 1) = "\") Then  
        'Carga la imagen  
        Picture1.Picture = LoadPicture(Dir1.Path & File1.filename)  
    Else  
        Picture1.Picture = LoadPicture(Dir1.Path & "\ & File1.filename)  
    End If  
End Sub
```

```
Private Sub File1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)  
SSPanel2.Caption = "Elija el archivo de imagen."  
End Sub
```

```
Private Sub Form_Load()  
    Comb0.AddItem "Mapa de Bits (*.bmp)"  
    Comb0.AddItem "Dibujos (*.dib)"  
    Comb0.ListIndex = 0  
    File1.Pattern = "*.bmp"  
    *****  
    'Text1 = UCase(Format(Date, "dd/mmm/yyy")) 'y la formatea.  
    Text1 = Format(Date, "dd/mmm/yyy") ' la formatea.  
    Text2 = Time 'Da la hora del día.  
End Sub
```

```
Private Sub SSCommand1_Click()
```

```
    If (Right(Dir1.Path, 1) = "\") Then  
        If Fot1.SSCheck1 = True Then  
            Fot1.Picture1.Visible = True  
            Fot1.Data1.Recordset.AddNew 'Pasa a la pantalla anterior con AddNew  
            Fot1.Text2.Text = Dir1.Path & File1.filename  
            Fot1.Command2.Visible = True  
        ElseIf Fot1.SSCheck2 = True Then  
            Fot1.Picture1.Visible = True  
            Fot1.Data2.Recordset.AddNew 'Pasa a la pantalla anterior con AddNew  
            Fot1.Text4.Text = Dir1.Path & File1.filename  
            Fot1.Command2.Visible = True  
        ElseIf Fot1.SSCheck3 = True Then  
            Fot1.Picture1.Visible = True  
            Fot1.Data3.Recordset.AddNew 'Pasa a la pantalla anterior con AddNew  
            Fot1.Text6.Text = Dir1.Path & File1.filename  
            Fot1.Command2.Visible = True  
        Else  
            Fot1.Picture1.Visible = True  
            Fot1.Data4.Recordset.AddNew 'Pasa a la pantalla anterior con AddNew  
            Fot1.Text10.Text = Dir1.Path & File1.filename  
            Fot1.Command2.Visible = True  
        End If  
    Else  
        'Solo para el data1  
        If Fot1.SSCheck1 = True Then
```

```

Fot1.Data1.Refresh
Fot1.Picture1.Visible = True
Fot1.Data1.Recordset.AddNew
Fot1.Text2.Text = Dir1.Path & "\" & File1.filename
Fot1.Command2.Visible = True
Fot1.Picture1.Visible = True
Unload Me
Fot1.X = ""
Fot1.Text1.SetFocus
ElseIf Fot1.SSCheck2 = True Then
Fot1.Data2.Refresh
Fot1.Picture1.Visible = True
Fot1.Data2.Recordset.AddNew
Fot1.Text4.Text = Dir1.Path & "\" & File1.filename
Fot1.Command2.Visible = True
Fot1.Picture1.Visible = True
Fot1.Text8.Visible = True
Fot1.Text9.Visible = True
Fot1.Comb1.Visible = True
Fot1.Text1.Visible = False
Fot1.Text3.Visible = True
Fot1.Text5.Visible = False
Fot1.Text7.Visible = False

Fot1.Data1.Visible = False
Fot1.Data2.Visible = True
Fot1.Data3.Visible = False
Fot1.Data4.Visible = False
Unload Me
Fot1.X = ""
Fot1.Text3.SetFocus
ElseIf Fot1.SSCheck3 = True Then
Fot1.Data3.Refresh
Fot1.Picture1.Visible = True
Fot1.Data3.Recordset.AddNew
Fot1.Text6.Text = Dir1.Path & "\" & File1.filename
Fot1.Command2.Visible = True
Picture1.Visible = True

Fot1.Text1.Visible = False
Fot1.Text3.Visible = False
Fot1.Text5.Visible = True
Fot1.Text7.Visible = False

Fot1.Data1.Visible = False
Fot1.Data2.Visible = False
Fot1.Data3.Visible = True
Fot1.Data4.Visible = False
Unload Me
Fot1.X = ""
Fot1.Text5.SetFocus
Else
Fot1.Picture1.Visible = True
Fot1.Data4.Refresh
Fot1.Data4.Recordset.AddNew
Fot1.Text10.Text = Dir1.Path & "\" & File1.filename
Fot1.Command2.Visible = True
Fot1.Picture1.Visible = True

Fot1.Text1.Visible = False
Fot1.Text3.Visible = False
Fot1.Text5.Visible = False
Fot1.Text7.Visible = True

Fot1.Data1.Visible = False
Fot1.Data2.Visible = False
Fot1.Data3.Visible = False
Fot1.Data4.Visible = True

```

Agregal - 5

```
        Fot1.SSPanel8.Caption = "Introduzca la clave de la Cédula."
        Unload Me
        Fot1.X = ""
        Fot1.Text7.SetFocus
    End If
End If
'Unload Me
'Fot1.X = ""
'Fot1.Text1.SetFocus
End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
SSPanel2.Caption = ""
End Sub

Private Sub Picture1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
SSPanel2.Caption = ""
End Sub

Private Sub SSPanel1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
SSPanel2.Caption = ""
End Sub

Private Sub SSPanel3_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
SSPanel2.Caption = ""
End Sub
```

Reportes - 1

```
Private Sub cero_Click(Index As Integer)
C exp.Show
C exp.C limpia
'C anula
Reportes.Hide
End Sub
```

```
Sub L combo()
Combo1 = ""
Combo2 = ""
Combo3 = ""
Combo4 = ""
Combo5 = ""
Combo6 = ""
Text1 = ""
End Sub
```

```
Sub liqh()
SSPanel6.Enabled = False
SSPanel6.BackColor = &HC0C0C0
End Sub
```

```
Sub light()
SSPanel6.Enabled = True
SSPanel6.BackColor = &HC0FFFF
End Sub
```

```
Private Sub Cmdsalir_Click()
Principal.Show
L combo
Reportes.Hide
liqh
End Sub
```

```
Private Sub siete_Click(Index As Integer)
End Sub
```

```
Private Sub Cmdsalir_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
SSPanel4.Caption = "De clic. para salir."
End Sub
```

```
Private Sub Combo1_Click()
Text1 = ""
Text1 = Combo1
light
End Sub
```

```
Private Sub Combo2_Click()
Text1 = ""
Text1 = Combo2
light
End Sub
```

```
Private Sub Combo3_Click()
Text1 = ""
Text1 = Combo3
light
End Sub
```

```
Private Sub Combo4_Click()
```

Reportes - 2

```
Text1 = ""
Text1 = Combo4
light
End Sub
```

```
Private Sub Combo5_Click()
Text1 = ""
Text1 = Combo5
'If Text1 = "" Then
'    Text1 = "CIF - Q01.RPT"
'End If
light
End Sub
```

```
Private Sub Combo6_Click()
Text1 = ""
Text1 = Combo6
light
End Sub
```

```
Private Sub Picture1_Click()
```

```
On Error GoTo Err_report
```

```
CrystalReport1.ReportFileName = "c:\siap\" + Text1.Text + ""
CrystalReport1.Destination = 0
CrystalReport1.Action = 1
```

```
Err_report:
    If Err = 32755 Then
        Exit Sub
    End If
```

```
L combo
ligh
End Sub
```

```
Private Sub SSCommand1_Click()
'CrystalReport1.ReportFileName = "c:\vb\prueba1.rpt"
'*****
'*Para que aparesca el menú de impresión
```

```
On Error GoTo Err_report
```

```
impresion.Flags = &H40
impresion.PrinterDefault = False
impresion.Action = 5
'*****
CrystalReport1.ReportFileName = "c:\siap\cinsf01.rpt"
CrystalReport1.Destination = 0
iResult = CrystalReport1.PrintReport
```

```
Err_report:
    If Err = 32755 Then
        Exit Sub
    End If
End Sub
```

```
Private Sub Picture1_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
SSPanel4.Caption = "De clic. En la imagen de Impresora para la impresión del Rep
orte."
End Sub
```

```
Private Sub SSPanel2_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
SSPanel4.Caption = ""
```

Reportes - 3

End Sub

```
Private Sub SSPanel3_DblClick()  
    lich  
End Sub
```

```
Private Sub SSPanel3_MouseMove(Button As Integer, Shift As Integer, X As Single,  
    Y As Single)  
SSPanel4.Caption = "Elija la clave para imprimir el reporte deseado."
```

End Sub

```
Private Sub Text1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y  
As Single)  
SSPanel4.Caption = "Introduzca el nombre del Reporte, Gracias."  
End Sub
```


Fot1 - 1

```
Dim sql, sql1, sql2, sql3 As String
Dim rs, r1, r2, r3 As Recordset
Dim db As Database
Dim Ejecuta As Integer
Dim respuesta As Variant
Private Sub cmdAdd_Click()
    Data1.Recordset.Update
End Sub
Sub Reinicia()
Data1.Refresh
X = ""
Command1.Visible = True
Command2.Visible = False
Command3.Visible = True
Cmdsalir.Visible = True
Consulta.Visible = True
Cmddelete.Visible = False
Text1 = ""
Picture1.Visible = False
Data1.Enabled = False

SSCheck1.Value = False
SSCheck2.Value = False
SSCheck3.Value = False
SSCheck4.Value = False

Text1.Visible = True
Text3.Visible = False
Text5.Visible = False
Text7.Visible = False

Text1.Enabled = False
Text3.Enabled = False
Text5.Enabled = False
Text7.Enabled = False

Text8.Visible = False
Text9.Visible = False
Combo1.Visible = False

Data1.Visible = True
Data2.Visible = False
Data3.Visible = False
Data4.Visible = False

SSPanel3.Enabled = False
SSPanel2.Enabled = True

End Sub

Private Sub cmdDelete_Click()
    Dim Mensaje As Variant
    Dim respuesta As Variant
    If Text1.Visible = True Then
        If Text1 = "" Then
            MsgBox "No puede Eliminar si no existen datos", vbYes, "siap"
            Exit Sub
        Else
            respuesta = MsgBox("Desea dar de baja esta Imagen?", vbYesNo, "siap")
            If respuesta = vbYes Then
                Data1.Recordset.Delete
                Reinicia
            End If
        End If
    ElseIf Text3.Visible = True Then
        If Text3 = "" Then
```

Fot1 - 2

```
        MsqBox "No puede Eliminar si no existen datos", vbYes, "siap"
        Exit Sub
    Else
        respuesta = MsgBox("Desea dar de baja esta Imagen?", vbYesNo, "siap"
    )
        If respuesta = vbYes Then
            Data2.Recordset.Delete
            Reinicia
        End If
    End If
ElseIf Text5.Visible = True Then
    If Text5 = "" Then
        MsqBox "No puede Eliminar si no existen datos", vbYes, "siap"
        Exit Sub
    Else
        respuesta = MsgBox("Desea dar de baja esta Imagen?", vbYesNo, "siap")
        If respuesta = vbYes Then
            Data3.Recordset.Delete
            Reinicia
        End If
    End If
Else
    If Text7 = "" Then
        MsqBox "No puede Eliminar si no existen datos", vbYes, "siap"
        Exit Sub
    Else
        respuesta = MsgBox("Desea dar de baja esta Imagen?", vbYesNo, "siap"
        If respuesta = vbYes Then
            Data4.Recordset.Delete
            Reinicia
        End If
    End If
Reinicia
End If
End If
End Sub
```

```
Private Sub cmdUpdate_Click()
' Data1.UpdateRecord
' Data1.Recordset.Bookmark = Data1.Recordset.LastModified
Ejecuta = 2
'Command1.Visible = False
Command2.Visible = False
Command3.Visible = False
Command5.Visible = True
Cmdsalir.Visible = False
Cmdupdate.Visible = False
Consulta.Visible = False
Cmddelete.Visible = False
'C habTot
Text1.SetFocus
Text1 = ""
'Picture1 = False
'C limpTot
End Sub
```

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

```
Private Sub Cmddelete_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
SSPanel8.Caption = "De Clic. para Borrar de una Imagen"
End Sub
```

```
Private Sub cmdRefresh_MouseMove(Button As Integer, Shift As Integer, X As Single
```

Fot1 - 3

```
e, Y As Single)
SSPanel8.Caption = "De Clic. para Reiniciar la pantalla."
End Sub
```

```
Private Sub Cmdsalir_Click()
MnuCaptura.Show
Text1 = ""
Picture1.Visible = False
Fot1.Hide
```

End Sub

```
Private Sub Cmdsalir_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
SSPanel8.Caption = "De Clic. para Salir del módulo de Imagenes."
End Sub
```

```
Private Sub Combol_Click()
Text8 = ""
Text9 = Combol
If Text9 = "SIN_TIPO" Then
Text8 = 0
ElseIf Text9 = "ACUERDO" Then
Text8 = 1
ElseIf Text9 = "CONTRATO" Then
Text8 = 2
ElseIf Text9 = "CONVENIO" Then
Text8 = 3
ElseIf Text9 = "DECRETO" Then
Text8 = 4
ElseIf Text9 = "DOCTO._ADMINISTRATIVO" Then
Text8 = 5
ElseIf Text9 = "ESCRITURAS" Then
Text8 = 6
End If
'If Option1 = True Then
' Combol.Visible = False
' Exit Sub
'End If
'ExpNum.SetFocus
Combol.Visible = False
End Sub
```

```
Private Sub Command_Click()
```

End Sub

```
Private Sub Command1_Click()
Ejecuta = 1
If Fot1.X = "" Then
MsgBox "Favor de indicar si la imagen será para:" _
& " Inmueble, Documento, Planos ó Croquis.", 48
Exit Sub
End If
```

```
Command1.Visible = False
Command2.Visible = True
Command3.Visible = False
Cmdsalir.Visible = False
Consulta.Visible = False
Cmddelete.Visible = False
Text1.Enabled = True
Text3.Enabled = True
Text5.Enabled = True
Text7.Enabled = True
Data1.Refresh
```

Fot1 - 4

```
Agregal.Show 1  
End Sub
```

```
Private Sub Command1_MouseMove(Button As Integer, Shift As Integer, X As Single,  
    Y As Single)  
SSPanel8.Caption = "De Clic. para capturar una Imagen"  
End Sub
```

```
Private Sub Command2_MouseMove(Button As Integer, Shift As Integer, X As Single,  
    Y As Single)  
SSPanel8.Caption = "De Clic. para Grabar la Imagen"  
End Sub
```

```
Private Sub Command3_Click()  
SSPanel3.Enabled = True  
'Ejecuta = 2  
Command1.Visible = False  
Command2.Visible = False  
Command3.Visible = False  
Cmdsalir.Visible = False  
Consulta.Visible = False  
Cmddelete.Visible = True  
SSPanel2.Enabled = False
```

```
End Sub
```

```
Private Sub Command5_Click()  
'On Error GoTo V error  
If Text1 = "" Then  
    MsgBox "No puede Actualizar si no existen datos", vbYes, "siap", 48  
    Exit Sub  
End If  
respuesta = MsgBox("Desea confirmar la actualización del registro?", vbYesNo, "s  
iap")  
If respuesta = vbYes Then  
    If Text1 = "" Then  
        MsgBox "No puede Actualizar si no existen datos", 48  
        Exit Sub  
    End If  
    Data1.UpdateRecord  
    Data1.Recordset.Bookmark = Data1.Recordset.LastModified  
    Command2.Visible = True  
    Cmdupdate.Visible = True  
    Comconfirm1.Visible = False  
    Cmdadd.Visible = True  
    Command1.Visible = False  
    Consulta.Visible = True  
    Data1.Refresh  
Else  
    Data1.Recordset.CancelUpdate  
    Data1.Refresh  
    Command2.Visible = True  
    Cmdupdate.Visible = True  
    Cmdsalir.Visible = True  
    Cmdsalir.SetFocus  
    Comconfirm1.Visible = False  
    Cmdadd.Visible = True  
    Command1.Visible = False  
    Cmddelete.Visible = False  
    Consulta.Visible = True  
End If  
'V error:  
'If Err.Number > 0 Then  
'    MsgBox "Hay un error por favor recapture con cuidado, posiblemente escribio  
mal una fecha, reintente y continúe" +  
'    " El formato de fecha es: dd/mm/aa", 48  
'    Data1.Refresh
```

Fot1 - 5

```
' Screen.MousePointer = vbDefault
' On Error Resume Next
'End If
End Sub
```

```
Private Sub Command3_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
SSPanel8.Caption = "De Clic. para iniciar el Borrado de una Imagen"
End Sub
```

```
Private Sub Consulta Click()
SSPanel3.Enabled = True
'Ejecuta = 2
Command1.Visible = False
Command2.Visible = False
Command3.Visible = False
Cmdsalir.Visible = False
Consulta.Visible = False
Cmddelete.Visible = False
SSPanel2.Enabled = False
```

End Sub

```
Private Sub Consulta_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
SSPanel8.Caption = "De Clic. para activar el bloque de Búsqueda de Imagen."
End Sub
```

```
Private Sub Data1_Error(DataErr As Integer, Response As Integer)
'Aquí es donde se coloca el código de control de errores
'Si desea ignorar los errores, comente la siguiente línea
'Si desea controlarlos, agregue código
MsgBox "Evento de error de datos:" & Error$(DataErr)
Response = 0 'desprecia el error
End Sub
```

```
Private Sub Data1_Reposition()
Screen.MousePointer = vbDefault
On Error Resume Next
'Mostrará la posición actual del registro
'para dynasets y snapshots
Data1.Caption = "Imagen : " & (Data1.Recordset.AbsolutePosition + 1) & " de "
& (Data1.Recordset.RecordCount)
'para el objeto tabla debe establecer la propiedad index al
'crear el recordset y usar la siguiente línea
'Data1.Caption = "Registro: " & (Data1.Recordset.RecordCount * (Data1.Recordse
t.PercentPosition * 0.01)) + 1
End Sub
```

```
Private Sub Data1_Validate(Action As Integer, Save As Integer)
'Aquí es donde va el código de validación
'Se llama a este evento cuando ocurren las siguientes acciones
Select Case Action
Case vbDataActionMoveFirst
Case vbDataActionMovePrevious
Case vbDataActionMoveNext
Case vbDataActionMoveLast
Case vbDataActionAddNew
Case vbDataActionUpdate
Case vbDataActionDelete
Case vbDataActionFind
Case vbDataActionBookmark
Case vbDataActionClose
End Select
Screen.MousePointer = vbHourglass
End Sub
```

Fot1 - 6

```
Private Sub oleField1 Db1Click()  
    'esta es la forma de obtener datos en controles OLE vacíos  
    'y devolverlos a la tabla  
    oleField1.InsertObjDlg  
End Sub  
  
Private Sub cmdRefresh_Click()  
Reinicia  
End Sub  
  
Private Sub Command2_Click()  
  
If SSCheck1 = True Then  
    If Text1.Text = "" Then  
        Beep  
        Beep  
        MsgBox "Favor de introducir la clave de Cédula para continuar.", 48  
        Text1.SetFocus  
        Exit Sub  
    End If  
    Data1.Recordset.AddNew  
    Fot1.Command1.Visible = True  
    Fot1.Command2.Visible = False  
    Text1.Enabled = False  
    Reinicia  
ElseIf SSCheck2 = True Then  
    If Text3.Text = "" Then  
        Beep  
        Beep  
        MsgBox "Favor de introducir la clave de Cédula para continuar.", 48  
        Text3.SetFocus  
        Exit Sub  
    End If  
    Data2.Recordset.AddNew  
    Fot1.Command1.Visible = True  
    Fot1.Command2.Visible = False  
    Text8.Visible = False  
    Text9.Visible = False  
    Combo1.Visible = False  
    Text3.Enabled = False  
    Reinicia  
ElseIf SSCheck3 = True Then  
    If Text5.Text = "" Then  
        Beep  
        Beep  
        MsgBox "Favor de introducir la clave de C'edula para continuar.", 48  
        Text5.SetFocus  
        Exit Sub  
    End If  
    Data3.Recordset.AddNew  
    Fot1.Command1.Visible = True  
    Fot1.Command2.Visible = False  
    Text5.Enabled = False  
    Reinicia  
Else  
    If Text7.Text = "" Then  
        Beep  
        Beep  
        MsgBox "Favor de introducir la clave de Cédula para continuar.", 48  
        Text7.SetFocus  
        Exit Sub  
    End If  
    Data4.Recordset.AddNew  
    Fot1.Command1.Visible = True  
    Fot1.Command2.Visible = False  
    Text7.Enabled = False  
    Reinicia  
End Sub
```

Fot1 - 7

End If
End Sub

```
Private Sub Data2 Error(DataErr As Integer, Response As Integer)
    'Aquí es donde se coloca el código de control de errores
    'Si desea ignorar los errores, comente la siguiente línea
    'Si desea controlarlos, agregue código
    MsgBox "Evento de error de datos:" & Error$(DataErr)
    Response = 0 'desprecia el error
End Sub
```

```
Private Sub Data2 Reposition()
```

```
*****
If Ejecuta <> 1 Then
'Con este select sql3 se engaña al data para que no se pare cuando entra***
sql2 = "SELECT * FROM Doc dig WHERE Doc_dig.Cedula = '" & Text1.Text & "'"
    Data2.RecordSource = sql2
    If Data2.Recordset.RecordCount < 1 Then
        'rs3.Close
        'MsgBox "No existe esta Clave de Colindancia,Revise.", 48, "siap"
        'C Restaura
        Exit Sub
    End If
*****
```

```
Data2.Visible = True
```

```
sql3 = "SELECT Sc Tipo Docto.Descripcion" & _
" FROM Sc Tipo Docto" &
" WHERE Sc Tipo Docto.Cve Tipo doc =" & Text8
Set rs2 = db.OpenRecordset(sql3, dbOpenSnapshot)
Text9.Text = rs2(0)
```

```
Screen.MousePointer = vbDefault
On Error Resume Next
Data2.Caption = "Imagen: " & (Data2.Recordset.AbsolutePosition + 1) _
& " de " & (Data2.Recordset.RecordCount)
rs2.Close
```

```
*****
Else
Screen.MousePointer = vbDefault
On Error Resume Next
'Mostrará la posición actual del registro
'para dynasets y snapshots
Data2.Caption = "Imagen : " & (Data2.Recordset.AbsolutePosition + 1) & " de "
& (Data2.Recordset.RecordCount)
'para el objeto tabla debe establecer la propiedad index al
'crear el recordset y usar la siguiente línea
'Data1.Caption = "Registro: " & (Data1.Recordset.RecordCount + (Data1.Recordse
t.PercentPosition * 0.01)) + 1
End If
End Sub
```

```
Private Sub Data2 Validate(Action As Integer, Save As Integer)
```

```
'Aquí es donde va el código de validación
'Se llama a este evento cuando ocurren las siguientes acciones
Select Case Action
    Case vbDataActionMoveFirst
    Case vbDataActionMovePrevious
    Case vbDataActionMoveNext
    Case vbDataActionMoveLast
    Case vbDataActionAddNew
    Case vbDataActionUpdate
    Case vbDataActionDelete
    Case vbDataActionFind
    Case vbDataActionBookmark
    Case vbDataActionClose
End Select
Screen.MousePointer = vbHourglass
End Sub
```

Fot1 - 8

```
Private Sub Data3 Error(DataErr As Integer, Response As Integer)
'Aquí es donde se coloca el código de control de errores
'Si desea ignorar los errores, comente la siguiente línea
'Si desea controlarlos, agregue código
MsgBox "Evento de error de datos:" & Error$(DataErr)
Response = 0 'desprecia el error
End Sub

Private Sub Data3 Reposition()
Screen.MousePointer = vbDefault
On Error Resume Next
'Mostrará la posición actual del registro
'para dynasets y snapshots
Data3.Caption = "Imagen : " & (Data3.Recordset.AbsolutePosition + 1) & " de "
& (Data3.Recordset.RecordCount)
'para el objeto tabla debe establecer la propiedad index al
'crear el recordset y usar la siguiente línea
'Datal.Caption = "Registro: " & (Data1.Recordset.RecordCount * (Data1.Recordse
t.PercentPosition * 0.01)) + 1
End Sub

Private Sub Data3 Validate(Action As Integer, Save As Integer)
'Aquí es donde va el código de validación
'Se llama a este evento cuando ocurren las siguientes acciones
Select Case Action
Case vbDataActionMoveFirst
Case vbDataActionMovePrevious
Case vbDataActionMoveNext
Case vbDataActionMoveLast
Case vbDataActionAddNew
Case vbDataActionUpdate
Case vbDataActionDelete
Case vbDataActionFind
Case vbDataActionBookmark
Case vbDataActionClose
End Select
Screen.MousePointer = vbHourglass
End Sub

Private Sub Data4 Error(DataErr As Integer, Response As Integer)
'Private Sub Data4 Error(DataErr As Integer, Response As Integer)
'Aquí es donde se coloca el código de control de errores
'Si desea ignorar los errores, comente la siguiente línea
'Si desea controlarlos, agregue código
MsgBox "Evento de error de datos:" & Error$(DataErr)
Response = 0 'desprecia el error

End Sub

Private Sub Data4_Reposition()

Screen.MousePointer = vbDefault
On Error Resume Next
'Mostrará la posición actual del registro
'para dynasets y snapshots
Data4.Caption = "Imagen : " & (Data4.Recordset.AbsolutePosition + 1) & " de "
& (Data4.Recordset.RecordCount)
'para el objeto tabla debe establecer la propiedad index al
'crear el recordset y usar la siguiente línea
'Datal.Caption = "Registro: " & (Data1.Recordset.RecordCount * (Data1.Recordse
t.PercentPosition * 0.01)) + 1
End Sub

Private Sub Data4 Validate(Action As Integer, Save As Integer)
'Aquí es donde va el código de validación
'Se llama a este evento cuando ocurren las siguientes acciones
```


Fot1 - 9

```
Select Case Action
    Case vbDataActionMoveFirst
    Case vbDataActionMovePrevious
    Case vbDataActionMoveNext
    Case vbDataActionMoveLast
    Case vbDataActionAddNew
    Case vbDataActionUpdate
    Case vbDataActionDelete
    Case vbDataActionFind
    Case vbDataActionBookmark
    Case vbDataActionClose
End Select
Screen.MousePointer = vbHourglass
End Sub

Private Sub Form_Load()
Set db = OpenDatabase("c:\siap\siap.mdb", False, False)
Fecha = Format(Date, "dd/mm/yy") ' la formatea.
Hora = Time 'Da la hora del día.
End Sub

Private Sub Image1_Click()

End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
SSPanel8.Caption = ""
End Sub

Private Sub Picture2_Click()
Ejecuta = 2
Command1.Visible = False
Command2.Visible = False
Cmdsalir.Visible = False

Text1.Visible = True
Text3.Visible = False
Text5.Visible = False
Text7.Visible = False

Text1.Enabled = True
Text1.SetFocus
Text1 = ""
Picture1.Visible = False

Data1.Visible = True
Data2.Visible = False
Data3.Visible = False
Data4.Visible = False

End Sub

Private Sub Picture2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
SSPanel8.Caption = "Oprima Clic. para activar Inmuebles."
SSPanel4.Visible = True
SSPanel5.Visible = False
SSPanel6.Visible = False
SSPanel7.Visible = False
End Sub

Private Sub Picture3_Click()
Ejecuta = 2

Text1.Visible = False
Text3.Visible = True
```

Fot1 - 10

```
Text5.Visible = False
Text7.Visible = False
```

```
Data1.Visible = False
Data2.Visible = True
Data3.Visible = False
Data4.Visible = False
```

```
Command1.Visible = False
Command2.Visible = False
Cmdsalir.Visible = False
```

```
Text3.Enabled = True
Text3.SetFocus
Text3 = ""
Picture1.Visible = False
```

End Sub

```
Private Sub Picture3_MouseMove(Button As Integer, Shift As Integer, X As Single,
    Y As Single)
SSPanel8.Caption = "Oprima Clic. para activar Documento."
SSPanel4.Visible = False
SSPanel5.Visible = True
SSPanel6.Visible = False
SSPanel7.Visible = False
End Sub
```

```
Private Sub Picture4_Click()
Ejecuta = 2
```

```
Text1.Visible = False
Text5.Visible = True
Text3.Visible = False
Text7.Visible = False
```

```
Data1.Visible = False
Data2.Visible = False
Data3.Visible = True
Data4.Visible = False
```

```
Command1.Visible = False
Command2.Visible = False
Cmdsalir.Visible = False
```

```
Text5.Enabled = True
Text5.SetFocus
Text5 = ""
Picture1.Visible = False
Data3.Enabled = False
End Sub
```

```
Private Sub Picture4_MouseMove(Button As Integer, Shift As Integer, X As Single,
    Y As Single)
SSPanel8.Caption = "Oprima Clic. para activar Planos."
SSPanel4.Visible = False
SSPanel5.Visible = False
SSPanel6.Visible = True
SSPanel7.Visible = False
End Sub
```

```
Private Sub Picture5_Click()
Ejecuta = 2
```

```
Text1.Visible = False
Text5.Visible = False
```

Fot1 - 11

Text3.Visible = False
Text7.Visible = True

Data1.Visible = False
Data2.Visible = False
Data3.Visible = False
Data4.Visible = True

Command1.Visible = False
Command2.Visible = False
Cmdsalir.Visible = False

Text7.Enabled = True
Text7.SetFocus
Text7 = ""

Picture1.Visible = False

End Sub

```
Private Sub Picture5_MouseMove(Button As Integer, Shift As Integer, X As Single,
    Y As Single)
SSPanel8.Caption = "Oprima Clic. para activar Croquis."
SSPanel4.Visible = False
SSPanel5.Visible = False
SSPanel6.Visible = False
SSPanel7.Visible = True
End Sub
```

```
Private Sub SSCheck1_Click(Value As Integer)
X = "X"
SSCheck2.Value = False
SSCheck3.Value = False
SSCheck4.Value = False
End Sub
```

```
Private Sub SSCheck2_Click(Value As Integer)
X = "X"
SSCheck1.Value = False
SSCheck3.Value = False
SSCheck4.Value = False
End Sub
```

```
Private Sub SSCheck3_Click(Value As Integer)
X = "X"
SSCheck1.Value = False
SSCheck2.Value = False
SSCheck4.Value = False
End Sub
```

```
Private Sub SSCheck4_Click(Value As Integer)
X = "X"
SSCheck1.Value = False
SSCheck2.Value = False
SSCheck3.Value = False
```

End Sub

```
Private Sub SSCommand1_Click()
```

End Sub

```
Private Sub SSFrame1_Click()
```

End Sub

```
Private Sub SSPanel1_MouseMove(Button As Integer, Shift As Integer, X As Single,
```

Fot1 - 12

```
Y As Single)
SSPanel8.Caption = ""
SSPanel4.Visible = False
SSPanel5.Visible = False
SSPanel6.Visible = False
SSPanel7.Visible = False
End Sub
```

```
Private Sub SSPanel2_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
SSPanel8.Caption = "Debe de marcar una opción para capturar."
End Sub
```

```
Private Sub SSPanel3_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
SSPanel4.Visible = False
SSPanel5.Visible = False
SSPanel6.Visible = False
SSPanel7.Visible = False
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
'*****Altas*****
****
```

```
On Error GoTo V error
'*****
```

```
If Text1 = "" Then
```

```
Beep
```

```
MsgBox "Favor de Introducir la Clave de la Cédula", 48, "siap"
```

```
Exit Sub
```

```
End If
```

```
Select Case Ejecuta
```

```
Case 1
```

```
sql = "SELECT S_Ubicacion.Cedula FROM S_Ubicacion WHERE S_Ubicacion.Cedu
la = ' ' & Text1.Text & ' ' "
```

```
Set rs = db.OpenRecordset(sql, dbOpenDynaset)
```

```
If rs.RecordCount = 0 Then
```

```
MsgBox "Esta Clave de Cédula No Existe, favor de verificar u
oprima Reiniciar, Gracias.", 48, "siap"
```

```
Exit Sub
```

```
Else
```

```
Text1.Text = rs(0)
```

```
End If
```

```
'*****consultas,bajas*****
```

```
Case 2
```

```
sql = "SELECT * FROM Fotografia WHERE Fotografia.Cedula = ' ' & Text1 & ' ' "
```

```
Data1.RecordSource = sql
```

```
Data1.Refresh
```

```
If Data1.Recordset.RecordCount < 1 Then
```

```
MsgBox "No existe esta Clave de Fotografia,Revise.", 48, "si
ap"
```

```
Exit Sub
```

```
End If
```

```
End Select
```

```
'*****
```

```
V error:
```

```
If Err.Number > 1 Then
```

```
MsgBox "Ha ocurrido un error,por favor oprima el botón de Reiniciar.Identifi
que el problema y/o Consulte a su Administrador...", 48
```

Fot1 - 13

```
        Data1.Refresh
        On Error Resume Next
End If
Picture1.Visible = True
Data1.Enabled = True
End If
End Sub
```

```
Private Sub Text1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
SSPanel8.Caption = "Introduzca la clave de la Cédula."
End Sub
```

```
Private Sub Text10_Change()
On Error GoTo Experror
'*****
        Picture1 = LoadPicture(Text10.Text)
'*****
Experror:
        If Err.Number = 53 Then
                MsgBox "Han Borrado una o varias Imagenes,no la(s) encuentro, por favor
Avisé a su Administrador"
                + " o intente abandonar el módulo,No se asuste, Gracias... ", 48
                Screen.MousePointer = vbDefault
                On Error Resume Next
                Data4.Refresh
                Command1.Visible = True
                Command2.Visible = True
                Command3.Visible = True
                Cmdsalir.Visible = True
                Consulta.Visible = True
                Cmddelete.Visible = True
                Text7 = ""
                Picture1.Visible = False
                Data4.Enabled = False
        End If
End Sub
```

```
Private Sub Text2_Change()
```

```
On Error GoTo Experror
'*****
        Picture1 = LoadPicture(Text2.Text)
'*****
Experror:
        If Err.Number = 53 Then
                MsgBox "Han Borrado una o varias Imagenes,no la(s) encuentro, por favor
Avisé a su Administrador"
                + " o intente abandonar el módulo,No se asuste, Gracias... ", 48
                Screen.MousePointer = vbDefault
                On Error Resume Next
                Data1.Refresh
                Command1.Visible = True
                Command2.Visible = True
                Command3.Visible = True
                Cmdsalir.Visible = True
                Consulta.Visible = True
                Cmddelete.Visible = True
                Text1 = ""
                Picture1.Visible = False
                Data1.Enabled = False
        End If
End Sub
```

Fot1 - 14

```
Private Sub Text3_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
'*****Altas*****
****
On Error GoTo V error
'*****
If Text3 = "" Then
    Beep
    MsgBox "Favor de Introducir la Clave de la Cédula", 48, "siap"
    Exit Sub
End If
Select Case Ejecuta
Case 1

    sql = "SELECT S_visita.Cedula FROM S_visita WHERE S_visita.Cedula = '" &
Text3.Text & "'"
    Set rs = db.OpenRecordset(sql, dbOpenDynaset)
    If rs.RecordCount = 0 Then
        MsgBox "Esta Clave de Cédula No Existe, favor de verificar u
oprima Reiniciar, Gracias.", 48, "siap"
        Exit Sub
    Else
        Text3.Text = rs(0)
    End If

'*****consultas,bajas*****
Case 2

    sql = "SELECT * FROM Doc dig WHERE Doc_dig.Cedula = '" & Text3 & "'"
Data2.RecordSource = sql
Data2.Refresh
If Data1.Recordset.RecordCount < 1 Then
    MsgBox "No existe esta Clave de Cédula para el Documento,Rev
ise.", 48, "siap"
    Exit Sub
Else
    Text8.Visible = True
    Text9.Visible = True
End If

End Select
sql2 = "SELECT Sc Tipo Docto.Descripcion FROM Sc_Tipo_Docto" & _
" WHERE Cve Tipo Doc =" & Text8
Set rs2 = db.OpenRecordset(sql2, dbOpenSnapshot)
Text9 = rs2(0)
rs2.Close
'*****
****
V error:
If Err.Number > 1 Then
    MsgBox "Ha ocurrido un error,por favor oprima el botón de Reiniciar.Identifi
que el problema y/o Consulte a su Administrador...", 48
    Data2.Refresh
    On Error Resume Next
End If
Picture1.Visible = True
Data2.Enabled = True
End If
End Sub

Private Sub Text3_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
SSPanel8.Caption = "Introduzca la clave de la Cédula."
End Sub

Private Sub Text4_Change()
On Error GoTo Experror
```

```

.....
Picture1 = LoadPicture(Text4.Text)
.....
Experror:
    If Err.Number = 53 Then
        MsgBox "Han Borrado una o varias Imagenes,no la(s)encuentro, por favor A
vise a su Administrador"
        + " o intente abandonar el módulo,No se asuste, Gracias... ", 48
        Screen.MousePointer = vbDefault
        On Error Resume Next
        Data2.Refresh
        Command1.Visible = True
        Command2.Visible = True
        Command3.Visible = True
        Cmdsalir.Visible = True
        Consulta.Visible = True
        Cmddelete.Visible = True
        Text3 = ""
        Picture1.Visible = False
        Data2.Enabled = False

    End If
End Sub

Private Sub Text5_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
'*****Altas*****
*****
On Error GoTo V error
'*****
If Text5 = "" Then
    Beep
    MsgBox "Favor de Introducir la Clavè de la Cédula", 48, "siap"
    Exit Sub
End If
Select Case Ejecuta
Case 1

        sql = "SELECT S_visita.Cedula FROM S_visita WHERE S_visita.Cedula = '" &
Text5.Text & "'"
        Set rs = db.OpenRecordset(sql, dbOpenDynaset)
        If rs.RecordCount = 0 Then
            MsgBox "Esta Clave de Cédula No Existe, favor de verificar u
oprma Reiniciar, Gracias.", 48, "siap"
            Exit Sub
        Else
            Text5.Text = rs(0)
            End If

'*****consultas,bajas*****
Case 2

        sql = "SELECT * FROM Planos WHERE Planos.Cedula = '" & Text5.Text & "'"
        Data3.RecordSource = sql
        Data3.Refresh
        If Data3.Recordset.RecordCount < 1 Then
            MsgBox "No existe esta Clave de Cédula para el Plano,Revise.",
48, "siap"
            Exit Sub
        End If

'*****
Case 3
' sql = "SELECT Ruta dos FROM Planos WHERE Planos.Cedula = '" & Text5.Text & "'"
'     Data3.RecordSource = sql
'     Data3.Refresh
'     If Data3.Recordset.RecordCount < 1 Then

```

Fot1 - 16

```
'
MsgBox "No existe esta Clave de Cédula para el Plano,Revise
", 48, "siap"
Exit Sub
End If
End Select
```

```
*****
V error:
If Err.Number > 1 Then
MsgBox "Ha ocurrido un error,por favor oprima el botón de Reiniciar.Identifi
que el problema y/o Consulte a su Administrador...", 48
Data2.Refresh
On Error Resume Next
End If
Picture1.Visible = True
Data3.Enabled = True
End If
End Sub
```

```
Private Sub Text5_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
SSPanel8.Caption = "Introduzca la clave de la Cédula."
End Sub
```

```
Private Sub Text6_Change()
On Error GoTo Experror
Picture1 = LoadPicture(Text6.Text)
Experror:
If Err.Number = 53 Then
MsgBox "Han Borrado una o varias Imagenes,no la(s)encuentro, por favor
Avisé a su Administrador"
+ " o intente abandonar el módulo,No se asuste, Gracias... ", 48
Screen.MousePointer = vbDefault
On Error Resume Next
Data3.Refresh
Command1.Visible = True
Command2.Visible = True
Command3.Visible = True
Cmdsalir.Visible = True
Consulta.Visible = True
Cmddelete.Visible = True
Text5 = ""
Picture1.Visible = False
Data3.Enabled = False

End If
End Sub
```

```
Private Sub Text7_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
*****Altas*****
On Error GoTo V error
*****
If Text7 = "" Then
Beep
MsgBox "Favor de Introducir la Clave de la Cédula", 48, "siap"
Exit Sub
End If

Select Case Ejecuta
```


Fot1 - 17

Case 1

```
sql = "SELECT S_visita.Cedula FROM S_visita WHERE S_visita.Cedula = '" &  
Text7.Text & "'" &  
Set rs = db.OpenRecordset(sql, dbOpenDynaset)  
If rs.RecordCount = 0 Then  
MsgBox "Esta Clave de Cédula No Existe, favor de verificar u  
oprima Reiniciar, Gracias.", 48, "siap"  
Exit Sub  
Else  
Text7.Text = rs(0)  
End If
```

*****consultas,bajas*****
Case 2

```
sql = "SELECT * FROM croquis WHERE Croquis.Cedula = '" & Text7.Text & "'" &  
Data4.RecordSource = sql  
Data4.Refresh  
If Data4.Recordset.RecordCount < 1 Then  
MsgBox "No existe esta Clave de Cédula para el Croquis,Revise."  
, 48, "siap"  
Exit Sub  
End If
```

End Select

V error:

```
If Err.Number > 1 Then  
MsgBox "Ha ocurrido un error,por favor oprima el botón de Reiniciar.Iden  
tifique el problema y/o Consulte a su Administrador....", 48  
Data4.Refresh  
On Error Resume Next  
End If
```

Picture1.Visible = True

Data4.Enabled = True

End If

End Sub

```
Private Sub Text7_MouseMove(Button As Integer, Shift As Integer, X As Single, Y  
As Single)  
SSPanel8.Caption = "Introduzca la clave de la Cédula."  
End Sub
```

Private Sub Text8_DblClick()

Combo1.Visible = True

End Sub

MnuCaptura - 1

```
Dim sql As String
Dim rs As Recordset
Dim db As Database
```

```
Sub Clean()
'cero(0) = ""
'cero(1) = ""
'cero(2) = ""
'cero(3) = ""
'cero(4) = ""
'cero(5) = ""
'cero(6) = ""
'cero(7) = ""
```

End Sub

```
Private Sub cero_Click(Index As Integer)
Texpediente.Show
Texpediente.Moped
Texpediente.Inhabilita
'cero(0).Value = False
MnuCaptura.Hide
'End If
End Sub
```

```
Private Sub cinco_Click(Index As Integer)
```

```
'cinco(5).Value = False
```

End Sub

```
Private Sub Cmdsalir_Click()
Principal.Show
MnuCaptura.Hide
```

End Sub

```
Private Sub Cmdsalir_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
salir.Visible = True
SSPanel9.Caption = "Salir del módulo de opciones de captura"
End Sub
```

```
Private Sub Form_Load()
Fecha = Format(Date, "dd/mm/yy") ' la formatea.
Hora = Time 'Da la hora del día.
End Sub
```

```
Private Sub Picture1_Click()
Texpediente.Show
Texpediente.Moped
Texpediente.Inhabilita
MnuCaptura.Hide
```

End Sub

```
Private Sub Picture1_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
```

```
SSPanel9.Caption = "De clic. Para acceder el módulo de Expedientes"
```

End Sub

```
Private Sub Picture2_Click()
Ubicacion.Show
Ubicacion.Data1.Refresh
```

MnuCaptura - 2

```
Ubicacion.C limpTot
Ubicacion.C InahTot
MnuCaptura.Hide
End Sub
```

```
Private Sub Picture2_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
```

```
SSPanel9.Caption = "De clic. Para acceder el módulo de Ubicación."
```

```
End Sub
```

```
Private Sub Picture3_Click()
```

```
Visita.Show
Visita.Data1.Refresh
Visita.C limpTot
Visita.C InahTot
MnuCaptura.Hide
End Sub
```

```
Private Sub Picture3_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
```

```
SSPanel9.Caption = "De clic. Para acceder el módulo de Visita."
```

```
End Sub
```

```
Private Sub Picture4_Click()
```

```
Juridico.Show
Juridico.Data1.Refresh
Juridico.C limpTot
Juridico.C InahTot
MnuCaptura.Hide
End Sub
```

```
Private Sub Picture4_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
```

```
SSPanel9.Caption = "De clic. Para acceder el módulo de Jurídico."
```

```
End Sub
```

```
Private Sub Picture5_Click()
```

```
Colinda.Show
Colinda.Data1.Refresh
Colinda.C limpTot
Colinda.C InahTot
MnuCaptura.Hide
End Sub
```

```
Private Sub Picture5_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
```

```
SSPanel9.Caption = "De clic. Para acceder el módulo de Colindancia."
```

```
End Sub
```

```
Private Sub Picture6_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
```

```
SSPanel9.Caption = "De clic. Para acceder el módulo de Imágenes para Inmuebles (
Estaciones)."
```

```
End Sub
```

MnuCaptura - 3

```
Private Sub Picture7_Click()  
Fot1.Show  
Fot1.Data1.Refresh  
Fot1.Text1 = ""  
Fot1.Text3 = ""  
Fot1.Text5 = ""  
Fot1.Picture1.Visible = False  
MnuCaptura.Hide  
End Sub
```

```
Private Sub Picture8_Click()  
Fot1.Show  
'siete(7).Value = False  
Fot1.Data1.Refresh  
Fot1.Text1 = ""  
Fot1.Text3 = ""  
Fot1.Text5 = ""  
Fot1.Picture1.Visible = False  
MnuCaptura.Hide  
End Sub
```

```
Private Sub Picture9_Click()  
Fot1.Show  
'siete(7).Value = False  
Fot1.Data1.Refresh  
Fot1.Text1 = ""  
Fot1.Text3 = ""  
Fot1.Text5 = ""  
Fot1.Picture1.Visible = False  
MnuCaptura.Hide  
End Sub
```

```
Private Sub seis_Click(Index As Integer)  
  
'seis(6).Value = False  
  
End Sub
```

```
Private Sub Picture6_Click()  
Fot1.Show  
Fot1.Data1.Refresh  
Fot1.Text1 = ""  
Fot1.Text3 = ""  
Fot1.Text5 = ""  
Fot1.Picture1.Visible = False  
MnuCaptura.Hide  
End Sub
```

```
Private Sub cuatro_Click(Index As Integer)  
Colinda.Show  
'cuatro(4).Value = False  
Colinda.Data1.Refresh  
Colinda.C limpTot  
Colinda.C InahTot  
MnuCaptura.Hide  
End Sub
```

```
Private Sub dos_Click(Index As Integer)  
Visita.Show  
Visita.Data1.Refresh  
Visita.C limpTot  
Visita.C InahTot  
'dos(2).Value = False  
MnuCaptura.Hide  
End Sub
```

MnuCaptura - 4

```
Private Sub SSCommand1_Click()
```

```
End Sub
```

```
Private Sub tres_Click(Index As Integer)
```

```
Juridico.Show  
Juridico.Data1.Refresh  
Juridico.C limpTot  
Juridico.C InahTot  
'tres(3).Value = False  
MnuCaptura.Hide  
End Sub
```

```
Private Sub uno_Click(Index As Integer)
```

```
Ubicacion.Show  
Ubicacion.Data1.Refresh  
Ubicacion.C limpTot  
Ubicacion.C InahTot  
'uno(1).Value = False  
MnuCaptura.Hide  
End Sub
```

```
Private Sub Picture7_MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)
```

```
SSPanel9.Caption = "De clic. Para acceder el módulo de Imagenes para: Documentos  
, Croquis y Planos."
```

```
End Sub
```

```
Private Sub SSPanel2_MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)
```

```
'SSPanel3.Visible = False  
'SSPanel4.Visible = False  
'SSPanel5.Visible = False  
'SSPanel6.Visible = False  
'SSPanel7.Visible = False  
'SSPanel8.Visible = False  
'SSPanel10.Visible = False
```

```
SSPanel9.Caption = "De clic. en la imagen que Ud. elija para entrar a cualquier  
módulo."
```

```
salir.Visible = False  
End Sub
```

Catálogos - 1

Dim rsval As Recordset

```
Sub Anula()  
'Catálogos.cero(0).Value = False  
'Catálogos.uno(1).Value = False  
'Catálogos.dos(2).Value = False  
'Catálogos.tres(3).Value = False  
'Catálogos.cuatro(4).Value = False  
'Catálogos.cinco(5).Value = False  
'Catálogos.seis(6).Value = False  
'Catálogos.siete(7).Value = False
```

End Sub

```
Private Sub cero_Click(Index As Integer)
```

End Sub

```
Private Sub Cmdsalir_Click()
```

```
    MenuMto.Show  
    Catálogos.Hide
```

End Sub

```
Private Sub Cmdsalir_MouseMove(Button As Integer, Shift As Integer, X As Single,  
    Y As Single)
```

```
    salir.Visible = True  
    mensajes.Caption = "Salir del módulo de Mantenimiento a Catálogos."  
End Sub
```

```
Private Sub CONSENV_Click()
```

```
    ConEnvio.Show  
End Sub
```

```
Private Sub CONSHAB_Click()
```

```
    Conhab.Show  
End Sub
```

```
Private Sub confirmar_MouseMove(Button As Integer, Shift As Integer, X As Single,  
    Y As Single)
```

```
    confirma.Visible = True  
    If salir.Visible = True Then  
        salir.Visible = False  
    End If
```

End Sub

```
Private Sub Optval_Click(Index As Integer)
```

```
    ConEnvio.Show  
End Sub
```

```
Private Sub Frame2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y  
    As Single)
```

```
    salir.Visible = False  
    confirma.Visible = False
```

End Sub

```
Private Sub historico_Click(Index As Integer)
```

End Sub

```
Private Sub Estaciones_Click()
```

Catálogos - 3

End Sub

```
Private Sub Picture5_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
mensajes.Caption = "De clic.En la Imagen para acceder el módulo de Tipos de Coli
ndancia."
End Sub
```

```
Private Sub Picture6_Click()
Tdocumento.Show
Catálogos.Hide
```

End Sub

```
Private Sub Picture6_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
mensajes.Caption = "De clic.En la Imagen para acceder el módulo de Tipos de Docu
mentos."
End Sub
```

```
Private Sub Picture7_Click()
Testación.Show
Catálogos.Hide
```

End Sub

```
Private Sub Picture7_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
mensajes.Caption = "De clic.En la Imagen para acceder el módulo de Tipos de Esta
ciones."
End Sub
```

```
Private Sub Picture8_Click()
Tmueble.Show
Catálogos.Hide
```

End Sub

```
Private Sub Picture8_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
mensajes.Caption = "De clic.En la Imagen para acceder el módulo de Tipos de Inmu
ebles."
End Sub
```

```
Private Sub SSPanel2_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
salir.Visible = False
mensajes.Caption = "Elija la imagen y de clic. para acceder el módulo deseado."
"
```

End Sub

```
Private Sub uno_Click(Index As Integer)
```

End Sub

```
Private Sub SSPanel4_Click()
```

End Sub

```
Private Sub SSPanel5_Click()
```

Catálogos - 2

End Sub

```
Private Sub Form_Load()  
Fecha = Format(Date, "dd/mmmm/yyyy") ' la formatea.  
Hora = Time 'Da la hora del día.  
End Sub
```

```
Private Sub Picture1_Click()  
Delegaciones.Show  
Catálogos.Hide
```

End Sub

```
Private Sub Picture1_MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)  
mensajes.Caption = "De clic.En la Imagen para acceder el módulo de Delegaciones."  
End Sub
```

```
Private Sub Picture2_Click()  
Edo fisico.Show  
Catálogos.Hide
```

End Sub

```
Private Sub Picture2_MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)  
mensajes.Caption = "De clic.En la Imagen para acceder el módulo de Estado Físico."  
End Sub
```

```
Private Sub Picture3_Click()  
Estaciones.Show  
Estaciones.Data1.Refresh  
Catálogos.Hide
```

End Sub

```
Private Sub SCommand1_Click()
```

End Sub

```
Private Sub Picture3_MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)  
mensajes.Caption = "De clic.En la Imagen para acceder el módulo de Estaciones."  
End Sub
```

```
Private Sub Picture4_Click()  
Linea.Show  
Catálogos.Hide
```

End Sub

```
Private Sub Picture4_MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)  
mensajes.Caption = "De clic.En la Imagen para acceder el módulo de Lineas."  
End Sub
```

```
Private Sub Picture5_Click()  
Colindancia.Show  
Catálogos.Hide
```


MantoDB - 1

```
Dim opcion, accion As Integer
Dim sql, sql1, sql2, sql3, sql4, sql5, sql6, sql7, sql8, sql9, nombre As String
Dim rs As Recordset
'Dim db As Database
```

```
Sub clean bd()
sql = "delete * from Croquis"
db.Execute sql
sql1 = "delete * from Doc_dig"
db.Execute sql1
sql2 = "delete * from Fotografia"
db.Execute sql2
sql3 = "delete * from Planos"
db.Execute sql3
sql4 = "delete * from S_Colindancia"
db.Execute sql4
sql5 = "delete * from S_Expediente"
db.Execute sql5
sql6 = "delete * from S_Juridico"
db.Execute sql6
sql7 = "delete * from S_Tipo_Exp"
db.Execute sql7
sql8 = "delete * from S_Ubicacion"
db.Execute sql8
sql9 = "delete * from S_Visita"
db.Execute sql9
End Sub
```

```
Sub Reiniciax().
DRIVE_A.Enabled = False
DRIVE_B.Enabled = False

respalda.Value = False
Restaura.Value = False
Imagen.Value = False
Borra.Value = False

respalda.Enabled = True
Restaura.Enabled = True
Imagen.Enabled = True
Borra.Enabled = True

Cmdconfirma.Enabled = False
txtarchivo.Text = ""
End Sub
```

```
Sub Restaura modulo()
DRIVE_A.Enabled = False
DRIVE_B.Enabled = False
respalda.Value = False
Restaura.Value = False
respalda.Enabled = True
Restaura.Enabled = True
Cmdconfirma.Enabled = False
txtarchivo.Text = ""
End Sub
```

```
Private Sub Borra_Click()
accion = 4
respalda.Enabled = False
Restaura.Enabled = False
Imagen.Enabled = False
DRIVE_A.Enabled = False
DRIVE_B.Enabled = False
txtarchivo.Enabled = True
Cmdconfirma.Enabled = True
```

MantODB - 2

```
Cmdcancela.Enabled = True
Cmdconfirma.SetFocus
```

End Sub

```
Private Sub Cmdcancela_Click()
Reiniciax
```

End Sub

```
Private Sub Cmdcancela_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
' cancelar.Visible = True
' If confirmar.Visible = True Then
'     confirmar.Visible = False
' ElseIf salir.Visible = True Then
'     salir.Visible = False
' End If
SSPanel8.Caption = "Reinicia la pantalla."
End Sub
```

```
Private Sub Cmdconfirma_Click()
```

```
Dim respuesta As String
```

```
respuesta = MsgBox("¿Esta seguro que desea realizar un Respaldo, Recuperación de archivo ó Borrar la Base de Datos? " & "Si no va realizar ninguna acción por favor oprima 'No' y salga del módulo. Gracias.", vbYesNo)
```

```
If respuesta = vbYes Then
```

```
Select Case accion
```

```
Case 1
```

```
db.Close
```

```
If opcion = 1 Then
```

```
msdos = Shell("C:\siap\Pkzip.EXE -ex a:\" + txtarchivo + " c:\siap\siap.mdb", 1)
MsgBox "Verifique que la Base de datos siap.mdb se haya respaldado con éxito."
```

```
Reiniciax
```

```
Else
```

```
msdos = Shell("C:\siap\Pkzip.EXE -ex d:\" + txtarchivo + " c:\siap\siap.mdb", 1)
MsgBox "Verifique que la Base de datos Siap.mdb se haya respaldado con éxito."
```

```
Reiniciax
```

```
End If
```

```
Case 2
```

```
If opcion = 1 Then
```

```
respuesta = MsgBox("Favor de verificar que el nombre del archivo sea correcto, continúe, gracias.", vbYesNo)
```

```
If respuesta = vbYes Then
```

```
msdos = Shell("C:\siap\Pkunzip.EXE -o a:\" + txtarchivo.Text + " *.*", 1)
```

```
MsgBox "Verifique que la Base de datos Siap.mdb se haya recuperado con éxito."
```

```
Reiniciax
```

```
Exit Sub
```

```
End If
```

```
Else
```

```
respuesta = MsgBox("Favor de verificar que el nombre del archivo sea correcto, continúe, gracias.", vbYesNo)
```

```
If respuesta = vbYes Then
```

```
msdos = Shell("C:\siap\Pkunzip.EXE -o d:\" + txtarchivo + " *.*", 1)
```

```
MsgBox "Verifique que la Base de datos Siap.mdb se haya recuperado con éxito."
```

```
Reiniciax
```

```
Exit Sub
```

```
End If
```

```
End If
```

```
Case 3
```

MantoDB - 4

```
' End If
  SSPanel8.Caption = "Salir del módulo."
End Sub

Private Sub DRIVE_A_Click()
  opcion = 1
  If accion = 1 Then
    If txtarchivo.Text = "" Then
      txtarchivo.SetFocus
      MsgBox ("Favor de introducir el nombre del archivo y confirme, gracias."
    )
    Else
      Cmdconfirma.Enabled = True
      DRIVE B.Enabled = False
      Cmdconfirma.SetFocus
    End If
  End If

  If accion = 2 Then
    If txtarchivo.Text = "" Then
      txtarchivo.SetFocus
      MsgBox ("Favor de introducir el nombre del archivo y confirme, gracias."
    )
    Else
      Cmdconfirma.Enabled = True
      DRIVE B.Enabled = False
      Cmdconfirma.SetFocus
    End If
  End If

  If accion = 3 Then
    If txtarchivo.Text = "" Then
      txtarchivo.SetFocus
      MsgBox ("Favor de introducir el nombre del archivo y confirme, gracias."
    )
    Else
      Cmdconfirma.Enabled = True
      DRIVE B.Enabled = False
      Cmdconfirma.SetFocus
    End If
  End If
End Sub

Private Sub DRIVE_A_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
  SSPanel8.Caption = "Utilizará el Drive 'A'"
End Sub

Private Sub DRIVE_B_Click()
  opcion = 2
  If accion = 1 Then
    If txtarchivo.Text = "" Then
      txtarchivo.SetFocus
      MsgBox ("Favor de introducir el nombre del archivo y confirme, gracias."
    )
    Else
      Cmdconfirma.Enabled = True
      DRIVE A.Enabled = False
      Cmdconfirma.SetFocus
    End If
  End If

  If accion = 2 Then
    If txtarchivo.Text = "" Then
      txtarchivo.SetFocus
      MsgBox ("Favor de introducir el nombre del archivo y confirme, gracias."
    )
    Else
      Cmdconfirma.Enabled = True
```

MantoDB - 5

```
        DRIVE A.Enabled = False
        Cmdconfirma.SetFocus
    End If
End If
If accion = 3 Then
    If txtarchivo.Text = "" Then
        txtarchivo.SetFocus
        MsgBox ("Favor de introducir el nombre del archivo y confirme, gracias."
    )
    Else
        Cmdconfirma.Enabled = True
        DRIVE A.Enabled = False
        Cmdconfirma.SetFocus
    End If
End If
End Sub
```

```
Private Sub DRIVE_B_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
SSPanel8.Caption = "Utilizará el Drive 'D'"
End Sub
```

```
Private Sub Form_Load()
    'Set db = OpenDatabase("c:\valija\valija.mdb", False, False)
Fecha = Format(Date, "dd/mm/yy") ' la formatea.
Hora = Time 'Da la hora del día.
End Sub
```

```
Private Sub Option1_Click()
    accion = 3

    respalda.Enabled = False
    DRIVE A.Enabled = True
    DRIVE B.Enabled = True
    txtarchivo.Enabled = True
    Cmdconfirma.Enabled = True
    Cmdcancela.Enabled = True

    DRIVE_A.SetFocus
End Sub
```

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
SSPanel8.Caption = ""
End Sub
```

```
Private Sub Imagen_Click()
    accion = 3
    respalda.Enabled = False
    Restaura.Enabled = False
    Borra.Enabled = False
    DRIVE A.Enabled = True
    DRIVE B.Enabled = True
    txtarchivo.Enabled = True
    Cmdconfirma.Enabled = True
    Cmdcancela.Enabled = True
    txtarchivo.SetFocus
End Sub
```

```
Private Sub respalda_Click()
    accion = 1
    Restaura.Enabled = False
    Imagen.Enabled = False
    Borra.Enabled = False
```

MantoDB - 6

```
DRIVE A.Enabled = True
DRIVE B.Enabled = True
Cmdcancela.Enabled = True
DRIVE A.SetFocus
'txtarchivo.Text = "Siap.mdb"
txtarchivo.Text = "SIAPBD"
End Sub

Private Sub Restaura_Click()
    accion = 2

    respalda.Enabled = False
    Imagen.Enabled = False
    Borra.Enabled = False
    DRIVE A.Enabled = True
    DRIVE B.Enabled = True
    txtarchivo.Enabled = True
    Cmdconfirma.Enabled = True
    Cmdcancela.Enabled = True
    'txtarchivo.Text = "Siap.mdb"
    txtarchivo.Text = "SIAPBD.ZIP"
    DRIVE_A.SetFocus
End Sub

Private Sub SSFrame1_MouseMove(Button As Integer, Shift As Integer, X As Single,
    Y As Single)
End Sub

Private Sub SSPanel2_MouseMove(Button As Integer, Shift As Integer, X As Single,
    Y As Single)
    ' LDRIVE A.Visible = False
    ' LDRIVE B.Visible = False
    SSPanel8.Caption = ""
End Sub

Private Sub SSPanel3_MouseMove(Button As Integer, Shift As Integer, X As Single,
    Y As Single)
    SSPanel8.Caption = "Al dar clic. estará efecundo el respaldo de la Base de Dato
s."
End Sub

Private Sub SSPanel4_MouseMove(Button As Integer, Shift As Integer, X As Single,
    Y As Single)
    SSPanel8.Caption = "Al dar clic. estará efectuando la recuperación de la Base de
datos."
End Sub

Private Sub SSPanel5_MouseMove(Button As Integer, Shift As Integer, X As Single,
    Y As Single)
    SSPanel8.Caption = "Al dar clic. estará efectuando la instalación de una imagen
en el Sistema"
End Sub

Private Sub SSPanel6_MouseMove(Button As Integer, Shift As Integer, X As Single,
    Y As Single)
    'LDRIVE A.Visible = False
    'LDRIVE B.Visible = False
    'confirmar.Visible = False
    'cancelar.Visible = False
    'salir.Visible = False
End Sub

Private Sub txtarchivo_KeyPress(KeyAscii As Integer)
```

MenuMto - 1

```
Dim rsval As Recordset
Private Sub Cmdconfirma_Click()
'Dim sqlval As String
'Dim sino As Integer

'If Optval(0).Value = True Then
'  Load Menucat
'  Menucat.Show
'ElseIf Optval(1).Value = True Then
'  Load Iniciar
'  Iniciar.Show
'ElseIf Optval(2).Value = True Then
'  Load Depura
'  Depura.Show
'ElseIf Optval(3).Value = True Then
'  If usuario = "ADMON" Then
'    Load Personal
'    Personal.Show
'  Else
'    MsgBox "No tiene autorización para personalizar", 38, "Observación"
'    Beep
'  End If
'End If

End Sub
```

```
Private Sub Cmdsalir_Click()
  Principal.Show
  MenuMto.Hide
  llave.Visible = False
End Sub
```

```
Private Sub Cmdsalir_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
  salir.Visible = True
  SSPanel6.Caption = "De clic. Para salir del módulo de Mantenimiento."
End Sub
```

```
Private Sub confirma Click()
'  Dim sqlval As String
'  Dim sino As Integer

'  If Optval(0).Value = True Then
'    Catálogos.Show
'    Optval(0).Value = False
'    MenuMto.Hide
'    Catálogos.cero(0).Value = False
'    Catálogos.uno(1).Value = False
'    Catálogos.dos(2).Value = False
'    Catálogos.tres(3).Value = False
'    Catálogos.cuatro(4).Value = False
'    Catálogos.cinco(5).Value = False
'    Catálogos.seis(6).Value = False
'    Catálogos.siete(7).Value = False
'  ElseIf Optval(1).Value = True Then
'    MsgBox "Que paso? que no ves que no tengo título?, por lo tanto no hay nada
...
'    MenuMto.Optval(1).Value = False
'    Iniciar.Show
'    Optval(1).Value = False
'    MenuMto.Hide
'  ElseIf Optval(2).Value = True Then
'    MantoDB.Show
'    Optval(2).Value = False
```

MenuMto - 2

```
'      MenuMto.Hide
'ElseIf Optval(3).Value = True Then
'      Secreto.Show
'      Optval(3).Value = False
'      MenuMto.Hide
'End If
End Sub

Private Sub confirma_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
    confirmar.Visible = True
    If salir.Visible = True Then
        salir.Visible = False
    End If
End Sub

Private Sub Form_Load()
Fecha = Format(Date, "dd/mmmm/yyyy") ' la formatea.
Hora = Time 'Da la hora del día.
End Sub

Private Sub Picture4_Click()
    If usuario = "ADMON" Then
        Load Personal
        Personal.Show
    Else
        MsgBox "No tiene autorización para personalizar", 38, "Observación"
        Beep
    End If
End Sub

Private Sub Frame2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
End Sub

Private Sub SSCommand1_Click()
End Sub

Private Sub Option1_Click()
End Sub

Private Sub llave_Click()
Secreto.Show
MenuMto.Hide
End Sub

Private Sub llave_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
SSPanel6.Caption = "De clic. Para acceder el módulo de Mantenimiento a Usuarios.
"
End Sub

Private Sub Picture1_Click()
Catálogos.Show
MenuMto.Hide
End Sub
```

MenuMto - 3

```
Private Sub Picture1_MouseMove(Button As Integer, Shift As Integer, X As Single,
    Y As Single)
SSPanel6.Caption = "De clic. Para acceder el módulo de Mantenimiento a Catálogos"
End Sub
```

```
Private Sub Picture2_Click(Index As Integer)
MantODB.Show
    MantODB.DRIVE_A.Enabled = False
    MantODB.DRIVE_B.Enabled = False

    MantODB.respalda.Value = False
    MantODB.Restaura.Value = False
    MantODB.Imagen.Value = False
    MantODB.Borra.Value = False

    MantODB.respalda.Enabled = True
    MantODB.Restaura.Enabled = True
    MantODB.Imagen.Enabled = True
    MantODB.Borra.Enabled = True

    MantODB.Cmdconfirma.Enabled = False
    MantODB.txtarchivo.Text = ""
MenuMto.Hide
End Sub
```

```
Private Sub Picture2_MouseMove(Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)
SSPanel6.Caption = "De clic. Para acceder el módulo de Mantenimiento a la Base de Datos."
End Sub
```

```
Private Sub SSPanel2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
SSPanel6.Caption = "De clic. en la imagen que Ud. elija para entrar a cualquier módulo."
salir.Visible = False
End Sub
```


Principal - 1

```
Private Sub Form Load()  
Fecha = Format(Date, "dd/mmmm/yyyy") ' la formatea.  
Hora = Time 'Da la hora del día.  
End Sub
```

```
Private Sub SSCommand1_Click()
```

```
Set db = OpenDatabase("c:\siap\siap.mdb", False, False)  
sql = "select usuario,privilegio from Usuario where usuario = '" + Pwd.Txtusuari  
o.Text + "'" "  
Set rs = db.OpenRecordset(sql, dbOpenSnapshot)  
If rs(1) = "B" Then  
MsgBox "Disculpe Ud. no tiene permiso para entrar a este módulo, gracias.",  
48, "siap"  
Else  
MnuCaptura.Show  
Principal.Hide  
End If  
End Sub
```

```
Private Sub SSCommand2_Click()  
Reportes.Show  
Principal.Hide  
End Sub
```

```
Private Sub SSCommand3_Click()  
Principal.Hide  
MenuHAB.Show  
End Sub
```

```
Private Sub SSCommand4_Click()  
Pwd.Show  
Principal.Hide  
Pwd!Txtusuario = ""  
Pwd!Txtpassword = ""  
Pwd!Cmddsalir.SetFocus  
End Sub
```

```
Private Sub SSCommand5_Click()  
Dim rs As Recordset  
Dim db As Database  
Dim sql As String  
Load MenuMto  
'****Checa que no capturen usuarios repetidos**LMB**  
Set db = OpenDatabase("c:\Siap\Siap.mdb", False, False)  
sql = "select usuario,password,privilegio from usuario where "  
sql = sql & "usuario.usuario = '" & Pwd.Txtusuario.Text & "'and "  
sql = sql & "usuario.password = '" & Pwd.Txtpassword.Text & "'" "  
Set rs = db.OpenRecordset(sql, dbOpenDynaset)  
If rs.RecordCount = 1 Then  
If rs(2) = "A" Then  
MenuMto.Show  
Principal.Hide  
MenuMto.llave.Visible = True  
Else  
MsgBox "Disculpe Ud. no tiene permiso para entrar a este módulo, gracias  
", 48  
Beep  
Exit Sub  
End If  
MenuMto.Show  
Principal.Hide
```

Principal - 2

End Sub

Private Sub Timer1 Timer()

Picture1.Move Picture1.Left + DeltaX, Picture1.Top + DeltaY

' If Picture1.Left < ScaleLeft Then DeltaX = 100

' If Picture1.Left + Picture1.Width > ScaleWidth + ScaleLeft Then

' DeltaX = -100

' End If

' If Picture1.Top < ScaleTop Then DeltaY = 100

' If Picture1.Top + Picture1.Height > ScaleHeight + ScaleTop Then

' DeltaY = -100

' End If

SSPanel4.Move SSPanel4.Left + DeltaX, SSPanel4.Top + DeltaY

If SSPanel4.Left < ScaleLeft Then DeltaX = 100

If SSPanel4.Left + SSPanel4.Width > ScaleWidth + ScaleLeft Then

DeltaX = -100

End If

If SSPanel4.Top < ScaleTop Then DeltaY = 100

If SSPanel4.Top + SSPanel4.Height > ScaleHeight + ScaleTop Then

DeltaY = -100

End If

End Sub

Presenta - 1

```
Private Sub Timer1_Timer()  
Static a As Integer  
a = a + 50  
Height = 140 + a  
If a = 18000 Then  
Timer1.Enabled = False  
a = 0  
Pwd.Show  
Pwd!Timer1.Interval = 1  
Pwd!Timer1.Enabled = True  
Presenta.Hide  
Exit Sub  
End If  
End Sub
```

```
Private Sub Form_Click()  
Pwd.Show  
Presenta.Hide  
End Sub
```

Colinda - 1

```
Dim sql, sql1, sql2, sql3 As String
Dim rs, r1, r2, r3 As Recordset
Dim db As Database
Dim Ejecuta As Integer
Dim respuesta As Variant
Sub C habTot()
Colinda.Text1.Enabled = True
Colinda.txtFields0.Enabled = True
Colinda.txtFields1.Enabled = True
Colinda.txtFields2.Enabled = True
Colinda.txtFields3.Enabled = True
Colinda.txtFields4.Enabled = True
Colinda.Text2.Enabled = True
Colinda.Text3.Enabled = True
Colinda.Text4.Enabled = True
Colinda.Text5.Enabled = True
Colinda.Text6.Enabled = True
Colinda.Text7.Enabled = True
Colinda.Text8.Enabled = True
Colinda.Text9.Enabled = True
Colinda.Text10.Enabled = True
Colinda.Text11.Enabled = True
'Colinda.Data1.Enabled = True
End Sub
Sub C limpTot()
Colinda.Text1 = ""
Colinda.txtFields0 = ""
Colinda.txtFields1 = ""
Colinda.txtFields2 = ""
Colinda.txtFields3 = ""
Colinda.txtFields4 = ""
Colinda.Text2 = ""
Colinda.Text3 = ""
Colinda.Text4 = ""
Colinda.Text5 = ""
Colinda.Text6 = ""
Colinda.Text7 = ""
Colinda.Text8 = ""
Colinda.Text9 = ""
Colinda.Text10 = ""
Colinda.Text11 = ""
End Sub

Sub C_Restaura()
Colinda.Data1.Recordset.CancelUpdate
Colinda.Data1.Refresh
Colinda.Command2.Visible = True
Colinda.Cmdupdate.Visible = True
Colinda.Cmdsalir.Visible = True
Colinda.Comconfirm1.Visible = False
Colinda.Cmdadd.Visible = True
Colinda.Command1.Visible = False
Colinda.Cmddelete.Visible = False
Colinda.Consulta.Visible = True
Colinda.C InahTot
Colinda.C limpTot
Colinda.Data1.Enabled = False
Combo1.Visible = False
End Sub

Sub C InahTot()
Colinda.Text1.Enabled = False
Colinda.txtFields0.Enabled = False
Colinda.txtFields1.Enabled = False
Colinda.txtFields2.Enabled = False
Colinda.txtFields3.Enabled = False
```

Colinda - 2

```
Colinda.txtFields4.Enabled = False
Colinda.Text2.Enabled = False
Colinda.Text3.Enabled = False
Colinda.Text4.Enabled = False
Colinda.Text5.Enabled = False
Colinda.Text6.Enabled = False
Colinda.Text7.Enabled = False
Colinda.Text8.Enabled = False
Colinda.Text9.Enabled = False
Colinda.Text10.Enabled = False
Colinda.Text11.Enabled = False
Colinda.Data1.Enabled = False
End Sub
```

```
Private Sub cmdAdd_Click()
Ejecuta = 1
Command2.Visible = False
Cmdupdate.Visible = False
Cmdsalir.Visible = False
Comconfirm1.Visible = True
Cmdadd.Visible = False
Consulta.Visible = False
C limpTot
C habTot
txtFields0.SetFocus
End Sub
```

```
Private Sub Cmdadd_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
SSPanel3.Caption = "Al dar clic. o Enter en este botón Iniciará una Captura."
End Sub
```

```
Private Sub cmdDelete_Click()
Dim Mensaje As Variant
Dim respuesta As Variant
If txtFields0 = "" Then
MsgBox "No puede Eliminar si no existen datos", vbYes, "siap"
Exit Sub
End If
respuesta = MsgBox("Desea dar de baja este Documento?", vbYesNo, "siap")
If respuesta = vbYes Then
Data1.Recordset.Delete
Command2.Visible = True
Cmdupdate.Visible = True
Cmdsalir.Visible = True
Cmdsalir.SetFocus
Comconfirm1.Visible = False
Cmdadd.Visible = True
Cmddelete.Visible = False
Consulta.Visible = True
C limpTot
C_InahTot
Else
C limpTot
C_InahTot
End If
End Sub
```

```
Private Sub Cmddelete_MouseMove(Button As Integer, Shift As Integer, X As Single
, Y As Single)
SSPanel3.Caption = "Al dar clic. o Enter en este botón Aceptará la Baja."
End Sub
```

```
Private Sub cmdRefresh_Click()
C_Restaura
```

```

Colinda - 3

Data2.Visible = False

End Sub

Private Sub cmdRefresh_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
SSPanel3.Caption = "Al dar clic. o Enter en este botón Reiniciará la pantalla."
End Sub

Private Sub Cmdsalir_Click()
MnuCaptura.Show
Colinda.Hide
C Restaura
End Sub

Private Sub Cmdsalir_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
SSPanel3.Caption = "Al dar clic. o Enter en este botón Saldrá del módulo."
End Sub

Private Sub cmdUpdate_Click()

Ejecuta = 2
Command2.Visible = False
Cmdupdate.Visible = False
Cmdsalir.Visible = False
Cmdadd.Visible = False
Command1.Visible = True
Consulta.Visible = False
C habTot
txtFields0.SetFocus
C_limpTot

End Sub

Private Sub cmdClose_Click()
Unload Me
End Sub

Private Sub Cmdupdate_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
SSPanel3.Caption = "Al dar clic. o Enter en este botón Iniciará una Modificación"
End Sub

Private Sub Combo1_Click()

txtFields3 = ""
Text1 = Combo1
If Text1 = "SIN COLINDANCIA" Then
txtFields3 = 0
ElseIf Text1 = "REAL" Then
txtFields3 = 1
ElseIf Text1 = "DOCUMENTAL" Then
txtFields3 = 2
End If
Combo1.Visible = False
End Sub

Private Sub Comconfirm1_Click()
If txtFields0 = "" Then
MsgBox "No puede Agregar un registro si no existen datos", vbYes, "siap"
Exit Sub
End If
respuesta = MsgBox("¿Desea confirmar el Alta?", vbYesNo, "siap")

```

Colinda - 4

```
If respuesta = vbYes Then
Text11 = txtFields4 * Text8
Text11 = Format(Text11, "##,##0")
Data1.Recordset.AddNew
Data1.Recordset.CancelUpdate
Command2.Visible = True
Cmdupdate.Visible = True
Cmdsalir.Visible = True
Cmdsalir.SetFocus
Comconfirm1.Visible = False
Cmdadd.Visible = True
Consulta.Visible = True
Data2.Visible = False
C InahTot
C Restaura
Beep
Beep
Beep
Else
C limpTot
C InahTot
Data2.Visible = False
End If
End Sub

Private Sub Comconfirm1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
SSPanel3.Caption = "Al dar clic. o Enter en este botón Aceptará la Captura."
End Sub

Private Sub Command1_Click()
If txtFields0 = "" Then
MsgBox "No puede Actualizar si no existen datos", vbYes, "siap"
Exit Sub
End If
respuesta = MsgBox("Desea confirmar la actualización del registro?", vbYesNo, "siap")
If respuesta = vbYes Then
If txtFields0 = "" Then
MsgBox "No puede Actualizar si no existen datos"
Exit Sub
End If
Text11 = txtFields4 * Text8
Text11 = Format(Text11, "##,##0")
Data1.UpdateRecord
Data1.Recordset.Bookmark = Data1.Recordset.LastModified
Command2.Visible = True
Cmdupdate.Visible = True
Comconfirm1.Visible = False
Cmdadd.Visible = True
Command1.Visible = False
Consulta.Visible = True
C InahTot
Data2.Visible = False
Data1.Refresh
Else
Data1.Recordset.CancelUpdate
Data1.Refresh
Command2.Visible = True
Cmdupdate.Visible = True
Cmdsalir.Visible = True
Cmdsalir.SetFocus
Comconfirm1.Visible = False
Cmdadd.Visible = True
Command1.Visible = False
Cmddelete.Visible = False
Consulta.Visible = True
```

Colinda - 5

```
Data2.Visible = False
C InahTot
C limpTot
End If
End Sub
```

```
Private Sub Command1_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
SSPanel3.Caption = "Al dar clic. o Enter en este botón Aceptará el Cambio."
End Sub
```

```
Private Sub Command2_Click()
Ejecuta = 2
Cmddelete.Visible = False
Cmdupdate.Visible = False
Cmdsalir.Visible = False
Cmddelete.Visible = True
Cmdadd.Visible = False
Command2.Visible = False
Consulta.Visible = False
Data1.Refresh
C habTot
C limpTot
txtFields0.SetFocus
End Sub
```

```
Private Sub Command2_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
SSPanel3.Caption = "Al dar clic. o Enter en este botón Iniciará una Baja."
End Sub
```

```
Private Sub Consulta_Click()
Ejecuta = 2
Command2.Visible = False
Cmdupdate.Visible = False
Cmdsalir.Visible = False
Consulta.Visible = True
Cmdadd.Visible = False
Consulta.Visible = False
C limpTot
C habTot
txtFields0.SetFocus
End Sub
```

```
Private Sub Consulta_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
SSPanel3.Caption = "Al dar clic. o Enter en este botón Iniciará una Consulta."
End Sub
```

```
Private Sub Data1_Error(DataErr As Integer, Response As Integer)
' This is where you would put error handling code
' If you want to ignore errors, comment out the next line
' If you want to trap them, add code here to handle them
MsgBox "Data error event hit err:" & Error$(DataErr)
Response = 0 'throw away the error
End Sub
```

```
Private Sub Data1_Reposition()
'*****
If Ejecuta <> 1 Then
'Con este select sql3 se engaña al data para que no se pare cuando no hay**
sql3 = "SELECT * FROM S_Colindancia WHERE S_Colindancia.Cedula = '" & txtFields0
& "'"
Data1.RecordSource = sql3
If Data1.Recordset.RecordCount < 1 Then
```


Colinda - 6

```
'rs3.Close
'MsgBox "No existe esta Clave de Colindancia,Revise.", 48, "siap"
'C Restaura
    Exit Sub
End If
'*****
Data1.Visible = True
sql2 = "SELECT Sc Tipo Colin.Descripcion" & _
" FROM Sc Tipo Colin" &
" WHERE Sc Tipo Colin.Cve Tipo colin =" & txtFields3
Set rs2 = db.OpenRecordset(sql2, dbOpenSnapshot)
Text1 = rs2(0)
    Screen.MousePointer = vbDefault
    On Error Resume Next
    Data1.Caption = "Total.:" & (Data1.Recordset.AbsolutePosition + 1) _
    & " de " & (Data1.Recordset.RecordCount)
rs2.Close
'*****
Else
    Screen.MousePointer = vbDefault
    On Error Resume Next
    'This will display the current record position
    'for dynasets and snapshots
    'Data1.Caption = "Record: " & (Data1.Recordset.AbsolutePosition + 1)
    Data1.Caption = "Total.:" & (Data1.Recordset.AbsolutePosition + 1) _
    & " de " & (Data1.Recordset.RecordCount)
    'for the table object you must set the index property when
    'the recordset gets created and use the following line
    'Data1.Caption = "Record: " & (Data1.Recordset.RecordCount * (Data1.Recordset.
PercentPosition * 0.01)) + 1
End If
End Sub

Private Sub Data1 Validate(Action As Integer, Save As Integer)
'This is where you put validation code
'This event gets called when the following actions occur
Select Case Action
    Case vbDataActionMoveFirst
    Case vbDataActionMovePrevious
    Case vbDataActionMoveNext
    Case vbDataActionMoveLast
    Case vbDataActionAddNew
    Case vbDataActionUpdate
    Case vbDataActionDelete
    Case vbDataActionFind
    Case vbDataActionBookmark
    Case vbDataActionClose
End Select
    Screen.MousePointer = vbHourglass
End Sub

Private Sub Data2 Error(DataErr As Integer, Response As Integer)
MsgBox "Data error event hit err:" & Error$(DataErr)
Response = 0 'throw away the error
End Sub

Private Sub Data2 Validate(Action As Integer, Save As Integer)
'This is where you put validation code
'This event gets called when the following actions occur
Select Case Action
    Case vbDataActionMoveFirst
    Case vbDataActionMovePrevious
    Case vbDataActionMoveNext
    Case vbDataActionMoveLast
    Case vbDataActionAddNew
    Case vbDataActionUpdate
    Case vbDataActionDelete
    Case vbDataActionFind
```

Colinda - 7

```
    Case vbDataActionBookmark
    Case vbDataActionClose
End Select
Screen.MousePointer = vbHourglass
End Sub
```

```
Private Sub DBGrid1_Click()
txtFields3 = DBGrid1.Columns(0).Value
Text1.Text = DBGrid1.Columns(1).Value
Screen.MousePointer = vbHourglass
End Sub
```

```
Private Sub Form_Load()
Set db = OpenDatabase("c:\siap\siap.mdb", False, False)
Fecha = Format(Date, "dd/mmmm/yyyy") ' la formatea.
Hora = Time 'Da la hora del día.
End Sub
```

```
Private Sub SSCommand1_Click()
```

```
End Sub
```

```
Private Sub txtFields_KeyPress(Index As Integer, KeyAscii As Integer)
```

```
End Sub
```

```
Private Sub SSCommand5_Click()
```

```
End Sub
```

```
Private Sub SSCommand3_Click()
```

```
End Sub
```

```
Private Sub SSCommand6_Click()
```

```
End Sub
```

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
SSPanel3.Caption = ""
End Sub
```

```
Private Sub SSPanel1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
SSPanel3.Caption = ""
```

```
End Sub
```

```
Private Sub Text10_KeyPress(KeyAscii As Integer)
If KeyAscii < Asc("0") Or KeyAscii > Asc("9") Then
    KeyAscii = 0
    Beep
End If
End Sub
```

```
Private Sub Text2_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Text5.SetFocus
End If
End Sub
```

```
Private Sub Text3_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
```

Colinda - 8

```
Text6.SetFocus
End If
End Sub
```

```
Private Sub Text4_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Text7.SetFocus
End If
End Sub
```

```
Private Sub Text5_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Text8.SetFocus
End If
End Sub
```

```
Private Sub Text6_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Text9.SetFocus
End If
End Sub
```

```
Private Sub Text7_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Text10.SetFocus
End If
End Sub
```

```
Private Sub Text8_KeyPress(KeyAscii As Integer)
If KeyAscii < Asc("0") Or KeyAscii > Asc("9") Then
KeyAscii = 0
Beep
End If
End Sub
```

```
Private Sub Text9_KeyPress(KeyAscii As Integer)
If KeyAscii < Asc("0") Or KeyAscii > Asc("9") Then
KeyAscii = 0
Beep
End If
End Sub
```

```
Private Sub txtFields0_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Select Case Ejecuta
Case 1
If txtFields0 = "" Then
Beep
MsgBox "Favor de Introducir la Clave de la Cédula", 48, "siap"
Exit Sub
Else
sql = "SELECT S_Visita.Cedula FROM S_Visita WHERE S_Visita.Cedula = '" &
txtFields0 & "'"
Set rs = db.OpenRecordset(sql, dbOpenDynaset)
If rs.RecordCount = 0 Then
MsgBox "Esta Clave de Cédula No Existe, favor de verificar,
Gracias.", 48, "siap"
C Restaura
Exit Sub
Else
Data1.Refresh

```

APÉNDICE B
GLOSARIO DE TÉRMINOS

GLOSARIO DE TERMINOS

ATRIBUTO

Identifica propiedades de las entidades.

CODASYL

Conference On Data Systems Languages
Conferencia sobre lenguajes de sistemas de datos.

DBA

Data Base Administrator
Administrador de Base de Datos

DBMS

Data Base Multituser System
Base de Datos de Sistemas Multiusuarios

DDL

Data Definition Language
Lenguaje de Definición de Datos

DEADLOCK

Bloqueo mortal. Contención de recursos (estancamiento) provocado por el uso de bloqueos de registros y archivos en aplicaciones multiusuario.

DICCIONARIO DE DATOS

Es un archivo que contiene "datos acerca de los datos"

DML

Data Manipulation Language
Lenguaje de Manejo de Datos

ENTIDAD

Objeto sobre el cual se almacena información. Puede representar algo real, tangible o abstracto.

FORMULARIO

Es un objeto que se puede utilizar para introducir, cambiar y ver registros de datos en la pantalla o en forma impresa.

LLAVE PRIMARIA

Es el o los atributos que definen unívocamente cada registro en las entidades

OBJETO

Es una unidad identificable, tal como una tabla, un formulario un informe o un control que se puede seleccionar y manipular como una unidad.

QUERY

Consulta

REGISTRO

Conjunto de atributos que pertenecen a cada elemento de una entidad.

SELECT

Operación de seleccionar columnas de una tabla.

SQL

Structured Query Language
Lenguaje de Consulta Estructurada

TABLA

Es un objeto de Microsoft Access que almacena datos en filas (registros) y columnas (campos). Generalmente la información se refiere a un tema en particular, tal como los empleados o las ventas.

UNLOCK

Desbloqueo

APÉNDICE C
DICCIONARIO DE DATOS

Propiedades

Def. actualizable: Verdadero Fecha de creación: 6/06/98 9:40:22 PM
RecordCount: 0 Última actualización: 6/06/98 10:44:30 PM

Columnas

Nombre	Tipo	Tamaño
Cedula	Texto	10
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Cedula	
ColumnHidden:	Falso	
ColumnOrder:	1	
ColumnWidth:	Predeterminado	
Orden de combinación:	General	
Posición ordinal:	0	
Required:	Falso	
Tabla de origen:	Croquis	
Ruta_uno	Texto	50
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Ruta_uno	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	2685	
Orden de combinación:	General	
Posición ordinal:	1	
Required:	Falso	
Tabla de origen:	Croquis	

Permisos de usuario

admin

Permisos de grupo

Admins
Users

Propiedades

Def. actualizable: Verdadero Fecha de creación: 7/03/97 6:06:13 PM
 RecordCount: 0 Última actualización: 29/06/98 6:27:42 PM

Columnas

Nombre	Tipo	Tamaño
Cedula	Texto	15
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Cedula	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Clave de la Cedula (Documento).	
Orden de combinación:	General	
Posición ordinal:	1	
Required:	Falso	
Tabla de origen:	S_Colindancia	
Punto_Cardinal	Texto	10
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Punto_Cardinal	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Orientación.	
Orden de combinación:	General	
Posición ordinal:	2	
Required:	Falso	
Tabla de origen:	S_Colindancia	
Colindancia	Texto	50
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Colindancia	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	2100	
Descripción:	Limita con	
Orden de combinación:	General	
Posición ordinal:	3	
Required:	Falso	
Tabla de origen:	S_Colindancia	
Punto_Cardinal1	Texto	10
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Punto_Cardinal1	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	

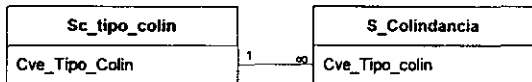
ColumnWidth:	Predeterminado		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	Falso		
Tabla de origen:	S_Colindancia		
Colindancia1		Texto	50
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Colindancia1		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	General		
Posición ordinal:	5		
Required:	Falso		
Tabla de origen:	S_Colindancia		
Punto_Cardinal2		Texto	10
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Punto_Cardinal2		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	General		
Posición ordinal:	6		
Required:	Falso		
Tabla de origen:	S_Colindancia		
Colindancia2		Texto	50
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Colindancia2		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	General		
Posición ordinal:	7		
Required:	Falso		
Tabla de origen:	S_Colindancia		
Punto_Cardinal3		Texto	10
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Punto_Cardinal3		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	General		
Posición ordinal:	8		
Required:	Falso		
Tabla de origen:	S_Colindancia		
Colindancia3		Texto	50
AllowZeroLength:	Falso		

Atributos:	Longitud variable		
Campo de origen:	Colindancia3		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	General		
Posición ordinal:	9		
Required:	Falso		
Tabla de origen:	S_Colindancia		
Cve_Tipo_colin		Número (largo)	4
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Cve_Tipo_colin		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Descripción:	Clave del tipo de colindancia.		
Lugares decimales:	Automático		
Orden de combinación:	Neutro		
Posición ordinal:	10		
Required:	Falso		
Tabla de origen:	S_Colindancia		
Valor predeterminado:	0		
Superficie		Número (largo)	4
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Superficie		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Descripción:	Superficie (lineal).		
Lugares decimales:	Automático		
Orden de combinación:	Neutro		
Posición ordinal:	11		
Required:	Falso		
Tabla de origen:	S_Colindancia		
Valor predeterminado:	0		
Superficie1		Número (largo)	4
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Superficie1		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Lugares decimales:	Automático		
Orden de combinación:	Neutro		
Posición ordinal:	12		
Required:	Falso		
Tabla de origen:	S_Colindancia		
Valor predeterminado:	0		
Superficie2		Número (largo)	4
AllowZeroLength:	Falso		

Atributos: Campo de origen: ColumnHidden: ColumnOrder: ColumnWidth: Lugares decimales: Orden de combinación: Posición ordinal: Required: Tabla de origen: Valor predeterminado:	Tamaño fijo Superficie2 Falso Predeterminado Predeterminado Automático Neutro 13 Falso S_Colindancia 0		
 		Superficie3	Número (largo)
AllowZeroLength: Atributos: Campo de origen: ColumnHidden: ColumnOrder: ColumnWidth: Lugares decimales: Orden de combinación: Posición ordinal: Required: Tabla de origen: Valor predeterminado:	Falso Tamaño fijo Superficie3 Falso Predeterminado Predeterminado Automático Neutro 14 Falso S_Colindancia 0		4
 		Superficie_tot	Número (largo)
AllowZeroLength: Atributos: Campo de origen: ColumnHidden: ColumnOrder: ColumnWidth: Formato: Lugares decimales: Orden de combinación: Posición ordinal: Required: Tabla de origen: Valor predeterminado:	Falso Tamaño fijo Superficie_tot Falso Predeterminado Predeterminado #,##0 Automático Neutro 15 Falso S_Colindancia 0		4

Relaciones

References



Atributos: Exigir
 Atributos: Uno a varios

Índices de tabla

<u>Nombre</u>	<u>Número de campos</u>
Reference5	1
Agrupado:	Falso
Cuenta distinta:	0
Extranjero:	Verdadero
Ignorar Nulos:	Falso
Nombre:	Reference5
Principal:	Falso
Required:	Falso
Única:	Falso
Campos:	Cve_Tipo_colin, Ascendente

Permisos de usuario

admin

Permisos de grupo

Admins
Users

Propiedades

Def. actualizable: Verdadero Fecha de creación: 19/11/96 5:43:27 PM
 RecordCount: 0 Última actualización: 29/05/98 11:44:50 PM

Columnas

Nombre	Tipo	Tamaño
Cedula	Texto	10
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Cedula	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Clave de la Cedula (Documento).	
Orden de combinación:	General	
Posición ordinal:	1	
Required:	Falso	
Tabla de origen:	S_Doc_Dig	
Documento	Objeto OLE	
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Documento	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Fotografia,Scan,Icono,.BMP	
Orden de combinación:	Neutro	
Posición ordinal:	2	
Required:	Falso	
Tabla de origen:	S_Doc_Dig	
Título:	Docto_1	
Cve_Tipo_Doc	Número (largo)	4
AllowZeroLength:	Falso	
Atributos:	Tamaño fijo	
Campo de origen:	Cve_Tipo_Doc	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Clave del tipo del docto.	
Lugares decimales:	Automático	
Orden de combinación:	Neutro	
Posición ordinal:	3	
Required:	Falso	
Tabla de origen:	S_Doc_Dig	
Valor predeterminado:	0	

<u>Nombre</u>	<u>Número de campos</u>
PrimaryKey	1
Agrupado:	Falso
Cuenta distinta:	0
Extranjero:	Falso
Ignorar Nulos:	Falso
Nombre:	PrimaryKey
Principal:	Verdadero
Required:	Verdadero
Única:	Verdadero
Campos:	Cedula, Ascendente

Permisos de usuario

admin

Permisos de grupo

Admins

Users

Propiedades

AltoDeLaFila: 195 Def. actualizable: Verdadero
 Fecha de creación: 19/11/96 5:59:12 PM RecordCount: 2
 Última actualización: 24/06/98 8:48:12 PM

Columnas

Nombre	Tipo	Tamaño
Cedula	Texto	10
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Cedula	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Clave de la Cedula (Documento).	
Orden de combinación:	General	
Posición ordinal:	1	
Required:	Falso	
Tabla de origen:	S_Expdiente	
Num_Expdte	Texto	10
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Num_Expdte	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Clave o Número del Expediente.	
Orden de combinación:	General	
Posición ordinal:	2	
Required:	Falso	
Tabla de origen:	S_Expdiente	
Archivero	Texto	10
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Archivero	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Clave del archivero.	
Orden de combinación:	General	
Posición ordinal:	3	
Required:	Falso	
Tabla de origen:	S_Expdiente	
Gaveta	Texto	10
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Gaveta	
ColumnHidden:	Falso	

ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Descripción:	Clave de la gaveta.		
Orden de combinación:	General		
Posición ordinal:	4		
Required:	Falso		
Tabla de origen:	S_Expediente		
Total_fojas		Número (largo)	4
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Total_fojas		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Descripción:	Total de hojas.		
Lugares decimales:	Automático		
Orden de combinación:	Neutro		
Posición ordinal:	5		
Required:	Falso		
Tabla de origen:	S_Expediente		
Valor predeterminado:	0		
Ext_Nuevo		Sí/No	1
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Ext_Nuevo		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Formato:	Sí/No		
Orden de combinación:	Neutro		
Posición ordinal:	6		
Required:	Falso		
Tabla de origen:	S_Expediente		
Ext_DDF		Sí/No	1
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Ext_DDF		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Formato:	Sí/No		
Orden de combinación:	Neutro		
Posición ordinal:	7		
Required:	Falso		
Tabla de origen:	S_Expediente		
Ext_Particular		Sí/No	1
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Ext_Particular		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		

Formato: Si/No
 Orden de combinación: Neutro
 Posición ordinal: 8
 Required: Falso
 Tabla de origen: S_Expediente

Ext_Federal Si/No 1

AllowZeroLength: Falso
 Atributos: Tamaño fijo
 Campo de origen: Ext_Federal
 ColumnHidden: Falso
 ColumnOrder: Predeterminado
 ColumnWidth: Predeterminado
 Formato: Si/No
 Orden de combinación: Neutro
 Posición ordinal: 9
 Required: Falso
 Tabla de origen: S_Expediente

Ext_STC Si/No 1

AllowZeroLength: Falso
 Atributos: Tamaño fijo
 Campo de origen: Ext_STC
 ColumnHidden: Falso
 ColumnOrder: Predeterminado
 ColumnWidth: Predeterminado
 Formato: Si/No
 Orden de combinación: Neutro
 Posición ordinal: 10
 Required: Falso
 Tabla de origen: S_Expediente

Ext_Desincorporado Si/No 1

AllowZeroLength: Falso
 Atributos: Tamaño fijo
 Campo de origen: Ext_Desincorporado
 ColumnHidden: Falso
 ColumnOrder: Predeterminado
 ColumnWidth: Predeterminado
 Formato: Si/No
 Orden de combinación: Neutro
 Posición ordinal: 11
 Required: Falso
 Tabla de origen: S_Expediente

Índices de tabla

Nombre	Número de campos
PrimaryKey	1
Agrupado:	Falso
Cuenta distinta:	2
Extranjero:	Falso
Ignorar Nulos:	Falso
Nombre:	PrimaryKey

Principal:	Verdadero
Required:	Verdadero
Única:	Verdadero
Campos:	Cedula, Ascendente

Permisos de usuario

admin

Permisos de grupo

Admins
Users

Propiedades

Def. actualizable: Verdadero Fecha de creación: 7/03/97 6:06:30 PM
 RecordCount: 0 Última actualización: 27/05/98 3:30:18 PM

Columnas

Nombre	Tipo	Tamaño
Cedula	Texto	10
AllowZeroLength: Falso Atributos: Longitud variable Campo de origen: Cedula ColumnHidden: Falso ColumnOrder: Predeterminado ColumnWidth: Predeterminado Descripción: Clave de la Cedula (Documento). Orden de combinación: General Posición ordinal: 1 Required: Falso Tabla de origen: S_Foto		
Foto	Objeto OLE	-
AllowZeroLength: Falso Atributos: Longitud variable Campo de origen: Foto ColumnHidden: Falso ColumnOrder: Predeterminado ColumnWidth: Predeterminado Orden de combinación: Neutro Posición ordinal: 2 Required: Falso Tabla de origen: S_Foto		

Índices de tabla

Nombre	Número de campos
PrimaryKey	1
Agrupado: Falso Cuenta distinta: 0 Extranjero: Falso Ignorar Nulos: Falso Nombre: PrimaryKey Principal: Verdadero Required: Verdadero Única: Verdadero Campos: Cedula, Ascendente	

Permisos de usuario

admin

Permisos de grupo

Admins

Users

Propiedades

Def. actualizable: Verdadero Fecha de creación: 29/04/98 7:53:18 PM
 RecordCount: 1 Última actualización: 22/05/98 12:28:45 AM

Columnas

Nombre	Tipo	Tamaño
Cedula	Texto	10
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Cedula	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Clave de la Cedula (Docto)	
Orden de combinación:	General	
Posición ordinal:	1	
Required:	Falso	
Tabla de origen:	S_Juridico	
Fecha_Convenio	Fecha/Hora	8
AllowZeroLength:	Falso	
Atributos:	Tamaño fijo	
Campo de origen:	Fecha_Convenio	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Fecha en que se hizo el convenio.	
Orden de combinación:	Neutro	
Posición ordinal:	2	
Required:	Falso	
Tabla de origen:	S_Juridico	
Fecha_Contrato	Fecha/Hora	8
AllowZeroLength:	Falso	
Atributos:	Tamaño fijo	
Campo de origen:	Fecha_Contrato	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Fecha en que se hizo el contrato.	
Orden de combinación:	Neutro	
Posición ordinal:	3	
Required:	Falso	
Tabla de origen:	S_Juridico	
Fecha_Decreto	Fecha/Hora	8
AllowZeroLength:	Falso	
Atributos:	Tamaño fijo	
Campo de origen:	Fecha_Decreto	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	

ColumnWidth:	Predeterminado		
Descripción:	Fecha en que se decreto		
Orden de combinación:	Neutro		
Posición ordinal:	4		
Required:	Falso		
Tabla de origen:	S_Juridico		
Fecha_Publicacion		Fecha/Hora	8
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Fecha_Publicacion		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Descripción:	Fecha de publicación (Conocimiento).		
Orden de combinación:	Neutro		
Posición ordinal:	5		
Required:	Falso		
Tabla de origen:	S_Juridico		
Ley_organica		Texto	30
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Ley_organica		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Descripción:	Ley orgánica.		
Orden de combinación:	General		
Posición ordinal:	6		
Required:	Falso		
Tabla de origen:	S_Juridico		
Fecha_Escritura		Fecha/Hora	8
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Fecha_Escritura		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Descripción:	Fecha de las escrituras(De su Expedición o Elaboración).		
Orden de combinación:	Neutro		
Posición ordinal:	7		
Required:	Falso		
Tabla de origen:	S_Juridico		
Nomb_Escritura		Texto	20
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Nomb_Escritura		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Descripción:	Descripcion		
Orden de combinación:	General		
Posición ordinal:	8		

Required:	Falso		
Tabla de origen:	S_Juridico		
Num_Notaria		Texto	20
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Num_Notaria		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Descripción:	Número de la Notaria.		
Orden de combinación:	General		
Posición ordinal:	9		
Required:	Falso		
Tabla de origen:	S_Juridico		
Nomb_Notario		Texto	50
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Nomb_Notario		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Descripción:	Nombre del Notario (Persona).		
Orden de combinación:	General		
Posición ordinal:	10		
Required:	Falso		
Tabla de origen:	S_Juridico		
Secion_Cpi		Texto	20
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Secion_Cpi		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	General		
Posición ordinal:	11		
Required:	Falso		
Tabla de origen:	S_Juridico		
Fecha_Cpi		Fecha/Hora	8
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Fecha_Cpi		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	Neutro		
Posición ordinal:	12		
Required:	Falso		
Tabla de origen:	S_Juridico		
Responsable_Cpi		Texto	50
AllowZeroLength:	Falso		
Atributos:	Longitud variable		

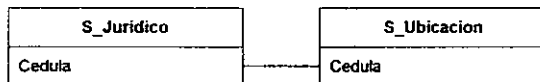
Campo de origen: ColumnHidden: ColumnOrder: ColumnWidth: Orden de combinación: Posición ordinal: Required: Tabla de origen:	Responsable_Cpi Falso Predeterminado Predeterminado General 13 Falso S_Juridico		
Propuesta_Cpi		Texto	30
AllowZeroLength: Atributos: Campo de origen: ColumnHidden: ColumnOrder: ColumnWidth: Orden de combinación: Posición ordinal: Required: Tabla de origen:	Falso Longitud variable Propuesta_Cpi Falso Predeterminado Predeterminado General 14 Falso S_Juridico		
Acuerdo_Cpi		Texto	30
AllowZeroLength: Atributos: Campo de origen: ColumnHidden: ColumnOrder: ColumnWidth: Orden de combinación: Posición ordinal: Required: Tabla de origen:	Falso Longitud variable Acuerdo_Cpi Falso Predeterminado Predeterminado General 15 Falso S_Juridico		
Permiso_Cpi		Texto	30
AllowZeroLength: Atributos: Campo de origen: ColumnHidden: ColumnOrder: ColumnWidth: Orden de combinación: Posición ordinal: Required: Tabla de origen:	Falso Longitud variable Permiso_Cpi Falso Predeterminado Predeterminado General 16 Falso S_Juridico		
Tipo_Cpi		Texto	20
AllowZeroLength: Atributos: Campo de origen: ColumnHidden: ColumnOrder: ColumnWidth: Orden de combinación: Posición ordinal: Required: Tabla de origen:	Falso Longitud variable Tipo_Cpi Falso Predeterminado Predeterminado General 17 Falso S_Juridico		

Elaboro		Texto	50
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Elaboro		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Descripción:	Nombre		
Orden de combinación:	General		
Posición ordinal:	18		
Required:	Falso		
Tabla de origen:	S_Juridico		
Fecha_Elaboracion		Fecha/Hora	8
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Fecha_Elaboracion		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	Neutro		
Posición ordinal:	19		
Required:	Falso		
Tabla de origen:	S_Juridico		
Autorizacion		Texto	50
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Autorizacion		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	General		
Posición ordinal:	20		
Required:	Falso		
Tabla de origen:	S_Juridico		
Fecha_Autorizacion		Fecha/Hora	8
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Fecha_Autorizacion		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	Neutro		
Posición ordinal:	21		
Required:	Falso		
Tabla de origen:	S_Juridico		
Observaciones		Texto	255
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Observaciones		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		

Orden de combinación: General
 Posición ordinal: 22
 Required: Falso
 Tabla de origen: S_Juridico

Relaciones

Reference4



Atributos: Único, No exigir
 Atributos: Uno a uno

Índices de tabla

Nombre	Número de campos
Cedula	1
Agrupado:	Falso
Cuenta distinta:	1
Extranjero:	Falso
Ignorar Nulos:	Falso
Nombre:	Cedula
Principal:	Falso
Required:	Falso
Única:	Falso
Campos:	Cedula, Ascendente
PrimaryKey	1
Agrupado:	Falso
Cuenta distinta:	1
Extranjero:	Falso
Ignorar Nulos:	Falso
Nombre:	PrimaryKey
Principal:	Verdadero
Required:	Verdadero
Única:	Verdadero
Campos:	Cedula, Ascendente

Permisos de usuario

admin

Permisos de grupo

Admins
Users

Propiedades

Def. actualizable: Verdadero Fecha de creación: 1/09/97 6:08:48 PM
 RecordCount: 0 Última actualización: 29/04/98 8:32:04 PM

Columnas

Nombre	Tipo	Tamaño
Cedula	Texto	10
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Cedula	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Clave de la Cedula (Documento).	
Orden de combinación:	General	
Posición ordinal:	1	
Required:	Falso	
Tabla de origen:	S_Planos	
Croquis	Objeto OLE	
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Croquis	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	3210	
Descripción:	Foto	
Orden de combinación:	Neutro	
Posición ordinal:	2	
Required:	Falso	
Tabla de origen:	S_Planos	
Titulo:	Croquis_1	
Plano	Objeto OLE	
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Plano	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Foto	
Orden de combinación:	Neutro	
Posición ordinal:	3	
Required:	Falso	
Tabla de origen:	S_Planos	
Titulo:	Plano_1	

Índices de tabla

Nombre	Número de campos
--------	------------------

PrimaryKey	1
Agrupado:	Falso
Cuenta distinta:	0
Extranjero:	Falso
Ignorar Nulos:	Falso
Nombre:	PrimaryKey
Principal:	Verdadero
Required:	Verdadero
Única:	Verdadero
Campos:	Cedula, Ascendente

Permisos de usuario

admin

Permisos de grupo

Admins

Users

Propiedades

AltoDeLaFila: 180 Def. actualizable: Verdadero
 Fecha de creación: 19/11/96 5:28:00 PM RecordCount: 4
 Última actualización: 24/06/98 8:48:36 PM

Columnas

Nombre	Tipo	Tamaño
Cedula	Texto	10
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Cedula	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	No. de Cedula del docto o expediente.	
Orden de combinación:	General	
Posición ordinal:	1	
Required:	Falso	
Tabla de origen:	S_Tipo_Exp	
Documentos	Texto	10
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Documentos	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Documentos.	
Orden de combinación:	General	
Posición ordinal:	2	
Required:	Falso	
Tabla de origen:	S_Tipo_Exp	
Fojas	Número (largo)	4
AllowZeroLength:	Falso	
Atributos:	Tamaño fijo	
Campo de origen:	Fojas	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Número de hojas de que consta.	
Lugares decimales:	Automático	
Orden de combinación:	Neutro	
Posición ordinal:	3	
Required:	Falso	
Tabla de origen:	S_Tipo_Exp	
Valor predeterminado:	0	
Cve_Tipo_doc	Número (largo)	4
AllowZeroLength:	Falso	
Atributos:	Tamaño fijo	

Campo de origen:	Cve_Tipo_doc
ColumnHidden:	Falso
ColumnOrder:	Predeterminado
ColumnWidth:	Predeterminado
Descripción:	Clave del Tipo de documento.
Lugares decimales:	Automático
Orden de combinación:	Neutro
Posición ordinal:	4
Required:	Falso
Tabla de origen:	S_Tipo_Exp

Permisos de usuario

admin

Permisos de grupo

Admins
Users

Propiedades

Def. actualizable: Verdadero Fecha de creación: 29/04/98 8:12:35 PM
 RecordCount: 1 Última actualización: 24/05/98 1:31:44 PM

Columnas

Nombre	Tipo	Tamaño
Cedula	Texto	10
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Cedula	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Clave de la Cedula (Documento).	
Orden de combinación:	General	
Posición ordinal:	1	
Required:	Falso	
Tabla de origen:	S_Ubicacion	
Cve_Estacion	Texto	8
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Cve_Estacion	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Orden de combinación:	General	
Posición ordinal:	2	
Required:	Falso	
Tabla de origen:	S_Ubicacion	
Calle	Texto	70
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Calle	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Orden de combinación:	General	
Posición ordinal:	3	
Required:	Falso	
Tabla de origen:	S_Ubicacion	
Numero	Texto	6
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Numero	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Orden de combinación:	General	

	Posición ordinal:	4		
	Required:	Falso		
	Tabla de origen:	S_Ubicacion		
Numero_Interior			Texto	6
	AllowZeroLength:	Falso		
	Atributos:	Longitud variable		
	Campo de origen:	Numero_Interior		
	ColumnHidden:	Falso		
	ColumnOrder:	Predeterminado		
	ColumnWidth:	Predeterminado		
	Orden de combinación:	General		
	Posición ordinal:	5		
	Required:	Falso		
	Tabla de origen:	S_Ubicacion		
Lote			Texto	6
	AllowZeroLength:	Falso		
	Atributos:	Longitud variable		
	Campo de origen:	Lote		
	ColumnHidden:	Falso		
	ColumnOrder:	Predeterminado		
	ColumnWidth:	Predeterminado		
	Orden de combinación:	General		
	Posición ordinal:	6		
	Required:	Falso		
	Tabla de origen:	S_Ubicacion		
Manzana			Texto	6
	AllowZeroLength:	Falso		
	Atributos:	Longitud variable		
	Campo de origen:	Manzana		
	ColumnHidden:	Falso		
	ColumnOrder:	Predeterminado		
	ColumnWidth:	Predeterminado		
	Orden de combinación:	General		
	Posición ordinal:	7		
	Required:	Falso		
	Tabla de origen:	S_Ubicacion		
Colonia			Texto	70
	AllowZeroLength:	Falso		
	Atributos:	Longitud variable		
	Campo de origen:	Colonia		
	ColumnHidden:	Falso		
	ColumnOrder:	Predeterminado		
	ColumnWidth:	Predeterminado		
	Orden de combinación:	General		
	Posición ordinal:	8		
	Required:	Falso		
	Tabla de origen:	S_Ubicacion		
Entre_Calles			Texto	80
	AllowZeroLength:	Falso		
	Atributos:	Longitud variable		
	Campo de origen:	Entre_Calles		

ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	General		
Posición ordinal:	9		
Required:	Falso		
Tabla de origen:	S_Ubicacion		
Codigo_postal		Texto	5
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Codigo_postal		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	General		
Posición ordinal:	10		
Required:	Falso		
Tabla de origen:	S_Ubicacion		
Cve_Delegacion		Número (largo)	4
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Cve_Delegacion		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Lugares decimales:	Automático		
Orden de combinación:	Neutro		
Posición ordinal:	11		
Required:	Falso		
Tabla de origen:	S_Ubicacion		
Valor predeterminado:	0		
Cta_Anterior		Texto	20
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Cta_Anterior		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	General		
Posición ordinal:	12		
Required:	Falso		
Tabla de origen:	S_Ubicacion		
Cuenta_Actual		Texto	20
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Cuenta_Actual		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	General		
Posición ordinal:	13		
Required:	Falso		

Tabla de origen:	S_Ubicacion	
Fecha_Captura	Fecha/Hora	8
AllowZeroLength:	Falso	
Atributos:	Tamaño fijo	
Campo de origen:	Fecha_Captura	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Orden de combinación:	Neutro	
Posición ordinal:	14	
Required:	Falso	
Tabla de origen:	S_Ubicacion	

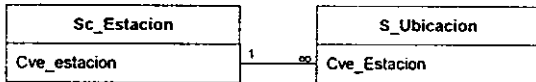
Relaciones

Reference



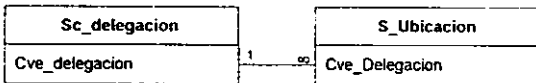
Atributos: Único, No exigir
Atributos: Indeterminado

Reference2



Atributos: Exigir
Atributos: Uno a varios

Reference3



Atributos: Exigir
Atributos: Uno a varios

Reference4



Atributos: Único, No exigir
 Atributos: Uno a uno

Índices de tabla

Nombre	Número de campos
PrimaryKey	1
Agrupado:	Falso
Cuenta distinta:	1
Extranjero:	Falso
Ignorar Nulos:	Falso
Nombre:	PrimaryKey
Principal:	Verdadero
Required:	Verdadero
Única:	Verdadero
Campos:	Cedula, Ascendente
Reference2	1
Agrupado:	Falso
Cuenta distinta:	1
Extranjero:	Verdadero
Ignorar Nulos:	Falso
Nombre:	Reference2
Principal:	Falso
Required:	Falso
Única:	Falso
Campos:	Cve_Estacion, Ascendente
Reference3	1
Agrupado:	Falso
Cuenta distinta:	1
Extranjero:	Verdadero
Ignorar Nulos:	Falso
Nombre:	Reference3
Principal:	Falso
Required:	Falso
Única:	Falso
Campos:	Cve_Delegacion, Ascendente

Permisos de usuario

admin

Permisos de grupo

Admins
Users

Propiedades

Def. actualizable: Verdadero Fecha de creación: 29/04/98 8:03:58 PM
 RecordCount: 1 Última actualización: 25/06/98 6:44:45 PM

Columnas

Nombre	Tipo	Tamaño
Cedula	Texto	10
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Cedula	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Clave de la Cedula (Documento).	
Orden de combinación:	General	
Posición ordinal:	1	
Required:	Falso	
Tabla de origen:	S_Visita	
Superficie	Número (largo)	4
AllowZeroLength:	Falso	
Atributos:	Tamaño fijo	
Campo de origen:	Superficie	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Lugares decimales:	Automático	
Orden de combinación:	Neutro	
Posición ordinal:	2	
Required:	Falso	
Tabla de origen:	S_Visita	
Valor predeterminado:	0	
Superficie_Construida	Número (largo)	4
AllowZeroLength:	Falso	
Atributos:	Tamaño fijo	
Campo de origen:	Superficie_Construida	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Lugares decimales:	Automático	
Orden de combinación:	Neutro	
Posición ordinal:	3	
Required:	Falso	
Tabla de origen:	S_Visita	
Valor predeterminado:	0	
Cve_Tipo_Inmueble	Número (largo)	4
AllowZeroLength:	Falso	
Atributos:	Tamaño fijo	
Campo de origen:	Cve_Tipo_Inmueble	

ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Lugares decimales:	Automático		
Orden de combinación:	Neutro		
Posición ordinal:	4		
Required:	Falso		
Tabla de origen:	S_Visita		
Valor predeterminado:	0		
Niveles		Número (largo)	4
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Niveles		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Lugares decimales:	Automático		
Orden de combinación:	Neutro		
Posición ordinal:	5		
Required:	Falso		
Tabla de origen:	S_Visita		
Valor predeterminado:	0		
Nombre_Inmueble		Texto	50
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Nombre_Inmueble		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	General		
Posición ordinal:	6		
Required:	Falso		
Tabla de origen:	S_Visita		
Uso_Inicial		Texto	50
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Uso_Inicial		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	General		
Posición ordinal:	7		
Required:	Falso		
Tabla de origen:	S_Visita		
Uso_Actual		Texto	50
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Uso_Actual		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	General		

	Posición ordinal:	8		
	Required:	Falso		
	Tabla de origen:	S_Visita		
Propietario			Texto	70
	AllowZeroLength:	Falso		
	Atributos:	Longitud variable		
	Campo de origen:	Propietario		
	ColumnHidden:	Falso		
	ColumnOrder:	Predeterminado		
	ColumnWidth:	Predeterminado		
	Orden de combinación:	General		
	Posición ordinal:	9		
	Required:	Falso		
	Tabla de origen:	S_Visita		
Usuario			Texto	70
	AllowZeroLength:	Falso		
	Atributos:	Longitud variable		
	Campo de origen:	Usuario		
	ColumnHidden:	Falso		
	ColumnOrder:	Predeterminado		
	ColumnWidth:	Predeterminado		
	Orden de combinación:	General		
	Posición ordinal:	10		
	Required:	Falso		
	Tabla de origen:	S_Visita		
Fecha_peticion			Fecha/Hora	8
	AllowZeroLength:	Falso		
	Atributos:	Tamaño fijo		
	Campo de origen:	Fecha_peticion		
	ColumnHidden:	Falso		
	ColumnOrder:	Predeterminado		
	ColumnWidth:	Predeterminado		
	Orden de combinación:	Neutro		
	Posición ordinal:	11		
	Required:	Falso		
	Tabla de origen:	S_Visita		
Fecha_Realizacion			Fecha/Hora	8
	AllowZeroLength:	Falso		
	Atributos:	Tamaño fijo		
	Campo de origen:	Fecha_Realizacion		
	ColumnHidden:	Falso		
	ColumnOrder:	Predeterminado		
	ColumnWidth:	Predeterminado		
	Orden de combinación:	Neutro		
	Posición ordinal:	12		
	Required:	Falso		
	Tabla de origen:	S_Visita		
Fecha_Conclusion			Fecha/Hora	8
	AllowZeroLength:	Falso		
	Atributos:	Tamaño fijo		
	Campo de origen:	Fecha_Conclusion		

ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	Neutro		
Posición ordinal:	13		
Required:	Falso		
Tabla de origen:	S_Visita		
Fecha_Confirmacion		Fecha/Hora	8
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Fecha_Confirmacion		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	Neutro		
Posición ordinal:	14		
Required:	Falso		
Tabla de origen:	S_Visita		
Elaboro		Texto	70
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Elaboro		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	General		
Posición ordinal:	15		
Required:	Falso		
Tabla de origen:	S_Visita		
Fecha_Elaboracion		Fecha/Hora	8
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Fecha_Elaboracion		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	Neutro		
Posición ordinal:	16		
Required:	Falso		
Tabla de origen:	S_Visita		
Autorizo		Texto	70
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Autorizo		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	General		
Posición ordinal:	17		
Required:	Falso		
Tabla de origen:	S_Visita		

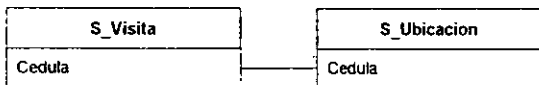
Fecha_Autorizacion		Fecha/Hora	8
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Fecha_Autorizacion		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	Neutro		
Posición ordinal:	18		
Required:	Falso		
Tabla de origen:	S_Visita		
Observaciones		Texto	255
AllowZeroLength:	Falso		
Atributos:	Longitud variable		
Campo de origen:	Observaciones		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Orden de combinación:	General		
Posición ordinal:	19		
Required:	Falso		
Tabla de origen:	S_Visita		
Agua		Sí/No	1
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Agua		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Formato:	Sí/No		
Orden de combinación:	Neutro		
Posición ordinal:	20		
Required:	Falso		
Tabla de origen:	S_Visita		
Luz		Sí/No	1
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Luz		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		
ColumnWidth:	Predeterminado		
Formato:	Sí/No		
Orden de combinación:	Neutro		
Posición ordinal:	21		
Required:	Falso		
Tabla de origen:	S_Visita		
Drenaje		Sí/No	1
AllowZeroLength:	Falso		
Atributos:	Tamaño fijo		
Campo de origen:	Drenaje		
ColumnHidden:	Falso		
ColumnOrder:	Predeterminado		

ColumnWidth: Predeterminado
 Formato: Si/No
 Orden de combinación: Neutro
 Posición ordinal: 22
 Required: Falso
 Tabla de origen: S_Visita

Cve_Edo_Fisico Número (largo) 4
 AllowZeroLength: Falso
 Atributos: Tamaño fijo
 Campo de origen: Cve_Edo_Fisico
 ColumnHidden: Falso
 ColumnOrder: Predeterminado
 ColumnWidth: Predeterminado
 Lugares decimales: Automático
 Orden de combinación: Neutro
 Posición ordinal: 23
 Required: Falso
 Tabla de origen: S_Visita
 Valor predeterminado: 0

Relaciones

Reference



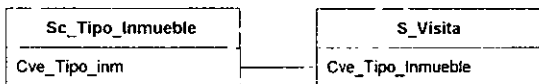
Atributos: Único, No exigir
 Atributos: Indeterminado

Reference1



Atributos: Único, No exigir
 Atributos: Uno a varios

Reference6



Atributos: Único, No exigir
 Atributos: Uno a varios

Índices de tabla

<u>Nombre</u>	<u>Número de campos</u>
Cedula	1
Agrupado:	Falso
Cuenta distinta:	1
Extranjero:	Falso
Ignorar Nulos:	Falso
Nombre:	Cedula
Principal:	Falso
Required:	Falso
Única:	Falso
Campos:	Cedula, Ascendente

Permisos de usuario

admin

Permisos de grupo

Admins

Users

Propiedades

Def. actualizable: Verdadero Fecha de creación: 22/02/97 2:03:49 PM
 RecordCount: 17 Última actualización: 8/05/98 12:18:20 AM

Columnas

Nombre	Tipo	Tamaño
Cve_delegacion	Número (largo)	4
AllowZeroLength:	Falso	
Atributos:	Tamaño fijo	
Campo de origen:	Cve_delegacion	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Clave de la delegación.	
Lugares decimales:	Automático	
Orden de combinación:	Neutro	
Posición ordinal:	1	
Required:	Falso	
Tabla de origen:	Sc_Delegacion	
Descripcion	Texto	40
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Descripcion	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	4245	
Descripción:	Detalle o nombre de la Delegación.	
Orden de combinación:	General	
Posición ordinal:	2	
Required:	Falso	
Tabla de origen:	Sc_Delegacion	

Relaciones

Reference3



Atributos: Exigir
 Atributos: Uno a varios

Índices de tabla

Nombre	Número de campos
--------	------------------

PrimaryKey	1
Agrupado:	Falso
Cuenta distinta:	17
Extranjero:	Falso
Ignorar Nulos:	Falso
Nombre:	PrimaryKey
Principal:	Verdadero
Required:	Verdadero
Única:	Verdadero
Campos:	Cve_delegacion, Ascendente

Permisos de usuario

admin

Permisos de grupo

Admins

Users

Propiedades

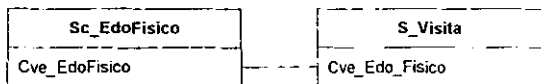
Def. actualizable: Verdadero Fecha de creación: 9/03/97 1:49:34 PM
 RecordCount: 4 Última actualización: 29/04/98 8:33:05 PM

Columnas

Nombre	Tipo	Tamaño
Cve_EdoFisico	Número (fargo)	4
AllowZeroLength:	Falso	
Atributos:	Tamaño fijo	
Campo de origen:	Cve_EdoFisico	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Clave del estado físico.	
Lugares decimales:	Automático	
Orden de combinación:	Neutro	
Posición ordinal:	1	
Required:	Falso	
Tabla de origen:	Sc_EdoFisico	
Valor predeterminado:	0	
Descripcion	Texto	40
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Descripcion	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	3750	
Descripción:	Detalle del estado físico.	
Orden de combinación:	General	
Posición ordinal:	2	
Required:	Falso	
Tabla de origen:	Sc_EdoFisico	

Relaciones

Reference1



Atributos: Único, No exigir
 Atributos: Uno a varios

Índices de tabla

Nombre	Número de campos
PrimaryKey	1
Agrupado:	Falso
Cuenta distinta:	4
Extranjero:	Falso
Ignorar Nulos:	Falso
Nombre:	PrimaryKey
Principal:	Verdadero
Required:	Verdadero
Única:	Verdadero
Campos:	Cve_EdoFisico, Ascendente

Permisos de usuario

admin

Permisos de grupo

Admins

Users

Propiedades

Def. actualizable: Verdadero Fecha de creación: 9/03/97 1:48:49 PM
 RecordCount: 176 Última actualización: 21/06/98 11:30:17 AM

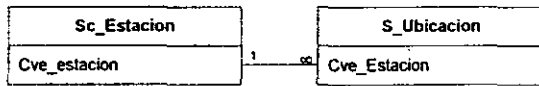
Columnas

Nombre	Tipo	Tamaño
Cve_estacion	Texto	8
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Cve_estacion	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Clave de la estación	
Orden de combinación:	General	
Posición ordinal:	1	
Required:	Falso	
Tabla de origen:	Sc_Estacion	
Linea	Texto	15
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Linea	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	3780	
Descripción:	Clave de la linea.	
Orden de combinación:	General	
Posición ordinal:	2	
Required:	Falso	
Tabla de origen:	Sc_Estacion	
Estacion	Texto	25
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Estacion	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Nombre de la estación.	
Orden de combinación:	General	
Posición ordinal:	3	
Required:	Falso	
Tabla de origen:	Sc_Estacion	
Cve_Tipo_Estacion	Número (largo)	4
AllowZeroLength:	Falso	
Atributos:	Tamaño fijo	
Campo de origen:	Cve_Tipo_Estacion	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	

ColumnWidth: Predeterminado
 Descripción: Clave del tipo de la estación.
 Lugares decimales: Automático
 Orden de combinación: Neutro
 Posición ordinal: 4
 Required: Falso
 Tabla de origen: Sc_Estacion
 Valor predeterminado: 0

Relaciones

Reference2



Atributos: Exigir
 Atributos: Uno a varios

Índices de tabla

Nombre	Número de campos
Cve_estacion	1
Agrupado:	Falso
Cuenta distinta:	176
Extranjero:	Falso
Ignorar Nulos:	Falso
Nombre:	Cve_estacion
Principal:	Falso
Required:	Falso
Única:	Falso
Campos:	Cve_estacion, Ascendente
PrimaryKey	1
Agrupado:	Falso
Cuenta distinta:	176
Extranjero:	Falso
Ignorar Nulos:	Falso
Nombre:	PrimaryKey
Principal:	Verdadero
Required:	Verdadero
Única:	Verdadero
Campos:	Cve_estacion, Ascendente

Permisos de usuario

admin

Permisos de grupo

Admins
Users

Propiedades

Def. actualizable: Verdadero Fecha de creación: 4/12/96 4:46:56 PM
 RecordCount: 12 Última actualización: 21/06/98 11:29:43 AM

Columnas

Nombre	Tipo	Tamaño
Linea	Texto	15
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Linea	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Linea	
Orden de combinación:	General	
Posición ordinal:	1	
Required:	Falso	
Tabla de origen:	Sc_linea	
Corresp	Texto	40
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Corresp	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	4320	
Descripción:	Correspondencia con	
Orden de combinación:	General	
Posición ordinal:	2	
Required:	Falso	
Tabla de origen:	Sc_linea	

Índices de tabla

Nombre	Número de campos
PrimaryKey	1
Agrupado:	Falso
Cuenta distinta:	12
Extranjero:	Falso
Ignorar Nulos:	Falso
Nombre:	PrimaryKey
Principal:	Verdadero
Required:	Verdadero
Única:	Verdadero
Campos:	Linea, Ascendente

Permisos de usuario

admin

Permisos de grupo

Admins
Users

Propiedades

Def. actualizable: Verdadero Fecha de creación: 24/02/97 12:18:47 PM
 RecordCount: 3 Última actualización: 8/05/98 10:39:21 PM

Columnas

Nombre	Tipo	Tamaño
Cve_Tipo_Colin	Número (largo)	4
AllowZeroLength: Falso Atributos: Tamaño fijo Campo de origen: Cve_Tipo_Colin ColumnHidden: Falso ColumnOrder: Predeterminado ColumnWidth: Predeterminado Descripción: Clave del Tipo de Colindancia. Lugares decimales: Automático Orden de combinación: Neutro Posición ordinal: 1 Required: Falso Tabla de origen: Sc_tipo_colin		
Descripción	Texto	40
AllowZeroLength: Falso Atributos: Longitud variable Campo de origen: Descripción ColumnHidden: Falso ColumnOrder: Predeterminado ColumnWidth: Predeterminado Descripción: Detalle de la ... Orden de combinación: General Posición ordinal: 2 Required: Falso Tabla de origen: Sc_tipo_colin		

Relaciones

Reference5



Atributos: Exigir
 Atributos: Uno a varios

Índices de tabla

Nombre	Número de campos
--------	------------------

PrimaryKey	1
Agrupado:	Falso
Cuenta distinta:	3
Extranjero:	Falso
Ignorar Nulos:	Falso
Nombre:	PrimaryKey
Principal:	Verdadero
Required:	Verdadero
Única:	Verdadero
Campos:	Cve_Tipo_Cotin, Ascendente

Permisos de usuario

admin

Permisos de grupo

Admins
Users

Propiedades

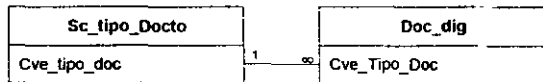
Def. actualizable: Verdadero Fecha de creación: 4/12/96 2:03:56 PM
 RecordCount: 13 Última actualización: 2/05/98 11:37:07 AM

Columnas

Nombre	Tipo	Tamaño
Cve_tipo_doc	Número (largo)	4
AllowZeroLength:	Falso	
Atributos:	Tamaño fijo	
Campo de origen:	Cve_tipo_doc	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Clave del tipo de docto.	
Lugares decimales:	Automático	
Orden de combinación:	Neutro	
Posición ordinal:	1	
Required:	Falso	
Tabla de origen:	Sc_tipo_Docto	
Descripcion	Texto	30
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Descripcion	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	3180	
Descripción:	Detalle del ...	
Orden de combinación:	General	
Posición ordinal:	2	
Required:	Falso	
Tabla de origen:	Sc_tipo_Docto	

Relaciones

Reference10



Atributos: Exigir
 Atributos: Uno a varios

Índices de tabla

Nombre	Número de campos
--------	------------------

PrimaryKey	1
Agrupado:	Falso
Cuenta distinta:	13
Extranjero:	Falso
Ignorar Nulos:	Falso
Nombre:	PrimaryKey
Principal:	Verdadero
Required:	Verdadero
Única:	Verdadero
Campos:	Cve_tipo_doc, Ascendente

Permisos de usuario

admin

Permisos de grupo

Admins
Users

Propiedades

Def. actualizable: Verdadero Fecha de creación: 19/11/96 5:29:51 PM
 RecordCount: 5 Última actualización: 20/06/98 12:36:11 AM

Columnas

Nombre	Tipo	Tamaño
Cve_Tipo_Est	Número (largo)	4
AllowZeroLength:	Falso	
Atributos:	Tamaño fijo	
Campo de origen:	Cve_Tipo_Est	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	Predeterminado	
Descripción:	Clave del Tipo de Estación.	
Lugares decimales:	Automático	
Orden de combinación:	Neutro	
Posición ordinal:	1	
Required:	Falso	
Tabla de origen:	Sc_Tipo_Estacion	
Descripcion	Texto	40
AllowZeroLength:	Falso	
Atributos:	Longitud variable	
Campo de origen:	Descripcion	
ColumnHidden:	Falso	
ColumnOrder:	Predeterminado	
ColumnWidth:	3435	
Descripción:	Detalle de la	
Orden de combinación:	General	
Posición ordinal:	2	
Required:	Falso	
Tabla de origen:	Sc_Tipo_Estacion	

Índices de tabla

Nombre	Número de campos
PrimaryKey	1
Agrupado:	Falso
Cuenta distinta:	5
Extranjero:	Falso
Ignorar Nulos:	Falso
Nombre:	PrimaryKey
Principal:	Verdadero
Required:	Verdadero
Única:	Verdadero
Campos:	Cve_Tipo_Est, Ascendente

Permisos de usuario

admin

Permisos de grupo

Admins
Users

Propiedades

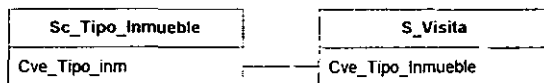
Def. actualizable: Verdadero Fecha de creación: 7/03/97 6:10:46 PM
 RecordCount: 18 Última actualización: 1/05/98 10:34:02 PM

Columnas

Nombre	Tipo	Tamaño
Cve_Tipo_inm	Número (largo)	4
AllowZeroLength: Falso Atributos: Tamaño fijo Campo de origen: Cve_Tipo_inm ColumnHidden: Falso ColumnOrder: 1 ColumnWidth: Predeterminado Descripción: Clave del tipo del inmueble. Lugares decimales: Automático Orden de combinación: Neutro Posición ordinal: 1 Required: Falso Tabla de origen: Sc_Tipo_Inmueble		
Descripción	Texto	30
AllowZeroLength: Falso Atributos: Longitud variable Campo de origen: Descripción ColumnHidden: Falso ColumnOrder: Predeterminado ColumnWidth: 3750 Descripción: Detalle de la ... Orden de combinación: General Posición ordinal: 2 Required: Falso Tabla de origen: Sc_Tipo_Inmueble		

Relaciones

Referencia6



Atributos: Único, No exigir
 Atributos: Uno a varios

Índices de tabla

Nombre _____ Número de campos _____

PrimaryKey	1
Agrupado:	Falso
Cuenta distinta:	18
Extranjero:	Falso
Ignorar Nulos:	Falso
Nombre:	PrimaryKey
Principal:	Verdadero
Required:	Verdadero
Única:	Verdadero
Campos:	Cve_Tipo_inm, Ascendente

Permisos de usuario

admin

Permisos de grupo

Admins

Users

APÉNDICE D
MANUAL DE USUARIO

**Sistema para la Administración del Patrimonio Inmobiliario
del STC Metro**

SIAP

Manual del Usuario

Introducción

CONTENIDO

Acerca del documento

Objetivo

Alcance

Audiencia

Prerequisitos

Como hacer uso del manual

SECCIÓN 1 Introducción al Sistema

SECCIÓN 2 Estructura y módulos del Sistema

SECCIÓN 3 Descripción por módulos

Acerca del Documento

Objetivo

Proveer a todos los usuarios una guía sobre la funcionalidad del Sistema para la Administración del Patrimonio Inmobiliario Metro para lograr una navegación ágil dentro del sistema.

Alcance

Descripción general del sistema.

Descripción detallada de la funcionalidad de cada uno de los siguiente módulos:

- Registro (expedientes, ubicación, visita, jurídico, colindancia, documentos, croquis y planos, estaciones)
- Reportes
- Mantenimiento

Audiencia

Dirigida al personal operativo y administrativo del STC.

Prerequisitos

Conocimiento elemental de una PC.

Conocimiento básico sobre el manejo de Windows 3.x o Windows 95

Control y manejo del Mouse

Conocimiento de políticas y procedimientos de la operación del STC.

Como hacer uso del manual

Para su mayor comprensión este manual esta integrado por las siguientes secciones:

Sección 1 Introducción al Sistema

Esta sección menciona el objetivo principal del Sistema para la Administración del Patrimonio Inmobiliario del STC.

Sección 2 Estructura y módulos de SIAP

En esta sección se menciona los elementos que integran la estructura del sistema y los módulos que lo componen. Contiene una explicación del manejo de las pantallas.

Sección 3 Descripción detallada de los módulos del sistema

Descripción

Sección 1 Introducción al Sistema

Este manual está dirigido al personal operativo y administrativo del STC metro que requiere de utilizar el Sistema para la Administración del Patrimonio Inmobiliario del STC Metro.

El diseño del sistema surge de la necesidad de obtener un adecuado control del registro de los predios que ha obtenido hasta hoy el STC metro.

El sistema permitirá realizar de forma automática el proceso que implica la captura de un predio adquirido por el STC metro y finalmente el estado del inmueble así como la ubicación exacta de cada uno de los predios que ha utilizado el STC metro para todas sus instalaciones.

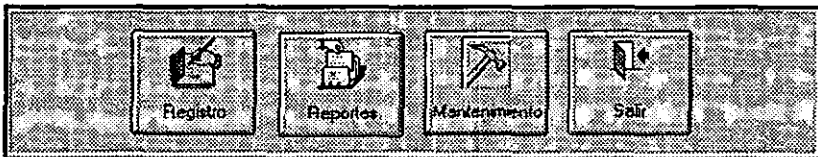
Este manual es una guía que lo llevará paso a paso para involucrarse en el manejo del sistema, tener una navegación ágil y ayudarle a resolver dudas. Servirá como una guía de consulta.

Sección 2 Estructura y módulos del sistema

El sistema SIAP está estructurado por los siguientes elementos: menús, pantallas, botones y mensajes de los cuales se dará a continuación una descripción más detallada.

Menús

Las pantallas del sistema cuentan con menús de iconos, los cuales pueden ser seleccionados al dar un clic en cada uno de ellos oprimiendo el botón izquierdo del Mouse. Estos iconos de menú tienen la particularidad de desplegar las pantallas subsecuentes una vez que el usuario selecciona algún icono del menú principal.



Menú de opciones

Estos menús se encuentran incluidos en la parte superior de la pantalla en forma de iconos integrado por diferentes opciones las cuales se explican a continuación:

Inicio de captura



Inicio de una baja



Modificaciones al expediente



Consulta de expedientes



Reinicio de la pantalla



Salir de este módulo



Casilla de verificación

Una casilla de verificación presenta opciones no excluyentes de las cuales puede seleccionar tantas como necesita. Cuando se selecciona una casilla de verificación, esta contiene una equis (X). Los nombres de las opciones no disponibles aparecen atenuados.

Para seleccionar o desactivar las casillas de verificación:

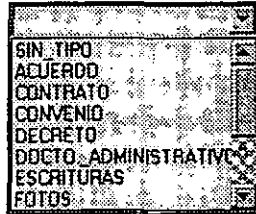
Haga clic en la casilla de verificación.

Nuevo	<input checked="" type="checkbox"/>
D.D.F.	<input checked="" type="checkbox"/>
Particular	<input type="checkbox"/>
Federal	<input checked="" type="checkbox"/>
S.T.C.	<input type="checkbox"/>

Listas desplegables.

Esta listas se presentan como un rectángulo con un cursor de selección. Al seleccionar el botón flecha abajo en el cuadro que aparece a la derecha, aparecerá

una lista de opciones disponibles. Si hay más elementos de los que caben en el cuadro, esta tendrá una barra de desplazamiento.



Barra de desplazamiento

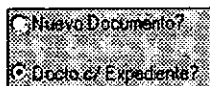
Para abrir un cuadro de lista desplegable y seleccionar un elemento usando el Mouse.

1. Haga clic en la flecha situada a la derecha del cursor.
2. Haga clic en la flecha de desplazamiento hacia arriba o hacia abajo. O bien, arrastre el cuadro de desplazamiento hasta tener a la vista el elemento que desee seleccionar.
3. Haga clic en el elemento.

Botones de opción

Los botones de opción representan grupos de opciones mutuamente excluyentes. Solamente puede seleccionar uno a la vez. Si ya ha seleccionado una, ésta será reemplazada por la opción seleccionada actualmente. La tecla de opción seleccionada contiene un punto negro. Los nombres de las opciones no disponibles aparecen atenuados.

Opción seleccionada



Para seleccionar un botón de opción:

Haga clic en el botón de opción.

Pantallas

El sistema está compuesto de varias pantallas a través de las cuales el usuario puede navegar para realizar tareas específicas.

Las pantallas podrán ser accesadas al oprimir cualquier botón del menú principal, el cual tiene una función específica.

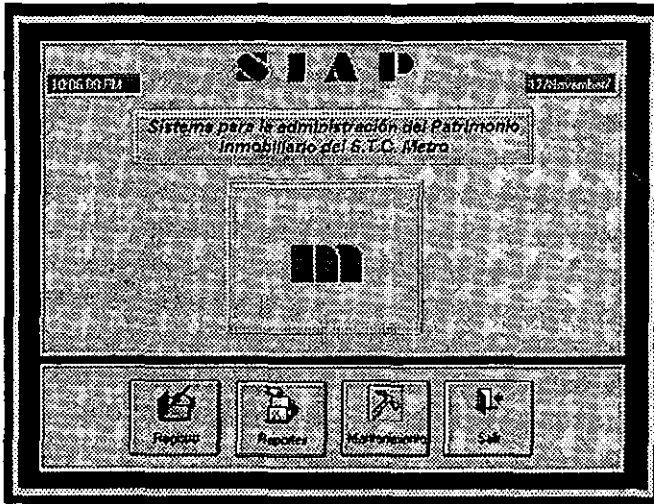
Pantalla inicial

Es la primer pantalla que se visualiza al acceder el sistema. En ella se despliega el logotipo de la empresa, la hora y fecha actuales. En esta misma pantalla aparecerá, de forma automática, una pequeña pantalla que solicitará la clave del usuario y la contraseña correspondiente. Esta le permitirá entrar al sistema y a los módulos autorizados. Lo anterior se explicara más a detalle en el módulo de seguridad.



Pantalla de Menú Principal

Esta pantalla contiene los iconos de los módulos que puede acceder el usuario. El número de iconos dependerá de las operaciones y funciones a las que esté autorizado a utilizar.



Botones

Las pantallas tienen disponibles una clase de botón, esta corresponde a los botones de decisión, estos están programados para realizar una acción específica.

Los siguientes son ejemplos de botones de decisión que se encontrarán dentro de las pantallas del sistema:

Botón de guardar: Al presionar este botón, el sistema guardará en la base de datos la información capturada o modificada en la pantalla, por lo que a partir de ese momento se tendrá la capacidad de consultar o generar reportes posteriormente.



Botón de Eliminar. Al presionar este botón, el sistema borra la información previamente guardada en la base de datos y que es mostrada en pantalla. Nota: Se

deberá tener cuidado al usar esta acción, ya que después de ejecutarla no se podrán recuperar los datos eliminados.



Botón de consultar: Al presionar este botón se le indica al sistema presentar en pantalla información referente a la función actual en base a criterios de búsqueda ya predefinidos.



Botón salir: Al presionar este botón se le pide al sistema que abandone la pantalla actual y regrese a la pantalla anterior o de la aplicación.



Cuadro de verificación: Al presionar este botón se le pide al sistema que seleccione y realice la función que tiene descrita.

Mensajes

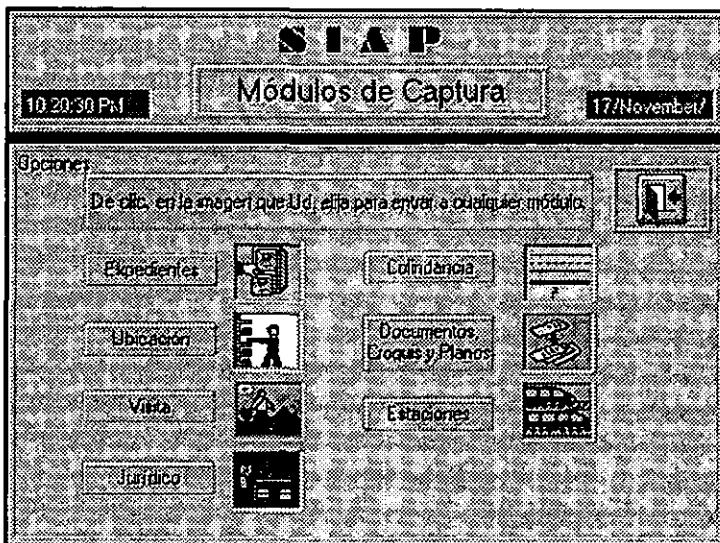
Son cuadros de diálogo, en los cuales es desplegado algún mensaje o aviso relacionado a un evento o proceso que se ejecutó o se va a ejecutar. Existirán casos en donde se pida la aceptación, confirmación o cancelación por parte del usuario.

Sección 3. Descripción de los módulos del sistema.

MÓDULO DE REGISTRO

Este módulo nos servirá para el registro de la información de los expedientes de los inmuebles.

Contiene un menú llamado Módulo de Captura el cual esta clasificado de la siguiente forma:



- Expedientes
- Ubicación
- Visita
- Jurídico
- Colindancia
- Documentos, Croquis y Planos

- Estaciones

Cuando usted coloque el cursor en cualquiera de las opciones anteriores se desplegará un mensaje donde le indicará los eventos a ejecutarse cuando usted de un clic en el icono.

EXPEDIENTES

De este menú se desplegaran los siguientes iconos, desplegados en la parte superior izquierda de la pantalla:

The screenshot shows a graphical user interface for 'SIAP Expedientes'. The window title is 'SIAP Expedientes'. The top bar displays the time '10:22:36 PM' and the date '17/11/2006'. Below the title bar is a toolbar with icons for home, edit, delete, print, and navigation. The main area is divided into sections: 'Tipo de Documento' with fields for 'Cédula', 'Doctos', 'Fojas', 'Clave', and 'Tipo de Docto'; 'Expediente:' with fields for 'Cédula', 'Núm. Expedite', 'Archivado', 'Gaveta', and 'Total de fojas'; and 'Estatus' with checkboxes for 'Nuevo', 'D.D.F.', 'Particular', 'Federal', 'S.T.C.', and 'Desincorporado'. There are also two buttons: 'Nuevo Documento?' and 'C. Docto. Expediente?'. At the bottom right, there is a 'Datos' button.

- Inicio de Captura
- Inicio de una Baja
- Modificaciones al Expediente

- Consulta de Expedientes
- Reiniciar la Pantalla
- Salir de este módulo

Para iniciar la captura de un nuevo "documento con expediente" se seleccionará el icono de "iniciar captura", posteriormente "Docto. c/Expediente", en donde se activarán automáticamente los campos de "Cédula, Doctos, Fojas, Clave y Tipo de docto (lista desplegable con varias opciones)"

A continuación se proporciona un ejemplo de la forma en como se dan de alta los documentos dentro del Sistema de Administración del Patrimonio:

CEDULA	DOCTOS	FOJAS	CLAVE	TIPO DE CONTRATO
L1	5	10	4	Decreto

Nota: El campo de clave se activa cuando usted ha seleccionado el tipo de contrato, el sistema automáticamente asigna un número en la clave correspondiente al tipo de documento.

SIAD

Expedientes

10:23 AM 17/November/

Tipo de Documento

Cedula	Dactos	Fojas	Clave	Tipo de Docto.
L1	10	20		SIN TIPO ACUERDO CONTRATO CONVENIO DECRETO DOCTO. ADMINISTRATIVO ESCRITURAS FOTOS

Expediente

Cedula:

Núm. Expedite:

Archivero:

Gaveta:

Total de fojas:

Nuevo:

D.D.F.:

Particular:

Federal:

S.T.C.:

Desincorporado:

Después que se proporcionaron estos datos se tendrá que proporcionar también la siguiente información:

NUM. EXPDTE	ARCHIVERO	GAVETA	TOTAL FOJAS	DE
L1	5	10	4	

también se tendrá que seleccionar el tipo de expediente, es decir si es del DDF, Nuevo, Particular, etc. Finalmente se tendrá que guardar la información capturada seleccionando el icono de "guardar" que se encuentra en la parte superior izquierda de la pantalla.

En caso de que se desee adicionar un nuevo documento al expediente solo tendrá que inicializar la captura y proporcionar el Número de la Cédula capturada en "Documento c/Expediente" y se proporcionará en el campo doctos el Número de doctos a capturar.

Cédula	Doctos	Fojas	Clave	Tipo de Docto
L1	10	20	2	CONTRATO

Expediente		Opciones	
Cédula	<input type="text"/>	Nuevo	<input checked="" type="checkbox"/>
Num. Expedite	235	D.D.F.	<input checked="" type="checkbox"/>
Archivo	15	Particular	<input type="checkbox"/>
Gaveta	5	Federal	<input checked="" type="checkbox"/>
Total de fojas	200	S.T.C.	<input type="checkbox"/>
		Desincorporado	<input type="checkbox"/>
			<input type="radio"/> Nuevo Documento? <input checked="" type="radio"/> Docto. c/ Expedite?

14/6/2000

Para realizar alguna modificación o consulta a los expedientes o documentos de estos solo tendrá que teclear el No. de Cédula y dar enter. En caso de que se realicen modificaciones se activaran todos los campos para que usted pueda agregar o modificar los datos de los documentos o expedientes (depende de los privilegios asignados al usuario).

Si se desea realizar una baja de algún documento o expediente tendrá que teclear el campo "cédula" el número de cédula del expediente.

Los campos que no son activados en una consulta son: "cédula" y "fojas" en la ventana de "Tipo de documento" y "Expediente".

UBICACION

En esta pantalla se proporcionaran datos para tener una mejor localización de los inmuebles del Sistema de Transporte Colectivo Metro, es decir los datos de la estación así como la dirección de esta.

Cédula:	L1	Cve. de la Estación:	
Datos de la Estación			
Cve. Estación:	ACUL	ACULCO	
Calle:	Zoologos		
Número:	10		
No. Interior:			
Lote:	7		
Manzana:	25		
Dirección:			
Colonia:	San José Aculco	C.P.:	09410
Entre Calles:	Eje 5 Sur y Río Churubusco		
Cve. Deleg.:	9	XTAPALAPA	Fec. de Captura:
Cta. Anterior:	3443322225	Cuenta Actual:	234423432
			17/11/1998

Aquí tendrá que proporcionar la clave de la cédula y la ubicación o datos de la estación, la calle , el número, No. interior, el lote y la manzana.

En esta sección se proporcionará la "Clave de la Estación" y la "Dirección de la Estación", seleccionando los botones que se despliegan en la parte superior derecha de la ventana.

SIAP	
10:34:23 PM	Ubicación
17/Noviembre/2017	
<input checked="" type="checkbox"/>	<input type="checkbox"/>
Cédula: [1]	Cve. de la Estación
Datos de la Estación	
Cve. Estación:	
Calle:	
Número:	Eve. Esta Estación
No. Interior:	▶ ACAT ACATHILA
Lote:	▶ AGU AGUICO
Manzana:	▶ AGOR AGRICOLA ORIENTAL
Dirección:	Colonia: C.P.
Entre Calles:	
Cve. Deleg.:	Fec. de Captura
Cta. Anterior:	Cuenta Actual

Finalmente se guardará la información con la fecha del día de la captura, aquí usted no tendrá que teclear nada ya que el sistema proporcionará esta automáticamente.

VISITA

Esta pantalla sirve para registrar todos los datos obtenidos durante la visita al inmueble.

Para iniciar la captura, consulta, modificaciones y bajas de la información solo tendrá que teclear el "Número de Cédula" y oprimir la tecla enter para obtener la información del inmueble.

SIAP			
3491231	Visita	18/11/1998	
<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Cedula: <input type="text" value="11"/>			
Datos generales:			
Nombre Inmueble:	<input type="text" value="Abrigotes 'Los Grupos'"/>	Propietario:	<input type="text" value="Claudio Veloz"/>
Uso Inicial:	<input type="text" value="Casa particular"/>	Uso Actual:	<input type="text" value="Local comercial"/>
Usuario:	<input type="text" value="Benjamin Garcia"/>	<input type="text" value="114"/>	<input type="text" value="111"/>
Características:			
Tipo Inmueble:	<input type="text" value="LOCAL COMERCIAL"/>	Estado Fracción:	<input type="text" value="EXCELENTES CONDICIONES"/>
Superficie:	<input type="text" value="250"/>	Mts:	
Construida:	<input type="text" value="200"/>	Mts:	
Niveles:	<input type="text" value="2"/>		
		Agua:	<input checked="" type="checkbox"/>
		Luz:	<input checked="" type="checkbox"/>
		Drenaje:	<input checked="" type="checkbox"/>
		Realización:	<input type="text" value="18/11/1998"/>
		Conclusión:	<input type="text" value="19/11/1998"/>
		Confirmación:	<input type="text" value="19/11/1998"/>
Fecha:			
Gestión:		Elaboración:	<input type="text" value="18/11/1998"/>
Elaboró:		Autorizó:	<input type="text" value="18/11/1998"/>
Aprobó:		Autorizó Sr.:	<input type="text" value="18/11/1998"/>
Observaciones: <input type="text" value="Se recomienda reubicar a las personas que laboran en este local ya que se verá afectada"/>			

Aquí se proporcionaran los datos generales del inmueble, se proporcionara el nombre del inmueble, propietario, uso inicial, uso actual, usuario y el número de visitas realizadas a este, así también se proporcionaran las características del inmueble, es decir el tipo de inmueble, el estado en que se encuentra, las dimensiones de este, y los servicios con que cuenta, es decir si el inmueble cuenta con agua, luz, drenaje.

Se registrará en esta pantalla la fecha en que se realizaron las visitas, la realización de esta, la conclusión y confirmación de las visitas, así también se registrarán los datos de la persona que elaboró la visita y quien autorizó esta.

JURIDICO

Aquí se proporcionaran los datos del jurídico, es decir el número de los documentos, las sesiones, los acuerdos, permisos, etc.

Documentos		
Cédula:	Lev. Dignica:	89
Fecha Escritura:	17/04/1970	Convenio:
Nomb. Escritura:	Abarrotes	Contrato:
Núm. Notaría:	8	Decreto:
Nomb. Notario:	Agustín Torres	Publicación:
897		
Antecedente del Comité del Patrimonio Inmobiliario (C.P.I.)		
Sesión:	Extraordinaria	Acuerdo:
Fecha:	18/11/1998	Compra del inmueble
Responsable:	Claudia Urzua	Permiso:
Propuesta:	Compra del inmueble	Gobernación
		Tipo:
		Jefe de Gobernación
Gestión		
Elaboró:	Claudia Vélez	Fec. de Elab:
Autorizó:	Carlos Roa	18/11/1998
Observaciones:	que las personas que la habitan se verán seriamente dañadas por la construcción de una estación del metro, por tal motivo se pide sea reubicada.	
		Fec. Autoriza:
		17/11/1998

En esta pantalla también se tendrá que proporcionar el nombre de la persona que autorizo, elaboro, hizo las observaciones y las fechas en que se realizaron o se autorizaron las visitas.

Se inicializará la captura de la información mediante los iconos que se encuentran en la parte superior izquierda de la pantalla que anteriormente se han definido.

COLINDANCIA

Esta pantalla servirá para ubicar las colindancias del inmueble y sus dimensiones con respecto a los puntos cardinales, aquí solo hay que introducir la información en los campos correspondientes.

Cédula: [L1]		
Punto Cardinal	Coordenada	Medida
Norte	Rio Churubusco	400 Mts.
Sur	Araucarias	600 Mts.
Oeste	Taquigrafos	300 Mts.
Este	Eje 5 sur	250 Mts.
Eve Tipo Colm: [1] [REAL]		

La forma de acceder a este modulo es utilizando el "Número de Cédula".

DOCUMENTOS, CROQUIS Y PLANOS

En esta pantalla se podrán capturar y buscar imágenes de los inmuebles, también se podrán realizar búsquedas de los documentos que previamente fueron digitalizados.

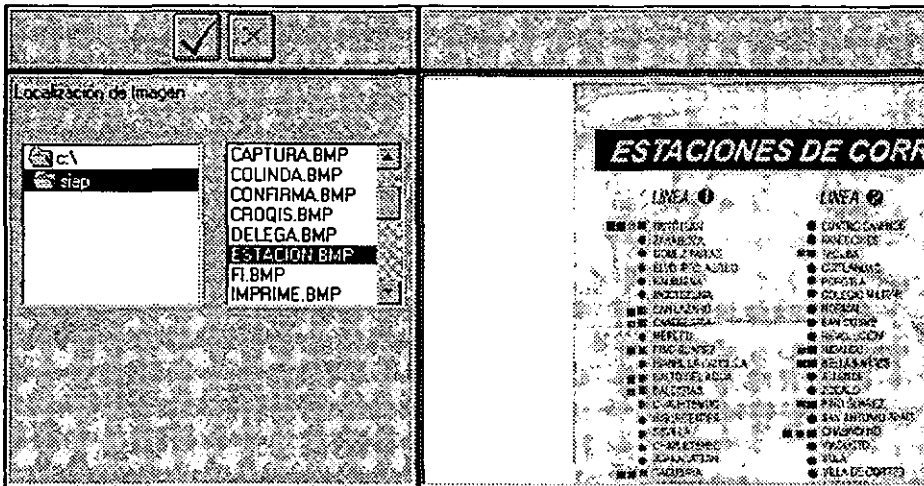
Cédula: [L1]	
Marca para Atlas: <input type="checkbox"/> Inmueble <input type="checkbox"/> Documento <input type="checkbox"/> Planos <input type="checkbox"/> Croquis	Búsqueda de Imagen:
Imagen:	

Esta pantalla aplica para la búsqueda de imágenes del inmueble, de documentos, planos y croquis.

Se desplegarán en la parte superior izquierda de la pantalla los siguientes iconos:

- Capturar una imagen
- Borrar una imagen
- Búsqueda de una imagen
- Reiniciar la pantalla
- Salir del módulo de imágenes

Aquí hay que introducir la clave la cédula y posteriormente que tipo de imagen se va a capturar, esta puede ser de documentos, inmueble, plano o croquis, el sistema automáticamente nos lleva a una pantalla en donde nosotros seleccionaremos la ruta de donde podemos obtener la imagen para cada inmueble una vez que esta se seleccione se presionará el icono de "aceptar" para regresar a la pantalla de origen.

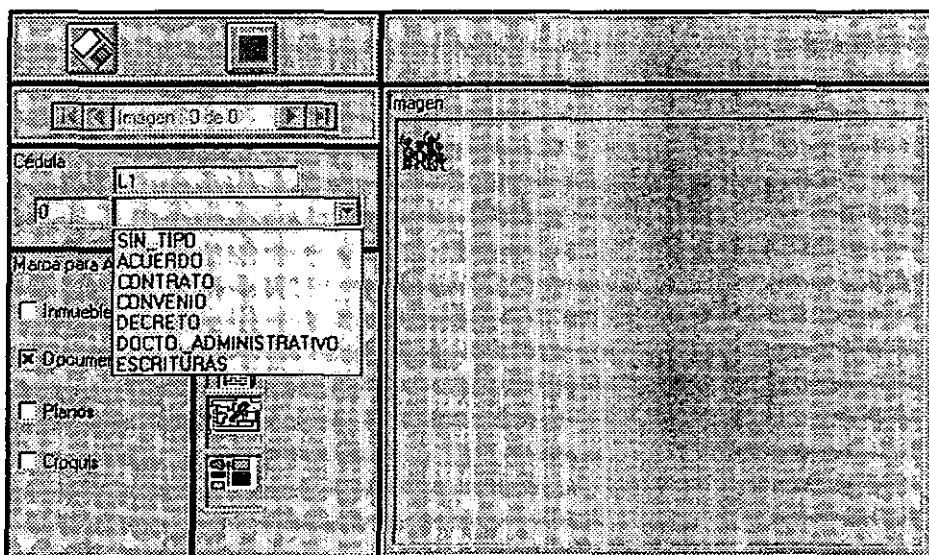


Una vez posicionados en esta pantalla en "cédula" escribiremos la clave y seleccionaremos en donde daremos de alta la imagen capturada anteriormente en opción de imagen: inmueble, documento, planos o croquis.

En la parte superior izquierda de la pantalla usted podrá seleccionar el icono de guardar o grabar la imagen.

Para realizar una búsqueda en esta pantalla se tendrá que seleccionar el icono "búsqueda de imagen" que se encuentra en la parte superior izquierda de la pantalla, automáticamente se activará en la parte inferior izquierda de la pantalla un recuadro de "búsqueda de imagen", por inmueble, documento, planos o croquis y finalmente se proporcionará la clave de la cédula y se dará enter.

Si se quiere dar de alta más de una imagen por inmueble, documento, etc. lo único que se tiene que hacer es presionar con el botón izquierdo del mouse el icono de "Reiniciar la pantalla"



En caso de que se quiera dar de alta una imagen de documento después de capturar la imagen se deberá teclear la clave de la cédula y de la lista desplegable el tipo de documento con el que se esta trabajando y finalmente guardamos o salvamos esta sesión.

MODULO DE REPORTES

En este modulo solo se tiene que seleccionar la clave del reporte que se desea imprimir y oprimir la imagen de impresora para la impresión del reporte estos reportes pueden ser:

De clic. en la imagen de impresora.

Clave del Reporte

Mapa	Inmueble	Croquis
C.F. M02 RPT		
Inf. del Inmueble	Antecedente	Expediente

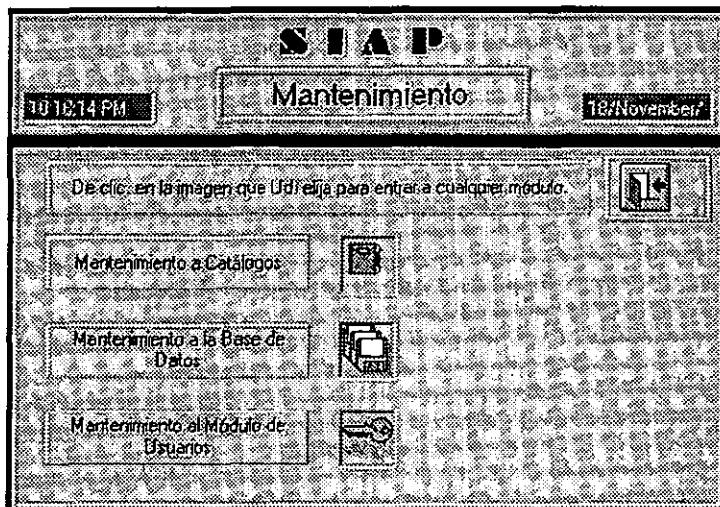
- Mapa
- Inmueble
- Croquis
- Información del Inmueble
- Antecedentes
- Expediente

Después de haber seleccionado el botón de impresora aparecerá una ventana en donde nos mostrará una presentación preliminar del reporte y finalmente un símbolo o imagen de una impresora para la impresión del reporte.

En caso de que usted no desee imprimir o seguir imprimiendo los reportes en la parte superior izquierda de la pantalla existe un botón para salir del módulo de reportes.

MODULO DE MANTENIMIENTO

Este módulo sirve para actualizar los catálogos que existen en el sistema, pero no solo podemos proporcionar mantenimiento a los catálogos también se puede dar mantenimiento a la Base de Datos y al módulo de usuarios que accesan al sistema.



Este módulo contiene lo siguiente:

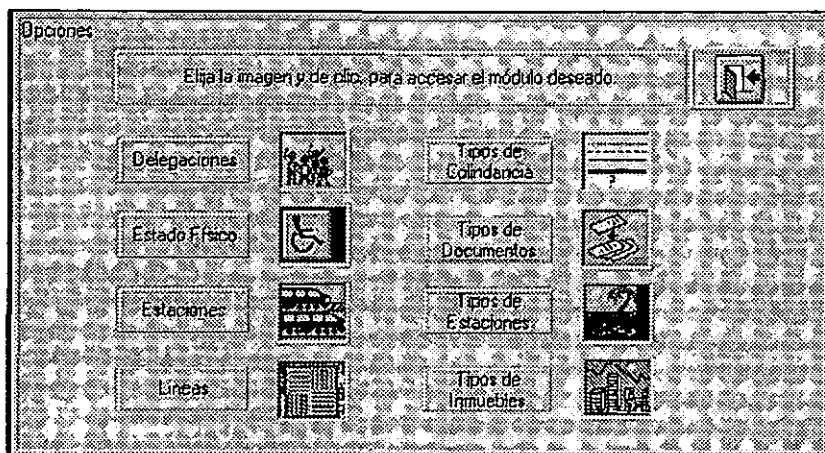
- Mantenimiento a Catálogos
- Mantenimiento a la Base de Datos

- Mantenimiento al Módulo de Usuarios

Aquí sólo tendrá que dar un clic en los iconos correspondientes para acceder a las pantallas o módulos antes mencionados.

Mantenimiento a Catálogos

Existen varios catálogos en el Sistema de Administración del Patrimonio, que a continuación se mencionan:



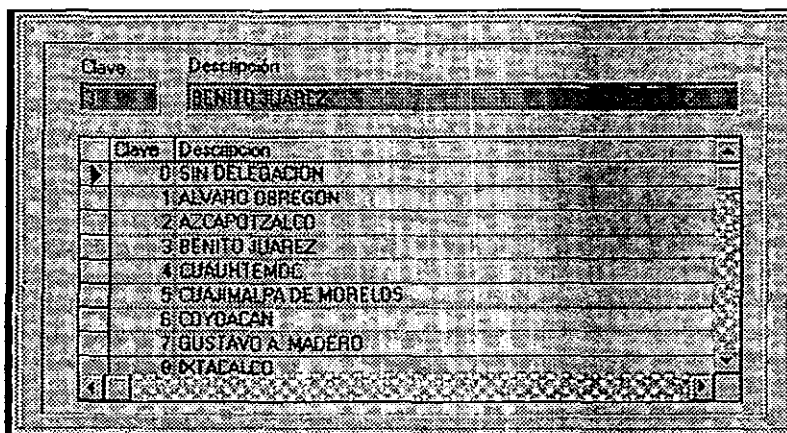
- Delegaciones
- Estado Físico
- Estaciones
- Líneas
- Tipos de Colindancia
- Tipos de Documentos
- Tipos de Estaciones
- Tipos de Inmuebles

En estas pantallas podremos dar de alta, baja o modificar los catálogos de estaciones, líneas, delegaciones, etc., siempre y cuando seamos usuarios autorizados para este tipo de modificaciones dentro del sistema.

A continuación se explicará la forma de dar mantenimiento al Catálogo de Delegaciones:

Primero se dará un clic en el icono de Delegaciones, en la pantalla siguiente se tendrá que dar clic en el icono de abrir sesión(Solo se activará dependiendo del perfil del usuario).

En la parte superior izquierda de la pantalla existen cinco iconos que nos servirán para dar de alta, baja, consultar o modificar los catálogos.



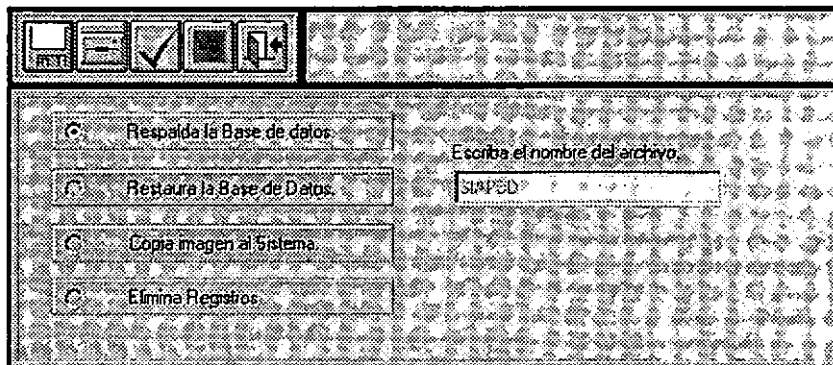
Para iniciar la captura de una nueva delegación solo se tendrá que oprimir el botón "Inicializar captura", posteriormente se proporcionará la clave y la descripción (nombre de la delegación).

En caso de que se quiera realizar una consulta, baja o modificación se tendrán que seleccionar los iconos correspondientes y en la lista de clave y descripción, se buscará la clave y delegación para que finalmente se de un doble clic y se coloque en el campo de captura para realizar la modificación o baja de este.

Mantenimiento a la Base de Datos

Este módulo contiene las siguientes opciones:

- Respalda la Base de Datos
- Restaura la Base de Datos
- Copia Imagen al Sistema
- Elimina Registros



Si queremos realizar el Respaldo de la Base de Dato solo tenemos que seleccionar esta opción y automáticamente se activarán los iconos ubicados en la parte superior de la pantalla, estos son cinco en donde el primero sirve para respaldar la base de datos en disquetes, el segundo en la unidad de cinta, el tercero para aceptar la acción, el cuarto para restaurar la pantalla y el quinto para salir del sistema.

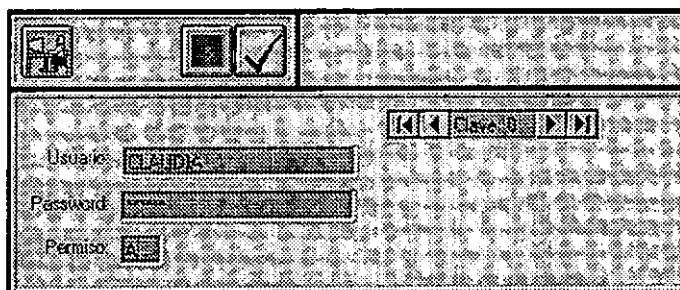
Los pasos para respaldar, restaurar, copiar imagen y eliminar el sistema son los siguientes: primero se tendrá que seleccionar alguna opción que anteriormente fueron definidas después se activara un campo en donde escribiremos el nombre del respaldo, posteriormente seleccionaremos el icono de "respaldo en disquete" o "en unidad de cinta" y finalmente el icono de "Aceptar la acción", cuando se acepta la acción se desplegará el siguiente mensaje:

¿Esta seguro que desea realizar un Respaldo, Recuperación de archivo ó Borrar la Base de Datos? si no va a realizar ninguna acción por favor oprima "NO" y salga del módulo gracias.

En caso de ser afirmativa la decisión solo se tendrá que introducir el disquete o dar la ruta en donde queremos que guarde el archivo con extensión ZIP.

Mantenimiento a Usuarios

En esta pantalla se podrán realizar altas, bajas y modificaciones a los perfiles de los usuarios.



The screenshot shows a graphical user interface for user management. At the top, there are three icons: a person, a square, and a checkmark. Below these is a navigation bar with left and right arrow buttons and a 'New User' button. The main area contains three input fields: 'Usuario' with the text 'RAIDC', 'Password' with a masked password, and 'Permiso' with a dropdown menu showing 'A'. The interface has a halftone or dithered background.

Para introducir una nueva cuenta de usuario se tendrá que oprimir el icono de "inicializar captura", se activaran los campos de nombre del usuario, password, permiso y finalmente se oprimirá el botón de "Aceptar", el sistema desplegará un mensaje de aviso, es decir si esta de acuerdo con la nueva cuenta de usuario que se esta dando

de alta, en caso de ser afirmativo se desplegará otro aviso indicando que el usuario ha sido dado de alta exitosamente, en caso contrario el sistema únicamente cancelará la acción antes descrita.