

19  
201



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

Sistema para determinar el valor  
unitario de un predio dentro del D.F.

T E S I S  
Que para obtener el título de  
INGENIERO EN COMPUTACION  
p r e s e n t a n

Humberto Blas Mancilla  
José Alejandro Bucio Martínez  
Bernabé Ignacio Peralta Piña  
Josefina Ramos Zambrano  
Elizabeth Romero Pérez

263593

Director de Tesis: M.I. Juan Carlos Roa Beiza



México, D. F.

1998



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

**Agradecimiento:**

**A la Universidad Nacional Autónoma de México y principalmente a la Facultad de Ingeniería por habernos abierto sus puertas y brindado una formación profesional.**

**Un reconocimiento especial al Ing. Juan Carlos Roa Beiza por haber aceptado la dirección de este trabajo, por la confianza que despertó en nosotros y por el apoyo que de él recibimos.**

**A los Profesores de la Facultad de Ingeniería por darnos la oportunidad de recibir una parte importante de sus conocimientos.**

**Bernabé, Josefina, Elizabeth, Alejandro y Humberto**

## **Gracias**

En primer lugar quiero darte las gracias a Dios, por haberme dado las fuerzas y el tiempo para llegar a esta etapa tan importante de mi vida. Éste es el final de un gran esfuerzo, pero también es el principio de un camino.

A Irena, mi mamá, quiero darte las gracias por tu apoyo incondicional, por tu paciencia y ejemplo. Y quiero decirte que tú eres la persona que más quiero y admiro. Todo este trabajo es realmente tuyo.

Mary, Angélica, Evelyn, Mary Carmen y Sonia, mis hermanas, les doy las gracias por estar cuando yo las necesito. En especial a tí Mary que hemos convivido durante más tiempo y nos hemos apoyado en diferentes situaciones, muy a pesar de todas las diferencias que hemos tenido.

Belmar, mi papá, que a pesar de haber estado ausente de mi vida durante mucho tiempo, al fin podemos llevarnos mejor.

A mis abuelos Pedro, Ninfa y Amparo(†), a mis tíos y tías y a mis primos les doy las gracias por haber creído en mí.

A mis amigos Arturo y Rosa, Adriana, Balta, Israel, Angeles, Carlos y cada una de las personas que saben que están en mi corazón pero que no las recuerdo en este momento, les doy las gracias por estar siempre junto a mí y por empujarme a seguir adelante.

A todas las personas que me apoyaron para la realización de este trabajo, como la Lic. Leonor G. y a cada uno de mis compañeros de la facultad, a mis maestros y un especial agradecimiento a mis compañeros de Tesis José, Alejandro, Elizabeth y Bernabé.

**Humberto Blas Mancilla**

**Abril de 1998.**

## **CON PROFUNDO AGRADECIMIENTO A:**

### **MIS PADRES.**

Angela y José Luis, quienes día tras día me brindaron su cariño, me alentaron y me apoyaron incondicionalmente durante todo el trayecto de mi vida como estudiante, haciendo posible el término de esta tesis, la cual representa también, el resultado de su esfuerzo.

### **MIS HERMANOS.**

Maribel, Edgard y Norberto(†), por ser los mejores amigos y porque alguna vez estuvieron conmigo en los momentos más difíciles de mi carrera.

### **MIS ABUELOS**

Juan(†), Esperanza(†), Efren y Felicitas(†) quienes aconsejaron a mis padres, para que nos apoyaran en todos los momentos para lograr todos nuestros objetivos y por los gratos recuerdos que tengo de ellos.

### **MIS FAMILIARES**

Por motivarme, brindarme su cariño y compartir conmigo el interés por llegar al fin en la elaboración de esta tesis.

### **MIS AMIGOS**

Gustavo, Guillermo, Víctor, Carlos y Jorge, porque con ellos he pasado momentos inolvidables, acompañándonos en las buenas y en las malas.

**JOSÉ ALEJANDRO BUCIO MARTÍNEZ**

## **Agradecimientos:**

### **A Jehová:**

Gracias Jehová, por haberme dado discernimiento al empezar, dirección al progresar y sabiduría al acabar.

### **A mis padres Jesús y Epifanía:**

Por todo el amor que me han brindado y el apoyo, en todos los lugares que los necesité, todas las veces que los solicité, durante todo el tiempo que me llevó llegar hasta este momento muy importante en mi vida y no lo mencionan por lo que han dado con el corazón como sólo los padres lo saben hacer.

### **A mis hermanos Leticia, Verónica, José Rafael y Rocío:**

Por ayudarme y alentarme a seguir en los momentos difíciles que pasé. Por el cariño y comprensión que siempre me han brindado.

### **A mis amigos:**

Marisol C., Janet O., Lina A., Rosa J., Zulema R., Ricardo M., Alberto A., Guillermo G., y todos aquellos que hicieron posible este trabajo.

Por su amistad sincera que me han brindado durante todo el tiempo que hemos estado juntos, compartiendo un poco de nuestras vidas.

**Bernabé Ignacio Peralta Piña**

A la persona más importante en mi vida y que más quiero, por el ejemplo de lucha constante y todo el apoyo incondicional que siempre nos ha brindado.

Gracias mamá.

En donde quiera que estés, por todo el cariño y la alegría que siempre me diste.

Gracias papá.

A mis hermanos: Eusebio, Damián, Luis, Lilia y Susy por el cariño y la unión que siempre hemos mantenido.

A mi esposo, Andrés por su amor, apoyo y comprensión.

Elizabeth.

## **AGRADECIMIENTO**

Quiero y deseo expresar mi más profundo y sincero agradecimiento a mis PADRES, ya que siempre han confiado en mí y sé que podré contar con su amor, cariño y apoyo por siempre.

Uno super especial para mis hermanos y para mi cuñada, para todos y cada uno de ellos que me alentaron y me brindaron siempre su ayuda.

Uno especial para mis sobrinos Gibrán y Mariana, por que he compartido y jugado muchas cosas con ellos y el verlos reír hace más feliz mi vida.

Y uno muy grande, especial y sobre todo importante: GRACIAS A DIOS, porque sin él no hubiera tenido la familia tan unida que tengo, la que ha estado conmigo en todo momento y a la cual quiero mucho.

**LOS QUIERO MUCHO ! M U C H A S   G R A C I A S ! .**

**Josefina Ramos Zambrano**



# ÍNDICE

## Introducción

### 1. Conceptos Generales

1.1. La valuación como guía de decisión	1
1.1.1. Antecedentes	2
1.1.2. Definición de valuación	4
1.1.3. Cómo influye en la toma de decisiones	7
1.2. Conceptos y variables que inciden en el valor de un predio	9
1.2.1. Terrenos	9
1.2.1.1. Terrenos Urbanos	10
1.2.1.2. Terrenos Suburbanos	11
1.2.1.3. Terrenos Industriales	11
1.2.1.4. Terrenos Rústicos	13
1.2.2. Factores	14
1.2.2.1. Factor de zona	15
1.2.2.2. Factor de ubicación	15
1.2.2.3. Factor de topografía	16
1.2.2.4. Factor de frente	16
1.2.2.5. Factor de forma	16
1.2.2.6. Factor de superficie	18
1.2.2.7. Factor de homologación por uso de suelo	19
1.2.2.8. Factor de homologación por servicios públicos	19
1.2.2.9. Factor de comercialización	19
1.2.3. Uso de suelo	20
1.2.4. Investigación de mercado	26
1.2.5. Tipos de valor	26
1.2.6. Comercialización	29
1.3. Tipos de avalúo	30
1.3.1. Avalúo catastral	30
1.3.2. Avalúo bancario	32
1.3.3. Avalúo económico	33

1.4.	Métodos de valuación	34
1.4.1.	Método del ingreso	35
1.4.2.	Método del mercado	37
1.4.3.	Método del costo	44
1.5.	Visual Basic 4.0	48
1.6.	Características de Access 7.0	78
<b>2.</b>	<b>Planteamiento del problema</b>	
2.1.	Definición del problema	99
2.2.	Descripción de objetivos	102
2.3.	Conceptos de bases de datos relacionales	103
2.3.1.	¿Qué es una base de datos?	104
2.3.2.	Modelos de bases de datos	123
2.3.2.1.	Modelo jerárquico	124
2.3.2.2.	Modelo de red	126
2.3.2.3.	Modelo relacional	128
2.3.3.	Diagrama Entidad-Relación	136
2.4.	Metodologías de diseño	139
2.4.1.	Metodologías de desarrollo	141
2.4.2.	Ciclo de Vida de un Sistema de Información	143
2.4.3.	Yourdon	145
2.4.4.	Pressman	148
2.4.5.	Merise	152
2.4.6.	Ciclo de vida en entorno Case	158
2.4.7.	Elección de metodología	160
2.5.	Análisis del sistema	163
2.5.1.	Situación actual y situación que se pretende alcanzar	163
2.5.2.	Requerimientos del usuario	166
2.5.3.	Recopilación y análisis de la información	168
2.5.4.	Propuesta de solución	172
2.5.5.	Diagramas de funcionalidad	175
2.5.5.1.	Técnicas de simplificación de operaciones	176

2.5.5.2. Diagrama	178
<b>3. Diseño e implementación del sistema</b>	
3.1. Diagrama de flujos de datos	181
3.2. Diagrama Entidad-Relación y Normalización de datos	199
3.3. Diccionario de datos	207
3.4. Diseño y construcción de la base de datos	216
3.5. Diseño y construcción del front-end	220
3.6. Integración y pruebas del sistema	284
3.6.1. Estrategia de prueba de software	285
3.7. Implantación del sistema	310
<b>4. Impacto del sistema</b>	
4.1. ¿Qué es el soporte de sistemas?	313
4.2. Mantenimiento del sistema: Corrección de errores	321
4.2.1. Objetivos y bloques elementales del mantenimiento de sistemas	321
4.2.2. Tareas, participantes y técnicas del mantenimiento de sistemas	323
4.3. Recuperación del sistema: Superar los fallos generales del sistema	334
4.4. Asistencia al usuario final	336
4.5. Mejoras y Reingeniería de Sistemas	336
4.5.1. Tipos de mantenimiento	337
4.5.1.1. Correctivo	337
4.5.1.2. Preventivo	337
4.5.1.3. Perfectivo	338
4.5.1.4. Adaptativo (aumentativo y tecnológico)	338
4.5.2. Objetivos y bloques elementales de las mejoras y la reingeniería de sistemas	339
4.5.3. Actividades, participantes y técnicas de las mejoras y la reingeniería de sistemas	342
Conclusiones	
Bibliografía	
Glosario	
Apéndice A Manual de Usuario	

## INTRODUCCIÓN

El presente trabajo, está enfocado al desarrollo de un sistema que permita estandarizar los criterios de valuación de predios y terrenos para algunos organismos en el Distrito Federal, empleando algunas de las metodologías clásicas del desarrollo de sistemas.

La Comisión de Avalúos de Bienes Nacionales (CABIN), es un organismo gubernamental encargado de realizar las valuaciones de los bienes propiedad de la nación, dentro de éstas, una de las problemáticas más fuertes que tienen, es la que involucra a los bienes inmuebles, específicamente, predios y terrenos. Esta problemática no es exclusiva de este organismo, pero sólo nos enfocaremos a éste, ya que controla una mayor cantidad de bienes y que de alguna forma pueden ser trasladados a organismos privados que manejen los mismos tipos de bienes u otros.

Actualmente, el proceso de avalúo que se lleva a cabo en CABIN, es de forma muy rudimentaria, es decir, en papel, con los criterios que cada valuador determina, originando resultados totalmente diferentes para un mismo caso.

Con el sistema desarrollado se pretende automatizar y estandarizar los criterios de valuación. Para la realización de este trabajo, se hizo una recopilación de los conceptos empleados por los valuadores, metodologías y consideraciones que se tienen para la realización de su trabajo. Además, se describen las herramientas empleadas para el desarrollo de sistema, mismos que conforman el capítulo 1.

En el capítulo 2, se realiza el planteamiento del problema, se definen objetivos y se retoman algunos conceptos generales del análisis de sistemas mismos que son aplicados, además se exponen algunas propuestas de solución.

El diseño y desarrollo del sistema se describe en el capítulo 3. También se retoman algunos conceptos generales, mismos que son aplicados en forma conjunta, en algunos casos, y otros en forma particular.

Este capítulo es el más importante, ya que en este se pone en práctica las metodologías, tanto de análisis, diseño y desarrollo al emplear los diagramas de flujo de datos (DFD), diagrama de contexto, modelo entidad-relación, diccionario de datos, etc. Por otro lado, se describe paso a paso el desarrollo del sistema con Access y Visual Basic, herramientas empleadas desde el prototipo hasta la finalización del sistema.

También en este capítulo se explican las pruebas de cada uno de los módulos que en algunos casos fueron selectivas y en otras ya establecidas. Por último se describe la implantación del sistema.

En el capítulo 4, se desarrollan algunos de los tipos de mantenimiento de sistemas.

Por último, se presentan cuatro apéndices con información complementaria y a detalle de algunas partes del sistema; el primero se refiere al manual de usuario, en donde se explica el empleo correcto del sistema; el segundo es el manual técnico, donde se hacen algunas observaciones para los desarrolladores para futuras ampliaciones del sistema o corrección de algunas fallas que pudiera tener; el tercero es el código del sistema en Visual Basic; y por último, referencias a la fuente de información.

Este trabajo fue realizado con la intención de resolver un problema cotidiano para el área de valuación, empleando los conocimientos adquiridos durante la carrera de Ingeniería en Computación, además de ponerlos en práctica y emplear tecnología de vanguardia en el desarrollo de sistemas. Otra de las consideraciones que se tomaron

en cuenta por la elección de este sistema fue la poca participación de los desarrollos para el área de avalúos en el mercado.

La principal limitación que se tuvo en el desarrollo del sistema fue la poca importancia que se le da a estas áreas dentro de cualquier organismo, mismos que se reflejan en la falta de equipo de cómputo y el poco presupuesto asignado para su adquisición.

Para poder recabar la información referente a la valuación de bienes, se realizaron entrevistas con personal que labora cotidianamente en estas tareas, los cuales en algunos casos tenían grados de estudio de licenciatura (Ingeniería o Arquitectura), y en otros casos niveles inferiores a estos, pero con bastante experiencia. Además se realizó la consulta en diferentes documentos desarrollados por estas áreas, donde muestran el desarrollo de valuaciones bajo diferentes criterios, pero en ningún caso se contó con un sistema de información desarrollado en computadora y si existía, no funcionaba como tal.

Una de las fuentes de información mas importante fue la consulta en CABIN e Internet.

# **I CONCEPTOS GENERALES**

## **1.1 La valuación como guía de decisión**

La valuación, entendida como el proceso de análisis que permite llegar a una estimación de valor, que puede servir como una guía de decisión, provee una base de información importante y frecuentemente determinante en la toma de decisiones relativas a un bien raíz, a programas de desarrollo o a muy diversos aspectos relacionados con inmuebles.

El análisis se hace en función del valor y de la productividad del objeto valuado. Aunque si bien éste no es el único factor determinante en la toma de decisiones inmobiliarias, generalmente es el más importante y en algunos casos el único.

Sin embargo, en sí, generalmente no son una decisión, proporcionan información con la cual, los ejecutivos, los inversionistas, los propietarios individuales, etc., tienen una guía para llegar a la mejor decisión posible. Así, si una hipotecaria solicita un avalúo, el valuador no se pondrá a considerar la conveniencia o inconveniencia que el conocer dicha hipoteca pueda tener para la empresa, simplemente realizará un avalúo con fines de préstamo hipotecario, que pondrá a disposición del ejecutivo, quien tendrá que dar aprobación o desaprobación para tal operación.

Un valuador podrá ayudar al vendedor a decidir qué tanto pedirá por una propiedad, o a un comprador a decidir que tanto ofrecerá, pero el valuador nunca tomará la decisión final. Este procedimiento se muestra en la figura 1.1.1.

Debido a lo anterior se ha considerado que el valuador no debería conocer el propósito de la valuación. Ya que según las diferentes interpretaciones de valor, es imposible para un valuador proceder a realizar su análisis sin el conocimiento del fin que se sigue y así se emitan posibles deformaciones o sesgos de información derivados de una tendencia natural.

## Diagrama para la toma de decisión con respecto a un predio.

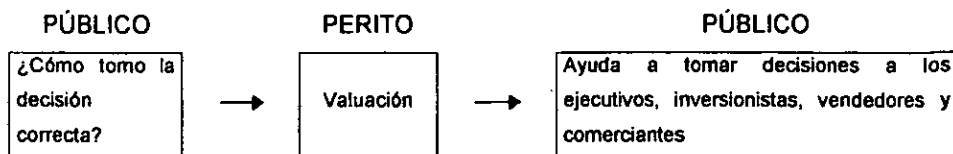


Figura 1.1.1 Los aspectos importantes para la toma de decisiones.

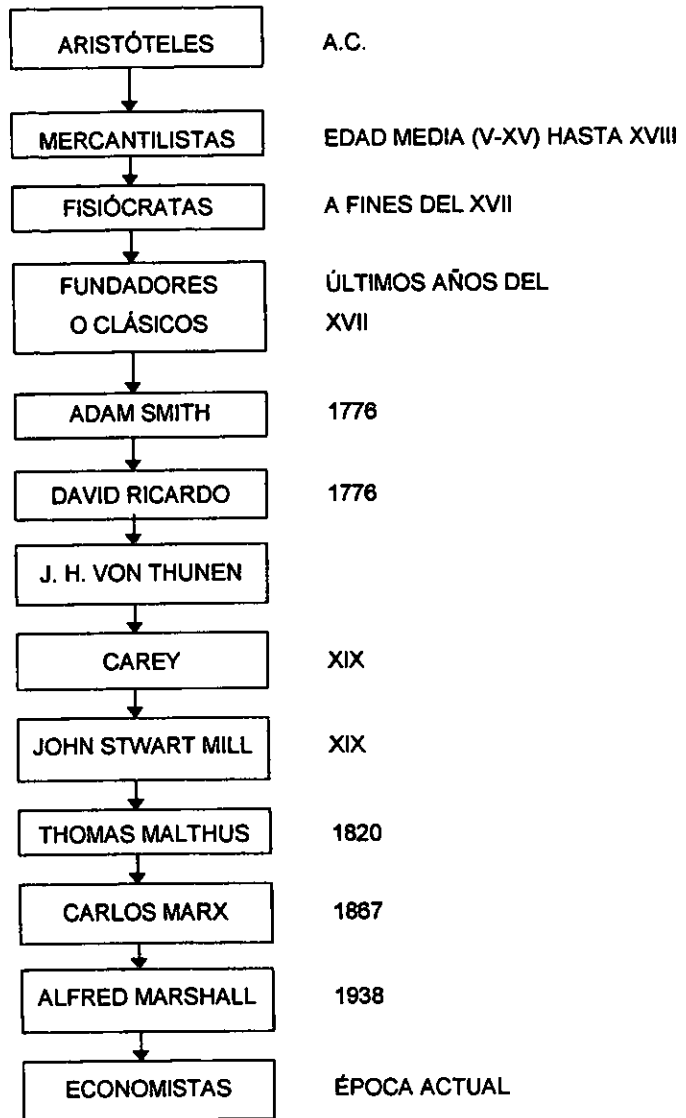
### 1.1.1 Antecedentes

A pesar que a lo largo de la historia, la disciplina de valuación se ha definido de varias maneras y para efectos del presente trabajo se ha mencionado que "tentativamente" definiríamos valuación ya que existirán varias maneras de definir esta disciplina, se ha examinado el común denominador del concepto pudiéndose definir.

Al intentar saber "Qué es la Valuación" el presente trabajo tendrá que referirse al conocimiento de valor, por lo que se debe empezar por distinguir cuáles y qué tipos de valores se va a encontrar. A través de la historia han existido etapas importantes para la definición de valor como lo muestra la figura 1.1.2.



## ANTECEDENTES DE LA VALUACIÓN (VALOR)



**Figura 1.1.2 Etapas importantes para la definición de valor.**

Ante la creciente necesidad de establecer valores a los bienes materiales que constituyen el patrimonio de una sociedad, es la valuación comercial la rama de la investigación económica la que proporciona las bases técnicas y las metodologías adecuadas para llevar a cabo tan importante y trascendental tarea.

Aún cuando los primeros trabajos de valuación inmobiliaria fueron realizados a fines del siglo pasado, es propiamente en la década de "los treintas" cuando se puede considerar el inicio formal del estudio de la valuación comercial en nuestro país.

Aquellos avalúos se referían exclusivamente a tasaciones cuya finalidad era la tributación predial. Los estudios posteriores, correspondieron a la fijación de garantías en los primeros créditos hipotecarios concedidos en ese entonces.

Otros dictámenes valuatorios y estudios financieros fueron realizados con el propósito de respaldar los créditos a gobiernos de estados y municipios, necesarios para llevar a cabo las obras públicas indispensables, tales como: introducción y abastecimiento de agua potable, alcantarillado, pavimentación, construcción de mercados, etc.

Cabe señalar que los procedimientos de valuación guían las normas establecidas por el Catastro del Departamento del Distrito Federal, pero ya con un criterio comercial, es decir con valores que debían reflejar la realidad del mercado inmobiliario.

Aún a la fecha, algunos criterios establecidos en tales Instructivos, constituyen una de las herramientas disponibles para realizar un estudio valuatorio. Sin embargo la práctica ha venido a demostrar en la actualidad que cada vez son menos operantes.

### **1.1.2 Definición de valuación**

Para definir el presente trabajo la **valuación** se define como "El procedimiento de estimar el precio justo de un bien, un producto o un servicio"<sup>1</sup>.

---

<sup>1</sup> Luis F. Cuellar. Introducción de la valuación de inmuebles urbanos. Mayo 1986. Pág. 2.

## **Avalúo**

Es el documento en el cual se ha reflejado la estimación del valor de un bien raíz en función de una unidad monetaria, para un mercado dado y a una fecha determinada.

La valuación dista mucho de ser una actividad simplista, derivada de la aplicación de fórmulas y llenado de cuestionarios. Es una especialidad en la que el buen juicio, la experiencia y principalmente la ética profesional se conjugan para la emisión de un dictamen.

El Código Civil para el Distrito Federal señala lo siguiente:

"ART. 750.- Son bienes inmuebles:

- a) El suelo y las construcciones adheridas a él.
- b) Las plantas y árboles, mientras estuvieren unidos a la tierra y los frutos pendientes de los mismos árboles y las plantas, mientras no sean separadas de ellos por cosechas o cortes regulares.
- c) Todo lo que esté unido a un inmueble de manera fija, de modo que no pueda separarse sin deterioro del mismo inmueble o del objeto a él adherido.
- d) Las estatuas, relieves, pinturas u otros objetos de ornamentación, colocados en edificios, en tal forma que revele el propósito de unirlos permanentemente al fundo (finca rústica), etc.

Así pues, un avalúo es una estimación del valor de un inmueble. El resultado de un avalúo puede ser dado en forma oral, pero generalmente es un reporte escrito elaborado por el valuador que ofrece un análisis del valor estimado de la propiedad debidamente identificada y a una fecha dada, justificándolo por la presentación y análisis de los diversos factores de mercado que intervienen.

Cualquier decisión acerca de un inmueble que involucre cambio de propiedad, renta del mismo, remodelación o destrucción, debe estar basada en un avalúo. Es frecuente que

una vez realizada una transacción que involucre un inmueble, las personas se percaten de que existían mejores soluciones.

## **Niveles de valuación**

La necesidad de los avalúos es universal y se realiza en tres niveles diferentes:

### **Primer nivel de actividad valuatoria**

Este nivel está comprendido por el público. Es la gente que usa el bien raíz, lo posee y lo emplea. La gente tiene que hacer decisiones para comprar, hipotecar, rentar, asegurar, calcular daños, desarrollar fraccionamientos, renovar, rehabilitar proyectos y otras construcciones que requieren de la inversión de sumas importantes de dinero.

Cuando un individuo se encuentra ante estos problemas, puede hacerlo con base en sus propias decisiones o bien acudir a un valuador profesional. De cualquier forma se hace un avalúo el cual estará en función de la integridad y capacidad de la persona que lo realiza.

La profesión de valuador es adquirida a través de una larga y costosa experiencia.

La gente que por error no acude a los servicios de un valuador, puede caer en equivocaciones, tales como:

- 1) Un comprador que paga más de lo que vale la propiedad que adquiere.
- 2) Un vendedor que vende su propiedad a un precio inferior de su verdadero valor.
- 3) Un propietario que firma un contrato que con el tiempo le reportará pérdidas.
- 4) Un inquilino que firme un contrato de rentas que su negocio no podrá sostener.
- 5) Individuos que realicen mejoras a sus inmuebles que no reeditarán en forma conveniente.

## **Segundo nivel de actividad valuatoria**

Este nivel está comprendido por las personas que intervienen en la valuación del bien raíz: vendedores, corredores, inversionistas, constructores, administradores y prestamistas. Asimismo, pueden considerarse valuadores de catastro, de derecho de vía y otras actividades gubernamentales.

Estos individuos no son valuadores profesionales, pero están en constante contacto con el valor de la propiedad. Son fuentes de información para el público, pero deben siempre limitar sus opiniones de acuerdo con sus posibilidades. Conviene, cuando el problema lo amerite, que recomienden los servicios de un valuador profesional.

## **Tercer nivel de actividad valuatoria**

El tercer nivel de actividad valuatoria está constituido por los valuadores profesionales. Este personal está dedicado exclusivamente para trabajos de valuación, contando con conocimientos, práctica, integridad y se identifican profesionalmente por su designación u otros medios académicos.

La necesidad para estos servicios es reconocida por los bancos, sindicatos, hipotecarias y compañías aseguradoras.

### **1.1.3 Cómo influye en la toma de decisiones**

#### **Objeto de un avalúo**

El objeto fundamental de un avalúo es estimar un valor determinado. Un avalúo provee bases firmes para tomar una decisión o establecer una política.

Las razones para la existencia de los avalúos entre otras muchas, pueden ser las siguientes:

- 1) Relación con el traslado de dominio.
  - a) Para ayudar a los vendedores a determinar precios de venta aceptables.
  - b) Para ayudar a los compradores a decidir si el precio de compra es correcto.
  - c) Para establecer un justo valor de la propiedad que va a ser motivo de compra-venta.
  - d) Para establecer valores de la unión o desintegración de propiedades múltiples.
  - e) Para distribuir los bienes de un conjunto.
  
- 2) En relación con financiamiento y crédito.
  - a) Determinar un valor que ofrezca seguridad con el propósito de conocer hipotecas.
  - b) Para proveer a un inversionista con bases firmes en la decisión de comprar cédulas hipotecarias o bonos.
  
- 3) Para establecer un valor justo en expropiaciones.
  - a) Estimar el valor del conjunto antes de la expropiación.
  - b) Estimar el valor después de la expropiación.
  - c) Para determinar el valor de la parte adquirida y el daño a la parte restante.
  
- 4) Para establecer bases de impuestos.
  - a) Para distribuir los bienes conociendo el valor de las partes destructibles y no destructibles, para aplicar diferentes tasas de capitalización.
  - b) Para determinar impuestos hereditarios o de donaciones.
  
- 5) Valor de Seguro.
- 6) Valor Contable.
- 7) Valor de Liquidación.
- 8) Valor Catastral.

Este listado no incluye todos los objetivos de los Avalúos pero indica un amplio margen de las actividades del valuador profesional.

El propósito más importante es diferenciar entre el objeto del avalúo (que es estimar un tipo específico de valor) y la función para que este valor estimado esté requerido.

Un requisito para practicar cualquier avalúo, es conocer el objeto del avalúo.

Para los efectos de la emisión de un dictamen valuatorio, es importante distinguir entre lo que entendemos por objeto de un avalúo y destino del mismo. A este respecto, cabe señalar que no se pretende cuestionar la ética del perito valuador, ni atender a los propósitos específicos de un solicitante, sino que en ocasiones es útil y en otras, indispensable, que se conozca la razón por la cual se está solicitando el avalúo.

## **1.2 Conceptos y variables que inciden en el valor de un predio**

### **1.2.1 Terrenos**

Un terreno es la porción ó porciones de tierra, incluyendo, en su caso, las construcciones ó mejoras, cuyos linderos con propiedades ajenas se encuentran totalmente definidos y determinan un perímetro sin solución de continuidad.

Existen diversos tipos de terrenos que van desde los urbanos, divididos en varias categorías que se analizan más adelante, hasta aquellos que por su ubicación geográfica, topográfica, problemas de subsuelo, etc., son suelos sin valor, pero que sin embargo dada la dinámica de las actividades y el desarrollo del país, con el tiempo van adquiriendo importancia "comercial".

Entre estos dos extremos se encuentran los terrenos suburbanos, industriales, rústicos, agrícolas, ganaderos, etc. Esta diversidad de terrenos esbozada a grandes rasgos que requiere de atención especial.

A continuación se desglosan las características de las diferentes categorías de terrenos.

### **1.2.1.1 Terrenos urbanos**

Son aquellos que se encuentran ubicados dentro de la zona urbana, fijada por las autoridades.

Cuentan con servicios municipales completos, tales como agua, luz, alumbrado, drenaje, alcantarillado, pavimentos, transporte, banquetas y guarniciones, etc.

Para los terrenos urbanos se determina un lote representativo de una zona determinada, al cual se le denomina "Lote Tipo". Los lotes tipo no deben tener frente menor de 7 mts. y tendrán una relación de frente a fondo de 1 a 3.

La valuación de los predios urbanos se rige por las siguientes bases:

1.- Según la zona:

- a) Ubicación y categoría de la zona.
- b) Densidad de población.
- c) Tipo de construcción dominante.
- d) Servicios municipales con que cuenta.

2.- Del lote:

- a) Lote tipo dominante.
- b) Proporción y forma del lote.
  - \* Predios regulares.
  - \* Predios irregulares.
  - \* Predios con poco frente.
  - \* Predios mixtos.
  - \* Predios con fondo excesivo.
  - \* Predios en esquina o esquinas.
  - \* Predios con 2 o más frentes.
  - \* Predios interiores o enclavados.



\* Predios con topografía accidentada.

- Desnivel en ascenso, aclave.
- Desnivel en descenso, declive.

### **1.2.1.2 Terrenos suburbanos**

Son aquellos que cuentan con los más indispensables servicios, aunque en algunos casos estos sean colectivos, caso de hidrantes públicos, energía eléctrica, etc.

Estos predios, originalmente de tipo rústico, por su ubicación han venido incorporándose a la zona urbana y se podrán llamar como de transición, se encuentran con frecuencia en la periferia de las poblaciones, aunque en algunos casos, son terrenos que han sido absorbidos por la ciudad y rodeados por ésta. Es conveniente hacer notar que este tipo de predios deben su estancamiento en relación a las zonas vecinas, a problemas de tipo legal, en ocasiones de muy difícil solución. Tal es el caso de ejidos que requieren para su legalización de un decreto presidencial, o de gestiones ante el departamento de asuntos agrarios, etc.

Como en los casos anteriores, es posible identificar un lote tipo y es factible también allegarnos a datos de operaciones recientes, sin embargo hay que considerar que estos predios frecuentemente cambian de propietario, pues la gente que los posee es por lo general inestable desde el punto de vista económico.

### **1.2.1.3 Terrenos industriales**

Son los terrenos destinados a construcción de industrias ya sea dentro o fuera de las zonas industriales. Como todos los tipos antes vistos, estos tienen a su vez características singulares.

El lote tipo industrial está fijado con dimensiones de 20 m. de frente por 70 m. de fondo. Los terrenos industriales no requieren de explicación especial, pues son fácilmente identificables sin embargo tienen unas condiciones de valor muy peculiares.

En primer término, su ubicación dentro de la zona urbana les da incremento o les resta valor según su caso.

Analizando para mayor objetividad la Ciudad de México, D.F., tenemos en ella zonas muy claramente marcadas, las cuales están formadas por áreas con diversas características especiales por su ubicación pero que comparten un común denominador que las identifica como pertenecientes a la misma zona.

En cuanto a su destino, tenemos por ejemplo las siguientes: al Norte, la Colonia Santa María; al noreste de la Merced, la Progresista, la 20 de Noviembre, la Moctezuma, la Granjas México, la Tres Estrellas, etc. Estas zonas industriales de tipo antiguo, están ya totalmente rodeadas por predios con usos diversos, principalmente de habitación obrera.

Existen también fraccionamientos industriales modernos como el Industrial Vallejo y algunas zonas aledañas al D.F., en las que se ofrecen a la industria servicios completos.

Conviene hacer notar los pros y contras de una de ellas.

Las zonas antiguas al estar dentro de la ciudad presentan la siguiente ventaja: mayor cercanía a los centros de consumo, a las zonas de habitación obrera, oficinas gubernamentales, mayor diversidad de vías de comunicación, transportes, etc.

En contraposición, estas zonas muestran los siguientes inconvenientes: por lo general la materia prima proviene de fuera de la ciudad y su acceso se dificulta, por tanto se encarece.

Las posibilidades de crecimiento de una industria se ven restringida al no haber oferta de predios vecinos, es decir, aumenta el valor de la tierra, llegando a ser la renta excesiva en función al capital que representa la inversión en terreno. Estos son solo ejemplos de los problemas que tiene la industria, aparte de las molestias que causa al resto de la zona.

Al hablar de las zonas industriales de formación reciente, prácticamente se podrían invertir los términos, o sea que lo que se vio como inconveniente, ahora resulta favorable, pero, sin embargo lo que se veía como conveniente en las antiguas zonas, poco a poco se va encontrando en las nuevas, al descentralizar las oficinas de gobierno, al aumentar los transportes hacia ellas, etc.

Es importante conocer los factores externos que pueden influenciar el valor de este tipo de inmuebles, basta como ejemplo el citar la nueva ley federal del trabajo, con las obligaciones obrero patronales enfocadas a la vivienda, las restricciones a la instalación y funcionamiento de industrias nuevas en el D.F., etc.

Como vemos el conocimiento de todos estos factores, aparentemente ajenos a nuestras actividades, son los que pueden, en determinado momento, cambiar totalmente los antecedentes de valor en estas zonas.

#### **1.2.1.4 Terrenos rústicos**

Son aquellos que se ubican fuera de las poblaciones y que carecen de los servicios fundamentales como pavimentos, alumbrado, agua, drenaje, etc. Por lo general son terrenos agrícolas, predios destinados a habitación campesina (pueblos, rancherías, villas, etc.). Como en todos los casos vistos con anterioridad cada terreno tendrá características propias, por lo que sólo se apuntarán sus directrices generales a modo de formar un criterio común.

Los valores de estos predios varían, siendo las principales causas las siguientes:

- 1.- Para predios destinados a habitación campesina:
  - I.- En relación al poblado.
    - b) Ubicación.
    - c) Servicios o su cercanía a ellos.
  - II.- En relación a ciudades próximas.
    - b) Vías de comunicación.
    - c) Transportes.
    - d) Importancia de las ciudades cercanas.
  - III.- En relación a los centros de trabajo.
  
- 2.- Predios destinados a la agricultura, en estos podemos agrupar para sus fines de estudio a los destinados a la ganadería.
  - I.- Ubicación en relación al poblado.
    - b) Servicios (riego, temporal, cerril, etc.).
    - c) Comunicaciones.
  - II.- Calidad de la tierra según sus rendimientos.
  - III.- En relación a la demanda del producto.
    - b) Costo de elaboración.
    - c) Costo de venta.
  - IV.- Futuro y plusvalía.

### **1.2.2 Factores**

Un factor es una variable que incide directamente en el valor de un predio. Con la obtención de los factores se puede aplicar un incremento o decremento al importe del bien evaluado. A continuación se mencionan los principales factores que se encuentran en un predio.

### 1.2.2.1 Factor de zona

Este factor toma en cuenta dónde está situado el predio con respecto a las calles y avenidas principales que se encuentran en esa zona. Esto se puede entender de forma más clara observando las características que se muestran en la figura 1.2.2.1.1.

CARACTERÍSTICAS	Factor (FZo)
Único frente a la calle moda de la zona.	1.00
Ningún frente a calle superior a la calle moda y al menos uno a la calle moda.	1.00
Al menos un frente a corredor de valor.	1.00
Sin frente a calle alguna.	1.00
Al menos un frente a calle superior a la calle moda o a un parque o plaza (y ninguno a corredor de valor).	1.00
Único frente o todos los frentes a calle inferior a la calle moda.	0.80

Figura 1.2.2.1.1 Consideraciones para la obtención del factor de zona (FZo).

### 1.2.2.2 Factor de ubicación

Este factor toma en cuenta la ubicación del lote respecto a la manzana y se obtiene según la figura 1.2.2.2.1.

CARACTERÍSTICAS	Factor de ubicación (FUb)
Sin frente a vía de circulación.	0.70
Con frente a una sola vía de circulación.	1.00
Con frente a dos vías de circulación.	1.15
Con frente a tres vías de circulación.	1.25
Con frente a cuatro o más vías de circulación.	1.35

Figura 1.2.2.2.1 Consideraciones para la obtención del factor de Ubicación (FUb).

### 1.2.2.3 Factor de topografía

Este factor toma en cuenta principalmente la pendiente del terreno y se procede como sigue:

$$FT_o = 1 - 0.56t;$$

donde  $t$  = talud equivalente:  $0 < t < 1$

Si la topografía del lote individual presentara mejores condiciones que la topografía de los lotes adyacentes, el factor de topografía puede ser mayor a 1 y se asignará directamente mediante la evaluación de las condiciones presentes.

### 1.2.2.4 Factor de frente

Este factor se determina tomando en cuenta la longitud de lo que se considera como parte frontal del terreno. Este factor es directamente proporcional a dicha longitud y varía según la siguiente tabla 1.2.2.4.1:

CARACTERÍSTICAS	Factor de frente (FFr)
Frente igual o mayor a 7.00 metros.	1.00
Frente igual o mayor a 4.00 y menor a 7.00 metros.	0.80
Frente menor a 4.00 metros.	0.60
Nota: Los predios cuyo frente tenga como mínimo una dimensión de 6.90 metros se considerarán con frente de 7.00 metros.	

Figura 1.2.2.4.1 Consideraciones para la obtención del factor de frente (FFr).

### 1.2.2.5 Factor de forma

Este factor considera la geometría del predio diferenciando zonas habitacionales de otras zonas, y se divide en dos casos principalmente:

- a) Para terrenos cuyo fondo sea igual o menor a tres veces el frente y cuya poligonal conforme ocho o menos ángulos. (Figura 1.2.2.5.1)

$$FFo = \sqrt{Ri/Sto}$$

Ri= Rectángulo inscrito.

Sto= Superficie total del predio.

Figura 1.2.2.5.1 Consideraciones para la obtención del factor de forma (FFo), para el caso a).

b).- Para terrenos cuyo fondo sea mayor a tres veces el frente o cuya poligonal conforme nueve o más ángulos se realiza la suma de las fórmulas que se presentan en las siguientes tablas de la figura 1.2.2.5.2:

NOMBRE	EFICIENCIA	CLAVE
Porción Anterior	$E_{Pa} = \frac{S_{Pa}}{S_{To}}$ $E_{Pa} = 1.00$	$E_{Pa}$ = Eficiencia de la porción anterior del rectángulo descrito $S_{Pa}$ = Superficie de la porción posterior $S_{To}$ = Superficie total del predio

NOMBRE	EFICIENCIA	CLAVE
Porción Posterior	$E_{Pp} = \frac{S_{Pp}}{S_{To}}$ $E_{Pp} = 0.70$	$E_{Pp}$ = Eficiencia de la porción Posterior $S_{Pp}$ = Superficie de la porción posterior $S_{To}$ = Superficie total del predio

NOMBRE	EFICIENCIA	CLAVE
Areas Irregulares Con frente a la vía de acceso	$E_{Ac} = \frac{S_{Ac}}{S_{To}}$ $E_{Ac} = 0.80$	$E_{Ac}$ = Eficiencia de las áreas irregulares con frente a la vía de acceso $S_{Ac}$ = Superficie de las áreas irregulares con frente a la vía de acceso $S_{To}$ = Superficie total del predio

NOMBRE	EFICIENCIA	CLAVE
Áreas Irregulares interiores	$S_{ai}$ $E_{ai}=0.50$ ----- $S_{to}$	$E_{ai}$ = Eficiencia de las irregulares Interiores $S_{ai}$ = Superficie de las áreas irregulares interiores $S_{to}$ = Superficie total del predio

$FFo = E_{Pa} + E_{Pp} + E_{Ac} + E_{ai}$
---

Figura 1.2.2.5.1 Consideraciones para la obtención del factor de forma (FFo), para el caso b).

### 1.2.2.6 Factor de superficie

Este factor es determinado en base a la superficie total del predio en estudio. La tabla se muestra a continuación. (Figura 1.2.2.6.1)

Factor de superficie (FSu)							
	RLm	Fsu	RLm	FSu			
Slo	Hasta 2.0	1.00	11.1	12.0	0.80		
RLm = -----	2.1 3.0	0.98	12.1	13.0	0.78		
SLm	3.1 4.0	0.96	13.1	14.0	0.76		
	4.1 5.0	0.94	14.1	15.0	0.74		
RLm = Relación con el	5.1 6.0	0.92	15.1	16.0	0.72		
lote moda	6.1 7.0	0.90	16.1	17.0	0.70		
Slo = Superficie del lote	7.1 8.0	0.88	17.1	18.0	0.68		
Que se está valuando	8.1 9.0	0.86	18.1	19.0	0.66		
	9.1 10.0	0.84	19.1	20.0	0.64		
SLm = Superficie del lote	10.1 11.0	0.82	20.1	y más	0.62		
moda							

Figura 1.2.2.6.1 Tabla que muestra las consideraciones para el cálculo del factor de superficie.



El factor resultante se obtiene multiplicando los diferentes factores como son: zona, ubicación, frente, forma, superficie, etc. Como se muestra en el recuadro de la figura 1.2.2.6.2.

<b>FACTOR RESULTANTE DE TIERRA (FRe)</b>
$FRe = FZo \times FUb \times FFr \times FFo \times Fsu$
Nota: Para efectos de revisión inicial, el factor resultante de tierra nunca será menor que 0.60; deberán utilizarse sólo dos decimales para cada factor.

Figura 1.2.2.6.2 Fórmula para la obtención del factor resultante en función de los demás factores.

### 1.2.2.7 Factor de homologación por uso de suelo

Este factor es determinado con base al uso de suelo que le corresponda a la zona en donde esté ubicado el predio; el uso de suelo puede ser comercial, industrial, habitacional, etc. Este factor algunas veces se determina de acuerdo con el criterio del perito valuador que esté realizando el avalúo del predio.

### 1.2.2.8 Factor de homologación por servicios públicos

Este factor es determinado con base en los servicios públicos con los que cuenta la zona en que se ubica el predio, como son: agua, drenaje, alumbrado, transporte, etc. También en este caso el criterio del perito valuador que esté realizando el avalúo del predio juega un papel determinante para este factor.

### 1.2.2.9 Factor por comercialización

En este factor se involucra una investigación exhaustiva del mercado, que permite identificar los precios a los que se están vendiendo los lotes, considerando los antecedentes del predio, así como las condiciones actuales y perspectivas físicas, políticas, sociales y jurídicas del mismo, para que determinado precio se disminuya o se eleve.

### 1.2.3      **Uso de suelo**

#### **Introducción a la planeación urbana en el D.F.**

Todo asentamiento humano, lugar donde se desarrolla la convivencia humana requiere de reglas y normas que ordenen y controlen su funcionamiento y crecimiento en beneficio de su población.

Es así que en los artículos 27, 73 y 115, de la Constitución Política de los Estados Unidos Mexicanos se concede el derecho a la nación para regular la propiedad privada imponiendo las modalidades que dicte el interés público con el objeto de ordenar los asentamientos humanos y establecer usos, reservas y destinos a la tierra, premisas bajo las cuáles la propiedad particular queda sujeta a que determinados usos y/o actividades que pretendan desarrollar queden condicionados o prohibidos.

#### **Conceptos Básicos del uso de suelo**

**Los usos del suelo** se refieren a aquella actividad específica a la que se encuentra dedicado o se pretende dedicar un predio en particular, los cuales pueden ser habitacional, comercial, restaurante, oficinas, clínica, etc.

**La densidad habitacional** es la relación que existe entre el número de viviendas y la superficie donde éstas se encuentran, considerando como promedio 5 integrantes por familia. Esta se expresa en habitantes por hectárea (hab/ha).

**La intensidad de construcción** es la relación que existe entre el número de metros cuadrados construidos y la superficie del predio en el cual se encuentran. Este valor se expresa en "n" veces el área del terreno y principalmente se refiere a usos de suelo no habitacionales.

La siguiente tabla (figura 1.2.3.1) ejemplifica los conceptos básicos dados anteriormente:

Clave	Tipo	Lote moda
H5	Habitacional hasta 50 hab./Ha	1,000 m <sup>2</sup>
H1	Habitacional hasta 100 hab./Ha	500 m <sup>2</sup>
H2	Habitacional hasta 200 hab./Ha	250 m <sup>2</sup>
H4	Habitacional hasta 400 hab./Ha	125 m <sup>2</sup>
H8	Habitacional plurifamiliar	600 m <sup>2</sup>
H2S	Habitacional con servicios hasta 200 hab./Ha	250 m <sup>2</sup>
H4S	Habitacional con servicios hasta 400 hab./Ha	125 m <sup>2</sup>
H2I, H4I	Habitacional industria mezclada	500 m <sup>2</sup>
H2IS, H4IS	Habitacional industria mezclada y servicios	500 m <sup>2</sup>
IV	Industria Vecinal	500 m <sup>2</sup>
IA	Industria aislada	2,000 m <sup>2</sup>

Figura 1.2.3.1 Tabla para la obtención de la superficie moda cuando no se puede determinar directamente.

### El uso del suelo y su clasificación

La delimitación de las colonias catastrales, se facilita fijando como criterio básico la clasificación del uso del suelo.

Los usos de suelo típicos en zonas urbanas son:

- **Uso residencial**, donde predominan viviendas propias o rentadas, con casas separadas o semiseparadas, casas construidas espalda con espalda, bungalows (fig. 1.2.3.2).



Figura 1.2.3.2 Ejemplo de uso de suelo residencial.

- **Uso residencial departamental**, donde se encuentran principalmente edificios de condominios o departamentos alquilados (fig. 1.2.3.3).

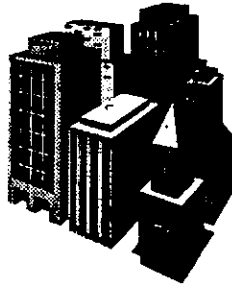


Figura 1.2.3.3 Ejemplo de uso de suelo residencial departamental.

- **Uso comercial**, que se caracteriza por la densidad de comercios, negocios, oficinas, hoteles, etc. (fig. 1.2.3.4).

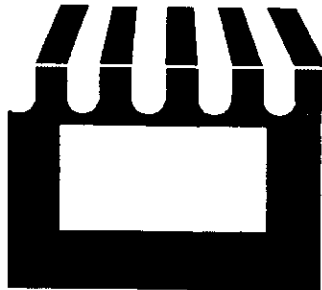
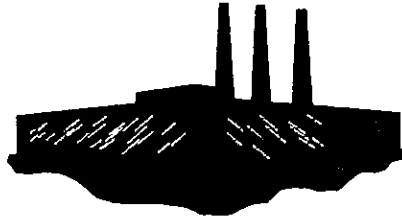


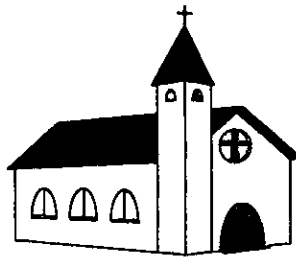
Figura 1.2.3.4 Ejemplo de uso de suelo comercial.

- **Uso industrial**, en zonas usadas principalmente para el establecimiento de fábricas, oficinas, almacenes, plantas de energía, depósitos de abastecimientos (fig. 1.2.3.5).



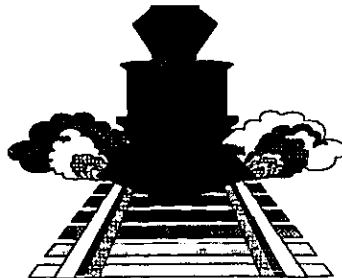
**Figura 1.2.3.5** Ejemplo de uso de suelo industrial.

- **Uso institucional**, áreas con edificios públicos gubernamentales, escuelas, hospitales, iglesias, etc. (fig. 1.2.3.6).



**Figura 1.2.3.6** Ejemplo de uso de suelo institucional.

- **Uso para transporte**, donde se encuentran vías de carreteras, ferrocarriles o fluviales (fig. 1.2.3.7).



**Figura 1.2.3.7** Ejemplo de uso de suelo para transporte.

## El análisis del uso del suelo

El análisis del uso del suelo sirve ante todo para establecer niveles económicos que inciden en forma determinante en los valores de la tierra. Un primer parámetro es el uso del suelo a nivel de una población o colonia.

Se distinguen:

- Poblaciones primarias, y
- Poblaciones secundarias.

- *Poblaciones primarias:*

Son aquellas que tienen una base económica propia. De acuerdo con su origen o su destino se pueden agrupar en:

- Poblaciones (o ciudades o colonias) industriales.
- Poblaciones (o ciudades o colonias) comerciales.
- Poblaciones (o ciudades o colonias) de recreo.
- Poblaciones (o ciudades o colonias) políticas.
- Poblaciones (o ciudades o colonias) educativas, etc.

- *Poblaciones secundarias:*

Son aquellas que dependen económicamente de otras, como por ejemplo ciudades satélites, ciudades residenciales, etc.

Un segundo parámetro en el análisis del uso del suelo son los **indicadores de crecimiento económico**, entre los más sobresalientes están:

- Número de bancos, depósitos y ahorros.
- Actividad del correo.

- Número de manifestaciones de obras.
- Número de automóviles registrados.
- Número de teléfonos.
- Porcentaje de la población económicamente activa.
- Número de librerías, etc.

Debido a la influencia de fuerzas socioeconómicas, las colonias evolucionan conforme a un patrón cíclico de edades que tiene un efecto directo en el nivel económico de la colonia.

El patrón muestra cinco etapas:

- Crecimiento.
- Madurez, a los 15-20 años de haberse formado.
- Declinación.
- Transición, a los 30-40 años.
- Rehabilitación, durante los siguientes 10 años.

Este patrón se repite después varias veces más con intervalos más cortos y con una tendencia general de declinación.

### **Clasificación del suelo según sus características**

Obviamente, algunos lotes urbanos dentro de una misma manzana, son superiores a otros porque sus características físicas son diferentes.

Estas características son:

- Localización dentro de la manzana.
- Forma.
- Topografía.
- Tamaño.

#### **1.2.4 Investigación de Mercado**

El punto inicial para un avalúo es el estudio de mercado, como parte de todo análisis.

El valor de los inmuebles se obtiene con base en su tipo, ubicación, edad, calidad, diversificación del uso del suelo, proyecto, estado de conservación, (referido a homologaciones de los valores de inmuebles en el mercado para llegar al valor justo del inmueble que se está valuando), esto incluye la demanda real, las ventajas competitivas, la posición actual y estimada del mercado de influencias primarias y secundarias.

#### **1.2.5 Tipos de Valor**

##### **Concepto de Valor usado en Valuación**

En el diccionario, valor se define como la relación entre un objeto deseado y un sujeto con posibilidad de adquirirlo. Si hay abundancia de productos es lógico que baje el valor de los mismos. Un producto no tiene valor si no tiene utilidad. Un objeto puede o no tener utilidad según las necesidades o deseos de adquirirlo.

Otro factor necesario para que un objeto tenga valor es que puede ser adquirido.

El valor no es una característica inherente a un objeto sino que depende de los deseos del hombre de adquirirlo y varía de persona a persona y de tiempo en tiempo como varían los deseos del individuo.

Una propiedad no puede tener valor, a menos que tenga utilidad y debe entenderse por utilidad la cualidad de fomentar el deseo en el hombre de poseerlo.

La escasez siempre está presente en el valor. La utilidad sola no le da al bien raíz ningún valor, ya que también debe ser relativamente escaso. Así la calidad más la escasez son dos de los elementos creadores del valor.



La utilidad y la escasez unidas no confieren valor a un inmueble, a menos que exista el deseo en el mercado de comprar y que a su vez se tengan los medios para hacerlo.

Para realizar un avalúo será indispensable tomar en cuenta las condiciones futuras, pudiendo fijarse las tendencias que pueden cambiar más o menos el valor de un inmueble a futuro.

### **Valor de Mercado**

Este es muy importante, ya que indica las relaciones entre compradores, valuadores, vendedores e inversionistas.

Definición de Valor de Mercado.

- 1) El Valor de Mercado es el precio más alto que se le da a una propiedad, expuesta a la venta en el mercado y cuya realización se lleva a efecto en un tiempo razonable.
- 2) Es el precio en que el vendedor y comprador aceptan realizar una operación de compra-venta sin ninguna presión.
- 3) Es el precio en el que se realiza una operación de compra-venta en un tiempo razonable siempre y cuando el comprador y vendedor estén bien informados del valor del inmueble.
- 4) Valor de mercado es el estimado para un inmueble y que se presume sería el precio de venta a la fecha del avalúo que lo determina, si fuera posible transcurrirse un tiempo razonable para encontrar un comprador y si la transacción fuese típica de las condiciones existentes en el mercado.

Podemos precisar que el "valor de mercado" es el más alto precio, estimado en términos de moneda en el cual una propiedad que ha sido expuesta en mercado abierto, durante un tiempo razonable, encuentra comprador, que libre de toda coacción

y por su propia voluntad adquiere el conocimiento de todas las órdenes a que se pueda adaptar el producto y para los que tiene capacidad de ser usado.

### **Valores de un Bien Raíz**

En relación a "Bienes Raíces", podemos tener diferentes valores según distintas finalidades que se persigan al efectuar la valorización, a saber:

**Valor en Plaza:** Es aquel establecido en un mercado abierto por un número de transacciones reales efectuadas entre vendedores y compradores, sobre inmuebles de semejantes características. Específicamente cada una de estas operaciones implica el importe expresado en dinero que un comprador ha pagado por una propiedad en un determinado momento, siempre y cuando se hubiera efectuado en forma justa.

Cuando se realizan numerosas ventas en el mercado, el resultado es establecer como base de las transacciones, valores de plaza correctamente definidos.

**Valor de Garantía:** Es el valor que tiene un inmueble estimado para cubrir riesgos futuros previsible. Este valor se expresa como resultado de un reajuste del valor comercial, el cual fue determinado en base al valor en plaza.

**Valor de Reposición:** Es el importe que será necesario invertir para reemplazar un inmueble, con las mismas características y amenidades que posee.

**Valor de Reproducción:** Es el importe que será necesario invertir para rehacer un inmueble por otro exactamente igual.

**Valor Catastral:** Es aquel que se establece tomando como base los valores dados por la oficina de catastro de la localidad que sirven generalmente para fijar los impuestos prediales.

**Valor Físico:** Es el que tiene una propiedad tomando en cuenta sólo su costo de reposición menos su depreciación.

**Valor de Rentabilidad:** Se obtiene por la capitalización adecuada de las rentas anuales recibidas de una propiedad, ya afectadas por las deducciones propias correspondientes.

**Valor Comercial:** Para los efectos del avalúo comercial de un bien raíz, es la cantidad expresada en moneda, justa y probable que se podrá obtener por una propiedad en un mercado abierto y en una fecha determinada.

**Valor de Capitalización:** Se determina con base en la rentabilidad del edificio que se supondría el ideal para el máximo aprovechamiento del terreno.

### **1.2.6 Comercialización**

La comercialización es probablemente uno de los términos más comúnmente utilizados en el vocabulario de mercadotecnia. Para algunas personas, comercialización es sinónimo de mercadotecnia. Los detallistas emplean de manera particularmente amplia la palabra. Un comprador comercial de ropa puede recibir felicitaciones por haber "comercializado" muy bien una nueva camisa deportiva en esta temporada. Otro ejecutivo puede declarar que una "comercialización" a fondo es la base del éxito de una tienda. Un tercero puede señalar que los planes de "comercialización" para la próxima temporada han sido terminados. Un ejecutivo de alto nivel en una tienda de departamentos es denominado por lo común, gerente de comercialización.

Comercialización es sinónimo de planeación de producto. Así pues, cualquier relación entre individuos u organizaciones que implique un intercambio (una transacción), es mercadotecnia. Es decir, la esencia de la mercadotecnia es una transacción; un intercambio, con el objetivo de satisfacer las necesidades o gustos de las personas.

Es por esto que, dentro del ámbito de avalúo el hablar de comercialización estamos hablando de negociación entre un vendedor y el cliente, y el predio. Se tiene que dar la valorización del predio, es decir, con base en las características y a los diversos factores que afectan el predio, hay que determinar el precio de éste, es decir, al realizar el estudio de mercado y determinar el valor unitario del predio, y lo cual es sabido por ambas partes, entra en marcha la negociación, el valuador que conoce el precio, determinará qué ganancia es la que desea obtener con la venta del terreno y junto con el cliente llegará a un acuerdo mutuo.

Dentro del ámbito de avalúos y sobre todo en la valuación de bienes inmuebles, para poder obtener ganancias sobre el terreno hay que tomar en cuenta diversos factores. Principalmente los factores que afectan el terreno, con base en estos, que forman una parte de investigación y análisis de mercado, el valuador puede determinar el valor o costo real del terreno y una vez conociendo el costo puede venderlo u ofrecerlo a gente que esté interesada (cliente) y mediante esto determinar el costo real.

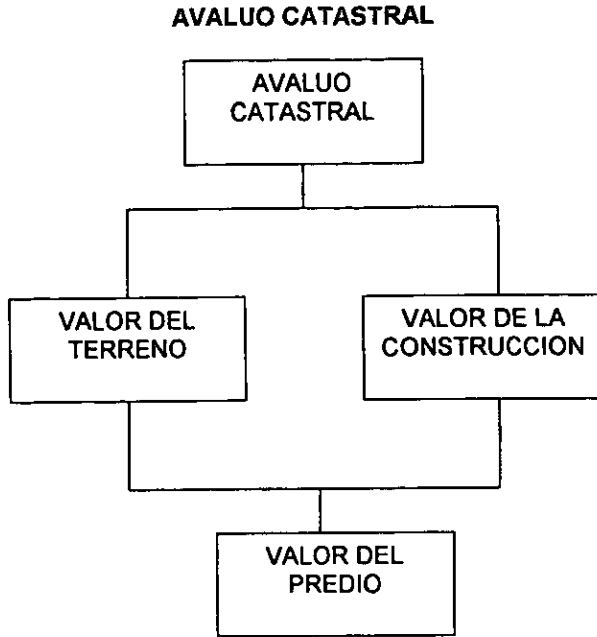
### **1.3 Tipos de avalúo**

#### **1.3.1 Avalúo catastral**

El avalúo catastral tiene por objeto determinar el valor del bien para efectos fiscales, impuesto predial ó indemnizaciones gubernamentales.

Actualmente la valuación catastral se efectúa en el D.F. de acuerdo al instructivo que para tal efecto se publicó en la Gaceta Oficial del Departamento del Distrito Federal el 15 de Julio de 1980.

El procedimiento consiste en determinar, por una parte, el valor del terreno y, por otra, el valor de la construcción, valores que sumados integran el valor del predio. Como se muestra en la figura 1.3.1.1.



**Figura 1.3.1.1 Procedimiento para la obtención del avalúo catastral.**

Para evaluar la tierra se determina el valor unitario medio "VUM" del terreno, que es igual para cada colonia catastral, definida ésta como una zona delimitada, con características de homogeneidad en su índice socioeconómico, en el tipo y edad de sus construcciones y en su desarrollo.

Así mismo, se deben tomar en cuenta características adicionales como instalaciones, tipo y calidad de mercancías cuando se trate de zonas comerciales.

Este tipo de avalúo lo realiza la Tesorería del Departamento del Distrito Federal a través de la Dirección de Catastros, en sus Departamentos de Avalúo Técnico.

El procedimiento consiste en:

- a) Determinar la región en estudio.

- b) Determinar colonias catastrales.
- c) Obtener el VUM de la colonia catastral.
- d) Determinar los factores físicos generales de la colonia.
- e) Calcular los factores físicos individuales de los lotes, si los hubiera.
- f) Calcular el valor catastral del terreno, que sumando el valor catastral de las edificaciones, dará como resultado el valor catastral del predio.

### 1.3.2 Avalúo bancario

Este tipo de avalúo se utiliza principalmente con objeto de fijar las bases para realizar transacciones, compra-venta, hipotecas, etc.; y es la Comisión Nacional Bancaria la que establece las normas para llevarlos a cabo. Cuando se utiliza con fines fiscales, deberá tener la aprobación de la Tesorería del Distrito Federal.

Los avalúos bancarios se apegan al sistema de avalúo catastral y en la actualidad se utiliza el Instructivo que antes de junio de 1980 se utilizaba en la Tesorería del Distrito Federal, sistema que, en resumen, consiste en lo siguiente:

El valor de un predio queda integrado por la suma del valor del terreno, el valor de las construcciones, y el valor de las instalaciones especiales como se muestra en la figura 1.3.2.1.

#### Obtención de un avalúo bancario.

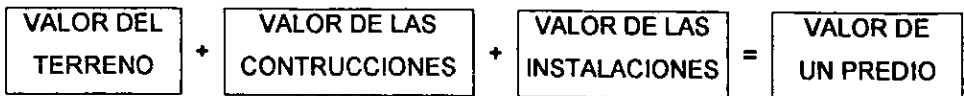


Figura 1.3.2.1 Esquema para la obtención de un avalúo bancario.

### **1.3.3 Avalúo económico**

#### **Principios:**

Generalmente las decisiones relativas a inmuebles giran en relación a los valores del mismo y a su recuperación a futuro. En esta parte se analizarán algunos conceptos relativos a diversos factores, que pueden ser usados como guías de valuación inmobiliaria.

A continuación se mencionan algunos de los conceptos fundamentales que conforman los principios de valuación:

- a) La recuperación es el factor fundamental en la vida económica.
- b) Como en otras áreas de la administración de negocios, en el campo de los bienes raíces se pone especial interés a la recuperación o ingreso que va a producir la propiedad, pero en este campo en especial, se toma en cuenta la localización del ingreso, debido a que la propiedad es un punto único en la superficie de la tierra.
- c) Las decisiones sobre bienes raíces se convierten a la larga en una reflexión sobre los factores relativos al ingreso, o al potencial de producción de ingreso, en su sitio determinado.
- d) La producción potencial de ingresos debe ser representada en valores presentes por el método de descuento de fondos; asimismo, el valor de una propiedad puede ser basado en el valor presente de tales recuperaciones a futuro.
- e) Las estimaciones del valor de los ingresos a futuro pueden estar basadas en el análisis de:
  - i) Tendencias económicas nacionales e internacionales.
  - ii) El panorama para ciertas industrias específicas.
  - iii) Las condiciones económicas regionales y locales.
  - iv) Los mercados para los diferentes tipos de bienes raíces y los costos de producción de los servicios de tales propiedades.
  - v) El panorama a futuro para los vecindarios sobre los cuales está la propiedad.

- vi) Las características de valor específicas de la propiedad.

Aunque es muy difícil en el campo de bienes raíces tener una seguridad de un valor acorde al mercado, frecuentemente es necesario hacer estimaciones de tal valor. Dichas estimaciones o valuaciones juegan un importante papel en la toma de decisiones, ya sea para comprar, vender, financiar, construir, rentar, etc. Como se apreciará mas adelante, las valuaciones pueden ser tomadas para una variedad de propósitos; por ejemplo: ¿En cuánto se podría vender tal propiedad rápidamente? ¿Cuál sería su valor para propósitos de préstamo, para cobertura de un seguro o su valor en términos catastrales? ¿Para efecto de impuestos? o bien, su valor a futuro. En fin una serie de situaciones que afectan o que intervienen en la forma de valorar tal propiedad.

#### **1.4 Métodos de valuación**

Para determinar el tipo de valuación a seguir, principalmente se toman en cuenta tres factores:

- a) El propósito de la valuación.
- b) El tipo de ingreso producido por la propiedad, o bien, que se presume que va a producir tal propiedad, y
- c) El tipo de datos con que se cuenta.

El método de valuación debe adaptarse al tipo de propiedad y al propósito de la valuación, y pueden existir o coexistir varios criterios de valuación, pues una de las cualidades de un Perito Valuador, es contar con un criterio sólidamente fundado en una preparación y experiencias adecuadas. Los métodos elegidos para encontrar el valor comercial de un inmueble, pueden ser muy variados, teóricos o prácticos, siempre y cuando los resultados sean similares y los métodos razonados y razonables; esto quiere decir, que dos peritos valuadores con preparación y criterios semejantes, deben



llegar a conclusiones semejantes y sus discrepancias de criterio no deben ir mas allá de  $\pm 10\%$  del resultado final.

Los principales métodos de valuación son:

- a) Capitalización estimada de los ingresos futuros, conocido como el método del ingreso.
- b) Determinación del costo de sustitución, conocido como el método del costo y
- c) Comparación de propiedades en precios, rentas y otros aprovechables del mercado, conocido como el método del mercado, método directo o de aplicación de valores unitarios.

#### **1.4.1 Método del ingreso**

Este método se funda en el valor futuro de ingresos presentes, es mucho más ágil, pero menos inexacto. El hecho de utilizar tasas de capitalización, no totalmente confiables, en tiempos actuales con monedas no estables, hacen que cualquier error en la apreciación del ingreso de los gastos propios de éste y de la tasa aplicable arroje, un error en la conclusión del valor.

Se recomienda que este método no se use en forma aislada, sino como auxiliar o comparativo con otros métodos. Su empleo combinado es sumamente útil en los estudios de preinversión.

La capitalización de ingresos futuros requiere la estimación de los ingresos, gastos y la estimación de los ingresos netos, sin considerar depreciación, así como la capitalización de tales ingresos a una tasa apropiada.

Para la valuación de este método se realizan tres pasos principalmente:

- a) Se estima el ingreso total, tomándolo como un ingreso posible al 100% de ocupación.
- b) Se calculan los gastos de operación, incluyendo los gastos relativos al mantenimiento.
- c) Se capitaliza el ingreso neto, resultante de la deducción de los gastos de operación del ingreso, calculado por medio de la aplicación de una tasa apropiada.

**Estimación del Ingreso:** La estimación del ingreso está basada en los datos que se tengan sobre lo que la propiedad ha producido o puede producir, tomando en consideración propiedades similares y posibilidades de futuros desarrollos. Las rentas anteriores solamente pueden servir de base si se considera que el futuro será igual que el pasado, con respecto a la propiedad; por lo tanto, el valuador debe considerar las tendencias económicas generales, las reglamentaciones locales, las tendencias del mercado local, etc. Uno de los principales errores en la estimación de ingresos futuros consiste en suponer periodos de bonanza en el futuro.

**Gastos de operación:** estos se definen como "todas las salidas actuales de caja que origina la propiedad y la operación de la propiedad, durante el resto de su vida económica, excepto aquellos gastos originados por un préstamo hipotecario"<sup>2</sup>. Los gastos de operación generalmente incluyen electricidad, agua, impuestos, publicidad, mantenimiento, etc. Los impuestos personales del propietario y los pagos por una hipoteca no constituyen gastos de operación.

Es aconsejable comparar los gastos de la propiedad con los de propiedades similares. A fin de determinar con mayor claridad los gastos que vendrán en el futuro. De igual manera los datos pasados sirven como base y pueden servir para una extrapolación a futuro.

---

<sup>2</sup> Francisco Solares Alemán. Apuntes de edificación. Tomo I. Avalúos. UNAM FI 1984. Pág 119.

**Capitalización:** El problema de la capitalización gira alrededor de los factores tiempo y tasa. El factor tiempo se refiere a la duración del inmueble, o bien, a la duración de una concesión específica. Se refiere también a los movimientos de ingresos y gastos. Existe una tendencia de los valuadores a capitalizar la perpetuidad, esto es, a suponer unos ingresos a futuro, estimando los ingresos del período siguiente al tomado como base y capitalizando esos ingresos netos a una tasa determinada.

Aunque este dato es indicativo, el tiempo a considerar debe ser el de la vida económica real de la propiedad, ya que al final de su vida económica únicamente podrá ser recuperado el valor del terreno. Aunque los terrenos tienden a incrementar su valor, el valor presente del precio de venta que a futuro pudiera tener el terreno, difícilmente puede ser igual al valor total de la construcción y terreno al momento de su valuación.

El factor tasa debe considerar la teoría de que un capital no debe ser improductivo, sino que debe estar produciendo siempre; por tanto, la tasa que debe emplearse es la tasa que dicho dinero está produciendo actualmente. Sin embargo, existen problemas de valuación que no son relativos a una empresa, cuyo capital produce una tasa determinada. Un factor a considerar en este momento es la inflación.

#### **1.4.2 Método del mercado**

Este método consiste en la comparación del inmueble valuado con otro u otros semejantes, cuyo valor real sea plenamente conocido y confiable.

El mercado inmobiliario, a través de la oferta y la demanda asigna a los predios, valores que se sustentan en el predio mismo, es decir, demuestran su preferencia, traduciéndola a costos, por determinadas situaciones.

Estas preferencias, y su monto en moneda, surgen espontáneas al comparar los valores obtenidos por los otros métodos y se consolidan a través de este método comparativo, el que en muchos países se considera como el más indicado para determinar con justicia el valor de un inmueble.

El proceso de estimación de valor por este método, incluye los siguientes cuatro pasos fundamentales:

1°. Análisis de la propiedad, considerando su uso actual y usos potenciales, características de la tierra, características de su estructura o de sus potencias de nuevas estructuras, factores de su localización, tendencias del mercado, reglamentaciones y restricciones que afecten o puedan afectar la propiedad.

El detalle con el que pueden ser hechos estos análisis estará en razón del tipo de propiedad y el propósito de la valuación. En muchos casos y debido a los propósitos de la valuación se usan formas impresas para la recolección de datos.

2°. Muestreo de propiedad con características similares a la que se está valuando. El valuador debe agenciarse información confiable acerca de tales propiedades, generalmente éstas son seleccionadas con base en precios de compra y de venta en el mercado; asimismo se investiga la oferta de ventas, la oferta de compras y la oferta de rentas. Los datos deben ser lo más recientes posibles.

3°. Análisis cuidadoso de la información obtenida. El valuador debe tomar en cuenta los datos, la fuente de los mismos y su confiabilidad, esto es, cuidar el máximo grado posible de confiabilidad, ya que los datos obtenidos pueden no ser representativos de las propiedades similares. El valuador deberá analizar también, las condiciones generales del mercado, las formas de venta y los financiamientos. Es conveniente hacer notar la importancia de la forma en la que se vende o se compra, ya que actualmente la venta masiva de bienes raíces destinadas a habitación, se realiza con facilidades financieras muy especiales, lo que da lugar a que la "elasticidad demanda - precio" esté no en razón del precio en sí, sino en razón de las facilidades con las que se paga dicho precio.

4°. Comparación de los datos obtenidos con el sujeto de la valuación. Esta comparación puede hacerse tomando los datos obtenidos, en relación con la propiedad, o bien, parte de la información con la propiedad en cuestión para reducirlo, que es lo más común, a unidades similares, aunque algunos factores involucren un aumento. Generalmente las comparaciones se hacen en razón de los factores físicos, de localización, de mercado, económicos y gubernamentales que a continuación se muestran:

Físicos	<ul style="list-style-type: none"> <li>a) Ubicación accesibilidad.</li> <li>b) Tamaño y medida del lote.</li> <li>c) Tamaño, estilo y funcionalidad del proyecto o del edificio.</li> <li>d) Condiciones en las que se encuentra la construcción.</li> <li>e) Materiales de la construcción.</li> <li>f) Número de departamentos, cuartos, oficinas, etc.</li> <li>g) Equipo con que cuenta el edificio, como la calefacción, aire acondicionado, hidroneumático, etc.</li> <li>h) Vida económica del edificio.</li> </ul>
Localización	<ul style="list-style-type: none"> <li>a) Imagen del vecindario en el que está ubicado.</li> <li>b) Las preferencias del público a ese vecindario, como lugar para vivir, para negocios, etc.</li> <li>c) La presencia o ausencia de construcciones o destinos de la tierra, que sean no armónicos con el vecindario.</li> <li>d) El futuro económico de la ciudad o de la región.</li> </ul>
Mercado	<ul style="list-style-type: none"> <li>a) Precio de venta.</li> <li>b) Rentabilidad.</li> <li>c) Facilidad o dificultad de financiamiento.</li> <li>d) Volumen de transacciones.</li> <li>e) Preferencias por varios tipos de propiedades.</li> <li>f) Costos de construcción.</li> <li>g) Tendencias del mercado hacia cambios.</li> </ul>

Económicos	<ul style="list-style-type: none"> <li>a) Ganancias.</li> <li>b) Gastos de operación.</li> <li>c) Posición de la propiedad dentro de la competencia y,</li> <li>d) Algunas condiciones especiales.</li> </ul>
Gubernamentales	<ul style="list-style-type: none"> <li>a) Impuestos.</li> <li>b) Zonificaciones y reglamentación de construcción.</li> <li>c) Restricciones.</li> <li>d) Reglamentación de tránsito.</li> <li>e) Marco urbano y servicios municipales como lo son: calles, escuelas, servicios que presta el municipio, etc.</li> </ul>

Otra forma de clasificar los factores para el aumento o decremento del valor de la valuación es como sigue:

Físicos	<ul style="list-style-type: none"> <li>a) Área (dentro del lote).</li> <li>b) Escasez de área.</li> <li>c) Exceso de área.</li> <li>d) Área utilizable.</li> </ul>
Infraestructura urbana	<ul style="list-style-type: none"> <li>a) Agua potable.</li> <li>b) Drenaje.</li> <li>c) Pavimentos (concretos, asfalto, etc.).</li> <li>d) Energía eléctrica.</li> <li>e) Cableados subterráneos.</li> <li>f) Redes telefónicas.</li> <li>g) T.V. por cable.</li> <li>h) Vialidades:</li> <li>i) Recolección de basura.</li> </ul>
	<ul style="list-style-type: none"> <li>a) Comercios y servicios.</li> <li>b) Educación.</li> <li>c) Asistencia.</li> </ul>

Equipamiento urbano	<ul style="list-style-type: none"> <li>d) Diversiones.</li> <li>e) Recreación.</li> <li>f) Trabajo.</li> <li>g) Parques públicos.</li> <li>h) Iglesias.</li> <li>i) Correos, telégrafos y teléfonos.</li> </ul>
Entorno urbano	<ul style="list-style-type: none"> <li>a) Características de las viviendas colindantes (escuelas, cantinas, edificios policiaicos, cementerios, bomberos, tugurios, etc.).</li> <li>b) Densidad de construcción.</li> <li>c) Densidad de población.</li> </ul>
Uso de suelo	<ul style="list-style-type: none"> <li>a) Habitacional.</li> <li>b) Comercial.</li> <li>c) Industrial.</li> <li>d) Campestre.</li> <li>e) Población rural.</li> <li>f) Agrícola.</li> <li>g) Reserva ecológica.</li> <li>h) Áreas verdes y espacios abiertos.</li> <li>i) Zonas restringidas.</li> </ul>
Régimen de propiedad	<ul style="list-style-type: none"> <li>a) Privado.</li> <li>b) Copropiedad.</li> <li>c) Condominio.</li> <li>d) Ejidal.</li> <li>e) Comunal.</li> <li>f) Comodato.</li> <li>g) Federales, estatales y municipios.</li> </ul>
Forma y topografía	<ul style="list-style-type: none"> <li>a) Frente y fondo.</li> <li>b) Dimensiones.</li> <li>c) Elevaciones.</li> <li>d) Depresiones.</li> </ul>

	<ul style="list-style-type: none"> <li>e) Irregularidades.</li> <li>f) Subsuelo.</li> <li>g) Panorama.</li> <li>h) Análisis geotécnico.</li> </ul>
<b>Ubicación</b>	<ul style="list-style-type: none"> <li>a) Esquina.</li> <li>b) Centro de manzana.</li> <li>c) Cabecera de manzana.</li> <li>d) Dos o más frentes.</li> <li>e) Predios interiores.</li> <li>f) Predios con plantas superpuestas.</li> <li>g) Orientación.</li> <li>h) Topografía de la colonia.</li> </ul>
<b>Contaminación</b>	<ul style="list-style-type: none"> <li>a) Mantos acuíferos.</li> <li>b) Ruido.</li> <li>c) Smog (contaminación por humo atmosférico).</li> <li>d) Basura.</li> <li>e) Visual.</li> <li>f) Olores.</li> </ul>
<b>Riesgos</b>	<ul style="list-style-type: none"> <li>a) Inundaciones.</li> <li>b) Desgajes de terreno.</li> <li>c) Vientos.</li> <li>d) Zonas sísmicas.</li> <li>e) Zonas minadas.</li> <li>f) Pandillerismo.</li> <li>g) Niebla.</li> <li>h) Vibraciones (motores, industrias).</li> <li>i) Clima.</li> </ul>
<b>Vialidad</b>	<ul style="list-style-type: none"> <li>a) De aproximación a vías rápidas.</li> <li>b) Sobre vías rápidas.</li> <li>c) Calles empedradas.</li> <li>d) Calles sin pavimento.</li> </ul>



	<ul style="list-style-type: none"> <li>e) Vías alternas.</li> <li>f) Amplitud de calles y banquetas.</li> <li>g) Calles vigiladas.</li> </ul>
Transporte	<ul style="list-style-type: none"> <li>a) Automóvil.</li> <li>b) Metro.</li> <li>c) Autobús.</li> <li>d) Trolebús.</li> <li>e) Microbús.</li> <li>f) Colectivos.</li> <li>g) Taxis.</li> </ul>
Tipo de zona	<ul style="list-style-type: none"> <li>a) Residencial (1ª 2ª y 3ª).</li> <li>b) Histórica.</li> <li>c) Típica.</li> <li>d) Antigua.</li> <li>e) Contemporánea.</li> <li>f) Suburbana.</li> <li>g) Campestre.</li> <li>h) Comercial.</li> <li>i) Industrial.</li> <li>j) Agrícola.</li> <li>k) Reserva ecológica.</li> </ul>
Limitantes	<ul style="list-style-type: none"> <li>a) Servidumbres de paso.</li> <li>b) Afectaciones.</li> <li>c) Restricciones (derecho de vía).</li> </ul>
Características de la oferta	<ul style="list-style-type: none"> <li>a) En el mercado local.</li> <li>b) En la zona.</li> <li>c) En la calle.</li> </ul>
Condiciones de la demanda	<ul style="list-style-type: none"> <li>a) Ingresos altos.</li> <li>b) Ingresos medios.</li> <li>c) Ingresos bajos.</li> <li>d) Interés social.</li> </ul>

Condiciones del mercado crediticio	<ul style="list-style-type: none"> <li>a) Favorable.</li> <li>b) Escaso.</li> <li>c) Nulo.</li> </ul>
Factores sociales	<ul style="list-style-type: none"> <li>a) Homogeneidad de las características sociales y económicas.</li> <li>b) Prestigio social.</li> <li>c) Aptitudes hacia el orden y la ley.</li> <li>d) Dimensiones de las familias.</li> <li>e) Edades de los grupos.</li> </ul>
Factores económicos	<ul style="list-style-type: none"> <li>a) Grado de prosperidad.</li> <li>b) Posesión de propiedad.</li> <li>c) Niveles de ingresos y rentas.</li> <li>d) Calidad de nuevas construcciones.</li> <li>e) Terrenos vacantes.</li> <li>f) Crecimiento de vecindario.</li> <li>g) Uso cambiante.</li> </ul>
Factores políticos	<ul style="list-style-type: none"> <li>a) Impuestos especiales.</li> <li>b) Carga de impuestos prediales y servicios.</li> <li>c) Códigos de zonificación y construcción.</li> <li>d) Construcción impositiva de edificios gubernamentales.</li> </ul>

### 1.4.3 Método del costo

Este método de valuación basa toda su estructura en el costo de reposición menos demérito y en algunos casos en el costo de reproducción menos demérito.

Este método adquiere toda su vigencia en casos tales como valuación de inmuebles de uso especializado en la valuación de monumentos y edificios históricos, en inmuebles de servicios públicos o privados, en mejoras diversas no rentables (plazas, parques, fuentes, etc.) y en general, en todos aquellos que sustentan su valor a hacerlos, sin

importar, al menos en forma fundamental, el producto que por concepto de uso o rentabilidad pudiesen obtener.

Este método, si lo analizamos, tiene un fundamento matemático-aritmético, en cuanto a la obtención del costo de reproducción, pero no tiene toda su vigencia (salvo en casos muy contados) pues los ajustes que deban hacerse por conceptos tales como: obsolescencia, deméritos por edad, uso, estado de conservación, etc., también propios del inmueble en sí, escapan a la manía de la matematización aislada, es decir, interviene en la aplicación de factores valorizantes o desvalorizantes, el criterio del valuador, como también intervendrá en el análisis al proyecto, adecuación de materiales, orientación y aprovechamiento de ésta, calidad de servicios, de vecindarios, etc., y en fin todo aquello que forma realmente el enfoque de un mercado inmobiliario, siempre cambiante, nunca estable.

Este método de valuación, se practica siguiendo la mecánica de un presupuesto, lo que lo hace laborioso, pues para ser exacto requiere de un estudio exhaustivo y acucioso de todas las partidas que intervendrían en la obra como *nueva*, amén de que hace suponer ciertos elementos que no son necesariamente visibles (cimentación, tuberías, obra de infraestructura en general) y que el valuador debe considerar como los más idóneos, de acuerdo a sus conocimientos en construcción. En este sentido es obvio que el valuador estima los elementos indicados, según el uso y el momento de la construcción, por lo que el técnico no sólo deberá reunir conocimientos en construcción sino datos fundados en la historia de la construcción y del sitio en que labora, pues materiales y procedimientos constructivos de moda en algún sitio, resultan obsoletos ya en otra. El detenido análisis de los materiales visibles de la construcción, indican al valuador, con una buena inspección, la calidad de éstos y por consiguiente del resto del inmueble, calificación de la mano de obra, la supervisión de la construcción, etc.

Este método, descrito en forma general, es sumamente preciso, pero muy laborioso. Tradicionalmente, no se estima el presupuesto exacto por partidas, sino que se considera un costo unitario por  $m^2$  de construcción, llegando el valuador cuando mucho

a dividir el predio en diversos tipos de construcción, de acuerdo con las características generales de cada uno, tomando como común denominador algún elemento constructivo dominante, que puede ser:

- a) Techumbre.
- b) Apoyos.
- c) Pisos.
- d) Instalaciones sanitarias.
- e) Acabados especiales.

o la incidencia de estos, etc.

Pero esto es a la vez impreciso, ya que para realizarse juntos, en el análisis, se deberá devaluar siguiendo la forma expresa de un presupuesto, en el que se cuantifica exactamente cada uno de los elementos que conforman la construcción en su totalidad, excavación y trazo, limpio de terreno, cimentación, estructura albañal, hasta llegar a detalles menores.

Obtenido de esta manera el costo de reposición total del inmueble, debe procederse a hacer un nuevo presupuesto, tendiente a cuantificar el monto necesario para poner en condiciones de *nuevo*, el inmueble estudiado.

El costo así obtenido se descontará del resultado anterior, llegando por este método al valor de reposición, menos demérito.

Pero esto, contempla exclusivamente el aspecto físico del bien valuado; a este valor deberán hacerse ciertos ajustes y quizás convenga seguir el método de puntuación o porcentual relativos a: deseabilidad, adecuación, calidad del proyecto, adaptabilidad o versatilidad, etc.

El método del costo de reposición requiere de la definición de lo que es el costo. Muchos términos son usados con referencia al costo, como costo actual, costo

histórico, costo original y costo de reposición, por lo que se explicará brevemente cada uno de éstos a continuación.

**Costo actual:** se considera como costo actual el monto que actualmente se paga por la adquisición de una propiedad o de un desarrollo. De esta manera una propiedad puede tener diferentes costos actuales bajo diferentes condiciones.

**Costo original:** Este término se refiere al costo original que dicha construcción tuvo, haciendo abstracción de los precios futuros que pudiera haber tenido dicha propiedad. Aunque se presenta el problema en la práctica contable, ya que la contabilidad tiende a identificar como costo original, el costo de adquisición para el propietario último.

**Costo histórico:** Este término es usado también para designar el costo original, pero también para considerar algunos costos que se hubieran hecho previamente a la edificación de tal propiedad, o bien a la adquisición original de la tierra.

Debido a lo anterior, es necesario usar un término diferente o bien definir claramente lo que para propósitos de valuación será el costo: " La estimación del costo de reposición, menos la depreciación, esto es el valor neto de reposición"<sup>3</sup>.

Es conveniente hacer notar que para hacer uso del método del costo es necesario partir del principio de la sustitución; esto es, que si la propiedad no pudiese ser substituida por una similar, el uso de este método se haría imposible. Para utilizar este método, lo más común es considerar los precios de mercado de construcción y, a la suma que resulte de estimar la realización de una construcción similar, aplicarle un castigo por la depreciación que ha acumulado el inmueble sujeto de la valuación.

Cabe hacer notar que el método del costo no presenta situaciones que afectan el valor de un inmueble, como son la obsolescencia y la falta de funcionalidad que la propiedad pudiera tener. En el método de los costos generalmente el valuador trabaja sobre

---

<sup>3</sup> Francisco Solares Alemán. Apuntes de edificación. Tomo I. Avalúos. UNAM FI 1984. Pág 126.

unidades similares o comparables, esto es sobre metros cuadrados de su superficie o metros cúbicos de espacio, con el objeto de trabajar con datos homogéneos.

Hay algunos factores que causan variaciones en los costos por metro cuadrado para diferentes tipos de construcción como son los siguientes:

- a) Dimensiones del terreno: Las dimensiones del terreno influyen en el costo, ya que generalmente a mayor superficie menor precio por metro cuadrado y a la inversa, a menor superficie mayor precio por metro cuadrado (precios de mercado).
- b) Una casa dúplex cuesta menos por metro cuadrado que una casa sola, debido a que las áreas de terreno se reparten entre dos unidades.
- c) Debido a que las instalaciones sanitarias representan uno de los más altos costos de la construcción, el costo de construcción, por metro cuadrado, de una casa que tuviese seis recámaras con un baño y una cocina, será menor que el costo por metro cuadrado de una casa, que teniendo el mismo baño y cocina tuviera una o dos recámaras.
- d) El costo varía según el material de que esté hecha la construcción.
- e) El costo varía con la calidad de los acabados y
- f) El costo de edificios comerciales varía según los "extras" que tenga incluidos, como elevadores, aire acondicionado, marquesinas, etc.

Por último se señala que en el método de costo, la valuación de la tierra se realiza en base a la comparación con valores en la zona; esto es, por el método del mercado.

## **1.5 Visual Basic 4.0**

La tendencia dominante de hoy en día es que las herramientas de desarrollo de aplicaciones generadas mediante lo que se denomina programación visual (entendida como el uso de expresiones tales como gráficas, dibujos, iconos, barra de menús, etc.), en el proceso de la programación de aplicaciones. Aunque existe una gran variedad de

lenguajes visuales uno de los que han destacado sobre todo para el desarrollo de aplicaciones "front-end" es el Microsoft Visual Basic.

Microsoft liberó el Visual Basic 1.0 en 1987 y fue el primer lenguaje visual desarrollado por esta compañía. Visual Basic fue un éxito y poco después salió al mercado la versión 2.0 (1991), sin embargo el verdadero potencial de esta herramienta fue vista por los usuarios hasta la versión 3.0. Algunas de las características de esta herramienta se listan a continuación.

Microsoft Visual Basic — Es la forma más fácil y rápida de crear aplicaciones poderosas para sistemas operativos Microsoft Windows. La programación en Visual Basic permite crear aplicaciones útiles y robustas que utilizan la **graphical user interface** (interface gráfica de usuario (GUI)).

Visual Basic nos facilita la producción de sistemas proporcionando las herramientas apropiadas para crear los diferentes aspectos de desarrollo de GUI. Se pueden crear interfaces gráficas de usuario para aplicaciones, dibujando objetos en forma gráfica. Se deben asignar ciertas propiedades sobre estos objetos para definir su apariencia y conducta. Para que la interface gráfica que se crea funcione es necesario escribir el código correspondiente a los eventos que ocurren en la interface.

Utilizando Visual Basic, se pueden crear aplicaciones poderosas y llenas de características.

- El Acceso a Bases de Datos permite crear bases de datos y aplicaciones front-end para los formatos de bases de datos más populares.
- OLE permite usar la funcionalidad provista por otras aplicaciones tales como Microsoft Word para Windows procesador de palabras, Microsoft Excel hoja de cálculo, y Microsoft Project sistema de planeación de proyectos de negocios.
- Al finalizar la aplicación se crea un archivo ejecutable .EXE que utiliza una librería dinámica (DLL).

## **Requerimientos del Sistema y del Hardware**

Para correr Visual Basic Professional o Enterprise Edition, se debe tener cierto hardware y software instalado en la computadora. Los requerimientos del sistema son:

- Máquina IBM-compatible con procesador 80386 o superior.
- Disco duro con 50 megabytes mínimo de espacio disponible para una instalación completa.
- Drive de 5.25 o 3.5, o un CD-ROM.
- Ocho megabytes de memoria (se requieren para Windows 95).
- Un mouse u otro dispositivo señalador.
- Sistema Operativo Windows versión 3.1 o superior.
- Para versiones Visual Basic de 32-bits, se debe contar con Windows 95 o superior, o Windows NT 3.51 o superior.
- Para versiones Visual Basic de 16-bits, se debe contar con Windows versión 3.1 o superior.

## **Instalando Visual Basic**

Cuando se corre el programa de instalación llamado Setup, se crea un directorio para Visual Basic; y se pueden instalar sólo los archivos de Visual Basic que se deseen.

La edición estándar sólo está disponible en versión de 32-bits. Con las otras ediciones, se pueden instalar versiones de 16-bits o 32-bits de Visual Basic. Las versiones de 16-bits y de 32-bits pueden instalarse en directorios separados o ambas en el mismo directorio.

## **Ediciones de Visual Basic**

- La edición Estándar permite a los programadores crear fácilmente poderosas aplicaciones para Microsoft Windows 95 y Windows NT. La edición Estándar está



disponible solamente en versiones de 32-bits. Esta incluye todos los controles, el grid, controles de liga, y el common dialog control.

- La edición Profesional da a los programadores profesionales un conjunto de herramientas para un fácil desarrollo de aplicaciones en ambiente Windows. Incluye todas las características de la edición Estándar, los custom controls y el Crystal Report Writer.
- La edición Enterprise permite a los profesionales crear aplicaciones robustas en un ambiente de empresa Cliente/Servidor. Incluye todas las características de la Edición Profesional, Automation Manager, Component Manager, herramientas para manejo de Base de Datos, el Microsoft Visual SourceSafe (Software generador de versiones) orientado a sistemas de control, y más.

### Ayuda de Microsoft Visual Basic

Una de las herramientas de ayuda que tiene, son las lecciones para aprender Visual Basic y para comenzar las lecciones, hay que escoger de la Barra de Menús la opción Help y seleccionar "Aprendiendo Microsoft Visual Basic" (fig. 1.5.1).

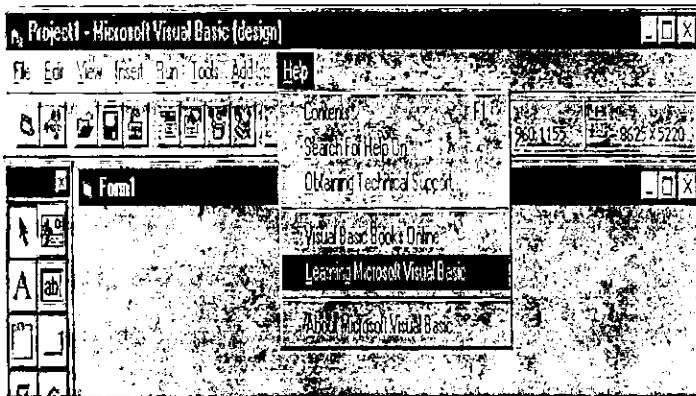


Figura 1.5.1 El menú Ayuda.

Las lecciones de "Aprendiendo *Microsoft Visual Basic*" son descritas brevemente en la siguiente tabla(Figura 1.5.2).

Lección	Contenido
"Como trabaja Visual Basic"	Introduce el ambiente de desarrollo de Visual Basic.
"Creando una Aplicación"	Explica el proceso de crear una aplicación simple en Visual Basic.
"Escribiendo un Programa de los Eventos"	Introduce a la programación y manejo de eventos, y lo compara con la programación tradicional.
"Trabajando con Formas y Controles"	Muestra cómo construir una forma, agregar controles y asignar propiedades.
"Agregando Menús"	Explica cómo agregar menús a las formas.
"Rastreado tu Aplicación"	Describe cómo escribir el código básico en Visual Basic y utiliza herramientas de rastreo.
"Accesando a Bases de Datos"	Introduce los data control y data-bound controls.
"Introducción a Objetos OLE Automation"	Explica cómo usar objetos de otras aplicaciones utilizando OLE Automation.
"Usando Color y Gráficos"	Demuestra ideas de diseño así como el uso de gráficos, color y formatos de salida.

Figura 1.5.2 Descripción de las opciones del tutorial de Visual Basic.

## Ayuda en Línea

Se puede acceder a esta Ayuda eligiendo la opción Contenido del menú de Ayuda, buscando por tópicos específicos con **Search tool** (la herramienta de Búsqueda), o

bien presionando F1 para conseguir ayuda de contexto sensitivo (hipertexto) sobre el ambiente de programación de Visual Basic.

### Contenido dentro del menú Ayuda

Para información sobre temas de Ayuda, se debe elegir la opción "Contenido" dentro del menú Ayuda o presionando F1 y seleccione la opción de Contenido con el botón del mouse. Puede usarse la pantalla de Contenido de la Ayuda (figura 1.5.3) para saltar a temas que indiquen como usar Visual Basic u obtener un rápido acceso a información de referencia llave.

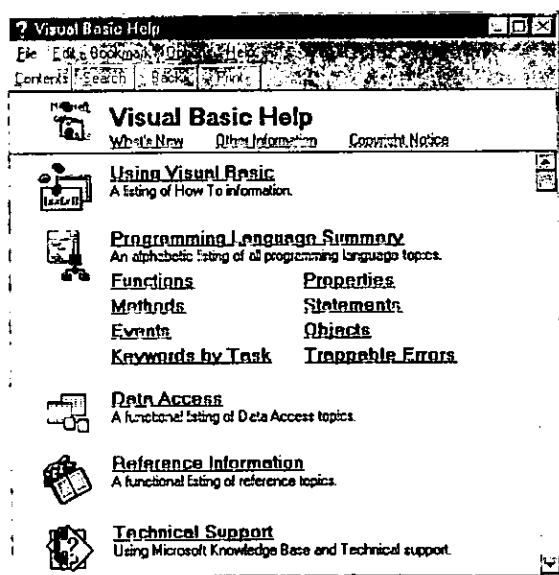


Figura 1.5.3 Pantalla del contenido de la Ayuda.

### Búsqueda dentro del menú ayuda (Help Search)

La forma más fácil de encontrar un tema en particular es utilizar **Search dialog box** (la caja de diálogo de Búsqueda) (figura 1.5.4). Para desplegar la Caja de Diálogo de

Búsqueda, puede elegirse Search (Búsqueda del menú de Ayuda) o bien seleccionar cualquier tema que se muestra en pantalla.

### Para buscar Ayuda

1. Del menú de Ayuda, elija **Search for Help On** (Búsqueda de Ayuda) o bien oprima el botón de Búsqueda desde cualquier tema de Ayuda.
2. En la Caja de Diálogo, escriba una palabra, o seleccione una de la lista que se muestra. Presione ENTER o elija el botón Muestra Temas para desplegar una lista de títulos relacionados con la palabra que se especificó.
3. Selecciona el nombre de un tópico, y presiona ENTER o elige **Go To** (Ir a ) para ver el tópico.

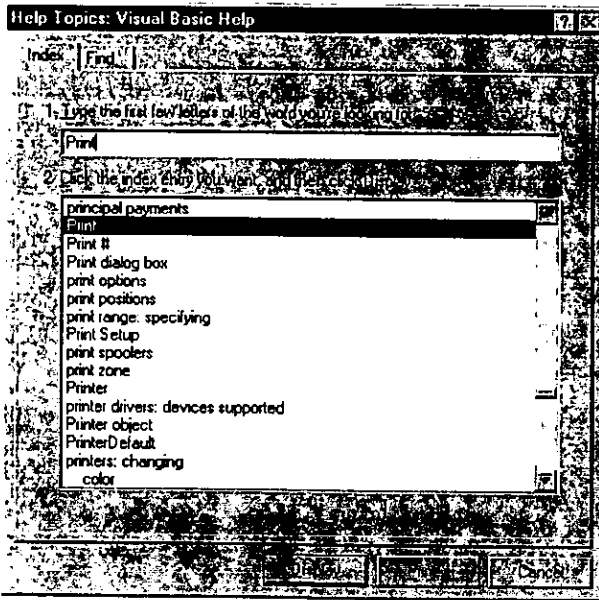


Figura 1.5.4 Caja de diálogo de Búsqueda.

## **Ayuda de Contexto-Sensitivo (F1)**

Muchas partes de Visual Basic están en contexto-sensitivo. *Contexto-sensitivo* significa que se puede obtener ayuda directamente sobre estas partes sin tener que ir al menú de Ayuda. Por ejemplo, para obtener Ayuda sobre cualquier palabra reservada dentro del lenguaje de Visual Basic, coloque el puntero del mouse sobre la palabra y presione F1.

Se puede presionar F1 desde cualquier parte de Visual Basic con contexto-sensitivo para desplegar información acerca de ese elemento. Las partes de contexto-sensitivo son:

- Cada ventana en Visual Basic (ventana de Propiedades, ventana de Código, y así sucesivamente).
- Controles en la barra de Herramientas.
- Objetos sobre una forma.
- Propiedades dentro de la ventana de Propiedades.
- Procedimientos de eventos en la ventana de Código.
- Palabras reservadas de Visual Basic (declaraciones, funciones, propiedades, métodos, eventos y objetos especiales).
- Mensajes de error.

Una vez que se haya abierto la Ayuda, puede presionarse F1 para obtener información acerca de cómo usar la propia ayuda.

## **Nuevas Características en Visual Basic 4.0**

Una de las nuevas características en la edición Estándar de Visual Basic 4.0 es la adición de un nuevo lenguaje de máquina — Visual Basic para aplicaciones. Esta versión de Visual Basic para aplicaciones es completamente compatible con versiones anteriores de Visual Basic, y con edición de aplicaciones Visual Basic, versión 1.0, la

cual se incluye en Microsoft Excel versión 5.0 y Microsoft Project versión 4.0. La adición de Visual Basic para aplicaciones hace más fácil el programa de objetos OLE Automation, esto significa que el código de Visual Basic puede ser más manejable entre módulos dentro de la aplicación que la soporta.

Visual Basic para aplicaciones también provee muchas mejoras o progresos al lenguaje y al ambiente de desarrollo. Muchas de estas mejoras aparecen en la siguiente tabla (Figura 1.5.5)

<b>Nueva Característica</b>	<b>Afecta en</b>
OLE custom controls (Controles OLE)	" <u>Creando y usando Controles</u> "
Insertable objects as controls (Objetos Insertables como controles)	" <u>Creando y usando Controles</u> "
Development environment extensibility (Ambiente de Desarrollo Extenso)	" <u>Manejo de Proyectos</u> "
Conditional compilation (Compilación Condicional)	" <u>Manejo de Proyectos</u> "
Letras fijas y tamaños de letra.	" <u>Manejo Proyectos</u> "
Property procedures (Procedimientos de Propiedad)	" <u>Programación Fundamental</u> "
Line continuation character (Continuación de Caracteres en línea)	" <u>Programación Fundamental</u> "
For Each...Next	" <u>Programación Fundamental</u> "
With...End With	" <u>Introducción a Objetos</u> "
Collection object (Colección de Objetos)	" <u>Introducción a Objetos</u> "
Creating reusable objects (Creando objetos reutilizables)	" <u>Introducción a Objetos</u> "
Object Browser	" <u>Introducción a Objetos</u> "

**Continuación...**

Nueva Característica	Afecta en
Forms as objects(Formas y Objetos)	"Introducción a <u>Objetos</u> "
OLE Automation	"Programando Otras aplicaciones <u>Objetos</u> "
Menu and toolbar negotiation(Menú y negociación de la Barra de Herramientas)	"Menús"
Improved Debug window(Ventana de rastreo)	"Rastreando"
Error object(Objetos de Error)	"Errores en la corrida"
Data access objects (DAO)( Objetos en Acceso de Datos)	"Accesando a las Bases de Datos con el <u>Data Control</u> "
New data-bound controls(Nuevos Controles de liga)	"Usando los <u>Data-Bound Controls</u> "
Soporte a 32-bits	"Programando para sistemas de 16- <u>Bits</u> "
Resource files(Archivos Fuente)	"Tratados internacionales"
What's This Help(Cuál es la Ayuda)	<i>Busca la ayuda en línea para saber qué es esto ?</i>
Improved Setup Toolkit (Manual de Instalación)	"Distribuye tus aplicaciones"

Figura 1.5.5 Mejoras de la edición de Visual Basic para aplicaciones.

**Primera aplicación en Visual Basic**

Toma pocos minutos construir una aplicación en Visual Basic. Puede crearse una interface de usuario dibujando "controles", así como cajas de texto, botones de comando, todo sobre una forma. Después, hay que asignar propiedades a las formas y a los controles para especificar captura de datos, color, y tamaño. Finalmente, escribir código para darle vida a la aplicación. Los pasos básicos que se realizan en la creación

de una primera aplicación indicarán los principios que deberán seguirse para una aplicación posterior.

### Comenzando Visual Basic

Cuando se realiza la instalación de Visual Basic mediante el programa Setup, éste permite colocar el icono del programa en un grupo de programas existente o crea un nuevo grupo de programas con los iconos para Visual Basic en Windows. Entonces se estará listo para comenzar Visual Basic desde Windows (fig. 1.5.6).

### Para Comenzar Visual Basic desde Windows

- 1) Hacer "click" con el mouse en Inicio sobre la Barra de Tareas
- 2) Seleccionar Programas y después elegir Visual Basic 4.0.

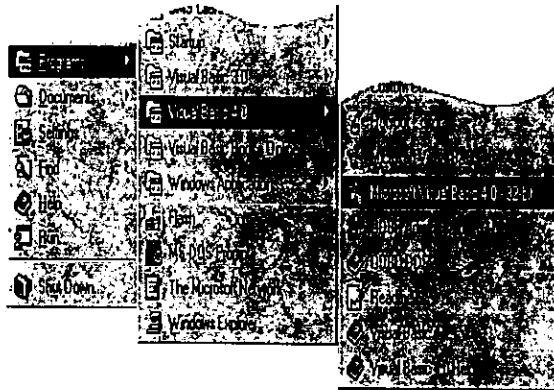


Figura 1.5.6 Menú inicio para abrir Visual Basic 4.0.

O también:

- 1) Hacer "click" en el botón Inicio sobre la Barra de Tareas.
- 2) Seleccionar Programas.
- 3) Usar el Explorador de Windows para encontrar el archivo ejecutable de Visual Basic (VB32.EXE).



4) Hacer doble-click sobre el icono de Visual Basic.

Cuando inicie Visual Basic, se verá la interface del ambiente de programación, tal como se muestra en la figura 1.5.7.

### Elementos de la Interface

La Interface de Visual Basic consta de los siguientes elementos:

**Toolbar (Barra de Herramientas)** Proporciona un acceso rápido a los comandos comúnmente usados en el ambiente de Programación (Ver figura 1.5.8). Una vez que se haya definido el uso y acción que tendrá ese comando habrá que elegirlo y acarrearlo a la forma.

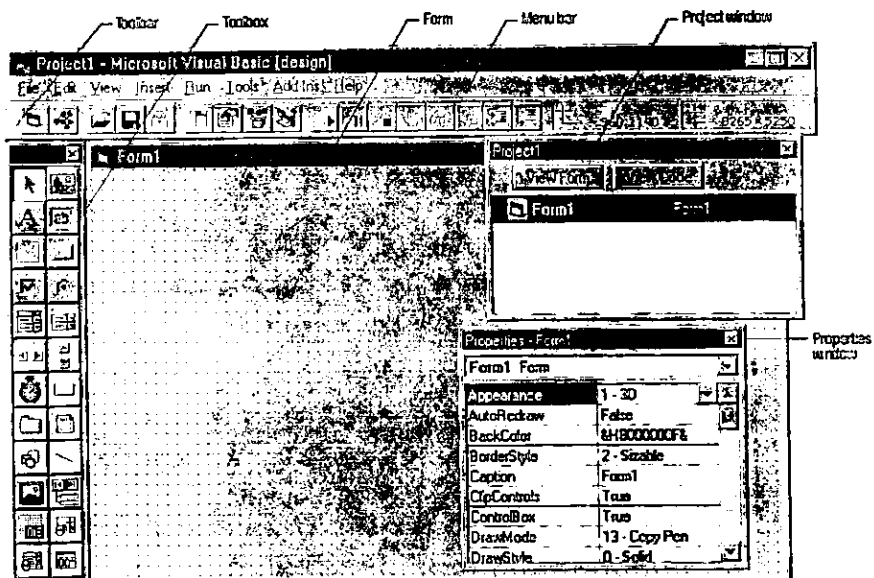












Figura 1.5.7 El ambiente de programación de Visual Basic.

Toolbar Botón	Acción	Menú equivalente
	Crea una nueva forma.	Comando Form (Forma) en el menú Insert (Insertar).
	Crea una nuevo módulo.	Comando Module (Módulo) en el menú Insert (Insertar).
	Abre un proyecto existente.	Comando Open Project (Abrir proyecto) en el menú File (Archivo).
	Salva el proyecto actual.	Comando Save Project (Salvar Proyecto) en el menú File (Archivo).
	Bloquea y desbloquea los controles en la forma activa.	Comando Lock Controls (Bloquea controles) en el menú Edit (Editar).
	Despliega el Editor de Menús.	Comando Menú Editor (Editor de Menús) en el menú Tools (Herramientas).
	Despliega la Ventana de Comando Properties.	Propiedades. (Propiedades) en el menú View (Ver).
	Despliega el Object Browser.	Comando Object Browser en el menú View (Ver).
	Despliega la ventana del Proyecto.	Comando Project (Proyecto) en el menú View (Ver).
	Comienza una aplicación en modo Diseño.	Comando Start (Inicio) en el menú Run (Ejecutar).

Continuación...








Toolbar Botón	Acción	Menú equivalente
	Usado para rastrear, parar la ejecución de un programa mientras está corriendo.	Comando Break en el menú Run (Ejecutar).
	Para la ejecución de una aplicación y retorna al modo diseño.	Comando End (Fin) en el menú Run (Ejecutar).
	Permite rastrear en una línea del programa.	Comando Toggles breakpoint en el menú Run (Ejecutar).
	Despliega el valor actual de una variable.	Comando Instant Watch en el menú Tools (Herramientas).
	Despliega la estructura de llamadas activas.	Comando Calls (llamadas) en el menú Tools (Herramientas).
	Rastrea cada línea de código dentro de procedimientos.	Comando Step Into en el menú Run (Ejecutar).
	Ejecuta el código de un procedimiento o declaración en la ventana de Código.	Comando Step Over en el menú Run (Ejecutar).

Figura 1.5.8 Barra de herramientas.

**Toolbox (Caja de Herramientas)** Provee un conjunto de herramientas que se utilizan en el modo de diseño de controles sobre una forma.

## Controles de Visual Basic

La caja de Herramientas de Visual Basic contiene las herramientas que se utilizan para dibujar controles sobre las formas. Cada herramienta en la caja de Herramientas representa un control (Figura 1.5.9).

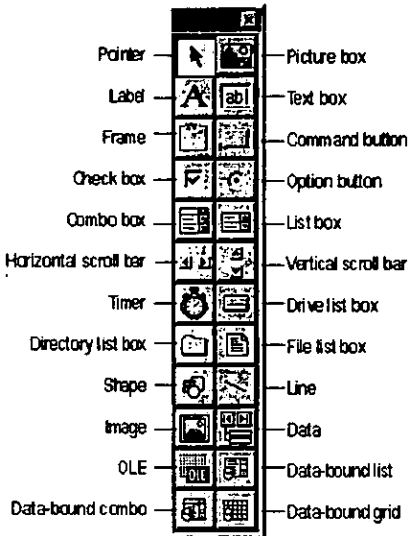














Figura 1.5.9 La caja de herramientas Visual Basic.












La siguiente tabla resume los controles básicos de la caja de Herramientas. Algunas de éstas herramientas son de versiones primitivas de Visual Basic.(Figura 1.5.10)

Botón	Acción y su descripción
	Apuntador: El apuntador (pointer) proporciona la selección de un control o alguna forma. (Nótese que esto no es un control).
	PictureBox: El control picture box (caja de imagen) despliega bitmaps, iconos o metafiles de Windows. Muestra texto o actúa como un contenedor visual de otros controles.

Continuación...

Botón	Acción y su descripción
	Label: El control label (etiqueta) despliega un texto que el usuario no puede modificar.
	TextBox: El control text box (caja de texto) provee un área para capturar o desplegar texto.
	Frame: El control Frame (marco) provee un contenedor visual y funcional para controles.
	Command Button: El control Command Button (botón de comando) ejecuta un comando o acción cuando el usuario acciona el botón.
	CheckBox: El control check box (caja de verificación) despliega la opción Verdadero/Falso o Si/No. Puede utilizarse cualquier número de cajas de verificación sobre una forma al mismo tiempo.
	RadioButton: El control option button (botón de opción), forma parte de un grupo de opciones junto con otros botones, despliega múltiples opciones, de las cuales el usuario puede elegir sólo una.
	ComboBox: El control combo box (caja de selección) combina una caja de texto con una caja de lista. Permite al usuario escribir o seleccionar una opción de la lista mostrada.
	ListBox: El control list box (Lista) despliega una lista de opciones que el usuario puede elegir.
	HScrollBar: El control de Barra Horizontal (horizontal scroll bar) y Barra Vertical (vertical scroll bar) permite al usuario seleccionar un valor dentro de un rango de valores. Se utilizan como controles separados y no son los mismos que aparecen junto con otros controles.
	VScrollBar: (Se describe junto con el anterior).

**Continuación...**

Botón	Acción y su descripción
	Timer: El control timer ejecuta eventos de tiempo en intervalos específicos.
	DriveListBox: El control drive list box (unidades de disco) despliega y permite al usuario seleccionar drives o unidades de disco válidas.
	DirListBox: El control directory list box (directorio) despliega y permite al usuario seleccionar directorios y rutas.
	FileListBox: El control file list box (archivo) despliega y permite al usuario seleccionar desde una lista de archivos.
	Shape: El control shape (figura) agrega un rectángulo, cuadrado, elipse o círculo a una forma.
	Line: El control line (línea) agrega un segmento de línea recta a una forma.
	Image: El control image (imagen) despliega bitmaps, iconos, o metafiles de Windows; actúa como un botón de comando cuando se selecciona.
	Data: El data control (control de datos) habilita la conexión hacia una base de datos existente y despliega información de ésta en la forma.
	OLE: El control OLE (OLE container) liga los datos dentro de una aplicación Visual Basic.
	DBCombo: El control data-bound combo box provee muchas de las características de un control combo box estándar, incrementando el acceso a bases de datos.
	DBGrid: El control data-bound grid provee muchas de las características del control grid estándar, se incrementó el acceso a las bases de datos.

### Continuación...



Botón	Acción y su descripción
	DBList: El control data-bound list box provee muchas de las características de una lista estándar, se incrementó su acceso a bases de datos.
	Menú: El control menú crea una barra de menús dentro de la aplicación Visual Basic.

Figura 1.5.10 Controles básicos de Visual Basic.

### Categorías de los Controles

Hay tres grandes categorías de controles en Visual Basic:

- *Standard controls (Controles estándar)*, tales como los controles botón de comando y frame. Estos controles están dentro del archivo ejecutable .EXE de Visual Basic. Los controles estándar están incluidos siempre en la caja de Herramientas, no así los *custom controls y objetos insertables*, los cuales pueden quitarse o agregarse a la caja de Herramientas.
- *Custom controls*, existen en archivos separados con extensión .VBX u .OCX. Estos incluyen controles especiales, como el *common dialog control y custom controls* en tercera dimensión, así también realza las versiones de controles estándar, como el control data-bound combo box.

Los Custom controls son archivos con extensión .OCX toman ventaja de la nueva tecnología OLE. Los Custom controls con extensión .VBX usan la tecnología vieja y se encuentran en aplicaciones primitivas de Visual Basic.

**Nota:** Se puede seguir utilizando custom controls con extensión .VBX en versiones de Visual Basic a 16 bits. Sin embargo estos controles no pueden utilizarse en versiones de Visual Basic de 32 bits.

Pueden utilizarse controles .OCX en versiones de Visual Basic de 16-bits y 32-bits, sin embargo, es necesario usar controles .OCX de 16 bits en versiones de Visual Basic de 16 Bits y controles .OCX de 32 bits con versiones de Visual Basic de 32 bits.

Cuando Visual Basic abre un proyecto contiene un control .VBX, si la versión de un control .OCX está disponible reemplaza el control .VBX con un control .OCX.

□ *Insertable Objects (Objetos Insertables)*, tales como una hoja de cálculo Microsoft Excel conteniendo una lista de empleados, o bien un horario de trabajo en Microsoft Project. Desde que estos pueden agregarse a la caja de Herramientas, se consideran custom controls. Algunos de estos objetos también soportan OLE Automation, lo cual permite programar otros objetos de la aplicación dentro de la aplicación en Visual Basic.

**Menu Bar** (Barra de Menús) Despliega los comandos que se usarán para construir alguna aplicación.

**Form** (Forma) Funciona como una ventana, la cual se puede personalizar como una interface para la aplicación. Pueden sumarse a esta forma controles, gráficos e imágenes para darle realce a la forma que se desea.

**Project Window** (Ventana del Proyecto) Lista las formas y módulos en el proyecto actual, es decir, el proyecto con el que se está trabajando. Un proyecto es la colección de archivos que se utilizan para construir una aplicación (fig. 1.5.11).



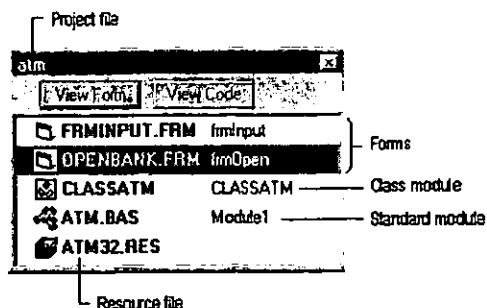


Figura 1.5.11 La ventana del Proyecto.

**Properties Window** (Ventana de Propiedades) Lista el conjunto de propiedades para la forma o el control seleccionado. Una propiedad es una característica de un objeto, tal como tamaño, etiqueta, color, etc.

**Object Browser** (Buscador de objetos) Se encuentra en el menú View (Ver), el Object Browser lista los objetos disponibles para usarse en el proyecto y proporcionar una forma más rápida de navegar dentro del código. El Object Browser explora los objetos en Visual Basic y otras aplicaciones, detecta que métodos y propiedades están disponibles para estos objetos y pega procedimientos de código dentro de la aplicación.

**Menús de Contexto** El ambiente de desarrollo de Visual Basic provee menús de contexto que contienen combinaciones de teclas para ejecutar rápidamente las acciones más frecuentes. Para abrir un menú de contexto, oprima el botón derecho del mouse sobre el objeto que se está utilizando. La lista de teclas rápidas disponibles del menú de Contexto dependerá de la parte que se haya elegido.

### Pasos para crear una aplicación Visual Basic

Existen tres pasos principales para crear una aplicación para Windows en Visual Basic:

1. Crear la interface.
2. Asignar propiedades.
3. Escribir el código.

## Creando la Interface

Las Formas son la base para crear la interface de una aplicación. Las formas pueden usarse para agregar ventanas y cajas de diálogo para la aplicación. Se pueden usar también como contenedores para iconos que no sean visibles en la aplicación de la Interface. Por ejemplo, puede tenerse una forma que sirva como un contenedor para gráficos que se deseen desplegar en otras formas.

El primer paso para construir una aplicación en Visual Basic es crear las formas que serán las bases para crear la interface de la aplicación. Después se dibujarán los objetos que conformaran la interface sobre las formas creadas.

### Para dibujar un control usando la Caja de Herramientas (Toolbox)

1. Seleccionar de la Caja de Herramientas el control que se desee dibujar.
2. Mover el apuntador dentro de la forma. El apuntador será una cruz, tal como se muestra en la figura 1.5.12.

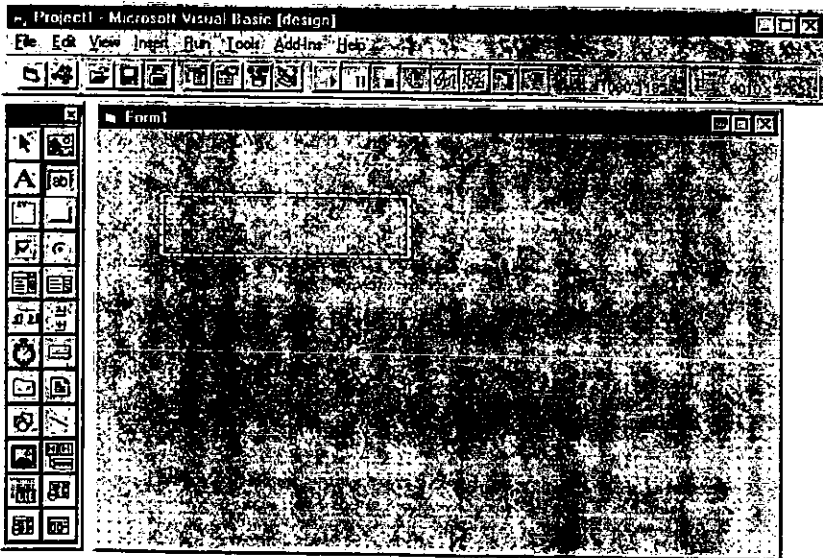


Figura 1.5.12 Dibujando una caja de texto con la caja de Herramientas (Toolbox).

3. Colocar el cursor en forma de cruz donde se desee poner el control.
4. Arrastrar el cursor hasta que el tamaño del control sea el que se desee.
5. Soltar el botón del mouse.

El control aparecerá sobre la forma.

Otra manera de agregar un control a la forma es hacer doble click al control que se desea dibujar. Esto crea un tamaño por omisión del control y lo coloca en el centro de la forma.

### Asignación de Propiedades

El siguiente paso consiste en asignar propiedades a los objetos que se han creado. La ventana de Propiedades (Properties window) (figura 1.5.13) provee una forma fácil de colocar las propiedades para todos los objetos sobre una forma. Para abrir la ventana de Propiedades, hay que elegir el comando Propiedades del menú Ver, hacer click al botón de propiedades sobre la barra de herramientas, o usar el menú de contexto para el control.

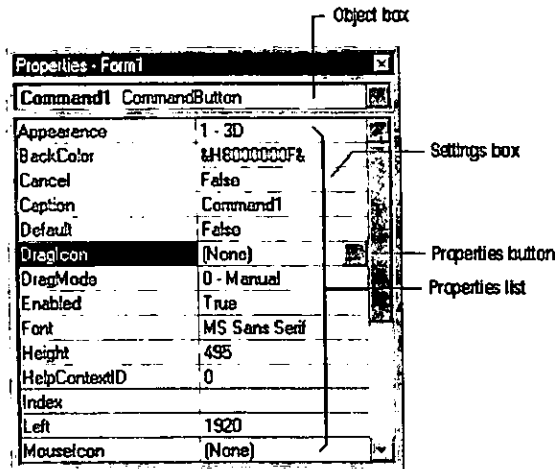


Figura 1.5.13 La ventana de Propiedades.

La Ventana de Propiedades consta de los siguientes elementos:

- **Object Box** (Caja de Objetos). — Despliega el nombre del objeto del cual se puede asignar sus propiedades. Hacer click a la flecha que está a la derecha de la Caja de Objetos para desplegar la lista de objetos de la forma.
- **Properties List** (Lista de Propiedades). — La columna izquierda despliega todas las propiedades para el objeto seleccionado. Se puede editar y asignar valores en la columna derecha.

### **Colocar propiedades desde la Ventana de Propiedades**

1. Desde el menú View (Ver), elegir Properties (Propiedades) o hacer click en el botón de Propiedades sobre la caja de Herramientas.  
La ventana de Propiedades despliega las asignaciones para el control o la forma seleccionados.
2. Desde la lista de Propiedades, seleccionar el nombre de una de ellas.
3. En la columna derecha, teclear o seleccionar el nuevo valor que tendrá esa propiedad.

### **Escribiendo el Código**

La ventana de Código es el lugar en donde se escribe el código en Visual Basic para la aplicación. El código consta de declaraciones, constantes, sentencias. Usando la ventana de Código se puede ver y editar cualquier código de la aplicación.

### **Para abrir la Ventana de Código**

Hacer doble-click sobre la forma o el control donde se desea escribir el código o desde la ventana de Proyecto, seleccionar el nombre de una forma o un módulo y elegir el botón de Código (View Code).

La figura 1.5.14 muestra la ventana de Código que aparece cuando se da doble click sobre un botón de Comando, y los eventos para ese comando.

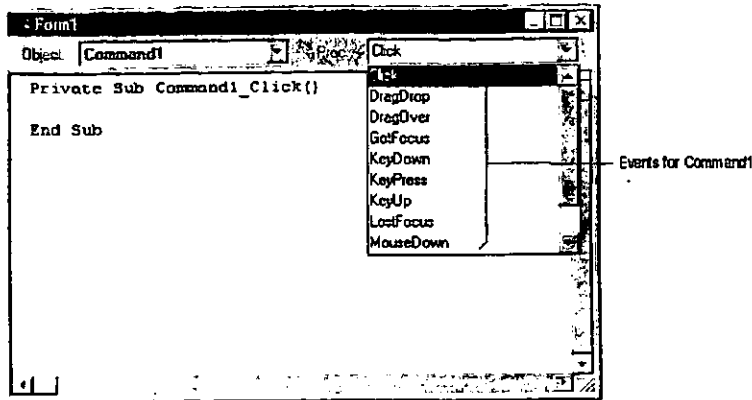


Figura 1.5.14 La ventana de Código.

Pueden desplegarse todos los procedimientos en la misma ventana de Código o desplegar un solo procedimiento.

#### Para desplegar todos los procedimientos en la misma ventana de Código

1. Desde el menú *Tools* (Herramientas) selecciona la caja de diálogo de Opciones.
2. Sobre el *Editor* en la caja de Diálogo de Opciones, selecciona el **check box** (la caja de verificación) que está a la izquierda del módulo *Ver*. La caja de verificación (check box) a la izquierda de un procedimiento agrega o remueve una línea que realiza la separación entre procedimientos.

#### Para desplegar un procedimiento en la ventana de Código

1. Desde el menú *Herramientas*, seleccionar la caja de Diálogo de Opciones.
2. Sobre el *Editor* en la caja de diálogo de Opciones, limpiar la caja de verificación a la izquierda del módulo *Ver*.

La ventana de Código incluye los siguientes elementos:

- **Caja de Objetos** — Despliega el nombre de un objeto seleccionado. Hacer click en la flecha de la derecha para listar todos los objetos asociados con la forma
- **Caja de Listas de Procedimiento** — Lista los procedimientos de un objeto. La caja despliega el nombre de un procedimiento seleccionado - en este caso Click. Elegir la flecha que se muestra a la derecha para mostrar todos los procedimientos para ese objeto.

### **Creando procedimientos de Eventos**

El código en una aplicación Visual Basic se divide en pequeños bloques llamados procedimientos. Un procedimiento de un evento contiene código que se ejecuta cuando un evento ocurre (así como cuando el usuario le da el click a un botón). Un procedimiento de un evento para un control combina el nombre del control actual (especificado en la propiedad Nombre), un subguión (\_) y el nombre del evento. Por ejemplo, si se quiere un botón de comando llamado Command1 para invocar un evento cuando éste es *clicked* (presionado), usa el procedimiento Command1\_Click.

### **Corriendo la Aplicación**

Para correr la aplicación elegir Inicio desde el menú *Run* (Ejecutar), o hacer click en el botón de la Barra de Herramientas, o bien presionar F5.

### **Bases de Datos**

1. Visual Basic incorpora un acceso directo a bases de datos de los siguientes formatos; Microsoft Access, Dbase III, Dbase IV, Btrieve, Microsoft FoxPro, Oracle, Paradox y Microsoft SQL server, sin embargo hay que mencionar que el formato de Base de Datos nativa es Microsoft Access, dándole a este la mayor flexibilidad y velocidad en el acceso a los datos.

2. Hacen uso de ODBC (Open Database Connectivity) con esto permite la utilización de bases de Datos Externas, esta categoría incluye las Bases de Datos en el modelo Cliente/Servidor (fig. 1.5.15).
3. La arquitectura de Bases de Datos es extensible, es decir, se pueden utilizar Drivers (manejadores) para adicionar formatos de Bases de Datos de otros fabricantes y sin la necesidad de realizar cambios en el código original.
4. Incorpora el uso del lenguaje SQL (Structure Query Lenguaje), este lenguaje es utilizado para la realización de consultas, actualizaciones y el manejo de Bases de Datos Relacionales. Para el caso de las Bases de Datos Externas (ODBC) los comandos SQL son mandados al Servidor para su procesamiento.
5. El lenguaje SQL de Visual Basic cumple con las normas de ANSI (ANSI-86, ANSI-89 y ANSI-9).
6. Posee una poderosa herramienta para la generación de reportes denominada Crystal Reports. Esta herramienta permite:
  - \* Realizar las conexiones necesarias entre Bases de Datos.
  - \* Realizar cálculos de las distintas bases de datos incluyendo cálculos de subtotales, comparaciones, conversiones de datos.
  - \* Realizar el formato de nuestros reportes incluyendo fuentes, márgenes, tamaño de hoja, etc.
  - \* Fijar encabezados y pies de páginas, títulos de columnas, etc.

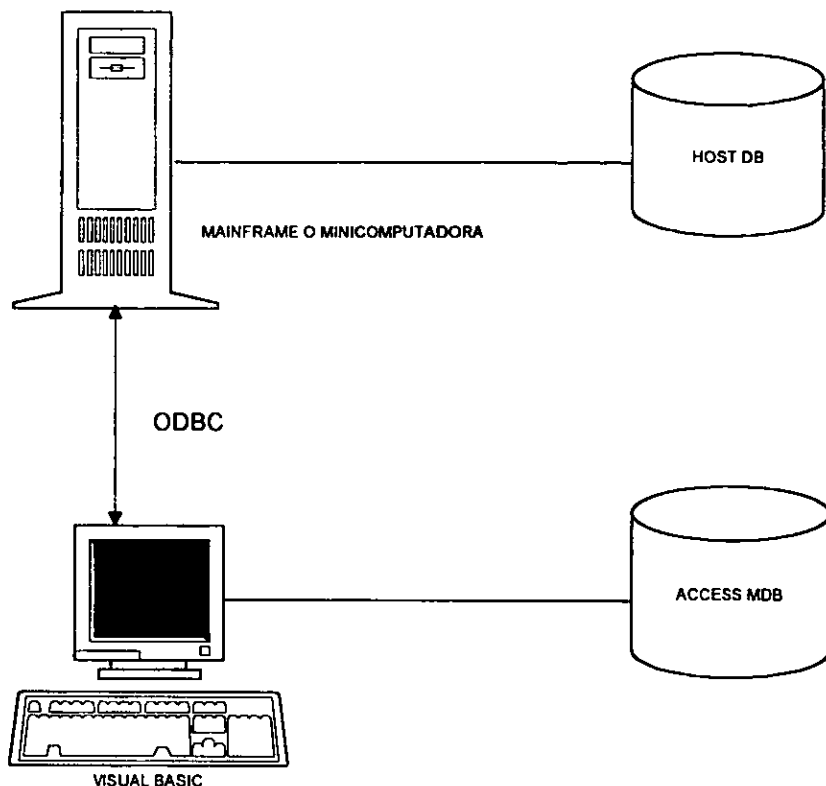


Figura 1.5.15 Acceso a diferentes Bases de datos con y sin ODBC.

### Realizar y Ejecutar un Archivo .EXE

Puede hacerse un archivo ejecutable desde Visual Basic eligiendo el comando Make File Exe desde el menú File.

### Para hacer un archivo ejecutable en Visual Basic

1. Desde el menú File, seleccionar Make EXE File.
2. Teclear un nombre de archivo o a través de los directorios seleccionar un nombre de archivo existente para reemplazar.



3. Eligiendo el botón de opciones, puede especificarse un número de versión específico, tal como se muestra en la figura 1.5.16.

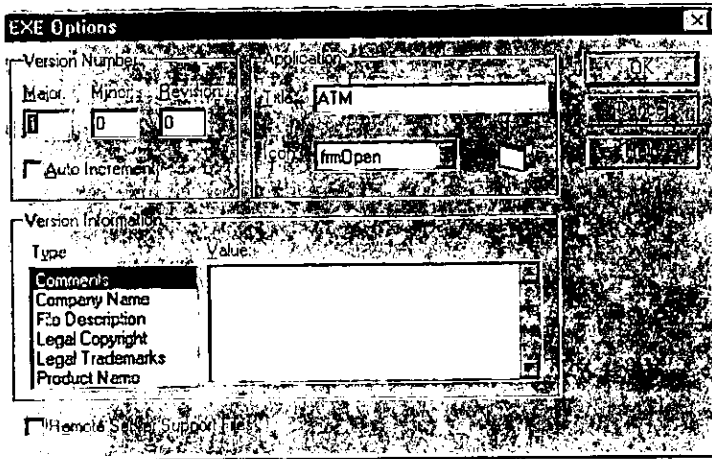


Figura 1.5.16 Caja de Diálogo de Opciones EXE.

## Ventajas de Visual Basic 4.0

En cuanto a programación.

1. Es un lenguaje de programación completo y permite la construcción de programas estructurados.
2. No es un lenguaje orientado a objetos porque carece de los componentes de herencia y polimorfismo, aunque no de encapsulamiento.
3. Su compatibilidad con las antiguas versiones de Visual Basic es casi completa acepta instrucciones tradicionales como print, input o data aunque estas instrucciones no son muy útiles.

4. Utiliza el tipo de Dato 'Variant' el cual es una variable de datos que puede almacenar información de cualquier tipo, ya sea numérico, cadenas o de fecha/tiempo.
5. La programación en Visual Basic es por eventos, es decir, que asocia código a las acciones posibles de los objetos de nuestra forma.
6. Utilización de las denominadas Visual Basic Extensión (VBX). Los VBX programas normalmente realizados en lenguaje C que realizan una función determinada y que pueden ser utilizados de manera directa en las Aplicaciones Realizadas en Visual Basic.
7. Posee una librería de rutinas especializadas con la cual el programador puede crear sus propios VBX, estas librerías están basadas en el software llamado 'Microsoft Windows Software Developed Kit (SDK)'.

En cuanto a Características Avanzadas.

1. Permite la creación de aplicaciones MDI (Multiple Document Interface). Una aplicación MDI permite al usuario desplegar múltiples documentos (formas) al mismo tiempo cada una de ellas en su propia ventana, estos documentos están contenidos en una ventana denominada 'Ventana Padre' que provee el espacio de trabajo para todas las 'Ventanas hijas'. Ejemplo de esto, el Manejador de Programas de Windows (Windows Program Manager) el cual contiene muchos diferentes grupos de programas. Cada grupo de programas es visualizado en su propia ventana y está confinada a la 'Ventana Padre' que es el Manejador de Programas de Windows).
2. Permite el intercambio Dinámico de Datos (DDE) con otras aplicaciones: Debido a que las aplicaciones generadas con Visual Basic se ejecutan en un Sistema Operativo Multitareas, es decir, que corren varias aplicaciones al mismo tiempo, el intercambio de datos con estas aplicaciones podría ser el conocido 'Cut and Paste'. Sin embargo, Visual

Basic nos permite automatizar este proceso usando el DDE (Dynamic Data Exchange) para extraer datos de otras aplicaciones y automáticamente actualizar la nueva aplicación.

3. Permite el uso de OLE (Object Linking and Embedding): OLE es un método de Intercambio e Interoperatividad entre aplicaciones. A través de OLE las aplicaciones generadas en Visual Basic podrán desplegar y manipular datos de otras aplicaciones Windows.

4. Llamadas a Procedimientos en DLL's: DLL (Dynamic-Link libraries) son una característica básica de Microsoft Windows. Las DLL's son librerías de procedimientos que las aplicaciones pueden enlazar y usar en el tiempo de ejecución, pueden ser compartidas por múltiples aplicaciones. En estas librerías se encuentran contenidos los procedimientos de manejo de memoria, graficación, etc. Las aplicaciones en Visual Basic pueden realizar llamadas a estos procedimientos para el desarrollo de acciones especiales que involucren DLL's del Ambiente Windows (USER.DLL, GDI.DLL, KERNEL.DLL, MMSYSTEM.DLL) u otros DLL's.

En cuanto a Hardware:

1. Los requerimientos de hardware para Visual Basic son comparativamente menores que otras herramientas de programación visual que existen en el mercado.

2. Otra de las grandes ventajas de Visual Basic es la utilización de las denominadas Visual Basic eXtension (VBX), así como OCX ó Custom Controls. Mediante un VBX se pueden desarrollar funciones que no están integradas en Visual Basic. Por ejemplo, si un programador necesita el uso en alguna aplicación de gráficas que se encuentran en formatos distintos a los que maneja Visual Basic (por ejemplo GIF o TIFF), mediante un VBX puede ser programado en un lenguaje distinto, y ser utilizado en las aplicaciones Visual Basic. Y aún mejor existen actualmente en el mercado más de 1000 VBX con

funciones que van desde la graficación, cálculos financieros o animación hasta la inteligencia artificial y el reconocimiento de voz.

3. Las licencias de distribución de las aplicaciones generadas en Visual Basic, son fáciles de obtener en comparación con algunos otros lenguajes, para los que hay que comprar una licencia por cada aplicación.

### **Desventajas.**

1. La capacidad de los 'custom controls' que traen de fábrica es limitada, sobre todo si el Visual Basic va a ser utilizado para el manejo de Bases de Datos de gran tamaño, y obliga a usuarios la adquisición de Custom Controls de terceros fabricantes.

2. Los requerimientos de Hardware aumentan considerablemente si se utilizan demasiados custom controls en un sólo programa.

## **1.6 Características de Access 7.0**

### **¿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 constituidas 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 (fig. 1.6.1).

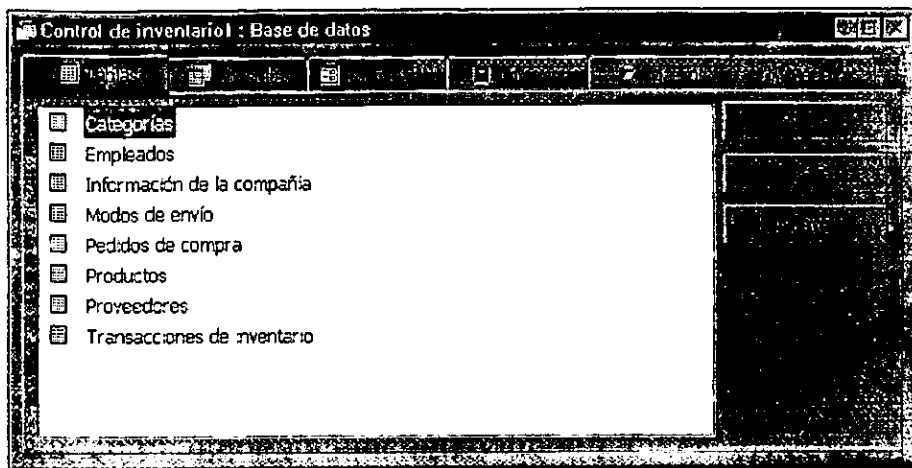


Figura 1.6.1 Objetos de Access.

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 1.6.2. nos muestra la estructura de Access 97.

#### 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 más 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 1.6.3).

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



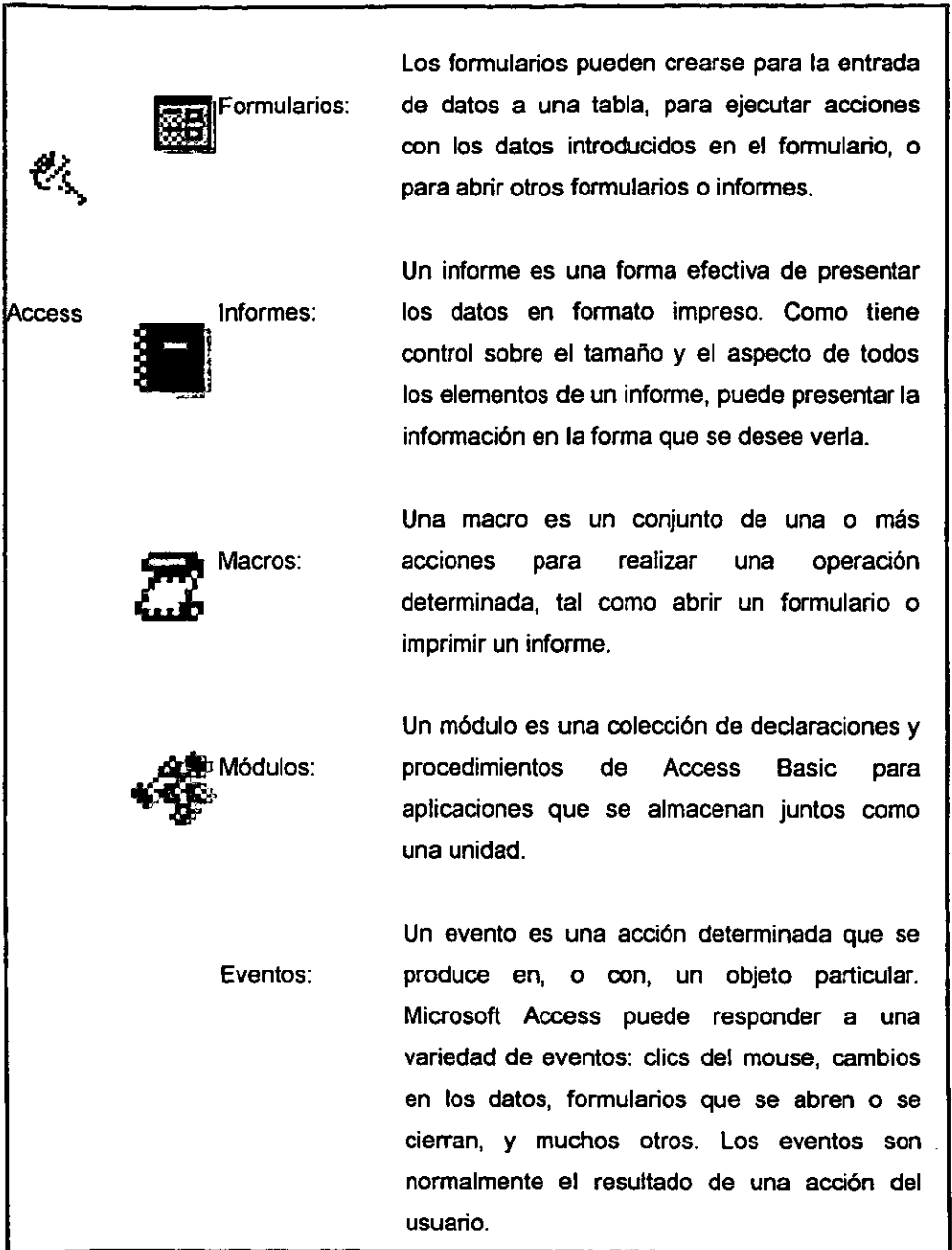
Tablas:

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.



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 1.6.2 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. De acuerdo con la definición que se ha proporcionado en la figura 1.6.2, una tabla 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 1.6.3.

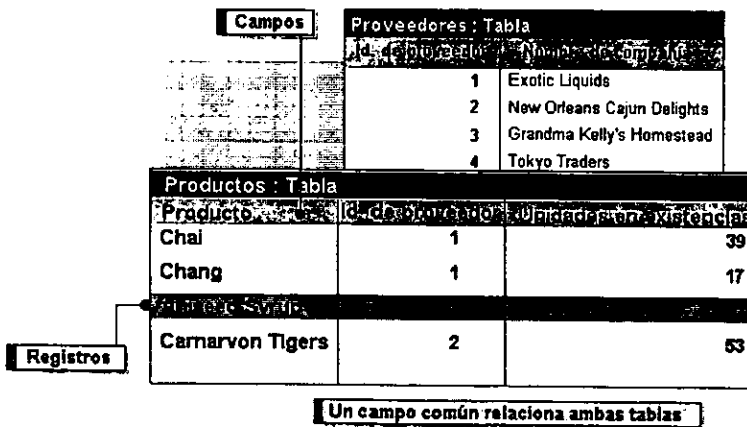


Figura 1.6.3 Estructura de una tabla de Microsoft Access.



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 se establezcan 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.

Cuando se crea una tabla, ésta no se encuentra 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 deseen 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 se puede actualizar un campo de una tabla y hacer que 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. (Ver figura 1.6.4).

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.

Cuando ejecuta la consulta, Microsoft Access recupera los registros que especifique...

... y a continuación muestra los datos combinados en el orden que quiera.

Proveedores: Tabla			
	Proveedor	Id. de proveedor	Teléfono
Exotic Liquids		1	(71) 555-2222
Alice Mutton		29	(514) 555-2965
Aniseed Syrup		14	(0544) 60323
Boston Crab Meat		34	(02) 555-6914
Camembert Pierrot		61	(03) 444-2343
			(06) 431-7877
			(503) 555-9931
			(03) 444-2343

Productos y proveedores. Consulta de selección		
Nombre de producto	Proveedor	Teléfono
Alice Mutton	Pavlova, Ltd.	(03) 444-2343
Aniseed Syrup	Exotic Liquids	(71) 555-2222
Boston Crab Meat	New England Seafood	(617) 555-3267
Camembert Pierrot	Gai pâturage	38.76.98.06

Figura 1.6.4 Consulta de Microsoft Access.

Existen cuatro 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 una 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 es posible incluir datos de varias tablas y establecer criterios para mostrar sólo un conjunto limitado de datos. También permite 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 se puede actualizar, eliminar o agregar un grupo de registros simultáneamente; También, es posible 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. (Ver figura 1.6.5).

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 para introducir los datos, pues ahorra tiempo y evita errores de mecanografía.

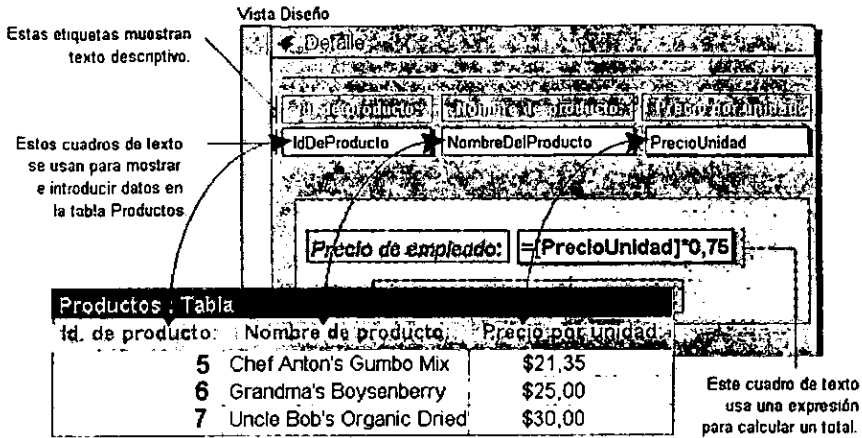


Figura 1.6.5 Formulario de Microsoft Access.

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*, permite 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*, sirve 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.
- Crear 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 se conocen como *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 en él 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 el 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 1.6.6).

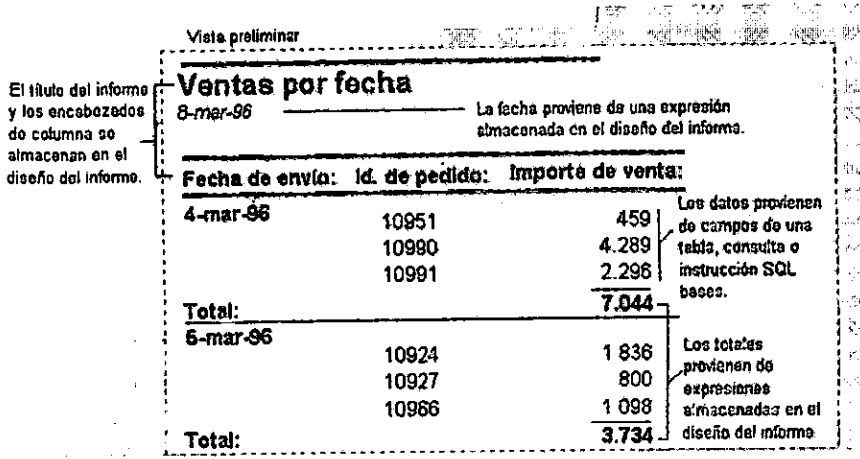


Figura 1.6.6 Informe de Microsoft Access.

Con Access se pueden diseñar informes que presenten la información de la forma deseada. Se pueden 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 para 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 mas para crear las macros. Cuando se ejecuta la macro, Access realizara 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, brindar las facilidades para importar o exportar datos entre diferentes formatos de archivos.
- Crear un 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 97 se asegura que los usuarios puedan darse cuenta de los beneficios de usar una base de datos rápidamente. Con estas nuevas tecnologías integradas Access 97 fue diseñado para hacer más fácil para todos los usuarios encontrar respuestas, compartir información al mismo tiempo y crear soluciones rápidas.

### **Nuevas Características de Access 97**

Las nuevas características de Access 97 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 97 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.

La figura 1.6.7 muestra un esquema de los elementos mejorados

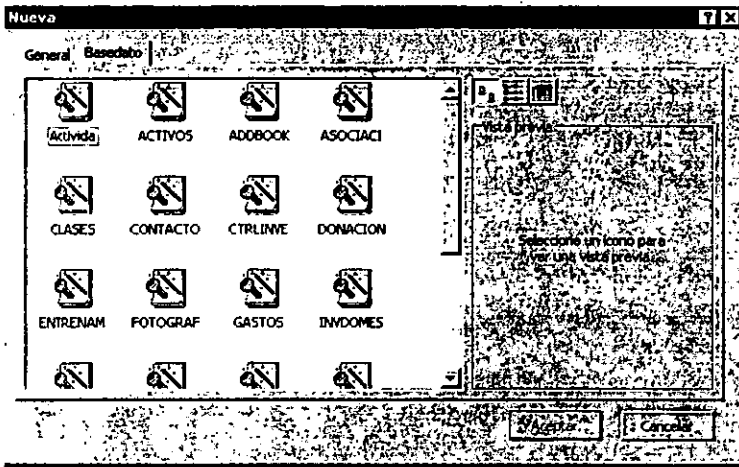


Figura 1.6.7 Asistente para crear bases de datos empleando plantillas.

*Asistente de Consultas:* Esta ayuda de usuario localiza rápida y exactamente la información que se busca, a pesar del nivel de experiencia en bases de datos con la que se cuenta.

*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 máximo valor. 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 entre éxitos y fracasos. La receta surge con la popularidad de Internet e Intranets. Como resultado Microsoft Access 97 incluyó nuevas características que permiten a los usuarios obtener más de sus ambientes computacionales ya instalados. Desde un principio, las bases de datos han sido herramientas que pueden ser usadas por más 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 97 ayudando a compartir a los usuarios información con un grupo de trabajo, en una Intranet o sobre Internet.

Microsoft Access 97 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, tales como ir a los URL's, 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 97 y trae nuevas herramientas orientadas al desarrollo. Estas mejoran el camino de los desarrolladores tanto en rapidez como en métodos para crean 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 a 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.

### **Especificaciones de uso de Microsoft Access.**

*Para utilizar Microsoft Access 97 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.



## II PLANTEAMIENTO DEL PROBLEMA

### 2.1 Definición del problema

Las necesidades actuales que tienen las empresas y compañías de integrarse al mundo de la computación, en todos los campos de acción en los que interviene el ser humano hace necesario que se desarrollen sistemas computacionales accesibles, sencillos y prácticos; que relacionen sin mucha dificultad a las personas que cuentan con poco conocimiento en el uso de sistemas computacionales.

Con ello surge la necesidad de crear sistemas que antes se hacían de forma manual y que para realizarlos llevaban demasiado tiempo. Para ello se han ido creando interfaces gráficas que involucran una serie de elementos tanto de hardware como de software sobre los que se apoya un sistema, que a la par con el desarrollo de sistemas de software, permitan a las personas obtener de una manera más sencilla y rápida, información para el mejor control de los sistemas.

Hoy en día con el surgimiento de lenguajes de programación visual en un ambiente como Windows, se incorporan características para crear interfaces basadas en conceptos de programación orientada a objetos, los cuales sirven de herramienta para el desarrollo del presente trabajo: **“Sistema para determinar el valor unitario de un predio dentro del Distrito Federal”**.

El sistema se enfoca hacia la problemática de que actualmente existen parámetros aislados que no unifican los criterios de los peritos valuadores para un determinado predio en una zona específica en el Distrito Federal, lo que provoca que difícilmente el cálculo de un avalúo hecho por un perito competente coincida con otro avalúo realizado para el mismo predio por otro perito competente. En el avalúo interviene un margen de apreciación personal, lo que nos trae como consecuencia el no contar con un valor real comercial de partida para negociar la venta o la compra del bien inmueble (predio).

Hoy en día, para llevar a cabo un avalúo, se elabora un documento en el que se especifica: la fecha, nombre del solicitante, los datos generales del predio (calle, colonia, ciudad, delegación, código postal, Entidad Federativa, boleta predial, propietario, régimen de propiedad), datos generales de la zona, uso de suelo, intensidad de construcción permitida, servicios públicos con que cuenta la zona en que se encuentra ubicado el predio, tipo de construcción dominante, saturación de construcción en la zona, densidad de población en la zona, calles que forman la manzana, descripción del predio (linderos, colindantes, según planos de la Secretaría de Desarrollo Urbano y Vivienda, área del terreno). También se incluye una descripción del predio, la que se especifica si cuenta con algún tipo de construcción (aunque en el avalúo no se considera dicha construcción), si se encuentra bardado, si es plano o no, si es regular o irregular en su forma. Posteriormente a manera de consideración previa, se formula la determinación de un precio mínimo de venta del terreno con base en la exhaustiva investigación de mercado sobre predios similares al de estudio, considerando ubicación, dimensiones, condiciones físicas, servicios públicos con que cuenta, además del uso de suelo que posee el predio de acuerdo con el Plan Parcial de Desarrollo Urbano Delegacional vigente.

En función de los factores antes mencionados se aplica un incremento o un decremento para el importe del bien que ha sido valuado.

Por otro lado, cuando en un nuevo avalúo se requiere consultar información acerca de otro similar, a fin de comparar datos de ambos, es necesario iniciar una búsqueda en los archivos donde se almacenan todos los documentos sobre los avalúos realizados anteriormente, lo cual requiere mayor cantidad de tiempo para la obtención de la información de apoyo para la realización de otros avalúos y esto a su vez se refleja en el retraso en la entrega del avalúo que se está efectuando. Por esta razón, muchos organismos públicos y privados se han hecho partícipes de las ventajas que trae consigo el uso de sistemas computarizados para llevar a cabo con un mayor grado de eficiencia y rapidez sus actividades cotidianas.

En el presente trabajo se pretende el desarrollo de un sistema que obtenga el valor unitario para un predio en el Distrito Federal de manera automatizada y homogénea con base en los factores definidos, y que cuente con una base de datos que contenga la información, y que permita consultas y actualizaciones de la misma, de acuerdo al comportamiento de la oferta y la demanda para cada zona específica del D.F.

Para lograr determinar un avalúo en forma rápida, el sistema muestra una interfaz gráfica mostrando distintas pantallas de acuerdo a un menú general. El análisis de la información que se obtenga permitirá lograr una buena clasificación, tal que permita un buen manejo de los datos y la obtención de un producto de calidad.

El sistema contará con menús y submenús, los cuales podrán ser accedidos por los usuarios a través de los dispositivos de entrada: teclado y mouse (ratón), siendo éste último de mayor utilidad, ya que sólo se requiere apuntar con el cursor hacia el menú u objeto deseado para hacer uso de él.

El sistema será desarrollado en el lenguaje Visual Basic versión 4.0 - cabe mencionar que Visual Basic no tiene un enfoque propiamente orientado a objetos, sino a eventos, aunque este lenguaje proporciona los elementos necesarios para realizar la interfaz gráfica, y un código de programación formal que requiere un cierto conocimiento del software.

En cuanto al volumen de información, debe ser analizado correctamente para determinar cuánta información se pretende manejar, ya que existen numerosos factores que pueden influir en el precio de un predio, así como varias organizaciones que se dedican a valorar bienes inmuebles entre las que se incluyen los predios. El punto de interés de este sistema, se enfoca a una selección de los factores más importantes y representativos de un avalúo. Al hacer la selección de la información, el objetivo es que ésta sea lo más breve y concisa, pero sin perder el contenido, para que el usuario cuente con los datos necesarios que le permitan realizar un buen trabajo.

Para ello, al diseñar la estructura de la base de datos, se deberá tomar en cuenta el análisis anterior y un buen diseño que cumpla con el alcance para el desarrollo de una base de datos óptima, buscando que el manejo de la información sea fácil, rápida y clara, hasta donde lo permita el manejador de la base de datos ACCESS, de tal manera que al ser desplegada en las pantallas creadas en Visual Basic, provoque el interés del usuario final del sistema, y proporcione los datos requeridos.

Por ello, no se debe olvidar que al presentar la información en las ventanas del entorno y adentrarse en el sistema, éste debe ser fácil de asimilar por los usuarios, tomando en consideración que la mayoría de ellos no están familiarizados con los términos usados en computación, aunque si deberían contar con un mínimo de conocimientos para el manejo y administración del sistema.

## **2.2 Descripción de objetivos**

A continuación se mencionan los objetivos que se pretenden cubrir con el desarrollo del presente sistema:

- Estandarizar los criterios de valuación.
- Disminuir el número de operaciones que se realizan en forma manual.
- Incrementar la eficiencia de las operaciones.
- Reducir el tiempo de entrega de un avalúo.
- Proporcionar un sistema que satisfaga los requerimientos de información en cuanto a almacenamiento, procesamiento y control.
- Mantener la compañía en un nivel competitivo dentro del área de avalúos.
- Establecer las bases para un sistema que realice avalúos de otros bienes inmuebles como son la rama de las construcciones (departamentos, edificios, bienes en renta y comerciales).

### **2.3 Conceptos de bases de datos relacionales**

El desarrollo de las bases de datos colectivas es y será sin duda una de las actividades más importantes en el campo de la computación en lo que resta del presente decenio. Los datos se mirarán cada vez con mayor interés como un recurso vital para la empresa que merece ser debidamente organizada con el fin de maximizar su valía.

Es impresionante observar como crecen en volumen e importancia los archivos de datos que utilizan las computadoras. La tasa de crecimiento anual de la capacidad de almacenamiento de estas máquinas es mayor que la del tamaño o la potencia de cualquier otro componente en la explosiva industria de la computación. Cuanto mayor es la cantidad de datos a que tiene acceso la computadora, tanto mayor es su poder potencial. En todas las esferas de la vida y en todas las áreas de la industria y el comercio las bases de datos ampliarán en gran medida las posibilidades de acción abiertas al hombre. En los siglos venideros, los historiadores contemplarán el surgimiento de los bancos de datos y las facilidades que ofrecen como un mito en la evolución de la sociedad, tal vez de mayor trascendencia aún para la condición humana que la invención de la imprenta.

Algunos de los aspectos más sobresalientes del desarrollo industrial de nuestros días arraigan en el extraordinario crecimiento de la disponibilidad de información. En su gran mayoría, esta información no tiene todavía nada que ver con la computadora. No obstante, el costo de los equipos para el almacenamiento de datos decrece más rápidamente que cualquier otro en el campo de la computación y, así, pronto resultará más barato almacenar datos en los archivos de las computadoras que sobre papel. Y no solo se almacenará la información escrita. La industria de la computación está aumentando su capacidad para almacenar digitalmente dibujos lineales, fotografías, voces humanas, etc. De hecho, es posible en nuestros días almacenar y transmitir digitalmente cualquier tipo de información.

La disminución del costo de almacenamiento de datos por bit está estrechamente vinculada con el hecho de que cada vez es mayor la cantidad de datos que se almacenan. Gracias a lo anterior y a la constante mejora de las facilidades para transmitirlos a distancia, las bases de datos desempeñarán un papel de primer plano en la evolución futura de la industria.

Es una tarea formidable la de identificar todos los datos que se necesitan para el buen control de una empresa y decidir cuál es la mejor manera de registrarlos y almacenarlos.

La tarea de diseñar una base de datos es cada vez más difícil, en particular cuando se pretende alcanzar soluciones óptimas. El software es cada vez más elaborado y a menudo se comprenden mal sus aptitudes, se le emplea erróneamente o no se le aprovecha a fondo. Hay muchas maneras de estructurar los datos y cada una de ellas ofrecen ventajas y desventajas. Un factor destacado entre los que contribuyen a dificultar la tarea es el hecho de que sean tantos los tipos de datos que reclaman ser organizados de manera particular. Cada tipo de datos exhibe características propias que afectan el modo de organizarlos y a menudo los usuarios tienen exigencias muy diversas. Los requisitos que se le imponen al diseñador son tan variados que a menudo resulta imposible satisfacer con una organización única.

### **2.3.1 ¿Qué es una base de datos?**

Hay quienes conciben la base de datos como un enorme receptáculo en el que un organismo guarda todos los datos procesables que reúne y al cual acuden muy diversos usuarios, esto se puede observar en la Fig. 2.3.1.1. Este gran almacén puede estar concentrado en una localidad determinada o distribuido en varias, todas ellas posiblemente interconectadas mediante un sistema de telecomunicación. Tienen acceso a la base de datos programas de la más diversa índole.

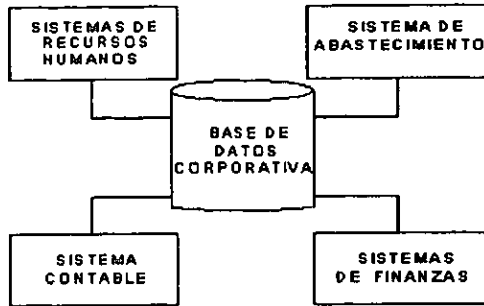


Figura 2.3.1.1 Enfoque colectivo o corporativo de una base de datos relacional.

Una base de datos puede definirse como una colección de datos interrelacionados almacenados en un conjunto, sin redundancias perjudiciales o innecesarias; su finalidad es la de servir a una aplicación o más, de la mejor manera posible; los datos se almacenan de modo que resulten independientes de los programas que los usan; se emplean métodos bien determinados para incluir datos nuevos y para modificar o extraer los datos almacenados. Una representación gráfica se muestra en la fig. 2.3.1.2.

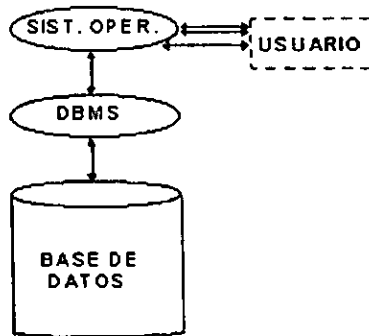


Figura 2.3.1.2 Representación gráfica del concepto de una base de datos.

Se dice que un sistema comprende una colección de bases de datos cuando éstas son totalmente independientes desde el punto de vista estructural.

En las organizaciones más sencillas, encontramos casi siempre una colección de registros organizados para una aplicación determinada. La idea básica en la implantación de una base de datos es la de que los mismos datos deben ser aprovechados para tantas aplicaciones como sea posible. Por eso, la base de datos se concibe a menudo como un repositorio donde se reúne la información necesaria para el ejercicio de las funciones propias de un organismo gubernamental, una empresa, una fábrica o cualquier otra organización. Semejante base de datos permitiría no solo la lectura de los datos almacenados, sino la continua modificación de los que son necesarios para el control de las operaciones. Será posible inspeccionar la base de datos en busca de la respuesta a los interrogantes que se planteen o de información para fines de planeamiento. La misma colección de datos servirá para varios departamentos, posiblemente sin tener en cuenta algunas fronteras administrativas.

La base de datos puede ser diseñada para el procesamiento por lotes, en tiempo real o en línea (en el que las transacciones se procesan una por una hasta completarse, pero sin los severos requisitos de tiempo propios de los sistemas de tiempo real). Muchas bases de datos sirven a una combinación de estos métodos de procesamiento y en ellas se emplean, por ejemplo, terminales de tiempo real al mismo tiempo que el procesamiento por lotes.

En la mayoría de las bibliotecas de cintas o discos anteriores con la llegada de la técnica de las bases de datos, hay una sorprendente cantidad de datos duplicados o redundantes. Muchos datos se hallan simultáneamente almacenados en varios volúmenes con distintas finalidades y también con diferentes fechas de actualización. En la mayoría de base de datos es deseable eliminar esta redundancia. Sin embargo en algunas de ellas se admite cierta redundancia con el objeto de reducir los tiempos de acceso o simplificar los métodos de direccionamiento. Algunos registros se duplican para facilitar la reconstrucción de la base de datos en caso de daño accidental. Hay



pues, necesidad de armonizar el grado de redundancia con otras características deseables de la base, de modo que es preferible hablar de *redundancia controlada* o *redundancia mínima* en lugar de *no redundancia* como criterio de diseño. En otros términos, en una base de datos bien diseñada se evita la redundancia perjudicial o innecesaria.

Una de las características más importantes de la mayoría de las bases de datos es la de mantenerse en plena crisis de cambio y crecimiento. La base de datos debe prestarse a una fácil reestructuración siempre que haya que agregar nuevos tipos de datos o utilizarla para nuevas aplicaciones. Esta reestructuración no debe originar la necesidad de volver a escribir los programas de aplicación y, en general, no debe ser fuente de trastornos. La facilidad con que pueda modificarse la base de datos tendrá siempre un efecto directo sobre la capacidad para desarrollar nuevas aplicaciones del procesamiento de datos dentro del organismo que la explota.

### **Objetivos de una organización de bases de datos**

Las bases de datos pueden organizarse de muchas maneras, pero los principales objetivos que debe satisfacer un Sistema de Gestión de Bases de datos deben ser los siguientes:

- 1) Versatilidad para la representación de relaciones.

Diferentes programadores requieren diferentes archivos. Estos archivos deben derivarse de la misma colección de datos. Existen diversas relaciones entre los rubros de datos almacenados. Algunas bases de datos comprenden un complejo entretejido de relaciones. El método de organización debe tener capacidad para representar estas relaciones y acomodar los posibles cambios en el futuro. El sistema de administración de datos debe tener a su vez capacidad para derivar, de datos y relaciones, los archivos que se requieran.

## 2) Desempeño.

Las bases de datos diseñadas para ser usadas por operadores de terminal deben asegurar un tiempo de respuesta adecuado para el diálogo entre el hombre y la terminal. Además, el sistema de bases de datos debe tener capacidad para manejar un adecuado caudal de transacciones. En los sistemas en el que el volumen de tráfico es reducido, el caudal de transacciones no tiene por qué imponer muchas restricciones al diseño de la base de datos. En cambio en los sistemas de alto volumen de tráfico, por ejemplo los de reserva de plazas en las aerolíneas el caudal de transacciones tiene una gran influencia sobre la organización del almacenamiento físico.

En los sistemas diseñados sólo para el procesamiento por lotes, el tiempo de respuesta no tiene una gran significación y el método de organización física se elegirá teniendo a la vista la mayor eficiencia con ese tipo de procesamiento.

El tiempo de respuesta adecuado en el caso de un sistema con terminales depende de la naturaleza del diálogo entre hombre y terminal. Para ciertas categorías de diálogo se requiere un tiempo de respuesta del orden de dos segundos, (es decir, dos segundos entre el instante en que el operador completa su mensaje de entrada en la terminal y el instante en que aparece en ésta el primer carácter de la respuesta). En el caso de averiguaciones simples pueden tolerarse, en general, tiempos de respuesta mucho mayores.

## 3) Costo mínimo.

Con el fin de mantener bajo el costo, hay que elegir técnicas que minimizan las necesidades totales de almacenamiento. Apelando a estas técnicas, la representación física de los datos en el almacén puede ser muy distinta de la representación que usa el programador. La conversión entre ambas representaciones se hace por medio del software, o posiblemente del hardware o de la microprogramación. Debe llegarse

siempre a una concordancia entre el costo de los algoritmos de conversión y la economía de almacén.

El costo por bit de almacenamiento está disminuyendo rápidamente gracias al progreso tecnológico, mientras que no ocurre lo mismo con el costo de la programación. Hay por lo tanto una necesidad creciente de mantener sencilla la programación de aplicación, y las organizaciones lógicas de datos deben diseñarse con este objetivo.

#### 4) Redundancia mínima.

Antes de la aparición de las técnicas de base de datos, existía un altísimo nivel de redundancia en los sistemas de procesamiento de datos. La mayoría de las bibliotecas de cinta contienen muchos datos redundantes. Aún con las técnicas de las bases de datos, a medida que crecientes volúmenes de información se combinan para formar bases de datos integradas, se crea una mayor posibilidad de redundancia. Como sabemos, la redundancia es onerosa porque ocupa más espacio de almacén que el necesario y requiere múltiples operaciones de actualización. Debido a que diferentes copias de la misma información suelen hallarse en diferentes etapas de actualización, la redundancia da a menudo origen a respuestas incoherentes.

Debería ser un objetivo de la organización de una base de datos el de eliminar los valores de datos redundantes, siempre que resulte económico, y controlar las incoherencias a que pueden dar lugar esos valores redundantes. Para ello es preciso identificar los datos que son utilizados en común por varias aplicaciones y almacenar una sola copia. Pero si hay una sola copia de ítems de datos que son necesarios para diversos fines, es preciso almacenar y mantener las relaciones pertinentes. Por esta razón, entre otras, puede no ser siempre razonable eliminar toda la redundancia en un sistema.

## 5) Capacidad de búsqueda.

El usuario de una base de datos suele plantear muchas interrogantes acerca de los datos almacenados. En la mayoría de las aplicaciones comerciales actuales, se han anticipado los tipos de averiguación y los datos se han organizado físicamente de modo de satisfacerlas con adecuada prontitud. Hay una creciente demanda para sistemas que sean capaces de atender consultas o producir informes que no han sido previstos detalladamente en la época del diseño. El usuario quiere presentar pedidos espontáneos de información en una terminal. Las averiguaciones no anticipadas (y algunas de las anticipadas) hacen necesario explorar algunas partes de la base de datos. Si en la terminal se pretende una respuesta inmediata, la exploración tiene que ser rápida. La capacidad para explorar una base de datos rápidamente y con diferentes criterios de búsqueda depende mucho de la organización física de los datos. Con algunas organizaciones los tiempos de búsqueda son excesivamente prolongados para poder considerar las respuestas como de tiempo real. Uno de los objetivos de la organización es, pues, el de lograr capacidad para la búsqueda rápida y flexible.

## 6) Integridad.

Cuando una base de datos incluye información utilizada por muchos usuarios, es importante que no puedan destruirse los datos almacenados ni las relaciones que existen entre los distintos datos. Ocasionalmente se producirán fallas de hardware y diversos tipos de accidente. El almacenamiento de los datos y los procedimientos de actualización e inserción deben asegurar que el sistema pueda recuperarse de estas contingencias sin daño para los datos. Toda instalación debe garantizar la integridad de la información que almacena.

Además de proteger los datos contra posibles problemas sistemáticos deben incluirse también procedimientos de chequeo que aseguren que los valores de los datos se ajusten a ciertas reglas prescritas de antemano. Estas pruebas podrán hacerse verificando las relaciones que existen entre varios valores de datos.

## 7) Reserva (<< privacidad >>) y seguridad.

Los datos almacenados en una base deben ser conservados con seguridad y reserva. La información almacenada es a menudo de gran valor para la empresa, por lo que deben estar protegidas contra su pérdida o robo. Cuanto más vital es la información almacenada, tanto más importante es protegerla contra los fallos del hardware o del software, contra las catástrofes y contra los criminales, vándalos, incompetentes y personas que pretendan darles un uso ilegítimo.

*La seguridad* de los datos se refiere a la protección de éstos contra el acceso accidental o intencional por parte de personas no autorizadas, y contra su indebida destrucción o alteración.

*La reserva* se refiere al derecho de los individuos y organismos para determinar por sí mismos cuándo, cómo y en qué medida se permitirá la transmisión a terceros de la información que les concierne.

Aunque la tecnología necesaria para asegurar este secreto está estrechamente relacionada con la de la seguridad, la reserva es una cuestión que traspasa los límites del centro de computación. En gran medida es un problema social. Para preservar el secreto de los datos personales, se necesitan recursos que escapan a la tecnología. La sociedad del futuro, cada vez más dependiente del uso masivo de datos, requerirá nuevos cuerpos legales y nuevos controles sociales si quiere mantener el grado de reserva que se pretende sobre la información personal.

La seguridad es un tema extremadamente complejo a causa de sus múltiples y variados aspectos. El analista de sistemas que se hace responsable de la seguridad debe estar familiarizado con todas las particularidades del sistema, porque éste puede ser atacado con fines ilícitos desde muchos ángulos. A veces se presta mucha atención a alguno de los aspectos del problema mientras se descuidan otros.

Los siguientes siete requisitos son esenciales para la seguridad de la base de datos:

- a) La base de datos debe estar protegida contra el fuego, el robo y otras formas de destrucción.
  - b) Los datos deben ser reconstruibles, dado que por muchas precauciones que se tomen, siempre ocurren accidentes.
  - c) Los datos deben poder ser sometidos a procesos de auditoría. La falta de auditoría en los sistemas de computación ha permitido la comisión de grandes delitos.
  - d) El sistema debe diseñarse a prueba de intromisiones. Los programadores, por ingeniosos que sean, no deben pasar por alto los controles.
  - e) Ningún sistema puede evitar de manera absoluta las intromisiones malintencionadas, pero es posible hacer que resulte muy difícil eludir los controles. Los usuarios de la base de datos deben ser sometidos a un proceso de identificación positiva antes de tener acceso a ella.
  - f) El sistema debe tener capacidad para verificar que sus acciones han sido autorizadas.
  - g) Las acciones de los usuarios deben ser supervisadas, de modo tal que pueda descubrirse cualquier acción indebida o errónea.
- 8) La interface con el pasado.

Los organismos que han estado usando el procesamiento de datos durante algún tiempo han invertido mucho dinero en los programas, procedimientos y datos existentes. Cuando un organismo se decide a instalar un nuevo software de base de datos, es importante que éste pueda trabajar con los programas y procedimientos existentes y que los datos ya almacenados puedan ser convertidos a las nuevas formas. Esta necesaria compatibilidad llega a menudo a tener el carácter de una severa restricción para pasar al nuevo sistema de base de datos. Es esencial, no obstante, que no se frene el avance de la tecnología de la informática por presentar una atención excesiva a esta compatibilidad con el pasado. El mayor obstáculo para el progreso del

software se encuentra posiblemente en la exageración de los problemas existentes y en tratar de remendarlos.

## 9) La interface con el futuro

Más importante que el vínculo con el pasado es la interface con el futuro. En el futuro, los datos y los medios de almacenamiento cambiarán de muchas maneras. Ninguna organización comercial es estática; el cambio es un modo de ser. Pero los cambios han resultado extremadamente costosos para los usuarios del procesamiento de datos. El enorme costo de cambios aparentemente triviales ha frenado seriamente el desarrollo de nuevas aplicaciones. El costo ha sido el resultado de tener que reescribir programas de aplicación, convertir datos y resolver los muchos problemas menores introducidos por los cambios. Con el transcurso de los años, la cantidad de programas de una organización va creciendo hasta el punto de que en algún momento no podrá ya pensarse en volverlos a escribir. Uno de los objetivos más importantes en el diseño de una base de datos será, el de planearla de manera que se le pueda modificar sin necesidad de tener que alterar los programas de aplicación en uso.

Para lograr esta esencial protección, el diseño debe prestar particular atención a dos características, ambas complejas y difíciles de implementar. Primero, la vista que de los datos tiene el programador de aplicaciones debe ser independiente de la representación física de aquéllos y el software de administración de datos debe ejecutar las conversiones necesarias para pasar de una a otra. Cuando se introduzcan cambios en la organización física de los datos o en el hardware, estos cambios se reflejarán en el software de administración de datos, pero no sobre los programas de aplicación. Es ésta la que llamamos independencia física de los datos.

Segundo, la vista de los datos que tiene el programador de aplicaciones debe estar protegida contra los cambios de la estructura lógica global y contra los requerimientos propios de otros programas de aplicación.

La representación lógica global de los datos en sí cambia frecuentemente en muchas instalaciones, a medida que se agregan nuevas aplicaciones o se modifican las viejas. Se agregan así nuevos campos en los registros lógicos y se crean nuevas relaciones entre los datos existentes. Lo que hasta entonces era una jerarquía de dos niveles se ha transformado posiblemente en una jerarquía de tres niveles, y así sucesivamente.

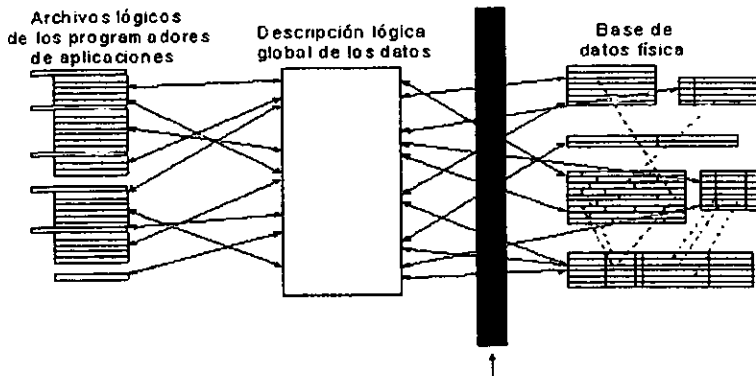
Es importante que estos cambios en la organización lógica de los datos se introduzcan sin tener que volver a escribir los programas de aplicación que no están directamente relacionados con aquéllos. Para lograr este tipo de independencia, la vista de los datos que corresponde a cualquier programa debe estar desvinculada de la representación lógica global de los datos. Debe ser posible agregar nuevos campos a un registro sin afectar los programas de aplicación existentes y que utilicen ese registro. Esta desvinculación es lo que llamamos independencia **lógica** de los datos.

Toda organización de base de datos requiere por lo tanto tres diferentes vistas de los datos:

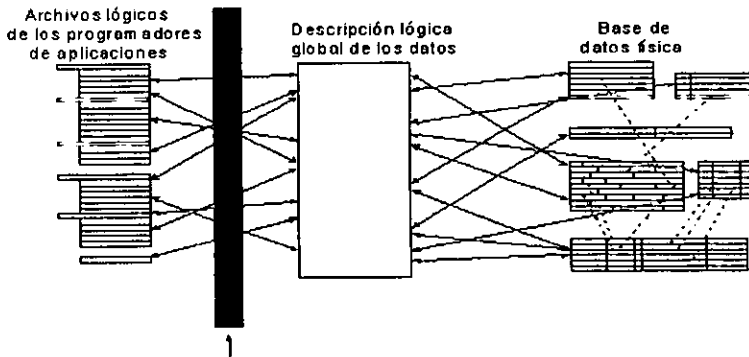
- a) La representación física.
- b) La representación lógica global de la base de datos.
- c) Las representaciones individuales de los programas de aplicación.

Debe ser posible modificar la primera de éstas sin afectar a las otras. Como lo indica la Fig. 2.3.1.3, la desvinculación de la organización física respecto de las otras dos vistas, ha de ser tan completa como sea posible. Debe ser también posible modificar la representación lógica global de los datos del sistema sin reescribir ningún programa de aplicación. Cuando las necesidades de información de un programa de aplicación cambian, el cambio puede exigir la modificación de la descripción lógica global, pero no debe dar lugar a la necesidad de modificar otros programas. Como lo indica la Fig. 2.3.1.4, la barrera entre los programas de aplicación y la descripción lógica global de los datos tiene que ser tan completa como sea posible.





**Figura 2.3.1.3** El software debería separar la organización física tan completa como fuese posible de la descripción lógica de los datos, de modo que la organización física pudiese ser afinada o cambiada totalmente sin necesidad de rehacer ni la descripción lógica global ni los programas de aplicación.



**Figura 2.3.1.4** El software debería hacer esta barrera tan impenetrable como fuese posible, de modo que cuando un programa de aplicación cambie los datos que utiliza, no se afecten los demás programas de aplicación, y no sea necesario rehacer los programas de aplicación cuando se reestructure la descripción lógica global de los datos.

En general deberá asegurarse la independencia de datos que sugiere la Fig. 2.3.1.5. Algunos sistemas actuales proveen cierta independencia física pero poca o ninguna independencia lógica.

	A	B	C
Se agrega un nuevo programa de aplicación, el que utiliza nuevos tipos de datos	X	X	X
Un programa de aplicación exige una representación modificada de los datos existentes (por ej., punto flotante en lugar de punto fijo)	X	X	X
Se agrega un nuevo programa de aplicación que utiliza los tipos de datos existentes	X		
Se insertan nuevas ocurrencias de registro se eliminan las viejas	X	X	X
Se mejora la descripción lógica global de los datos o se crean nuevas relaciones entre los datos		X	
Se consolidan dos bases de datos	X		
Se mejora la organización física de los datos; posiblemente se usan diferentes representaciones	X	X	
Se modifican los métodos de direccionamiento	X	X	
Los datos se mudan a un tipo diferente de volumen		X	X
Se modifica el software	X	X	
Se modifica el hardware	X	X	

Columna A. No hay cambios en los otros programas de aplicación.

Columna B. No hay cambios en la descripción lógica global de los datos.

Columna C. No hay cambio en la organización del almacenamiento físico de los datos.

**Figura 2.3.1.5 Características de independencia de datos deseable en los sistemas de bases de datos modernos.**

## 10) Afinación.

Cuando es indispensable asegurar el acceso en tiempo real a los datos, los usuarios del sistema se interesan al extremo en el tiempo de respuesta del sistema. En los sistemas de procesamiento por lotes, el interés se traslada al caudal de transacciones que el sistema admite o al tiempo que se tarda en despachar la carga de trabajo. Estos factores dependen del tiempo necesario para acceder a los datos requeridos y dependen también de la organización de éstos y de su localización en las unidades de almacenamiento. La diferencia entre una organización adecuada y una organización inadecuada se reflejan como una gran diferencia en los tiempos de respuesta y en los caudales de transacciones.

Cuando el almacenamiento se prevé para un conjunto específico y bien entendido de operaciones, es posible optimar, la organización del almacén y la localización de los datos en él. El diseñador sabe perfectamente qué es lo que tiene entre manos. No siempre el proyectista es tan afortunado. En muchos casos ni siquiera sabe cómo se utilizarán en realidad los archivos, cómo se interrogará la base o con qué frecuencia. Resulta por lo tanto necesario ajustar, y hasta cambiar fundamentalmente, la organización del almacén después de que el sistema ha entrado en servicio y se han aclarado suficientemente las pautas de uso. En muchos casos, el uso de la base de datos evoluciona continuamente, a medida que más personas se van familiarizando con ella y se crean más programas de aplicación. El ajuste de la organización del almacén con el objeto de mejorar su desempeño se convierte así en un proceso continuo.

Este proceso de la base de datos se llama *afinación*. En la práctica, la afinación ha conducido a menudo a importantes economías. A veces, éstas han sido tan importantes como para marcar la diferencia entre lo que es una aplicación rentable y lo que no lo es. El responsable de la afinación de la base de datos es el administrador o su grupo, y es importante que éste tenga libertad para introducir los cambios que estime necesarios, sin hacer estragos en los programas de aplicación. Sin un software

apropiado, la afinación suele incurrir en costos inadmisibles en lo que se refiere al mantenimiento y la prueba de programas.

La correcta afinación tiene dos requisitos. Primero, necesita la independencia física de los datos. Segundo, requiere medios para supervisar automáticamente el uso de la base de datos con el fin de que puedan hacerse los ajustes necesarios. En las futuras bases de datos se incorporarán posiblemente algunos medios para la afinación automática, por ejemplo, en lo que se refiere a la migración de datos. Se pretende o no que la afinación sea automática, el sistema debe ser diseñado de modo que ella sea fácil.

#### 11) Migración de datos.

Algunos datos se usan con mucha frecuencia y otros sólo ocasionalmente. Es deseable almacenar los datos de uso frecuente de manera que resulte fácil y rápido acceder a ellos. Los datos de uso ocasional se almacenarán, en cambio, de manera más económica. En una oficina, la información que se usa diariamente se halla en los archivadores de las secretarías; la información que sólo se consulta accidentalmente se guardará probablemente en el sótano, de modo que no estorbe y su almacenamiento no cueste mucho. El equivalente del sótano en la computadora podría ser la cinta magnética, mientras que los datos de uso frecuente se hallarán en discos o tambores de modo que se les pueda leer siempre en una fracción de segundo. Toda base de datos más o menos compleja tendrá múltiples niveles de facilidad de acceso.

Algunos datos almacenados, por ejemplo la música pop, gozan de una popularidad efímera. En la bolsa, por ejemplo, suele surgir repentinamente un gran interés por obtener información acerca de acciones que el mes pasado no tenían prácticamente movimiento alguno. En las aerolíneas hay un gran interés por los registros de un vuelo y sus pasajeros durante los últimos días anteriores al despegue. Diez meses antes sólo alguna que otra persona se interesaba por ese vuelo. Pero dos días después del

despegue se ha perdido todo interés en los datos pertinentes, aunque éstos deben ser conservados de alguna manera.

A medida que cambie la popularidad de un conjunto de datos, será conveniente mudarlos dentro del almacén a posiciones más o menos accesibles, de acuerdo con su actividad. En algunos casos no se mudan los datos, pero sí se modifican los índices que se utilizan para direccionarlos, de manera que puedan ser alcanzados más rápidamente. Este proceso de ajustar el almacenamiento de los datos de acuerdo con su popularidad se llama *migración de datos*. En algunos sistemas se hace automáticamente. En otros, la operación está a cargo de los programadores de sistema o del administrador de datos. En ocasiones se le considera como parte del proceso de afinación de la base de datos.

## 12) Simplicidad.

Los medios que se utilizan para representar la vista general de los datos deben ser concebidos de manera simple y nítida. En muchos sistemas se utilizan punteros en la representación lógica con el fin de patentizar las relaciones que existen entre distintos datos. Un problema que surge con el uso de punteros es que, a medida que se agregan más y más, la colección de punteros resulta tan complicada que se hace difícil representar la vista lógica general de la base de datos con suficiente claridad. En algunos casos, el abuso de los punteros múltiples en la representación de los datos para el usuario resulta engañoso. No se necesita, en realidad, tanta complejidad como la que se encuentra en ciertas estructuras lógicas de datos.

## Facilidades y limitaciones

Como se ha indicado antes, hay cierto número de Sistemas de Gestión de Bases de Datos disponibles actualmente. Las facilidades ofrecidas por ellos varían muchísimo su nivel de sofisticación. Sin embargo, en general, un buen Sistema de Gestión de Bases

de Datos debería proporcionar las siguientes ventajas respecto de un sistema convencional:

a) Independencia de datos y programas.

Esta es una primera ventaja de una base de datos. Tanto la base de datos como el programa de usuario pueden alterarse independientemente uno de otro. Esto ahorra tiempo y dinero, que de otro modo sería necesario para modificarlo, precisamente para mantener su consistencia.

b) Fácil diseño de sistemas.

Aun diseñador de sistemas en un entorno de una base de datos no le conciernen extensos diseños de archivos, ni la duplicación de datos, ni su inconsistencia, ni su mantenimiento, ni las facilidades de copia de seguridad, etc., de un sistema convencional. En una base de datos, los datos existen en una forma apropiada para todas las aplicaciones, el diseñador sólo tiene que escoger la que necesita. El diseño de sistemas es, por lo tanto, más sencillo.

c) Fácil de programar.

La tarea de programación es mucho más reducida, porque los programadores están relevados de los detalles de procesamiento de archivos, actualización de archivos, y de numerosos procesos de ordenación. El nivel de destreza de programación requerido puede ser menor para una base de datos.

d) Múltiples lenguajes anfitrión.

Se espera que un Sistema de Gestión de Bases de Datos soporte cierto número de lenguajes anfitrión, de modo que un usuario pueda elegir aquel que es más conveniente para una aplicación particular.

e) Datos consistentes y actualizados.

Una base de datos reduce la duplicación de datos y ayuda a mantener los datos consistentes y actualizados. En un sistema convencional de datos, la duplicación de datos en varios archivos conduce frecuentemente al caos y a la inconsistencia.

f) Utilización concurrente.

Una base de datos puede permitir que acudan a ella más de un programa al mismo tiempo, maximizando, por tanto, la utilización de recursos.

g) Protección de los datos.

La privacidad e integridad de los datos pueden controlarse más efectivamente en una base de datos.

h) Facilidades de interrogación.

Se espera que una base de datos soporte facilidades de interrogación de alto nivel.

i) Sistema evolutivo.

Se exige que las bases de datos sean sistemas evolutivos, para permitir que el usuario construya gradualmente su propia base de datos y que aprenda de su experiencia. También se espera que se proporcionen facilidades para reorganizar la base de datos para optimizar el rendimiento total.

La desventaja de una base de datos es su costo. Es costosa de instalar, de funcionar y de mantener. Los principales factores de costo son:

a) Gran memoria.

Las bases de datos necesitan gran memoria para albergar las rutinas del Sistema de Control de Base de Datos, los programas de aplicación, las tablas de la base de datos, los directorios, los buffers del sistema, etc., además del sistema operativo anfitrión.

b) Dispositivos de almacenamiento.

Una base de datos requiere una gran capacidad de almacenamiento y gran parte de ella se pierde como gasto extra (overheads), más que en datos reales.

c) Capacidad de canal.

Puede necesitarse una capacidad adicional del canal de entrada/salida para tener suficiente potencia para hacer frente al incremento de movimiento de datos entre la base de datos sobre los dispositivos físicos de almacenamiento y la memoria del ordenador.

d) Velocidad de procesado lenta.

La velocidad de procesado, especialmente la velocidad de actualización, es muy lenta en la base de datos, debido a unos importantes gastos extra.

e) Gastos extra de personal.

Una base de datos necesita personal adicional para su administración. El personal existente requiere nuevo adiestramiento y puede necesitarse también personal adicional, según la capacidad de la base de datos.



f) Implementación.

La implementación de un sistema de base de datos es una organización de usuarios y es una empresa importante y costosa. Nada de esto ocurre en los sistemas convencionales de procesado de datos.

g) Incompatibilidad.

La incompatibilidad de los modelos de datos disponibles es una desventaja para el usuario; una vez que ha instalado un Sistema de Gestión de Bases de Datos, llega a ser muy difícil y costoso para él cambiarlo por otro.

Como ventajas de las bases de datos se citan: el incremento de productividad de los programadores y el control de los recursos de datos. En muchos casos, las bases de datos ofrecen comodidades que son muy difíciles de cuantificar. El valor asignado a una comodidad depende fundamentalmente de para qué se le va a utilizar. Una organización que tiene instalada y utiliza satisfactoriamente una base de datos durante algún tiempo, es probable que olvide el costo y tenga en cuenta los beneficios. El grado de justificación requerido para cambiar de un sistema convencional computarizado a un sistema de base de datos es similar al requerido para cambiar un sistema manual a otro computarizado. Si consideramos las operaciones manuales como la primera etapa en el proceso de la información, el uso de una base de datos es la tercera etapa lógica.

### **2.3.2 Modelos de bases de datos**

Los modelos de base de datos son herramientas conceptuales para describir los datos, sus relaciones, su semántica y sus limitantes. Permiten interpretar de una manera abstracta fenómenos del mundo real.

De los 4 modelos de bases de datos más importantes, 3 de ellos han sido muy usados en su tiempo: jerárquicos, red y relacional. Debemos aclarar que el relacional hoy en día es el más popular.

El cuarto modelo, el orientado a objetos esta en sus primeros pasos. Aunque esta teniendo un gran auge, ya se cuentan con algunas bases de datos, sin embargo el mercado continua siendo dominado por las relacionales.

### 2.3.2.1 Modelo Jerárquico

En un modelo jerárquico los datos están estructurados en forma arborescente y las relaciones entre los diferentes tipos de registros se resuelven mediante punteros o enlaces entre ellos. Se establece una jerarquía de modo que las relaciones entre un registro y otro relacionado con él (relación padre - hijo) tienen como condición que un registro "hijo" no puede existir si no existe el registro "padre" asociado a él. La estructura es como la que se ve en la Fig. 2.3.2.1.1.

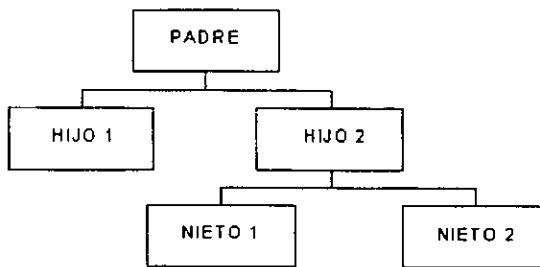


Figura 2.3.2.1.1. Estructura jerárquica.

Los arboles están compuestos por una jerarquía de elementos llamados nodos, el nivel más alto de la jerarquía tiene un sólo nodo llamado raíz.

Con excepción de la raíz, todo nodo está relacionado con otro nodo de nivel más alto llamado padre. Ningún elemento puede tener más de un padre, sin embargo, todo elemento puede tener uno o más elementos de nivel más bajo llamados hijos. Los elementos que se encuentran en las puntas de las ramas, es decir, que no tienen hijos se les llama hojas.

Los arboles se utilizan tanto para las descripciones lógicas, donde describen las relaciones que existen entre tipos de segmentos o tipos de registro; como para las descripciones físicas donde se les emplea para describir conjuntos de apuntadores y relaciones entre entradas en los índices.

### **Jerarquías balanceadas y binarias**

En una jerarquía balanceada todos los nodos deben tener el mismo número de ramas, el árbol empieza por la raíz en forma descendente y va avanzando de izquierda a derecha.

Es fácil implementar una organización física de datos para los árboles que tienen un número fijo de ramas por nodo que para los que tienen números variables.

La mayoría de las organizaciones de datos por supuesto que no encajan en una estructura jerárquica balanceada sino que requiere números de ramas diferentes en los diversos nodos.

La jerarquía binaria es una categoría especial de la estructura jerárquica balanceada, que admite solo dos nodos por rama. Las jerarquías binarias al igual que otras jerarquías balanceadas son principalmente utilizadas en la representación física de los datos y no en las lógicas.

## Correspondencia simple y compleja

Las estructuras jerárquicas implican por lo general que hay una correspondencia simple de hijo a padre, es decir, cada hijo tiene sólo un padre. Sin embargo, la correspondencia inversa es compleja, de uno a muchos, porque cada padre puede tener varios hijos.

### 2.3.2.2 Modelo de red

Si en la estructura arborescente anterior, permitimos relaciones entre "hermanos", es decir, entre registros de un mismo padre lógico, tendremos una base de datos en red. En ellas se pueden construir esquemas como el que se muestra en la fig. 2.3.2.2.1.

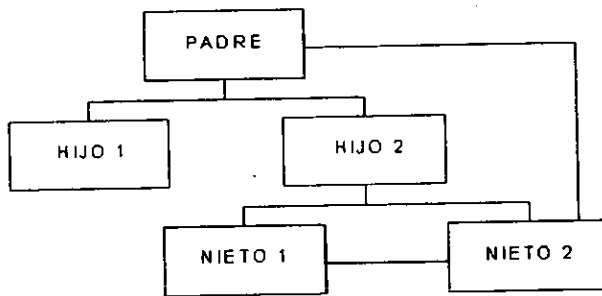


Figura 2.3.2.2.1 Ejemplo de modelo en red.

Debe considerarse al modelo jerárquico como un tipo de modelo de datos en red con ciertas restricciones.

Una desventaja de este modelo es que algunas estructuras de red comienzan a tomar una apariencia de "telaraña", con apuntadores que salen en todas direcciones, por lo cual podría llegar a ser compleja y difícil de modificar.

Si en una relación entre datos un hijo tiene más de un padre, la relación no puede ser descrita por medio de un árbol o estructura jerárquica, se le describe en cambio, como una estructura de red.

En una estructura de red cualquier elemento puede vincularse con cualquier otro elemento. Al igual que la estructura jerárquica, la estructura de red puede ser descrita en términos de padres e hijos, y dibujada de cierta forma que los padres siempre quedan arriba de los hijos. Pero en la estructura en red, un hijo puede tener más de un padre.

En algunas estructuras de red es natural referirse a niveles, como ocurre con las estructuras jerárquicas.

### **Estructuras de red simples y complejas**

En una buena cantidad de las estructuras de red que representan relaciones entre tipos de registros o tipos de agregados de datos, la correspondencia entre padres e hijos es similar a la que existe en un árbol, la correspondencia de hijo a padre es simple y la de padre a hijo es compleja.

Sin embargo, en un modelo de red es común que la estructura sea compleja, es decir, que los hijos tengan muchos padres, así como los padres tengan muchos nodos hijos.

La razón que justifica el patentizar la diferencia entre estructuras simples y complejas reside en el hecho de que las últimas requieren métodos más elaborados para su representación física.

## **Descomposición**

Las estructuras de red se pueden reducir a una forma más simple introduciéndoles algo de redundancia. En algunos casos la redundancia involucrada en la transformación no es mucha y se tolera sin problemas.

Una estructura de red compleja no es tan fácil de convertir en una estructura de árbol, se requieren dos arboles para ser representada.

En general, cada relación con correspondencia compleja en los dos sentidos tiene que ser reemplazada por dos vinculaciones del tipo de estructura jerárquica.

### **2.3.2.3 Modelo relacional**

El modelo relacional de datos es definido en 1970 por el Dr. F. Codd que lo define por medio de una serie de reglas cuyo objetivo es lograr la independencia de la representación lógica de los datos de su almacenamiento físico.

Esta independencia física/lógica se refiere a tres aspectos:

- Independencia de la ordenación, es decir, que el resultado obtenido en un acceso no dependa de cómo estén ordenados los datos físicamente.
- Independencia de la indexación, separando los índices de los datos haciendo que la creación y mantenimiento sean manejados por el sistema.
- Independencia de los caminos de acceso, haciendo que la navegación a través de los datos no tenga que estar previamente establecida consiguiendo así unas formas de acceso más flexibles.

El modelo relacional es una estructura lógica en vez de física. A continuación se mencionarán algunas ventajas de este modelo:

- La representación tabular usada en el esquema relacional es fácil de comprender por los usuarios y fácil de implementar en el sistema físico de bases de datos.
- Es relativamente fácil convertir casi cualquier otro tipo de estructura o modelo de base de datos al esquema relacional, por lo que este modelo puede considerarse como una forma de representación universal.
- Las operaciones de proyección y reunión son fáciles de implementar, por lo que la creación de nuevas relaciones necesarias para aplicaciones particulares resulta fácil de realizar.
- El control de acceso a datos sensibles es de implementación inmediata. Los datos sensibles se colocan simplemente en relaciones separadas y el acceso a estas relaciones se controlan mediante algún tipo de autoridad o esquema de acceso.
- Las búsquedas pueden ser mucho más rápidas que en los sistemas que deben seguir una cadena de apuntadores.
- Las estructuras relacionales son mucho más fáciles de modificar que las estructuras jerárquicas o de red. En ambientes donde la flexibilidad es importante, esto es primordial.
- La claridad y visibilidad de la base de datos mejora con la estructura relacional. Es mucho más fácil buscar datos tabulados que desarrollar posibles interconexiones arbitrarias y complejas de elementos de datos dentro de un mecanismo de apuntadores.

## **Elementos**

- **Tablas o relaciones:** Son arreglos bidimensionales o una colección de registros. Las tablas bidimensionales han sido una herramienta muy utilizada para representar datos de los usuarios de una manera muy fácil. Las tablas deberán organizarse de forma que no se pierda ninguna de las relaciones existentes entre datos.
- **Tupla:** Es un renglón de la tabla.
- **Atributo:** Es una columna de la tabla.

- **Cardinalidad:** La cardinalidad de una relación es el número de tuplas que la componen.
- **Grado:** Es el número de atributos o columnas de la relación.
- **Llave o clave primaria:** Por lo general en una tabla o relación existe un atributo cuyos valores son únicos, por lo cual, se pueden usar para identificar cada una de las tuplas de esa tabla, a este tipo de atributos se les conoce como llaves primarias. No siempre en una relación existe un sólo atributo que identifique a cada tupla, por lo que para identificar a cada tupla se requieren conjuntar más de un atributo formando la llave primaria.
- **Llave o clave ajena:** Es un atributo o conjunto de atributos de una relación que a su vez son clave primaria en otra relación.
- **Esquemas:** Es la descripción lógica de la base de datos, es un diagrama de los tipos de bases de datos que se usan. Proporcionan los nombres de las entidades y sus atributos y especifica las relaciones que existen entre ellos.

### **Dinámica del modelo**

La parte dinámica del modelo relacional está constituida por una serie de operadores que permiten hacer transformaciones y manipular las relaciones definidas.

Estos operadores permitirán insertar una tupla en una relación, modificarla, borrarla, etc.

Codd define esta dinámica del modelo relacional y clasifica los operadores de la siguiente forma:

#### **a) Operadores del álgebra relacional:**

- Unión.
- Intersección.
- Diferencia.



- Producto cartesiano.
- Proyección.
- Selección.

- b) Operadores de combinación.
- c) Operadores con valores nulos.

Veamos estos operadores con algunos ejemplos. Sean las relaciones R1 y R2 (fig. 2.3.2.3.1).

**R1 Empleados andaluces**

# EMPLEADO	NOMBRE	PROV. NAC.
4	JUAN	GRANADA
7	PEPE	SEVILLA
8	LUIS	SEVILLA
15	CONCHA	CORDOBA

**R2 Empleados casados**

# EMPLEADO	NOMBRE	PROV. NAC.
2	ANTONIO	MADRID
8	LUIS	SEVILLA
10	LUIS	BURGOS
13	RAQUEL	TOLEDO

**Figura 2.3.2.3.1 Ejemplos de relaciones.**

Unión: La relación R3, Unión de las relaciones R1 y R2, contendrá las tuplas de R1 más las de R2, eliminando después las posibles tuplas duplicadas. Las relaciones a las

que se le aplica deben tener el mismo número de atributos, en el mismo orden y deben corresponder a iguales dominios, es decir, a iguales conjuntos de posibles valores.

El resultado de la operación es el que muestra en la fig. 2.3.2.3.2.

Intersección: Dadas dos relaciones definidas sobre el mismo conjunto de atributos, la intersección de las mismas será otra relación que contiene las tuplas comunes a las dos relaciones primitivas. El resultado se muestra en la fig. 2.3.2.3.3.

R3 Empleados andaluces o casados

# EMPLEADO	NOMBRE	PROV. NAC
2	ANTONIO	MADRID
4	JUAN	GRANADA
7	PEPE	SEVILLA
8	LUIS	SEVILLA
10	RAQUEL	TOLEDO
13	LUIS	BURGOS
15	CONCHA	CORDOBA

Figura 2.3.2.3.2 Relación Unión.

R4 Empleados andaluces y casados

# EMPLEADO	NOMBRE	PROV. NAC
8	LUIS	SEVILLA

Figura 2.3.2.3.3 Relación Intersección.

Diferencia: R5 como relación diferencia de R1 y R2, contendrá las tuplas de R1 exceptuando las que sean iguales a las contenidas en R2 (fig. 2.3.2.3.4).

R5 Empleados andaluces no casados

# EMPLEADO	NOMBRE	PROV. NAC.
4	JUAN	GRANADA
7	PEPE	SEVILLA
15	CONCHA	CORDOBA

Figura 2.3.2.3.4 Relación Diferencia.

Producto cartesiano: Si definimos la relación R6 de coches de los empleados podemos ver el ejemplo del producto cartesiano de R5 y R6 (fig. 2.3.2.3.5 y 6).

R6 Coches de empleados

MATRICULA	MARCA
M -9978-FF	SEAT
SE-6655-AF	FORD

Figura 2.3.2.3.5 Relación de coches de empleados.

Producto cartesiano

# EMPLEADO	NOMBRE	PROV. NAC.	MATRICULA	MARCA
4	JUAN	GRANADA	M -9978-FF	SEAT
4	JUAN	GRANADA	SE-6655-AF	FORD
7	PEPE	SEVILLA	M -9978-FF	SEAT
7	PEPE	SEVILLA	SE-6655-AF	FORD
15	CONCHA	CORDOBA	M -9978-FF	SEAT
15	CONCHA	CORDOBA	SE-6655-AF	FORD

Figura 2.3.2.3.6 Producto Cartesiano.

Proyección: Con este operador aplicado sobre una relación conseguimos seleccionar determinados atributos, formando una nueva relación (fig. 2.3.2.3.7).

R7 Provincias donde han nacido empleados

PROV. NAC.
GRANADA
SEVILLA
CORDOBA

Figura. 2.3.2.3.7 Proyección.

Selección: Si aplicamos este operador a una relación obtendremos otras cuyas tuplas son aquellas de la relación original que cumplen unas condiciones establecidas (fig. 2.3.2.3.8).

R8 Empleados nacidos en Sevilla

# EMPLEADO	NOMBRE	PROV. NAC.
7	PEPE	SEVILLA
8	LUIS	SEVILLA

Figura 2.3.2.3.8 Selección en una relación.

Combinación (Join): Si definimos las relaciones siguientes podemos ver el ejemplo de la combinación (fig. 2.3.2.3.9).

# EMPLEADO	NOMBRE	PROV. NAC.
2	ANTONIO	MADRID
4	JUAN	GRANADA
7	PEPE	SEVILLA
8	LUIS	SEVILLA

Continuación...

MATRICULA	MARCA	PROV. MAT
M -9978-FF	SEAT	MADRID
SE-6655-AF	FORD	SEVILLA

Figura 2.3.2.3.9 Ejemplos de relaciones.

La combinación o Join de estas dos relaciones podría ser la siguiente (fig. 2.3.2.3.10). En ella se ha requerido la condición de igualdad entre los valores de las provincias de nacimiento de los empleados y las de matriculación de los coches .

R9 Empleados y coches nacidos y matriculados en la misma provincia

# EMPLEADO	NOMBRE	PROV. NAC.	MATRICULA	MARCA	PROV. MAT.
2	ANTONIO	MADRID	M -9978-FF	SEAT	MADRID
7	PEPE	SEVILLA	SE-6655-AF	FORD	SEVILLA
8	LUIS	SEVILLA	M -9978-FF	SEAT	SEVILLA

Figura 2.3.2.3.10 Combinación entre relaciones.

Operaciones booleanas con valores nulos. A continuación se observan los cuadros de resultados de los operadores lógicos AND, OR, y NOT cuando interviene como operando el valor nulo, es decir, un valor desconocido o no aplicable (fig. 2.3.2.3.11).

En los cuadros se utilizan los siguientes símbolos: V - Verdadero, F - Falso, ? - Valor nulo.

AND	V	?	F
V	V	?	F
?	?	?	F
F	F	F	F

OR	V	?	F
V	V	V	V
?	V	?	?
F	V	?	F

Continuación...

NOT	V
V	F
?	?
F	V

Figura. 2.3.2.3.11 Operadores lógico con valores nulos.

### 2.3.3 Diagrama Entidad - Relación

A la hora de representar mediante estructuras de datos el mundo real, es decir, cuando se realiza el diseño del modelo lógico de datos, podemos optar por cualquier tipo de base de datos: en red, jerárquica o relacional. No obstante, encontramos bastantes problemas ya que estamos pasando de un mundo real a estructuras físicas de datos con sus limitaciones y restricciones. En estas habrá que tener en cuenta accesos a los datos y cuestiones de eficiencia por lo que el mundo real se ve deformado y difícil de reconocer en el modelo lógico de datos que obtenemos como resultado del diseño.

Por otro lado, hay relaciones en el mundo real que no son contempladas, al menos directamente, por los sistemas de bases de datos. Tal es el caso de una relación **N a M** entre dos entidades. Por ejemplo la relación entre profesor y alumno: un profesor puede tener varios alumnos y un alumno varios profesores.

El modelo ENTIDAD/RELACION (del inglés ENTITY/RELATIONSHIP) surge con la idea de separar el diseño conceptual de datos del diseño físico, es decir, con el fin de plasmar en primer lugar, mediante técnicas de diagramación, el mundo real, con sus entidades o agrupaciones de datos con significado dentro del sistema y las relaciones entre ellas y después, en un segundo paso, y una vez elegida la base de datos sobre la que vamos a implementar el modelo, pasar ya al diseño físico del mismo.

No hay que confundir en este modelo el significado de la palabra RELACION con el que tenía en el modelo relacional. Si en éste último RELACION era sinónimo de

agrupación de datos, en el modelo ENTIDAD/RELACION su sentido es reflejar la interrelación o asociación entre dos o más entidades del sistema. Por ello algunos autores prefieren llamar a este modelo ENTIDAD/INTERRELACION, a fin de diferenciar estos dos significados.

El modelo conceptual de datos así conseguido es, por lo general, más estable que el modelo físico y además es independiente del sistema de base de datos elegido.

Por otro lado, a la hora de la confirmación por el usuario de la representación del modelo de datos, el modelo entidad/relación es más fácil de comprender ya que se acerca más a la realidad.

### Técnicas de representación

Básicamente en un diagrama del modelo E/R se deben representar los conceptos siguientes:

Entidad. La componen una serie de datos que, agrupados, tienen un cierto significado para la empresa. Se representan por medio de un rectángulo en cuyo interior figura el nombre de la entidad. Cada entidad está compuesta por una serie de atributos que se representarán por una serie de elipses enlazados con la entidad, esto se puede observar en la fig. 2.3.3.1.

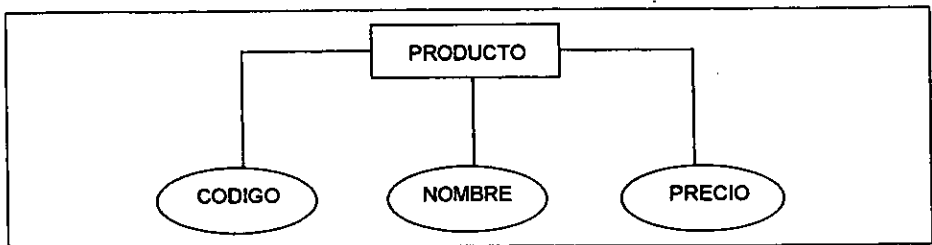


Figura 2.3.3.1 Entidad y sus atributos.

Relaciones o Asociaciones. Constituyen el enlace entre las diferentes entidades del sistema. Se representan mediante líneas que unen las entidades o a través de un

rombo en cuyo interior se pone el nombre de la relación. Así, entre un PRODUCTO y un PROVEEDOR existirá la relación "ES SERVIDO POR", ver la fig. 2.3.3.2.

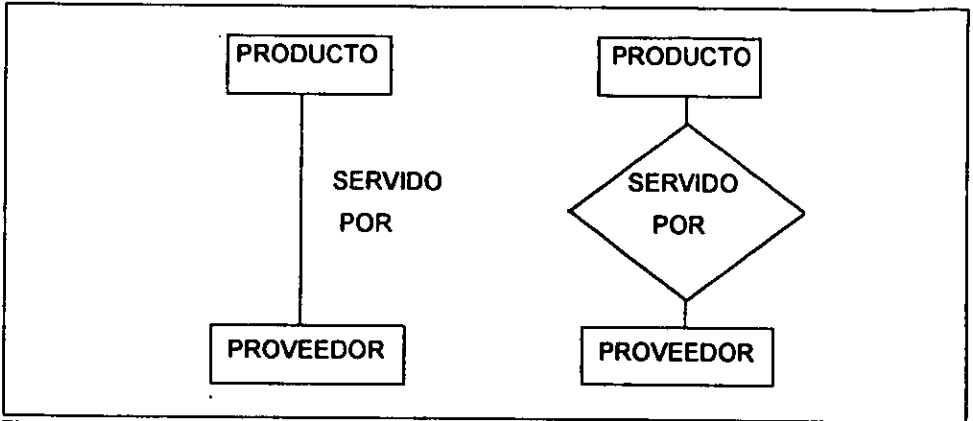


Figura 2.3.3.2. Relación entre entidades.

A continuación se mencionan los tipos de relaciones:

- UNO A UNO (1:1). Una entidad en A está relacionada únicamente con una entidad de B y una entidad de B con sólo una de A.
- UNO A MUCHOS (1:M). Una entidad de A está relacionada con cualquier número de entidades de B, pero una entidad de B sólo está relacionada con una de A.
- MUCHOS A UNO (M:1). Una entidad de A está relacionada únicamente con una entidad de B, pero una entidad de B está relacionada con cualquier número de entidades en A.
- MUCHOS A MUCHOS (M:M). Una entidad en A está asociada con cualquier número de entidades en B, y una entidad de B está vinculada con cualquier número de entidades de A.

La figura 2.3.3.3 muestra los tipos de relaciones mencionadas.



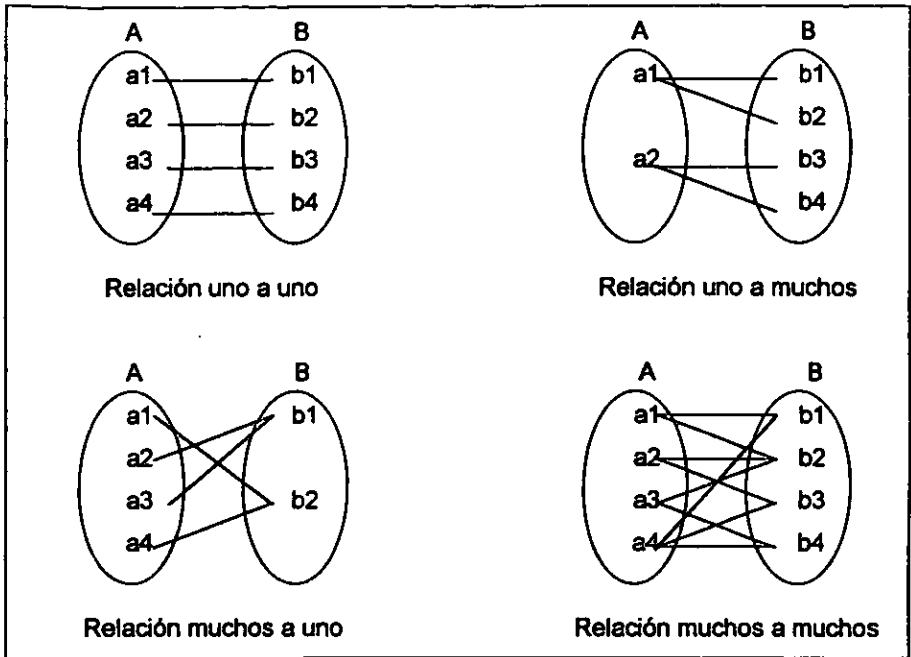


Figura 2.3.3.3. Tipos de relaciones.

## 2.4 Metodologías de Diseño

Si se analizan, las causas de este hecho se observa que hay varias razones por las que, en la década de los ochenta, algunas técnicas vuelven a estar de moda y, por tanto, a utilizarse.

Evidentemente, la revolución informática de estos años viene marcada por el uso generalizado de los ordenadores personales basándose en un costo cada vez menor de las máquinas que ha hecho accesible la informática a casi todos: desde el estudiante al curioso, desde el jefe de proyecto al pequeño empresario, desde la secretaria al administrativo.

Probablemente tardará asimilar el cambio de hábitos tan grande llevado a cabo en estos años, quizá comparable a la revolución industrial del siglo pasado.

Esta revolución, al fin, ha llegado también a los informáticos que, empeñados en automatizar el mundo y sus procesos administrativos, en organizar empresas antes de automatizar, se olvidan permanentemente de trabajar de forma organizada y, por supuesto, de automatizar las fases de producción de software.

Ha tenido que ser la pura necesidad la que ha empujado al informático a dedicar recursos a la investigación de este terreno. La carencia de personal con experiencia ante las peticiones cada vez mayores de la sociedad, ha hecho que se contemple la producción de aplicaciones de la misma forma que cualquier otro proceso de producción industrial.

Tradicionalmente el informático ha sido considerado como un tipo extraño de artista. Era capaz, en ciertos momentos críticos, con una pincelada de ingenio, de arreglar una situación de desastre. En un momento, por obra y gracia de esos superhombres, las cosas empezaban a funcionar.

Dicha imagen, sin embargo, está condenada a desaparecer. Este tipo de profesional resulta cada vez menos rentable a las empresas, sabedoras de la movilidad del personal informático debido a la gran demanda del mercado.

Los grandes avances en el mundo de las PC'S han hecho posible que se desarrollen herramientas destinadas a una producción más automatizada de aplicaciones informáticas.

La utilización de las técnicas y teorías que hasta ahora parecían difíciles de implantar en una empresa informática, por el costo añadido que suponían y por el retraso en los plazos de entrega de los proyectos, empieza a ser posible a medida que dichas

técnicas son soportadas por estaciones gráficas de no muy alto costo que hacen más fácil su uso.

Así se debe hablar del CASE, la ingeniería del software, como el terreno con mayores innovaciones en los últimos dos años, y de las metodologías soportadas por herramientas CASE como el entorno de trabajo de los informáticos en los próximos tiempos.

La eliminación de los inconvenientes que tenían las metodologías, la facilidad de uso dentro de un ambiente de producción integrado e informatizado, han hecho y van a hacer estos años que la metodología como MERISE (1977) vuelva a ser de actualidad. Técnicas de representación como las de YOURDON, PRESSMAN, etc., pueden emplearse de modo sencillo desde una PC. La modelización de datos y estructuras según diversas técnicas empieza a ser posible sin gran esfuerzo, y el mantenimiento de un diccionario de datos y de la generación de la documentación de una aplicación dejará de ser la cruz con la que siempre han cargado los informáticos.

Este subtema pretende, en un momento de gran actividad en el entorno de producción, aportar información sobre las metodologías y técnicas más empleadas en cada una de las fases del desarrollo de aplicaciones. Pretende ser una guía en un momento de gran difusión de estos temas en una etapa en la que muchos estarán planteando la necesidad de implantar en su empresa ésta forma de trabajo.

Existen diversas metodologías de desarrollo de software, por lo que es necesario analizarlas y evaluarlas, para seleccionar la que mejor se adapte a las necesidades planteadas. Normalmente se selecciona una metodología estándar con las variaciones y modificaciones requeridas. En el presente trabajo se analizarán los conjuntos metodológicos de Yourdon, Pressman y Merise, los cuales presentan características comunes en los aspectos generales, aunque difieren en la profundidad con que abordan las diversas etapas del desarrollo de sistemas de información.

### 2.4.1 Metodología de Desarrollo

Generalmente se entiende por metodología, refiriéndose a cualquier ámbito o trabajo, a un sistema ordenado de proceder para la obtención de un fin.

Si se considera el entorno informático, es decir, la producción o desarrollo de sistemas informáticos, es evidente que el uso de una metodología en este proceso aporta algunas ventajas que hacen aconsejable su uso.

El problema en el momento actual es elegir una de las metodologías disponibles en el mercado con el suficiente conocimiento de causa.

Una primera observación, permite una separación entre metodologías PÚBLICAS, o sea, aquellas cuya utilización no lleva al usuario al pago de ninguna cantidad a las entidades que la crearon y PRIVADAS, aquellas desarrolladas por entidades del mismo tipo y que, por tanto, basan sus beneficios en el cobro de licencias de uso como si de cualquier otro producto software se tratara.

Dentro de las metodologías públicas, actualmente se pueden distinguir tres corrientes actualmente:

- ☐ La francesa, que dio como fruto la metodología MERISE, potenciada por la administración francesa a partir del año 1977.
- ☐ La de PRESSMAN.
- ☐ La americana, basada en las teorías de EDWARD YOURDON, y que tiene algunas variantes aportadas por otros autores como DEMARCO, GANE Y JAMES MARTIN.

El resto de las metodologías existentes, tanto públicas como privadas, deben considerarse como adaptaciones, más o menos mejoradas, de las citadas anteriormente.

También hay que considerar que los objetivos que persiguen todas ellas son parecidos, por tanto, es evidente que para hacer un desarrollo estructurado y ordenado de una aplicación, los caminos seguidos, o sea, las fases y su cronología, no pueden ser muy diferentes.

Uno de los posibles enfoques de un libro sobre metodología, sería definir una propia y divulgarla como la metodología ideal.

Pero el futuro del uso de las metodologías basa su soporte en herramientas informatizadas, de las englobadas actualmente bajo el nombre de CASE. Es por esto que la tendencia general indica que, con menos frecuencia, cada vez menos, cada cual haga la guerra por su cuenta. Afortunadamente, van pasando los tiempos en que cada casa fabricante seguía su línea, y se imponen líneas comunes a la hora de buscar estándares en casi todos los terrenos.

A nivel europeo, todos trabajan en la elaboración del EUROMETODO, proyecto impulsado por la comunidad Europea en un intento de unificar las tendencias actuales sobre metodología.

No parece buen momento, por tanto, para aportar nuevas vías y por el contrario parece conveniente, en un momento de explosión de las herramientas CASE en el mercado, clarificar ideas y documentar técnicas preparándonos para el nuevo entorno de desarrollo en los 90.

#### **2.4.2 Ciclo de Vida de un Sistema de Información**

Por ciclo de vida de un sistema de información se entiende el conjunto de fases por las que pasa a lo largo del tiempo, desde la fase de estudio y concepción, hasta la de realización, explotación y mantenimiento.

Las fases o etapas de este ciclo de vida son definidas por las diversas metodologías y hay diferentes variantes de las mismas.

En general, todas las metodologías coinciden en una secuencia lógica de fases, y en los párrafos siguientes veremos a detalle cada una de las mismas. No obstante, podemos citar el enfoque que tradicionalmente se ha dado al desarrollo de un sistema (fig. 2.4.2.1).



Fig. 2.4.2.1 Fases tradicionales de desarrollo

A continuación se muestra el esquema del ciclo de vida propuesto por algunas metodologías actuales.

### **2.4.3 Yourdon**

Yourdon llama a su metodología de desarrollo de sistemas "ciclo de vida estructurado del proyecto", y propone los siguientes pasos para llevar a cabo satisfactoriamente cualquier producto de software (fig. 2.4.3.1).

#### **✓ La encuesta.**

Este punto se conoce también como el estudio de factibilidad o como estudio inicial del negocio. Comúnmente comienza cuando el usuario solicita que una o más partes de los procesos que realiza se automaticen. Entre los objetivos de la encuesta se tienen los siguientes:

1. Identificar a los usuarios responsables y crear un entorno inicial del sistema.
2. Establecer metas y objetivos para el nuevo sistema.
3. Determinar si es factible automatizar el sistema y, de ser así, sugerir escenarios aceptables.
4. Preparar el esquema que se usará para guiar el resto del proyecto.

#### **✓ Análisis estructurado.**

El propósito principal de la actividad de análisis es transformar las políticas del usuario y el esquema del proyecto, en una especificación estructurada. Esto implica modelar el ambiente del usuario con diagramas de flujo de datos, diagramas entidad-relación, etc.

Adicionalmente al modelo del sistema que describe los requerimientos del usuario, generalmente se prepara un conjunto de presupuestos y cálculos de costos y beneficios más precisos y detallados al final de la actividad de análisis.

✓ Diseño estructurado.

La actividad de diseño se dedica a la creación de una jerarquía apropiada de módulos de programas y de interfaces entre ellos para implementar la especificación creada durante el análisis. Además la actividad de diseño se ocupa de la transformación de modelos de datos entidad-relación en un diseño de base de datos.

✓ Codificación.

Durante esta actividad se desarrolla la programación y la integración de módulos. En este punto se consideran herramientas tales como la programación estructurada. La implantación se propone que sea descendente, es decir, se elaboren los esqueletos de los módulos principales, y posteriormente se realice la programación de los submódulos que componen cada módulo principal.

✓ Elaboración de pruebas de aceptación.

La especificación estructurada debe contener toda la información necesaria para definir un sistema que sea aceptable desde el punto de vista del usuario. Por eso, una vez generada la especificación puede realizarse la actividad de producir un conjunto de casos de prueba de aceptación a partir de la especificación estructurada.

✓ Garantía de calidad.

Se requiere de alguna actividad que verifique que el sistema tenga un nivel apropiado de calidad, es decir, que cumpla no sólo con los requisitos del usuario si no que adicionalmente cuente con buena documentación, tiempo de respuesta aceptable, codificación entendible, etc. Para que los puntos anteriores sean posibles de alcanzar, es importante llevar a cabo actividades de garantía de calidad en cada una de las fases de la metodología puesto que con ello se asegura en gran medida la calidad final del producto.



✓ **Elaboración de manuales de usuario.**

Una de las actividades importantes a realizar es la generación de una descripción formal de las partes del sistema que se harán en forma manual, lo mismo que la descripción de cómo interactuarán los usuarios con la parte automatizada del nuevo sistema.

✓ **Conversión de base de datos.**

Esta actividad requiere como entrada la base de datos actual del usuario, al igual que la especificación del diseño. El diseño de la base de datos se establece a partir del diagrama entidad-relación. Durante la implementación física de la base de datos se pueden utilizar las ventajas otorgadas por cada manejador DBMS, por ejemplo creación de roles de usuarios, establecimiento de restricciones, indicaciones de cómo utilizar el espacio en disco, etc.

✓ **Instalación.**

La instalación representa el último paso de la metodología de Yourdon. Las entradas de la instalación son el manual de usuario, la base de datos convertida y el sistema aceptado. La puesta en marcha del nuevo sistema puede acompañarse de ejecuciones en paralelo del sistema anterior (en caso de que exista) en tanto que se comprueba que funciona adecuadamente en un medio real.

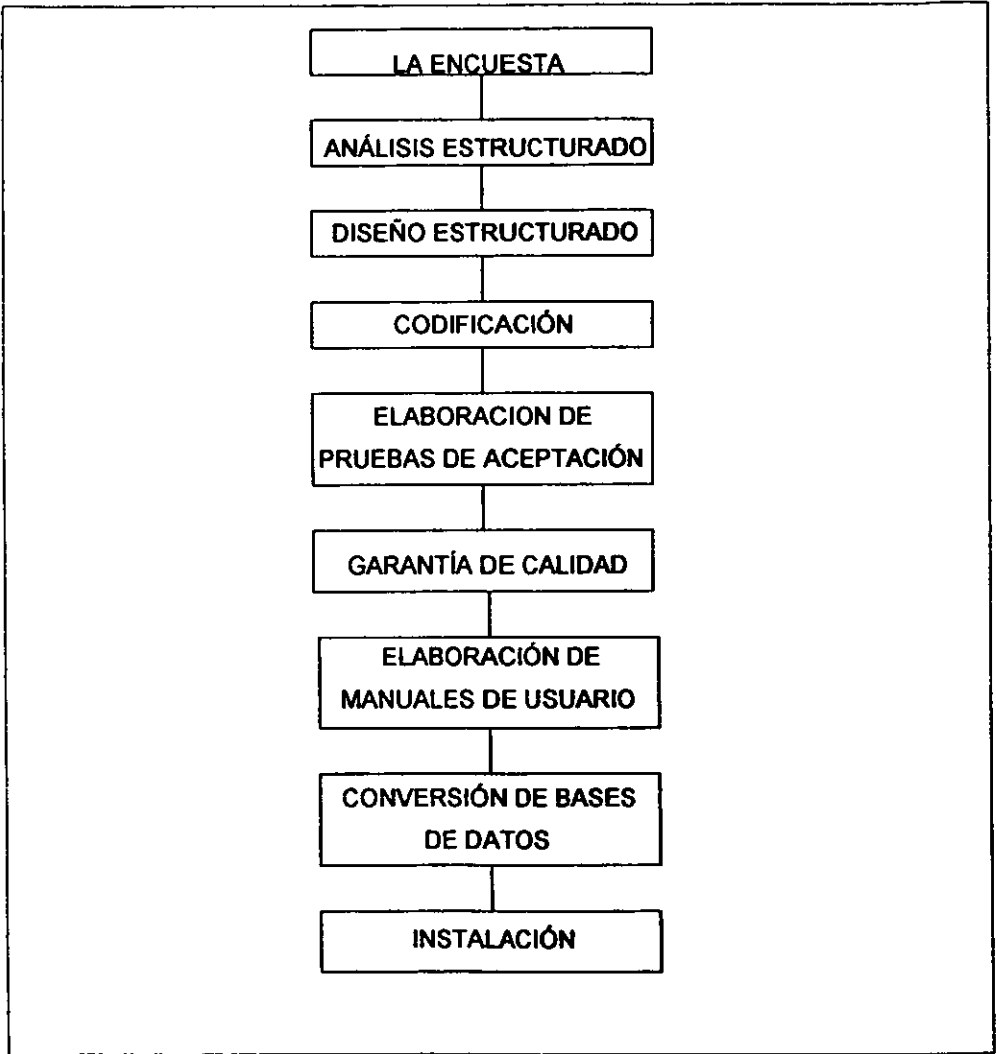


Figura 2.4.3.1 Fases de desarrollo según Yourdon.

#### **2.4.4 Pressman**

Pressman sugiere las siguientes etapas para el desarrollo adecuado de software de sistemas (fig. 2.4.4.1).

✓ **Definición.**

1. Análisis del sistema.
2. Planificación del Proyecto de Software.
3. Análisis de los requerimientos.

✓ **Desarrollo.**

1. Diseño del software.
2. Codificación.
3. Prueba del software.

✓ **Mantenimiento.**

1. Corrección.
2. Adaptación.
3. Aumento.
4. Prevención.

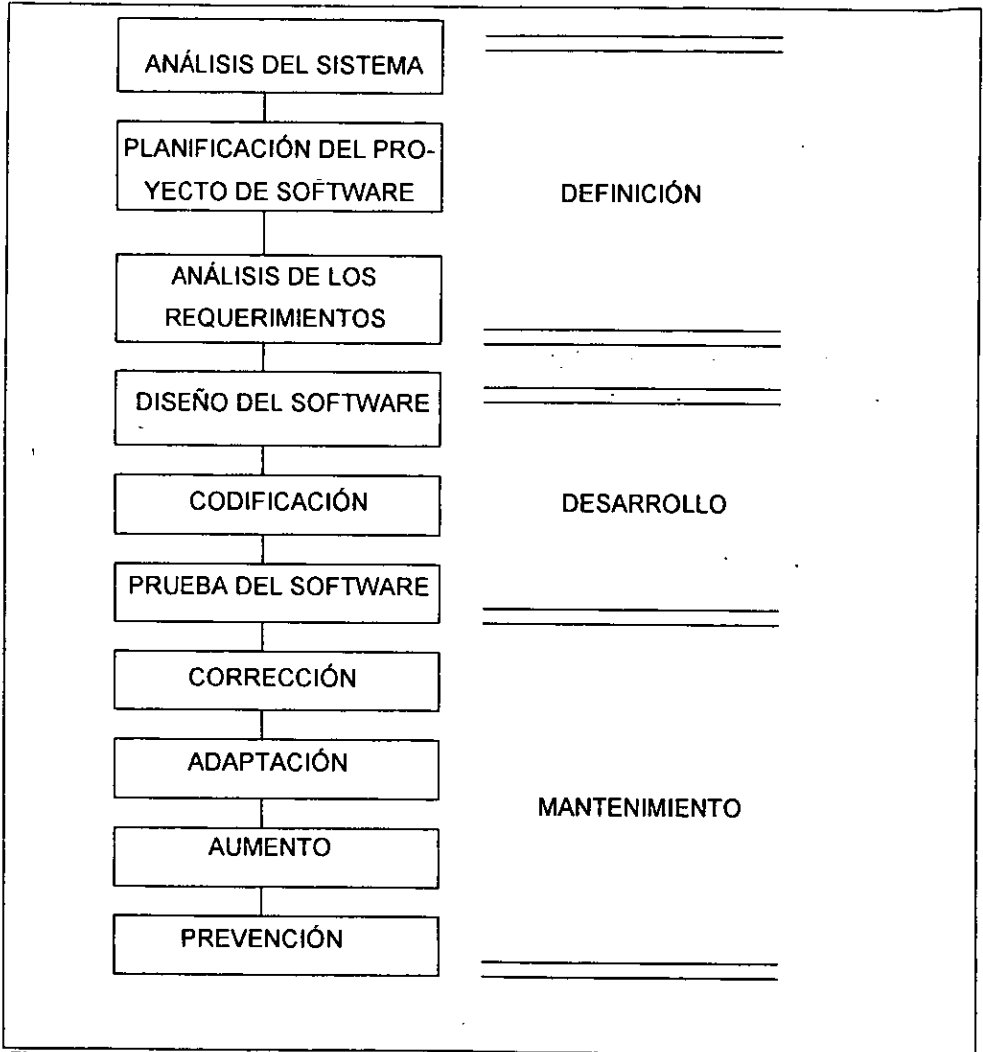


Figura 2.4.4.1 Fases de desarrollo según Pressman.

Para la etapa de definición, Pressman menciona que es necesario desarrollar un enunciado definitivo del problema por resolver. También es necesario incluir una descripción de la situación actual, restricciones del problema y las metas que se lograrán. Además de justificar que es necesaria una estrategia computacional en la solución del problema. Se deben identificar las funciones por realizar, las restricciones,

el equipo electrónico con que se cuenta, el personal, etc. Es necesario determinar los objetivos y requisitos en el nivel del sistema para el proceso de desarrollo y los productos finales y, por último, establecer criterios de alto nivel para la aceptación del sistema. Según Pressman se analizan los requerimientos para especificar las necesidades funcionales y de desempeño del sistema, así como para la detección de excepciones.

Pressman proporciona la siguiente lista de especificaciones de los requisitos para la producción de software:

1. Resumen y panorama del producto.
2. Ambientes de desarrollo, operación y mantenimiento.
3. Interfaces externas y flujo de datos.
4. Requisitos de operación.
5. Manejo de excepciones.
6. Subconjuntos iniciales y prioridades instrumentales.
7. Modificaciones y mejoras previsibles.
8. Criterios de aceptación.
9. Guías y sugerencias de diseño.
10. Índice cruzado.
11. Glosario de términos.

El diseño del software requiere de concebir, planear y especificar las características del producto final, las cuales incluyen la definición de las pantallas de presentación, de los formatos de reportes, la definición de las entradas y salidas de datos, entre otras cosas. Es necesario incluir la concepción, planeación y especificación de la estructura interna y de los detalles del proceso del producto, así como los detalles de los algoritmos, las estructuras de datos y planes de pruebas.

En resumen, el proceso de diseño comprende el desarrollo de una visión conceptual del sistema, el establecimiento de una estructura, la identificación de las cadenas de

datos y su almacenamiento, la descomposición de funciones de alto nivel en subfunciones, el establecimiento de las relaciones e interconexiones entre componentes, el desarrollo de la representación de datos en forma concreta y la especificación de los detalles de los algoritmos.

La fase de codificación tiene por objetivo principal escribir el código fuente y la documentación interna de modo que sea fácil de verificar y corregir posibles errores, o dar mantenimiento, se debe contar con un conjunto de requisitos de programación bien definidos y una descripción del diseño detallado.

Las pruebas del software sirven para determinar el apego a las especificaciones, incluyéndose las especificaciones de requisitos, la documentación del diseño, diversos principios generales de estilo, estándares del lenguaje de programación, estándares del proyecto, estándares organizacionales y expectativas del usuario.

Los errores ocurren cuando una parte del producto está incompleta. Los tres errores más comunes son: los *errores de requisito* que se provocan por una propuesta incorrecta de las necesidades del usuario, o bien, por falta de una especificación completa de los requisitos funcionales y de desempeño, entre otras cosas. *Los errores de diseño*, los cuales se producen por fallas al traducir los requisitos en estructuras de solución correctas y completas, por inconsistencias, etc. Por último *los errores de codificación*, que son los cometidos al traducir las especificaciones del diseño en código fuente.

Las actividades de mantenimiento del software implican realizar mejoras en los productos de software, adaptar productos a nuevos ambientes y corregir problemas. El mantenimiento consume en la mayoría de los casos el 50 % del presupuesto del ciclo de vida de un producto. La mejora de los productos puede dar como resultado proporcionar nuevas capacidades funcionales.

### 2.4.5 Merise

La metodología Merise surge en Francia a partir del año 1977, como un intento de definir una metodología a emplear en la Administración Pública para el desarrollo y diseño de sistemas informáticos.

Los principios generales en que se apoya Merise son:

- ✓ Desglose del desarrollo por etapas.
- ✓ Definición de los documentos estándar de cada etapa.
- ✓ Uso del modelo Entidad-Relación y sus principios para la representación de datos.
- ✓ Uso de las redes de Petri para la representación de procesos y tratamientos.
- ✓ Definición de grupos de trabajo y reparto de las responsabilidades y funciones a lo largo del desarrollo.
- ✓ Especificación del reparto de tareas y tratamientos entre los usuarios y el ordenador.
- ✓ Definición de los flujos de información entre las unidades del sistema.

La concepción del sistema para un análisis y construcción consiste en trabajar sobre dos elementos: datos y tratamientos sobre los mismos. Los primeros son la información que se maneja en el entorno del sistema y las relaciones dentro de ella. Los tratamientos reflejan los procesos que se aplican a la información y su secuencia en el tiempo para obtener los resultados deseados.

Estos dos elementos del sistema son tratados en tres niveles: conceptual, lógico y físico.

La metodología utiliza técnicas para modelar ambos elementos a todos los niveles.

Algunas técnicas utilizadas son las siguientes:

- ✓ Utilización de grupos de trabajo para el desarrollo del proyecto (comité de directores, comité de usuarios y grupo de desarrolladores).
- ✓ Representación gráfica de diagramas de flujo.
- ✓ Modelado de datos (conceptual, lógico y físico).
- ✓ Modelado de tratamientos (conceptual, organizativo y operacional).

La metodología propone que la concepción de los datos y los tratamientos pueden ser llevadas a cabo de manera paralela y con algunas interfaces entre sí; sin embargo, es muy importante tomar en cuenta el orden de los niveles de concepción del sistema. En primer lugar realizar la conceptualización de los dos elementos, después la concepción lógica y al final obtener un resultado físico.

Merise sugiere una secuencia de etapas, fases y pasos a seguir en el desarrollo de un proyecto informático como metodología para el desarrollo de proyectos y distingue cuatro etapas a lo largo de un proyecto (fig. 2.4.5.1).

- ✓ Estudio Preliminar.

**Fase 1: Recolección de Datos:**

- Recolección Inicial.
- Estudio de la Situación Actual.
- Síntesis y Crítica de la Situación Actual.

**Fase 2: Concepción de la nueva solución.**

- Objetivos a alcanzar.
- Descripción de la solución.

**Fase 3: Evaluación y plan de desarrollo.**

- Evaluación de la nueva solución.
- Plan de desarrollo.



✓ Estudio Detallado.

Fase 1: Concepción general.

Fase 2: Concepción detallada de las fases.

Realización de las especificaciones detalladas de procesos.

Fase 3: Plan de desarrollo.

✓ Realización.

Fase 1: Estudio Técnico.

Fase 2: Producción.

✓ Puesta en marcha.

Fase 1: Preparación de los recursos físicos y humanos.

Fase 3: Recepción y lanzamiento del sistema.

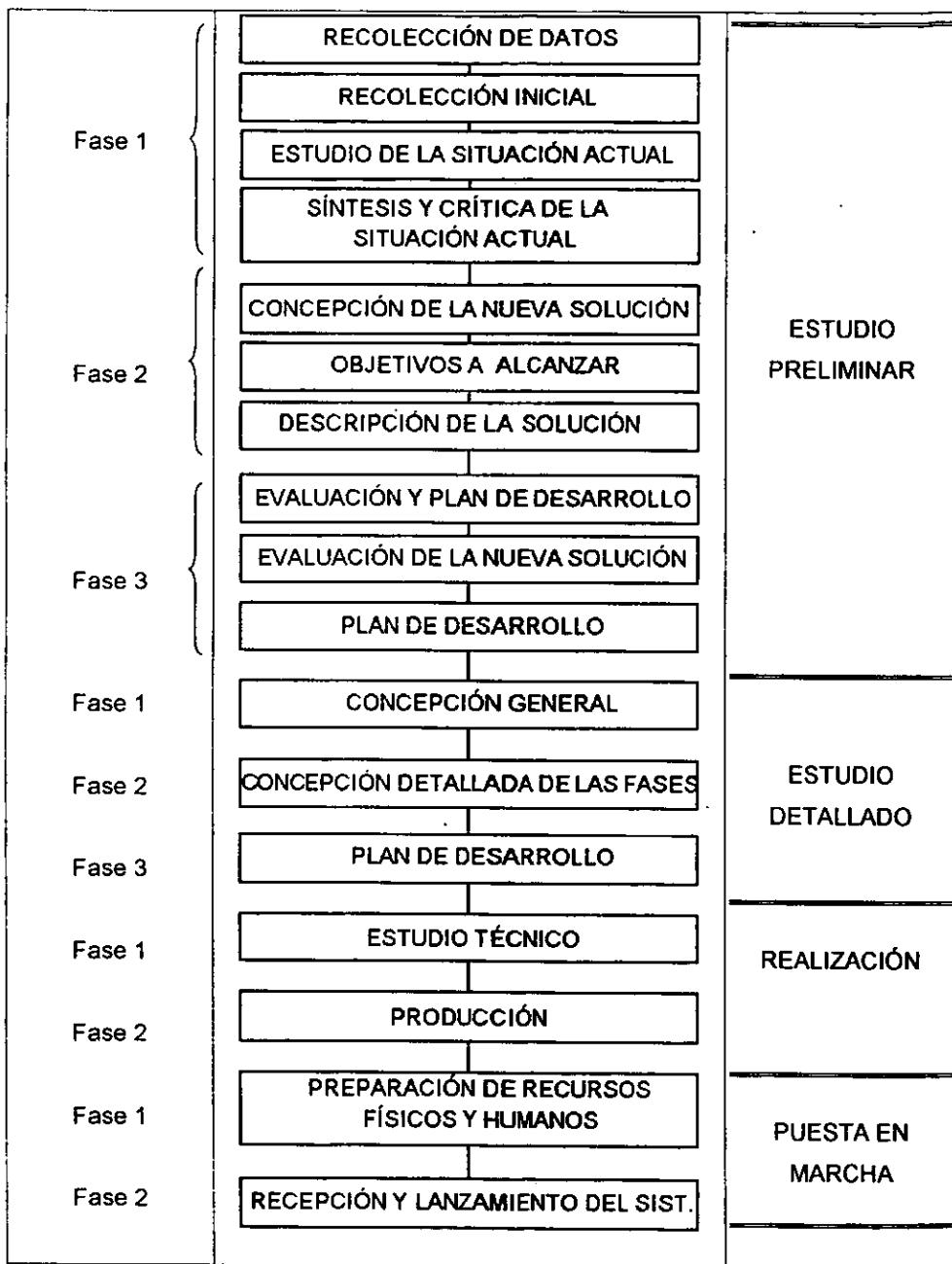


Figura 2.4.5.1 Fases de desarrollo según Merise.

El modelo Entidad-Relación es una representación gráfica con el fin de mostrar, en primer lugar mediante técnicas de diagramación del mundo real las entidades o agrupaciones de datos con significado dentro del sistema y las relaciones entre ellas.

Las redes de Petri es un método de representación muy completo ya que por cada acción-operación a realizar determina:

- ✓ Qué sucesos desencadenan la ejecución de procesos.
- ✓ Bajo qué condiciones de esos sucesos desencadenantes se ejecuta el proceso.
- ✓ En qué consiste el proceso, es decir, qué operaciones hay que realizar.
- ✓ Quién ejecuta el proceso.
- ✓ Resultados obtenidos o salidas del proceso.

Para la estructuración de las acciones a realizar se define "Proceso" como toda actividad de la empresa con sus entradas y salidas establecidas, y "Operación o Procedimiento" como una serie de acciones ejecutables sin interrupción y que se ejecutan bajo las mismas condiciones en cuanto a factores desencadenantes.

### 2.4.6 Ciclo de vida en entorno Case

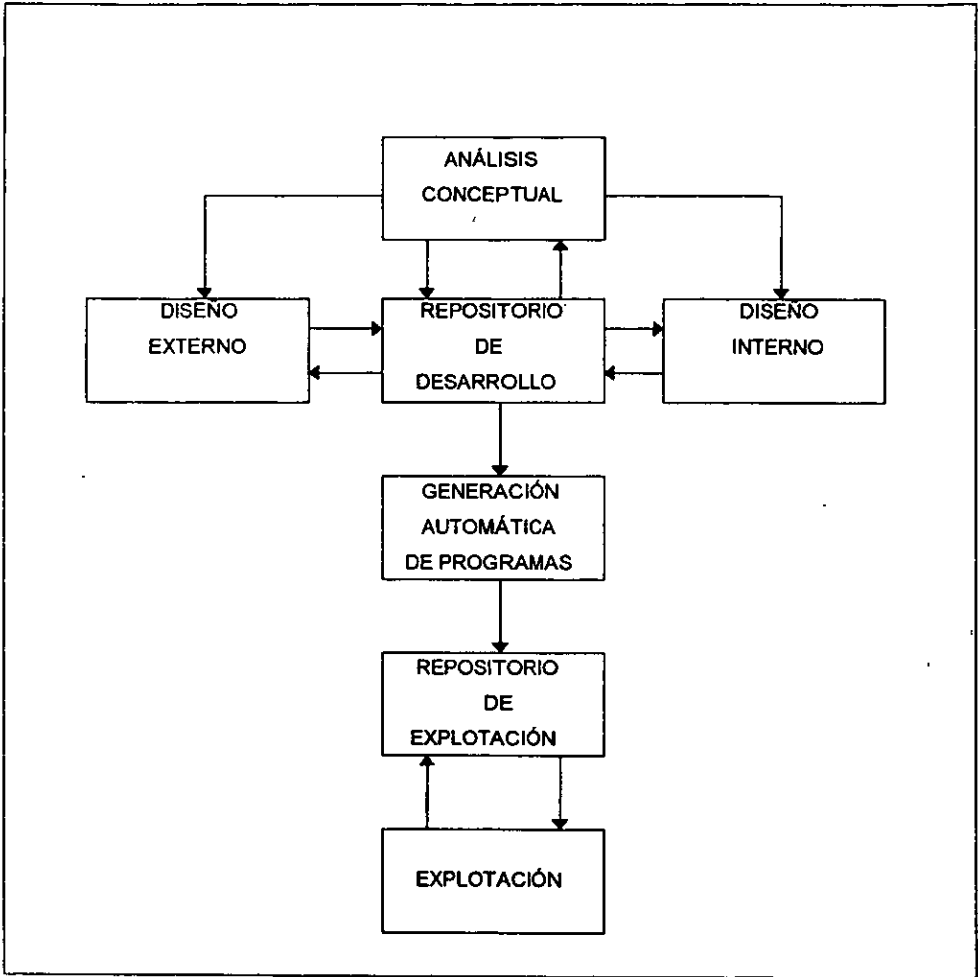


Figura 2.4.6.1 Desarrollo en un entorno CASE

A continuación se analiza el contenido que tienen algunas fases que, con carácter general, se repiten en todas las metodologías:

## 1) Estudio de Viabilidad o Estudio Preliminar:

Antes de comenzar el desarrollo de un proyecto, se realiza una toma inicial de datos y se define el marco de aplicación del sistema. En este punto se deben fijar las siguientes tareas:

- ⇒ Definición del problema y de necesidades.
- ⇒ A quién afecta.
- ⇒ Estudio de la situación actual.
- ⇒ Crítica de esta situación y propuesta de soluciones.
- ⇒ Viabilidad del proyecto.

Para llevar a cabo este primer contacto y para el estudio de la situación actual, la propuesta general es la realización de una serie de entrevistas con directivos y usuarios responsables a los que afectará el proyecto. Este primer contacto no debe pasar de eso, una recopilación de la información suficiente como para poder estimar la envergadura del proyecto, la posibilidad de acometerlo y fijar unos objetivos muy generales a cubrir con el desarrollo del mismo.

## 2) Análisis Lógico o Análisis Funcional:

En esta segunda etapa se hará una toma de datos exhaustiva, determinando los siguientes aspectos:

- ☒ Especificación de requerimientos.
- ☒ Funcionalidad del sistema.
- ☒ Selección de opción técnica más apropiada.

En esta fase se podrán utilizar como ayuda técnicas, de elaboración de prototipos del sistema, a manera de paso previo para la producción del sistema y simulación de funcionamiento en las reuniones con usuarios.

### 3) Diseño Físico y Realización:

Es una etapa en la que con mayor nivel de detalle se estructuran y definen datos y procesos. Es la de producción real del sistema. Hay variantes sobre las tareas a realizar y el orden en que deben hacerse, pero lo que está claro es que al final de esta fase se debe haber producido:

- ◆ Documentación sobre el modelo de datos.
- ◆ Especificaciones de procesos.
- ◆ Programas.
- ◆ Documentación interna del sistema.
- ◆ Documentación de usuario.
- ◆ Pruebas del sistema.

### 4) Puesta en Marcha o Implantación:

El sistema debe estar ya acabado y aprobado, y en esta fase deben acometerse posibles conversiones de sistemas antiguos al nuevo y dar el suficiente apoyo al usuario para el arranque del proyecto.

### 5) Explotación y Mantenimiento:

Una vez pasada la fase de implantación, el sistema sigue vivo y genera, por un lado, trabajos de explotación, para lo que se debe de disponer de la infraestructura necesaria y, por otra parte, peticiones de mejoras o ampliaciones del sistema primitivo que se irán incorporando al sistema.

El ciclo de vida del sistema acabaría cuando, por diversas razones, como son la obsolescencia del sistema o un cambio de estrategia en la empresa, el sistema es abandonando o sustituido por otro.

## 2.4.7 Elección de metodología

Las tres metodologías expuestas en los apartados anteriores mencionan facetas similares para el desarrollo de sistemas, sintetizando podemos distinguir las fases generales como sigue:

- ✓ Análisis.
- ✓ Diseño y Codificación.
- ✓ Pruebas e Instalación.

Sin embargo, cada metodología resalta ciertos aspectos, por ejemplo, Pressman da gran importancia a la fase de mantenimiento y corrección del software; en tanto que Yourdon plantea que si en cada fase de la metodología se aplican reglas de calidad (las cuales consisten en una revisión detallada de los documentos generados en cada fase y la congruencia de los mismos), el sistema calificará de acuerdo a los requerimientos iniciales, y por lo tanto la fase de mantenimiento y corrección estará limitada.

Merise menciona la necesidad de tener grupos organizados de trabajo para desarrollar proyectos y de las tres metodologías analizadas es el único autor que detalla este paso como parte esencial de la misma, en tanto que los otros dos autores no ponen énfasis en este aspecto, aunque implícitamente dejan ver que el desarrollo de sistemas se lleva a cabo en grupos de trabajo, sin embargo, no exponen reglas claras de cómo formar y administrar éstos.

Un aspecto importante de la implantación de sistemas corresponde a la capacitación de los usuarios y la entrega de manuales de usuario. Yourdon marca como parte de su metodología la elaboración de manuales de usuario, no obstante, en la fase de instalación no habla explícitamente de la capacitación a los usuarios finales, aunque se

entiende que si se quiere implantar un sistema que tenga ciertos parámetros de calidad se hará necesaria la capacitación de los usuarios. En tanto, Merise menciona en su metodología la capacitación del usuario, sin detallar los aspectos del manual de usuario, sin embargo, este documento podría ser cubierto en la fase de capacitación.

En cuanto a las herramientas de análisis y diseño, Merise se enfoca en las redes de Petri para conceptualizar los procesos del sistema y utiliza el diagrama Entidad-Relación para mostrar los datos. Queda confuso el saber qué herramientas maneja Merise para el diseño, así también no profundiza en aspectos y técnicas de codificación.

Pressman enfoca el análisis y diseño de sistemas en un concepto llamado jerarquía de datos, manejado inicialmente por Jackson y Warnier. Los datos son definidos dentro de los procesos y esto vuelve difícil su manejo.

Los diagramas de flujos de datos son la herramienta con la cual Yourdon nos muestra los procesos y flujos entre entidades pertenecientes al sistema. Al igual que Merise maneja el diagrama Entidad-Relación para modelar los datos del sistema, y agrega un concepto muy importante el diccionario de datos. Bajo la metodología de Yourdon las herramientas de análisis y diseño están íntimamente ligadas y deben estar "balanceadas", es decir, todo flujo de datos debe estar especificado en el diccionario de datos, todo almacén de datos debe estar plasmado en el diagrama Entidad-Relación, etc.

Para efecto del presente trabajo, los autores consideran que la metodología de Yourdon ofrece las siguientes ventajas sobre las otras dos metodologías:

La metodología se presenta:

- ✓ Menos compleja y, por lo tanto, más clara.
- ✓ Manejo de herramientas de análisis y diseño, fáciles de entender por parte del usuario.



- ✓ Permite pasar fácilmente a la codificación.
- ✓ Documenta coherentemente la aplicación y hace fácil el mantenimiento.
- ✓ Garantiza la calidad en cada etapa del desarrollo.

Por las ventajas anteriormente expuestas, se concluye que Yourdon ofrece el mejor conjunto metodológico de desarrollo de sistemas de información y que de acuerdo a nuestras necesidades (sistema con un enfoque administrativo) nos proporciona las mejores herramientas para llevar a cabo exitosamente el sistema que queremos generar. Lo anterior no excluye por supuesto, que consideren ideas de los otros dos autores, así pues, el resto del documento estará basado en la metodología de Yourdon complementable con aspectos de relevancia de las otras dos.

## 2.5 Análisis del sistema

En esta fase se realiza un estudio general del sistema, en donde se define la situación actual, la situación que se pretende alcanzar, las necesidades o requerimientos de la empresa, y se plantean alternativas para su solución.

### 2.5.1 Situación actual y situación que se pretende alcanzar

El procedimiento que se lleva a cabo actualmente para la valuación de predios es el que se muestra en la Fig. 2.5.1.1.



**Figura 2.5.1.1 Esquema actual del proceso de avalúos.**

El valuador verifica en campo la homogeneidad de la zona en estudio, y en su caso subdivide las áreas, procurando tomar en cuenta el máximo posible de características de homogeneidad, como pueden ser: por sus habitantes, por sus construcciones preponderantes, en zonas comerciales, etc.

Enseguida, el valuador registra todos los factores físicos generales de la colonia, como pueden ser: facilidad de acceso a la zona, tipo de uso de suelo, servicios urbanos, nivel social, contaminación, etc.

Teniendo ya registrados los factores físicos generales de la colonia, procede a registrar los factores físicos individuales del predio, entre los que se pueden encontrar: el factor de superficie, de forma, de frente, de ubicación, de zona, de topografía, etc.

Hasta este punto el valuador termina de obtener y registrar todos los datos necesarios de la investigación de campo para proceder a realizar el cálculo del predio. Los datos que se han obtenido de la investigación de campo, normalmente son anotados en hojas con un formato ya establecido para facilitar su manejo y tener una mejor identificación de cada uno.

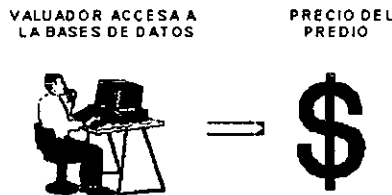
Posteriormente, el valuador procede a calcular el Valor Unitario Medio de la colonia cuyos muestreos se obtendrán de: avalúos directos realizados por otros valuadores, ofertas de terrenos de compra o venta a través de anuncios al público, resultados de registros estadísticos y proyecciones.

Una vez que ya se obtuvieron los datos necesarios se procede a realizar un análisis de éstos para verificar que sean lo mas reales y recientes posible. Ya habiendo realizado el análisis se realiza el cálculo del Valor Unitario del predio, En caso de no presentar factores físicos particulares en el terreno, el Valor Unitario del predio se obtendrá multiplicando el Valor Unitario Medio de la colonia por la superficie total del terreno.

Cuando se tienen factores físicos que afectan el valor del terreno, el Valor Unitario Medio de la colonia deberá multiplicarse por un factor resultante y por la superficie total del terreno. El factor resultante que se mencionó anteriormente se obtiene de la operación matemática de los factores físicos del predio.

Todas y cada una de las actividades que se mencionó anteriormente se realizan cada vez que se solicita un avalúo, esto como se puede observar requiere un lapso de tiempo grande.

Con el desarrollo del sistema se desea que el proceso de valuación funcione como se muestra en la Fig. 2.5.1.2.



**Figura 2.5.1.2** Nuevo esquema del proceso de avalúos.

El valuator verifica en campo las características del predio y registra todos los factores físicos del mismo. Enseguida con la utilización del sistema se busca el Valor Unitario Medio de la colonia donde se encuentra ubicado el predio en cuestión. Esto es debido a que el sistema contará con una base de datos en la cual ya se tendrán registrados y actualizados para cada colonia que existe en el D.F., las características de homogeneidad, los avalúos anteriores, las ofertas de compra y venta de terrenos, los registros estadísticos y los factores físicos generales.

Posteriormente, con los factores físicos generales de la colonia y los factores físicos individuales del predio, estos se le proporcionan al sistema para que realice todas las operaciones necesarias y obtenga el factor resultante.

Una vez que el sistema presenta el factor resultante, el Valor Unitario Medio y la superficie total del predio, realiza las operaciones para obtener el Valor Unitario del predio y un informe detallado de todas las características que tiene.

En caso de que se llegara a presentar una situación donde una determinada colonia no cuente con información en la base de datos del sistema, o no existieran registros de avalúos anteriores, no haya ofertas de compra o venta de terrenos, el sistema deberá tomar un lote tipo que se apegue a las características del terreno que se está analizando y de ahí partir para obtener el Valor Unitario de ese predio.

Como se puede observar con la implantación del sistema se pretende agilizar y facilitar el proceso de valuación de un predio en el D.F., además de que el sistema sea confiable, seguro y que cubra todos los requisitos de información necesarios.

### **2.5.2 Requerimientos del usuario**

Los alcances y limitaciones de cualquier proyecto se definen cuando se conocen los requerimientos del usuario. Siendo esta la parte más importante en el ciclo de vida del desarrollo de sistemas, la información sobre la factibilidad y alcance del proyecto, el control, los requerimientos de operatividad de datos y los datos mismos, son los elementos que permiten al diseñador conocer a fondo toda la problemática para posteriormente ofrecer una solución. La responsabilidad del usuario es comunicar todo lo que el desea que haga el sistema. En este punto es de vital importancia recopilar toda la información que se pueda. La duración del proyecto puede verse afectada si el análisis carece de elementos significativos de los requerimientos, o bien, si el usuario falló al proporcionar las respuestas detalladas durante el análisis.

Una de las tareas en la definición de un sistema es la de enfocar y comprender el problema que viene a solucionar, para poder emitir una justificación de la solución propuesta. Este proceso requiere de una intensa comunicación entre el usuario y el diseñador del sistema.

De esta forma, el sistema de valuación de predios pretende cumplir con los siguientes requerimientos:

- 1) Diseñar un sistema que permita unificar los criterios de valuación, para evitar que se tengan diferentes valores para un predio, independientemente de que el avalúo sea realizado por cualquier valuador.
- 2) Proporcionar con el desarrollo del sistema, el valuador tenga que registrar la mínima información indispensable en su investigación de campo.
- 3) Que el sistema defina entradas y salidas de información en forma estandarizada, para poder conjugar la información de diferentes bases de datos locales o de otras empresas.
- 4) El sistema debe contar con una base de datos que contenga la mayor cantidad de información específica, para verificar más rápidamente las características de cada colonia del D.F.
- 5) Que el sistema permita realizar una actualización de la base de datos, para tener registrados los datos más recientes.
- 6) Con el sistema todas las operaciones matemáticas y cálculo deben realizarse en el menor tiempo posible y con la mayor exactitud.
- 7) El sistema debe tener la capacidad de verificar que los datos que se introduzcan sean los adecuados para que pueda funcionar correctamente.
- 8) El sistema debe permitir almacenar toda información relacionada con el predio en estudio y recuperarla en forma detallada, resumida, estratificada y organizada para que sea comprendida fácilmente.
- 9) Que el sistema sea lo mas sencillo de manejar, tanto para la captura, modificación, consulta y reportes, de tal forma que no se necesiten mayores conocimientos de computación para su correcta operación.
- 10) El sistema debe permitir agregar otros módulos, ya que se desea que en un futuro el sistema también pueda realizar el cálculo del Valor Unitario para casas, departamentos, edificios, bienes en renta y comerciales.
- 11) Para garantizar el funcionamiento eficaz del sistema se deben elaborar manuales y procedimientos detallados que definan y normen los datos de entrada y salida,

asegurando la continuidad de la información, tanto para el personal del área de avalúos como para los demás usuarios.

### 2.5.3 Recopilación y análisis de la información

Un punto que no se debe perder de vista en todo análisis es el de informar a las personas que están ligadas con los procesos a automatizar, sobre la importancia del estudio que se efectuará para implantar el nuevo sistema, así como los beneficios que se obtendrán al termino de éste y, por consiguiente, la necesidad de contar con información verídica proveniente de ellos para poder cumplir con los objetivos.

Para el presente estudio se llevaron a cabo una serie de entrevistas con las personas autorizadas a fin de conocer en forma más específica cada uno de los procesos, relevantes para el diseño.

La identificación de procesos que se presenta en la figura 2.5.3.1 se determinó con base en la experiencia del personal del área de avalúos y con la investigación de las tareas que están involucradas en el proceso de avalúos.

La figura 2.5.3.1 presenta cada uno de los procesos de la realización de avalúos divididos en etapas y a su vez ordenados por precedencia en cada una de sus etapas.

PRIMERA ETAPA	SEGUNDA ETAPA
Solicitud de avalúo	Obtener la ubicación del predio
Asignación del valuador.	Estudio de mercado.
Determinación de los datos generales del predio.	Factores físicos individuales del predio.
Determinar la región en estudio y delimitar la colonia.	Análisis de datos.
	Obtención del valor unitario del predio.
	Reporte detallado del avalúo.

Figura 2.5.3.1. Identificación de procesos divididos en etapas.

La división de las etapas se determinó en función de todas las actividades que se efectúan dentro de la realización de un avalúo, dicha separación se debió a que es más sencillo analizar un sistema por módulos con pocas tareas que un sólo módulo con un gran número de actividades, pero tomando en cuenta que también es contraproducente tener un número indeterminado de módulos con pocas tareas.

Ya que se tiene identificado cada uno de los procesos que intervienen en la realización de un avalúo, es necesario identificar cuál es la información que se maneja en cada uno de los procesos, por lo que se procedió a analizar cada una de las actividades en forma detallada.

*Solicitud de avalúo.* El cliente genera una petición a la compañía de avalúos para que le realice una valuación a un predio. El cliente debe proporcionar todos sus datos generales, así como una solicitud por escrito del avalúo que requiere.

*Asignación del valuador.* En este proceso la compañía asigna al perito que será el responsable de llevar a cabo el proceso de valuación del predio y la compañía le proporciona los datos que han sido proporcionados por el cliente.

*Determinación de los datos generales del predio.* En este proceso el valuador obtiene todos los datos generales del inmueble, estos datos pueden ser: ubicación del predio, nombre del dueño, régimen de propiedad, inmueble que se evalúa, objeto del avalúo.

*Determinación de la región en estudio y delimitación de la colonia.* En este proceso lo que realiza el valuador es determinar la zona en que se encuentra ubicado el predio al cual se le va a realizar el avalúo, por lo general se ayuda con mapas de la zona para ubicar el predio y obtiene todas las características de la zona como pueden ser:

- Tipo y calidad de los servicios públicos, tales como: limpia, vigilancia, redes de agua potable, alcantarillado, corriente eléctrica, gas y alumbrado público.
- Zonificación, distribución y mejoras de las calles.

- Tipo, estado, actividad y tendencias de desarrollo urbano.
- Influencia de molestias (contaminación).
- Tipo, calidad, frecuencia y costos de transportes.
- Tránsito.
- Tipos de usos de suelo.
- Tipo, calidad, condición, tamaño y edad de las construcciones.
- Características de la lotificación y distribución de áreas verdes.
- Cercanía de escuelas, iglesias, comercios, bancos, centros de diversión.

Delimita la colonia, con el fin de obtener una unidad uniforme para la valuación. En cada zona las características de urbanismo y de población deben ser homogéneas por lo que se refiere a su calidad. Para la definición de los límites de la colonia debe considerarse las siguientes características:

- Raza, nacionalidad, prestigio, ocupación, ingresos, costumbres, intereses culturales y cívicos, así como educación de los residentes.
- La densidad de la población.
- Su actitud hacia la ley y el gobierno y su vida económica respecto al nivel y estabilidad de ingresos profesionales o por ocupación.
- Intensidad y calidad comercial.
- Porcentaje de casas propias y vacías, y frecuencia de las ventas.

*Ubicación del predio.* En este proceso, el valuador obtiene la ubicación del predio, tiene que registrar las calles que forman la manzana, una descripción general del terreno y una descripción general del uso actual.

*Estudio de mercado.* Este proceso lo realiza el valuador con el fin de obtener más información para poder obtener el Valor Unitario Medio de la colonia. La información se obtiene de las siguientes fuentes de información:



- Avalúos directos realizados por otros valuadores profesionales ó personas relacionadas con el medio de los Bienes Raíces (corredores, técnicos de hipotecas, etc.).
- Ofertas de terrenos en venta a través de anuncios al público.
- Resultados de registros estadísticos y proyecciones.

*Factores físicos individuales del predio.* En este proceso el valuador tiene que obtener y registrar los factores físicos individuales que afectan al predio, los principales factores físicos que se encuentran en los predios son:

- Factor de frente.
- Factor de ubicación.
- Factor de superficie.
- Factor de forma.
- Factor de topografía.
- Factor de zona.
- Factor de homologación por uso de suelo.
- Facto de homologación por servicios públicos.
- Factor por comercialización.

*Análisis de datos.* En este proceso lo que realiza el valuador es un análisis de los datos que obtuvo de la investigación de campo y del estudio de mercado, para verificar que sean lo más reales y recientes posibles, así como los factores particulares del predio sean correctos.

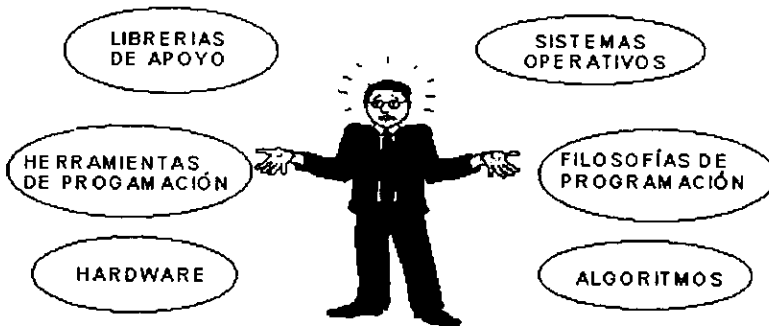
*Obtención del valor unitario del predio.* En este proceso se llevan a cabo todas las operaciones matemáticas que se requieren para obtener como resultado final el Valor Unitario del predio al que se le está realizando la valuación.

*Reporte detallado del avalúo.* En este proceso el valuador genera un documento detallado, en el cual debe mencionar todas las características que influyen en la obtención del Valor Unitario del predio.

Una vez que se terminó el proceso de entrevistas, ya se cuenta con una gran cantidad de información necesaria para poder seguir con el análisis del sistema.

### 2.5.4 Propuestas de solución

Debido al avance a pasos acelerados de la computación actualmente se cuenta con muchas opciones para la solución de problemas relacionados con el proceso de un sistema computacional, desde múltiples opciones de hardware, sistemas operativos, hasta las herramientas de programación. Estas opciones se multiplican más considerando que dentro de una herramienta de programación hay varias opciones de: librerías de apoyo, algoritmos, filosofías de programación, etc. como se puede observar en la figura 2.5.4.1.



**Figura 2.5.4.1 Elección de una operación.**

Un enfoque para encontrar la mejor solución, es centrarse en el problema principal y a partir de él derivar la opción más idónea para el desarrollo; por supuesto previendo todos los requerimientos que debe tener un buen sistema como lo son: buena relación

costo-beneficio, tiempos de entrega, tiempos en adecuaciones del sistema, compatibilidad con otras plataformas, etc.

El elegir una opción no es del todo libre, hay que considerar que se puede contar con la opción de comprar nuevos recursos o utilizar la infraestructura con la que se cuenta. En caso de elegir utilizar los recursos ya existentes, se acortan las opciones de solución, como lo muestra la figura 2.5.4.2 y que de alguna manera encaminan a un universo menor de soluciones, pero dentro de los cuales se puede encontrar la optima, es decir dar el mayor rendimiento con el mínimo de recursos.

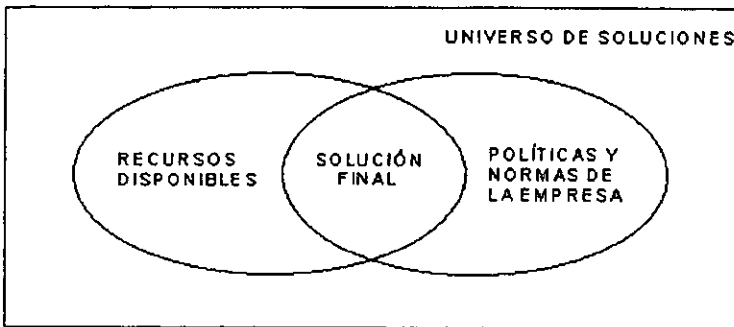


Figura 2.5.4.2 Libertad en la solución.

A continuación se presentan las diferentes opciones a tomar en cuenta analizadas en el presente proyecto para el desarrollo del sistema de avalúos y que se cumplan todos los objetivos y requerimientos del usuario.

#### Opción 1.

En esta alternativa se contempla el utilizar una base de datos como Oracle, Informix o Sybase que son bases de datos que tienen grandes características y ventajas como: seguridad, ejecución, almacenamiento, etc., además de que son de las bases de datos comunes que se manejan dentro del mercado.

Observación: Esta alternativa es buena en cuanto a la base de datos que se desea manejar, pero el hecho de seleccionar alguna de ellas acarrea el problema de que la relación costo-beneficio estaría desequilibrado, ya que se tendría que realizar una inversión bastante considerable en equipo de cómputo que soportara cualquiera de las tres bases de datos y también en la compra del software. Otro punto a considerar es que el tiempo de adecuación del sistema tardaría demasiado ya que varias de las personas que intervienen en el proyecto no tienen conocimiento suficiente de esas bases de datos, por lo que el tiempo de entrega se incrementaría también. Dadas estas desventajas se considera que no es la alternativa más adecuada.

### Opción 2.

Otra opción es la de utilizar los recursos con los que cuenta la empresa, entre los que destaca Microsoft Visual Basic 4.0, Microsoft Access 7.0, Windows 95 y varios equipos de cómputo que soportan el software que se mencionó.

Observación: En esta opción la relación costo-beneficio es muy conveniente para la empresa ya que no tendría que realizar una inversión tan grande en el desarrollo del sistema, además de que si se utiliza Visual Basic y Access se tendría un sistema en un ambiente gráfico que facilitaría al usuario el manejo del sistema. Con este tipo de opción además se tendría resuelto el manejo de la base de datos, puesto que Visual Basic y Access se pueden comunicar de una forma muy sencilla. Otro punto a favor de esta opción es que la mayoría de las personas que intervienen en el proyecto, conocen las herramientas que se pretende utilizar, lo que se reflejaría en un tiempo menor en la adecuación del sistema y el tiempo de entrega.

### Opción 3.

En la tercera opción se pretende utilizar una base de datos de las más comunes en computadoras personales como son: Dbase, Foxpro o Clipper.

Observación: Esta opción es una buena alternativa, pero en lo que respecta al código de programación, este se incrementaría en gran forma, lo que generaría programas muy largos, complicados para las labores de mantenimiento preventivo y correctivo. Además estos programas en su compilación archivos con código ejecutable de un tamaño bastante grandes, dando como consecuencia una saturación rápida del espacio en los medios de almacenamiento secundario de los equipos que se utilizan.

Conclusiones.

Después de haber analizado las opciones que se tienen para el desarrollo del presente proyecto, se llegó a la conclusión de que la mejor, es la opción 2, por que en ella encontramos las siguientes características:

- La relación costo-beneficio es muy buena para la empresa ya que tendría que realizar una inversión de dinero mínima para el desarrollo del sistema, además de que se utilizarían los recursos con los que cuenta la empresa.
- Los tiempos de adecuación del sistema y de entrega disminuyen en forma substancial ya que las personas que intervienen en el desarrollo del proyecto tienen un buen conocimiento del software que se va a utilizar.
- Debido a que el sistema se desarrollara en un ambiente gráfico, resulta más amigable y fácil de utilizar para el usuario que lo utilice.
- El sistema para avalúos definitivamente es rentable no por el hecho de los ingresos, sino por el tiempo que se disminuirá en algunas etapas del proceso de avalúos, lo que se reflejara en el tiempo de entrega del avalúo.

### **2.5.5 Diagramas de funcionalidad**

Un sistema o procedimiento es un conjunto de operaciones o etapas que en forma cronológica se establecen para llevar a cabo un determinado tipo de trabajo.

Los diagramas operacionales o de funcionalidad son diagramas que representan en forma gráfica la secuencia en que se realizan las operaciones de un determinado procedimiento y/o el recorrido de las formas o materiales. Con el fin de tener una visión de conjunto de un sistema o procedimiento se utilizan los diagramas de flujo donde se muestran las unidades orgánicas o puestos que intervienen en el procedimiento, las operaciones que realizan, la secuencia de las mismas y el equipo utilizado en cada caso.

### **2.5.5.1 Técnicas de simplificación de operaciones**

El análisis del sistema o procedimiento implica descomponerlo en sus partes para conocer la naturaleza, los objetivos, los responsables de la ejecución, el lugar y tiempo en que se debe realizar.

Para analizar la información que permita después llevar a cabo un proceso de simplificación de operaciones se debe buscar las respuestas a las siguientes preguntas:

- ¿Qué trabajo se hace?. Se pretende determinar el propósito y objetivo de la naturaleza o tipo de labores que se realizan en la unidad orgánica y los resultados que se obtienen.
- ¿Quién lo hace?. Se intenta definir qué unidades orgánicas o personas intervienen en el procedimiento y además las aptitudes del personal para la realización de una tarea específica, así como la actitud que adopta en la relación de la misma.
- ¿Cómo se hace?. Se busca determinar los medios con los que se cuenta para realizar la tarea, que comprende los métodos y técnicas aplicados y adaptados a la dependencia y a los equipos e instrumentos utilizados.
- ¿Cuándo se hace?. Se determina el orden o secuencia en que se desarrollan las actividades de la unidad orgánica, es decir, la ubicación geográfica y domicilio de las oficinas, la funcionalidad de éstas y la distribución del espacio de las mismas.

- ¿Porqué se hace?. Se busca la justificación de la existencia de ese trabajo o de su procedimiento. Se pretende conocer los objetivos de las acciones que integran el trabajo o procedimiento al igual que la primera pregunta.

El analista establece como resultado no sólo la descripción de las operaciones del sistema o procedimiento en forma más precisa o definida, sino también la posibilidad de mejorarlo, y para ello se tienen las siguientes alternativas:

- Eliminar el trabajo innecesario. Se parte del principio de que todo sistema o procedimiento es perfectible y por ello es necesario mejorarlo, eliminando esfuerzos y tareas innecesarias. Por ejemplo, la obtención de factores generales de la colonia, cada vez que se realiza un avalúo.
- Modificar las operaciones o partes de ellas. Es conveniente subdividir el trabajo pero no tan detalladamente que haga ineficiente el procedimiento o sistema, al someterlo a repetidas revisiones que repercutan en pérdida de tiempo y energía. Por ejemplo se pueden cambiar dos operaciones en una sola.
- Cambiar el orden de las operaciones. El simple cambio en el orden de las operaciones puede hacer más eficiente un procedimiento.
- Simplificar las operaciones necesarias. Aquí es necesario estudiar más detalladamente cada una de las operaciones del procedimiento para determinar posibilidades de simplificación. Pueden aplicarse los estudios de tiempo y movimientos, de esquemas del lugar de trabajo, etc.
- Modificar las actividades de los puestos y equilibrar cargas de trabajo. Cuando se observa en algunos puestos que se consignan los asuntos por negligencia, falta de adiestramiento del personal o mala distribución de la carga de trabajo. Es necesario en los dos primeros casos controlar, motivar y capacitar al personal y en el último caso, replantear las actividades entre los empleados que participan en los procedimientos.

Para llevar a cabo la simplificación de operaciones se necesita establecer un programa que pueda contener los siguientes temas:

- Seleccionar un trabajo importante. El seleccionar un trabajo requiere de una consideración detenida, ya que sólo un trabajo de gran volumen o técnica difícil ofrecerá la oportunidad para efectuar una simplificación de trabajo considerable.
- Dividirlo. Es el proceso de clasificar los detalles o componentes de un trabajo, sistema o procedimiento, de tal manera que pueda examinarse en detalle. Para ello, es posible utilizar un diagrama.
- Hacer preguntas sobre los detalles. Constituye un requisito absolutamente necesario para la preparación de los diagramas, ya que sólo un completo conocimiento puede producir una visión total de las operaciones para pasar después al análisis que conduce a las mejoras.
- Desarrollar las mejoras que se propondrán. Constituye el resultado final de la división, cuadros y cuestionarios, y requiere una buena cantidad de ingenio, imaginación y lógica.
- . Instalar las mejoras. Requiere la participación y cooperación de los empleados para que se pueda instalar la o las mejoras del sistema o procedimiento.

#### **2.5.5.2 Diagrama**

Para el sistema que se pretende desarrollar en la presente tesis se tiene el siguiente diagrama de funcionalidad o flujo de operaciones (ver fig. 2.5.5.2.1). El diagrama muestra el flujo de actividades que deben realizar las áreas para obtener su información así como las demás áreas involucradas para cada caso.



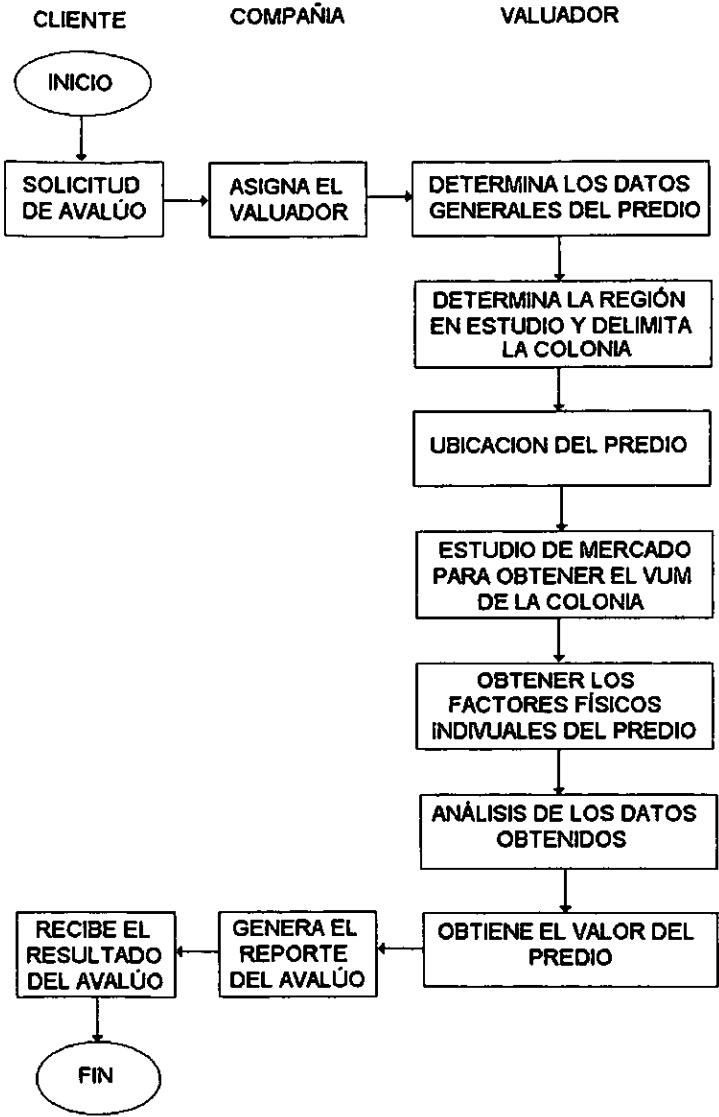


Figura 2.5.5.2.1 Diagrama de funcionalidad del proceso de avalúos.

### **III ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL SISTEMA**

#### **3.1 Diagramas de flujos de datos**

Los primeros trabajos sobre modelos de análisis aparecieron a finales de los 60 y principios de los 70, pero la primera aparición del enfoque de análisis estructurado fue como complemento de otro tema importante - el "diseño estructurado".

El término "análisis estructurado" fue popularizado por DeMarco. En su libro sobre esta materia, DeMarco presentó y denominó los símbolos gráficos clave que permitirían a un analista crear modelos de flujo de información; sugirió heurísticas para utilizar esos símbolos; sugirió el uso de un Diccionario de Datos y narrativas de procesamiento como complementos a los modelos de flujo de información. En los años siguientes, Page-Jones, Gane y Sarson y muchos otros propusieron variaciones del enfoque de análisis estructurado.

El diagrama de flujo de datos es una técnica gráfica que representa el flujo de la información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida. En la figura 3.1.1 se muestra la forma básica de un diagrama de flujo de datos.

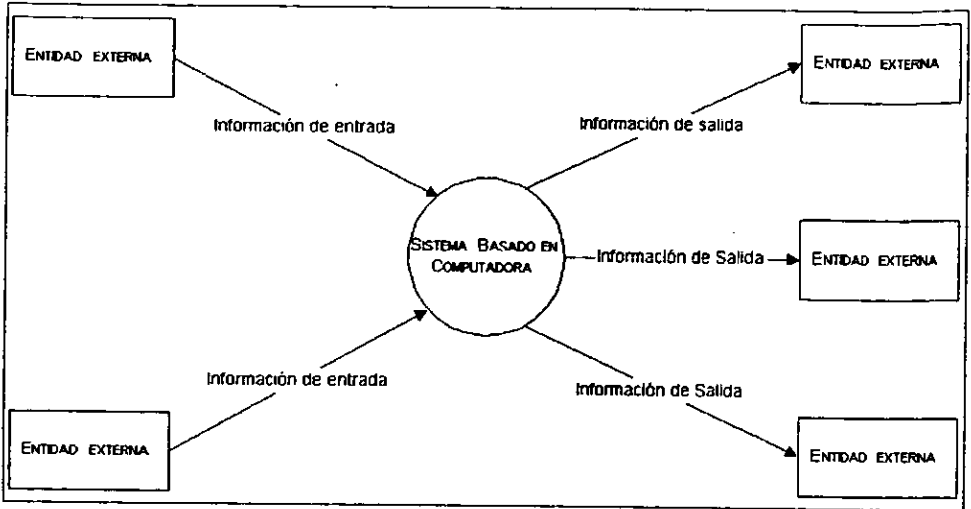


Figura 3.1.1 Modelo de flujo de información.

### Elementos del diagrama de flujo de datos

Los diagramas de flujo de datos consisten en procesos, almacenadores, flujos de información y entidades externas.



- Las entidades externas "External entity", también denominados productores o consumidores de información, son las entidades con las que el sistema se comunica, es por ello que residen fuera de los límites del sistema a ser modelado. Generalmente se trata de personas, organizaciones, documentos, sistemas de cómputo externos u otros objetos que proporcionen o consuman información del sistema.



- Los procesos se representan por medio de círculos, o burbujas en el diagrama. Representan las diversas funciones individuales que el sistema lleva a cabo.
- Los flujos de datos "Data Flow" se muestran por medio de flechas curvas; la cabeza de la flecha indica la dirección del flujo de datos, el cual representa la información que dichos procesos requieren como entrada o la información que generan como salida.
- Los almacenadores "Data store", se representan por medio de dos líneas paralelas, y se utilizan para almacenar datos que serán utilizados por uno o más procesos.

### **Niveles de un Diagrama de Flujo de Datos**

A través del uso de los diagramas de flujo de datos se puede representar un sistema o aplicación a cualquier nivel de abstracción, dado que, los DFD's pueden ser refinados en niveles que representen un mayor flujo de información y un mayor detalle funcional. Estos niveles comprenden desde el diagrama de contexto, diagrama de nivel 0 y diagramas de nivel 1,2,3...N.

## **Diagrama de Contexto**

El primer nivel de un DFD es el diagrama de Contexto, en donde a través de una sola burbuja se representa al sistema completo, con datos de entrada y de salida respectivamente.

El diagrama de contexto enfatiza las siguientes características importantes del sistema:

- Las personas, organizaciones y sistemas con los que se comunica el sistema; Se conocen como entidades externas, agentes externos o actores.
- Los datos que el sistema recibe del mundo exterior y que deben procesarse de alguna forma.
- Los datos que el sistema produce y que se envían al mundo exterior.
- Los almacenes de datos que el sistema produce y que se envían al mundo exterior.
- La frontera entre el sistema y el mundo exterior.

## **Diagrama de Nivel 0**

A través del diagrama de nivel 0 se representan los procesos principales de la Aplicación que se esta modelando, los cuales al llevarse a cabo el refinamiento sucesivo se irán explotando en diagramas de nivel inferior.

## **Diagramas de Nivel 1,2,3...N**

Estos diagramas son el resultado del refinamiento de cada una de las burbujas en distintos niveles para mostrar un mayor detalle. Es importante señalar que se debe mantener la continuidad del flujo de información, es decir, que las entradas y las salidas de cada refinamiento deben ser las mismas.

## Reglas para la construcción de un DFD

- ∅ Asignar nombres con significado para los procesos, flujos, almacenes y agentes externos.
- ∅ Numerar los procesos en forma decimal, p.e. 1.0, 1.1, 1.2. Los números sirven para relacionar adecuadamente una burbuja con el siguiente nivel del DFD que la describe más a fondo.
- ∅ Redibujar el DFD tantas veces como sea necesario.
- ∅ Evitar los DFD excesivamente complejos.
- ∅ Asegurarse de que el DFD sea internamente consistente y que también lo sea cualesquiera DFD relacionado con él.

## Características de los DFD

En la elaboración de los DFD's, se debe observar que cumplan con las siguientes características:

### 1. Consistencia.

La consistencia se obtiene al evitar sumideros infinitos, burbujas que tienen entradas pero no salidas, burbujas de generación espontánea que tienen salidas sin tener entradas.

### 2. Balance.

El nivel de balance compara las entradas a, y salidas de un proceso en un diagrama padre con los flujos de salida de un diagrama hijo. Es decir, los flujos de información que entran a un DFD hijo, deben estar representados en el padre por los mismos flujos de información.

La figura 3.1.2 muestra las características que un DFD debe poseer y otras que deben observarse en la construcción de este.

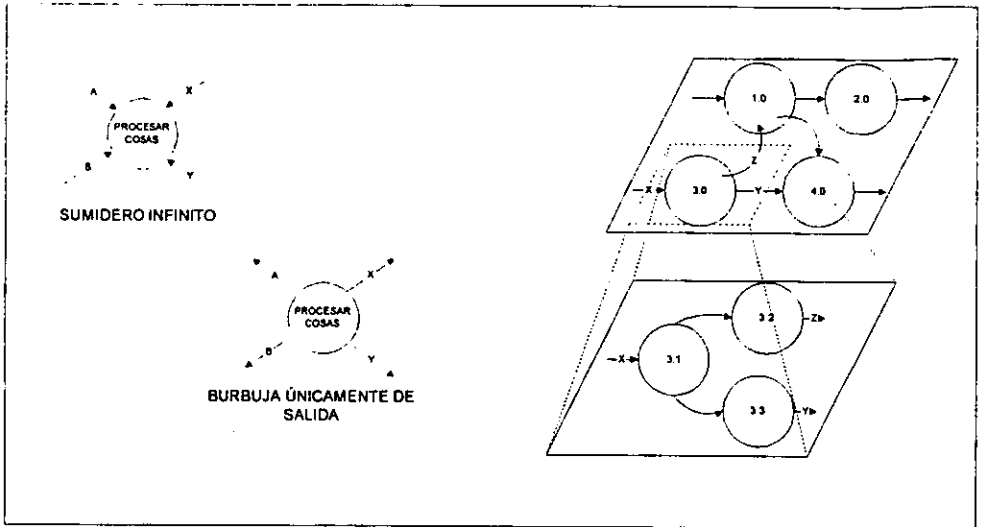


Figura 3.1.2 Características de un diagrama de flujo de datos.

### Descripción de procesos

La descripción de procesos, se utiliza para describir todos los procesos del modelo funcional que aparecen en el nivel final de refinamiento, con el propósito de definir lo que debe hacerse para transformar entradas en salidas, a través de narrativas textuales o descripciones en lenguaje estructurado, que servirán como guía para el diseño del componente de programa que implementará el proceso.

En la especificación de procesos en general se describe lo siguiente:

- La información que entra y la que sale del módulo (una descripción de la interfaz).
- La información que es retenida en el módulo.

- ❑ Una explicación del procedimiento, que indique los principales puntos de decisión y las tareas.
- ❑ Un breve tratamiento de las restricciones.

Una descripción de procesos no necesariamente debe realizarse empleando el lenguaje estructurado, también pueden emplearse narrativas textuales que indiquen la función del proceso, las funciones o procedimientos y los parámetros de entrada y salida de tales funciones, sin llegar a detallar.

La figura 3.1.3 resume el procedimiento a través del cual se realiza la especificación de procesos.

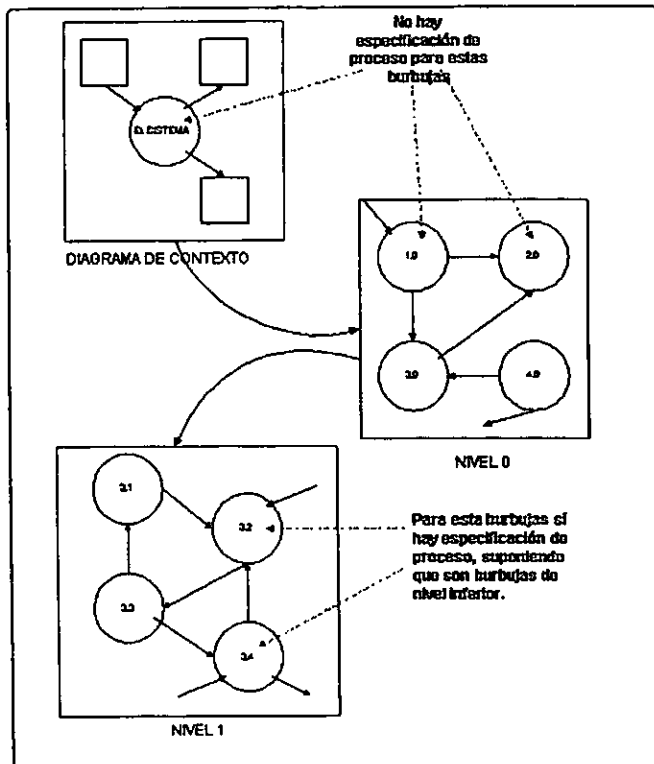
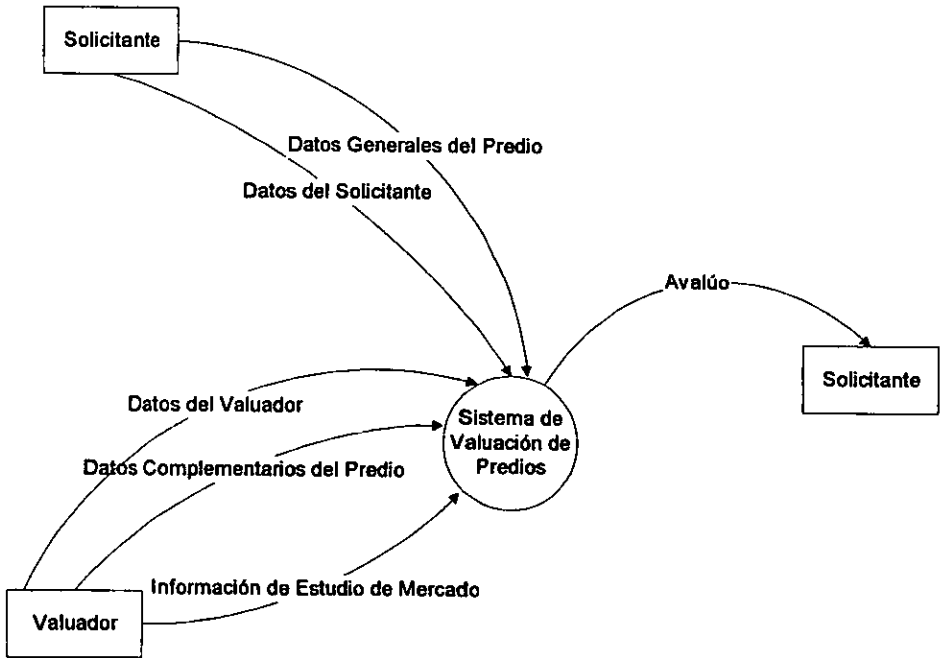


Figura 3.1.3 Especificación de procesos.



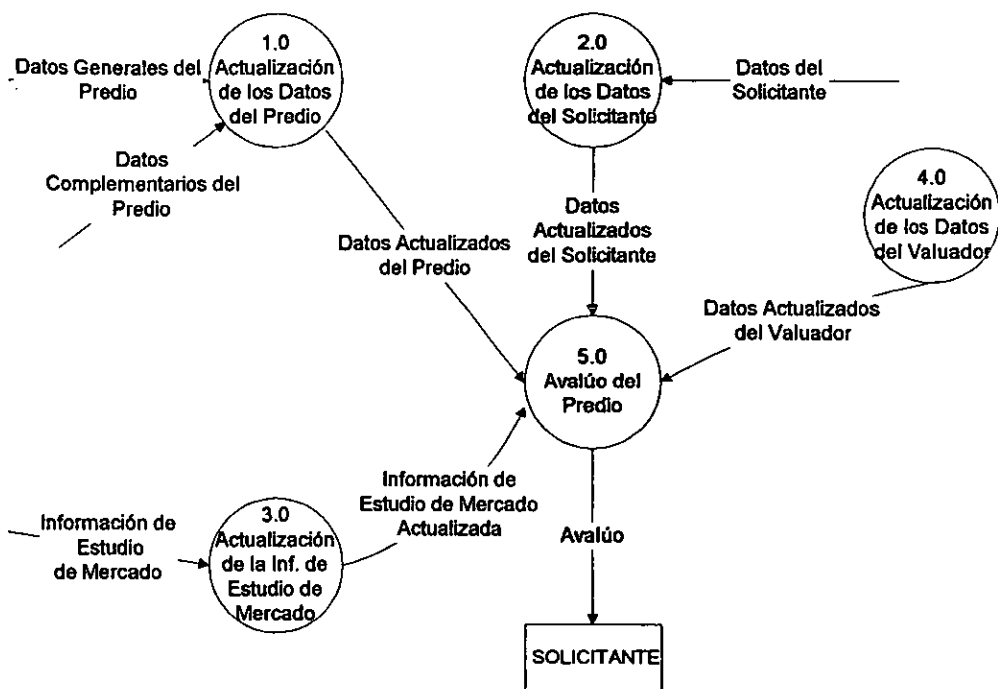
A continuación se muestran los DFD's obtenidos durante el diseño del sistema de Valuación de Predios.

### Diagrama de Contexto



---

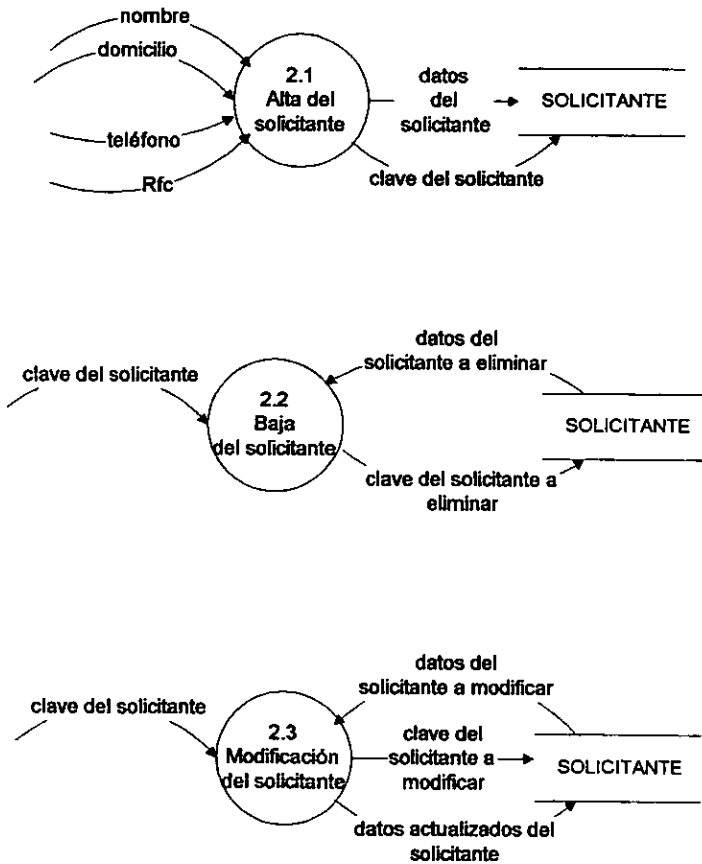
## Diagrama de Flujo de Datos "Sistema de Valuación de Predios"



Nivel 0

---

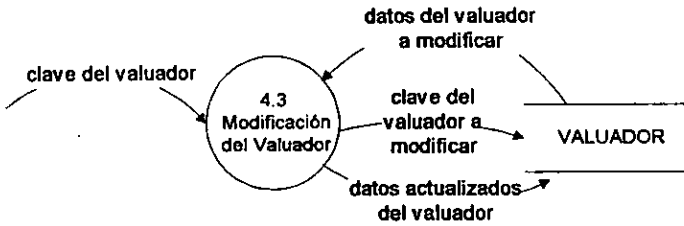
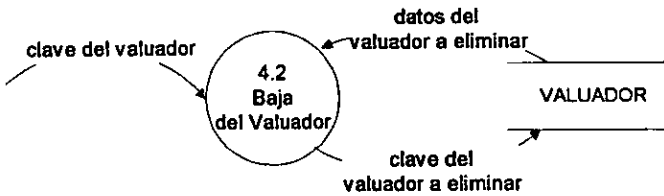
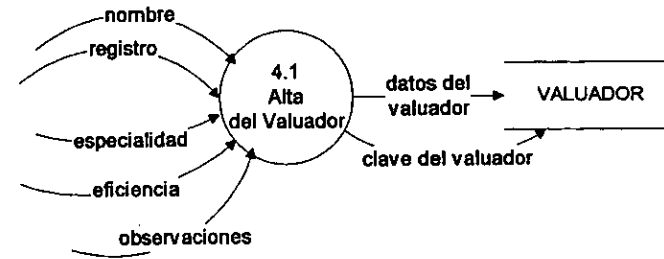
## Diagrama de Flujo de Datos "Actualización de los Datos del Solicitante"



**Nivel 1**

---

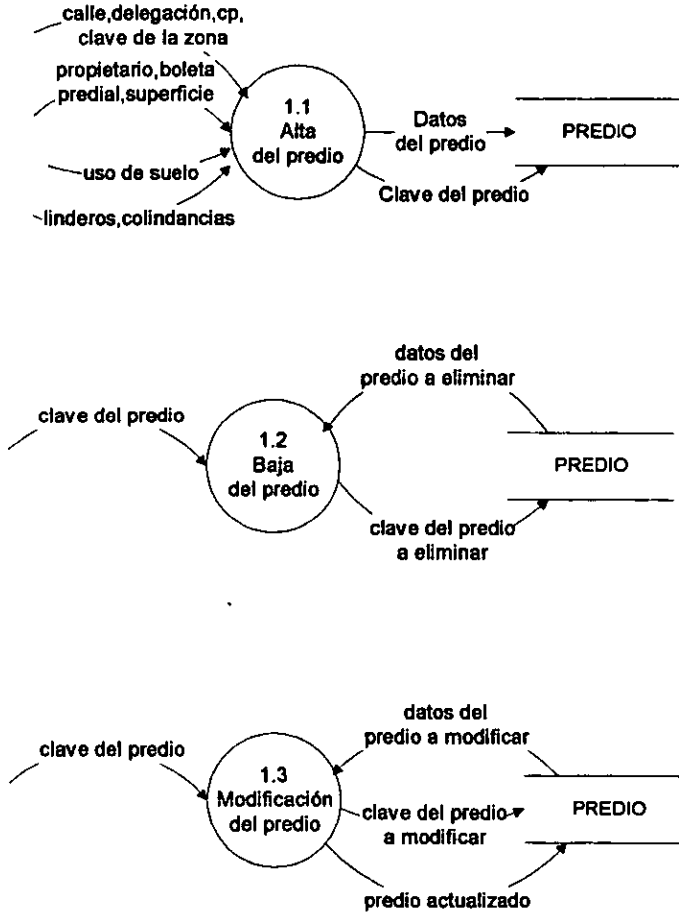
## Diagrama de Flujo de Datos "Actualización de los Datos del Valuador"



Nivel 1

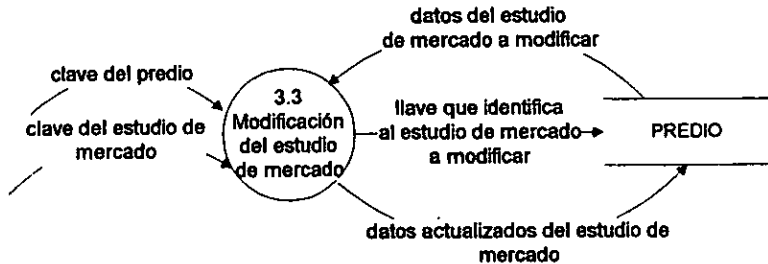
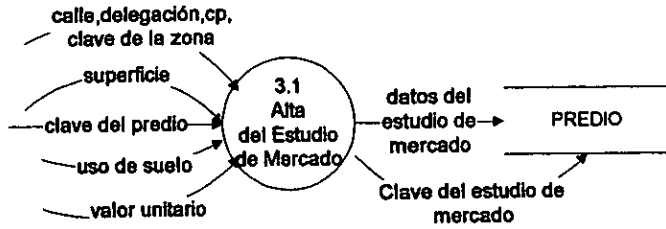
---

## Diagrama de Flujo de Datos "Actualización de los Datos del Predio"



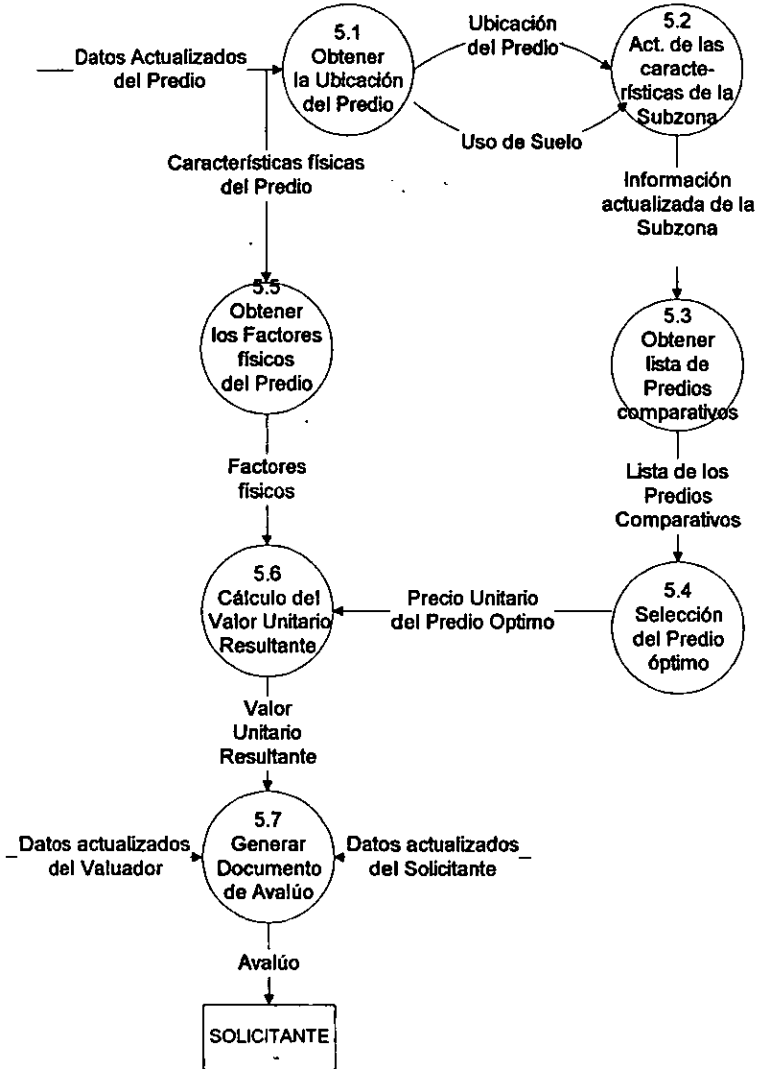
Nivel 1

## Diagrama de Flujo de Datos "Actualización del Estudio de Mercado"



Nivel 1

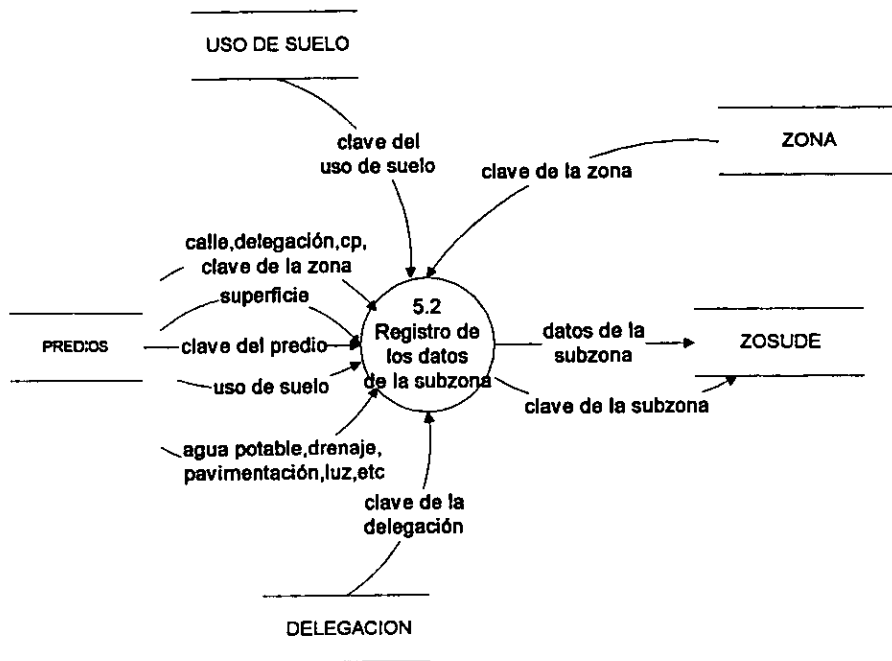
## Diagrama de Flujo de Datos "Avalúo"



**Nivel 1**

---

## Diagrama de Flujo de Datos "Actualización de la Subzona"

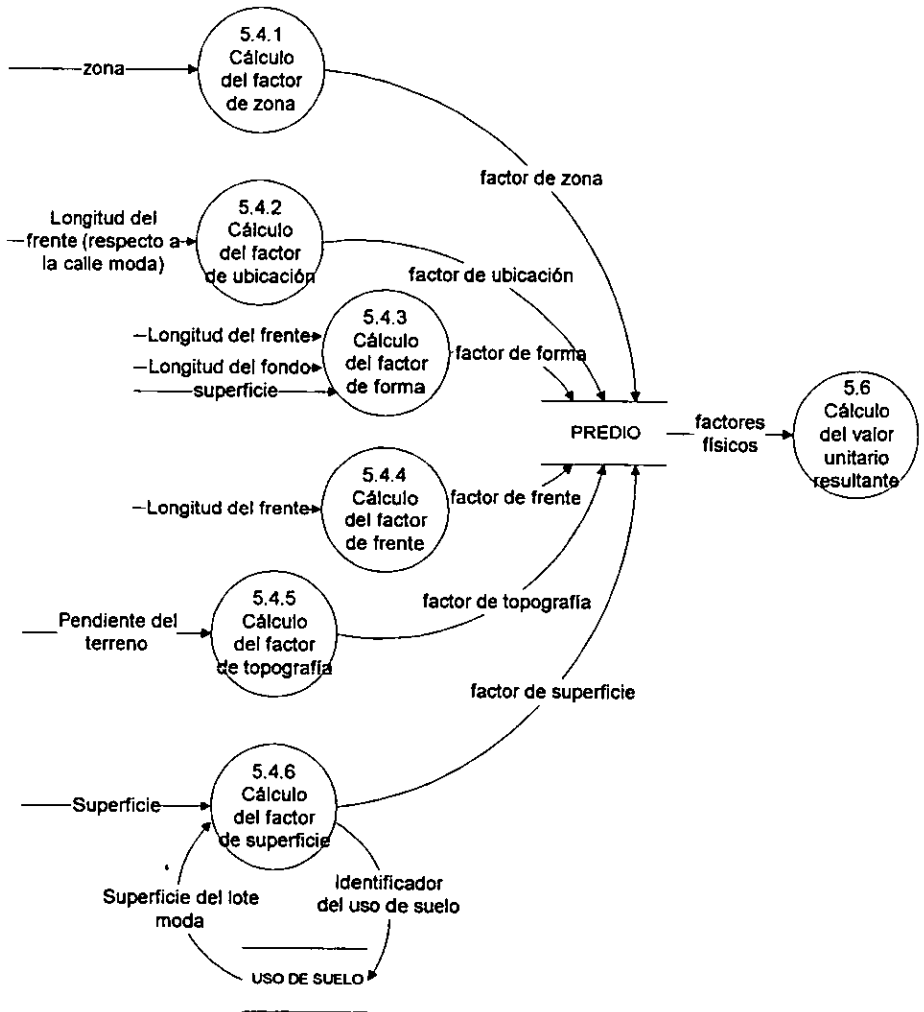


Nivel 2

---



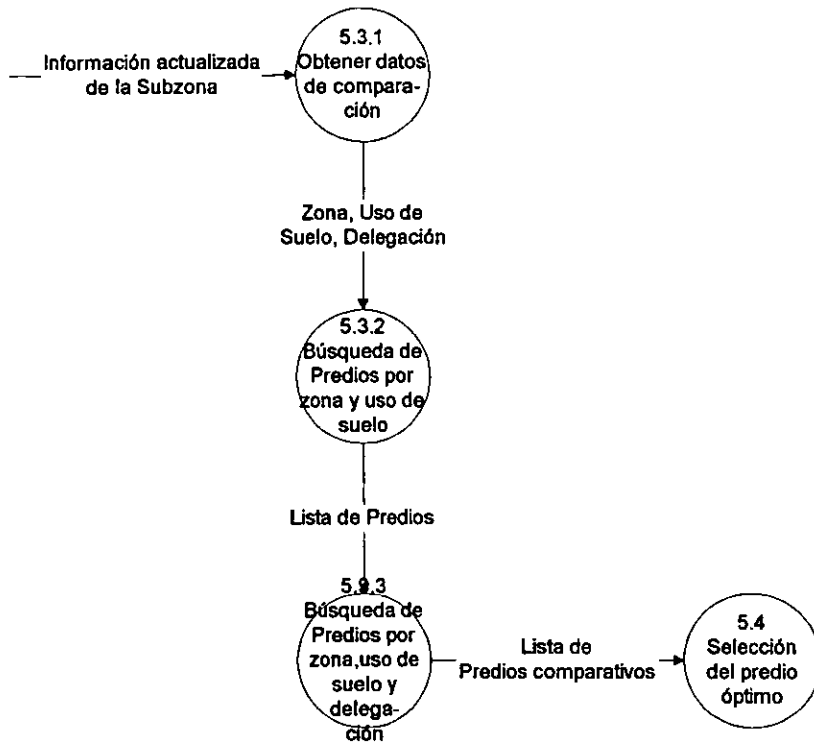
**Diagrama de Flujo de Datos  
"Obtención de los Factores físicos del Predio"**



**Nivel 2**

---

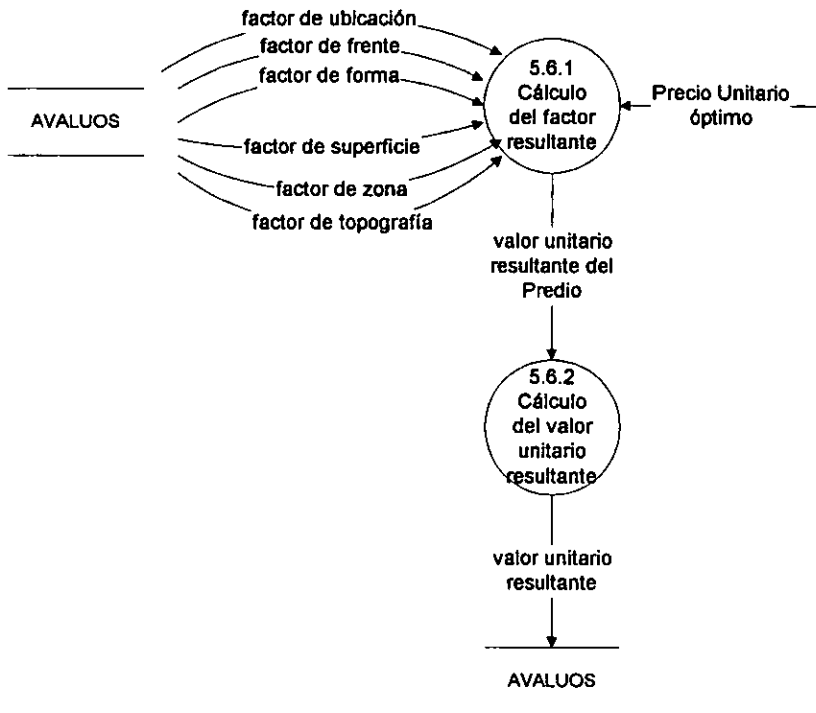
**Diagrama de Flujo de Datos**  
**"Obtención de la Lista de Predios**  
**Comparativos"**



**Nivel 2**

---

**Diagrama de Flujo de Datos**  
**"Cálculo del Valor Unitario Resultante"**



**Nivel 2**

---

### 3.2 Diagrama Entidad-Relación y Normalización de datos

El diagrama entidad relación es una herramienta del análisis estructurado que proporciona un conocimiento más amplio sobre los detalles de los almacenadores de datos y sus relaciones con los procesos dentro del modelo funcional.

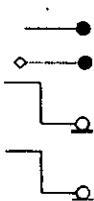
El diagrama entidad relación es un modelo de red que describe con un alto nivel de abstracción la distribución de datos almacenados en un sistema.

#### Elementos del Diagrama de Entidad Relación

Un diagrama entidad relación esta integrado por dos elementos básicos: entidades y relaciones.



- Se representan por medio de un rectángulo en el diagrama. Esto representa una colección o conjunto de objetos reales o abstractos (personas, lugares, eventos, etcétera) del mundo real que tienen atributos comunes y que juegan algún papel en el desarrollo del sistema; pueden además ser identificados de manera única.



- Una relación es una asociación entre dos entidades y representa un conjunto de conexiones entre objetos, instancias o entidades. Las relaciones se representan a través de líneas, las cuales una vez establecidas, se evalúa cada extremo para

determinar la cardinalidad, la cual puede ser:

- ◊ Simple: Para una relación 1:1
- ◊ Múltiple: Para las relaciones 1:N, 0:N y N:N.

## **Normalización de datos**

La normalización es el proceso mediante el cual un diseñador de bancos de datos puede transformar cualquier estructura de datos no plana, a un conjunto de relaciones normalizadas, es decir a un conjunto de tablas planas que no contengan grupos repetitivos.

Se dice que una relación no plana o no normalizada contendrá al menos un dominio que será en realidad otra relación. Una relación normalizada tiene únicamente dominios simples, es decir, dominios que no son a su vez otra relación.

Un archivo que sea plano excepto por un grupo repetitivo, se normalizará al quitarlo y formar con él una relación separada. Esta deberá tener un nombre propio así como un campo llave.

La normalización, es utilizada generalmente para poder obtener el enfoque relacional que posee bases matemáticas rigurosas, mismas que respaldan la teoría relacional, proporcionando simplicidad en las estructuras de datos utilizadas, facilitando su uso y modificaciones, además de que evita que los archivos de las bases de datos tengan redundancias, incoherencias o inconsistencias.

## **Definiciones previas:**

**Dependencia funcional.** Dado un archivo, sean CAMPO1 y CAMPO2 dos campos de ese archivo, diremos que CAMPO2 tiene dependencia funcional de CAMPO1 cuando dado un valor de CAMPO1 se obtiene un valor único de CAMPO2 o que a cada valor de CAMPO2 le corresponde un único valor de CAMPO1.

**Dependencia funcional completa.** Si en un archivo, CAMPO1 es un conjunto de campos, entonces diremos que CAMPO2 tiene dependencia funcional de todo el conjunto de CAMPO1 consta de un único atributo entonces coincide la dependencia funcional y la dependencia funcional completa.

**Candidato a llave.** Candidato a llave es un atributo o conjunto de atributos que identifica de forma unívoca un registro. Uno de los candidatos a llave es utilizado con llave principal o primaria.

**Atributo no llave.** Es aquel que no forma parte de la llave o de un candidato a llave.

**Determinante.** Es un atributo o un conjunto de atributos del que dependen otros atributos.

## **Primera Forma Normal**

Esta es la forma más común de llegar a la Primera Forma Normal. La redundancia puede reducirse separándola en otro grupo aparte de la entidad debiendo contener el identificador de la entidad original para mantenerse relacionadas, por lo que se dice que una relación está en primera forma normal si para cada valor específico de un identificador, existe uno y sólo un valor de cada atributo. Es decir, no hay "grupos repetitivos", pero se mantiene un alto grado de redundancia.

## **Segunda Forma Normal**

Para determinar que una entidad esté en Segunda Forma Normal, se requiere que el valor de cualquier atributo que no es llave dependa de todos los atributos que forma la llave. Para normalizar una entidad en la Segunda Forma, se crea una nueva entidad de los atributos que dependen parcialmente de una llave, siendo parte del identificador de esta nueva entidad el atributo, del cual depende para mantenerla relacionada con la original.

## **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.

## **Cuarta forma normal**

No se cumple con esta forma normal cuando una relación contiene dos o más datos multivaluados de una entidad.

## **Quinta forma normal**

Establece que un registro no puede estar estructurado de tal forma que sus contenidos se pueden reconstruir a partir de registros más pequeños. Por su puesto, siempre es posible reestructurar un registro en otros más pequeños con el mismo campo llave. Este caso es una excepción de la restricción anterior.

Las formas normales se aplican secuencialmente, sobre un diseño de base de datos, primero debe comprobarse la primera, luego la segunda, seguida de la tercera, y así sucesivamente. Aunque aquí se ha mencionado hasta una quinta forma normal, con aplicar la tercera forma normal se obtiene un buen esquema de base de datos, por lo que aquí normalizaremos hasta esta forma normal.

### **Objetivos de la normalización**

- Eliminar en lo posible todos los datos que mantengan anomalías.

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 debe 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:

*Anomalia de Inserción.* No cabe almacenar nueva información sobre una entidad en particular hasta que se establece su relación con otra entidad.

*Anomalia de Eliminación.* La eliminación de un solo registro puede ocasionar la eliminación de toda la ocurrencia de una entidad.

*Anomalia de Actualización.* Si el valor de un atributo cambia, debe cambiar en los múltiples sitios donde se encuentra definido.

- Conservar toda la información.
- Maximizar la flexibilidad.



La capacidad de adaptabilidad ante los cambios, maximizan la independencia del uso particular de los datos.

La normalización requiere tres (de ahí el nombre de Primera, Segunda y Tercera Forma Normal), o más acciones sobre un atributo de una entidad, hasta que el modelo alcance su última etapa de no redundancia, por lo regular la normalización está comprendida hasta la tercera etapa, donde el modelo ya alcanza un estado manejable y entendible para el usuario y el programador.

Esto se establece, en las dependencias funcionales que existen 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 ésta.

A continuación, normalizaremos las bases de datos que contienen la información de la valuación de predios.

Los atributos requeridos para un avalúo son los que se muestran en la figura 3.2.1.

NoSecuencial	Número del avalúo
FechaRecepción	Fecha de recepción de la solicitud del avalúo.
MotivoAvaluo	Motivo del avalúo.
NomSolicitante	Nombre de la persona o Institución que solicita el avalúo.
NumOficio	Número de oficio.
DomicilioSolici	Domicilio del solicitante.
TelSolicitante	Teléfono del solicitante.
FechaEntrega	Fecha de entrega del avalúo.

**Continuación...**

NomValuador	Nombre del perito que realiza el avalúo.
RegistroValuador	Número de registro del valuador.
EficienValuador	Eficiencia desarrollada en los trabajos que se le encomiendan.
EspecialidadValuador	Especialidad del valuador.
ObservaValuador	Observaciones sobre el cumplimiento del valuador.
Calle	Calle en donde se ubica el predio.
Delegacion	Delegación.
Cp	Código Postal.
Colonia	Colonia.
BoletaPredial	Número de boleta predial.
Propietario	Propietario.
RegimenPropiedad	Régimen de propiedad.
UsoDeSuelo	Uso de suelo.
IntenConstruc	Intensidad de construcción.
ClasifZona	Clasificación de la zona.
ServPublicos	Servicios públicos.
TpoDeConstruc	Tipo de construcción.
SaturaDeConstruc	Saturación de construcción.
DensiDePoblacion	Densidad de la población.
CalleMznaNte	Calle que forma la manzana al norte.
CalleMznaSureste	Calle que forma la manzana al sureste.
CalleMznaSuroeste	Calle que forma la manzana al suroeste.
ColinNoreste	Colindancia al noreste.
ColinNoroeste	Colindancia al noroeste.
ColinSureste	Colindancia al sureste.
ColinSuroeste	Colindancia al suroeste.

**Continuación...**

DistNoreste	Distancia al noreste.
DistNoroeste	Distancia al noroeste.
DistSureste	Distancia al sureste.
DistSuroeste	Distancia la suroeste.
Superficie	Superficie del terreno.
FactorDeZona	Factor de zona.
FactorDeUbica	Factor de ubicación.
FactorDeFrente	Factor de frente.
FactorDeForma	Factor de forma.
FactorDeSup	Factor de superficie.
FactorRte	Factor resultante.
ValorUnitario	Valor unitario.
ValorUnitarioRte	Unitario resultante.
Conclusiones	Conclusiones.

Figura 3.2.1 Requisitos para un avalúo.

A continuación identificaremos los atributos que se pueden separar, para ello agregaremos algunos atributos que nos servirán para identificar a las relaciones que se originen.

**Avalúos      Identificador: **Avaluo****

IdAvaluo	MotivAva	IdSolici	FechRece	IdPredio	IdValuad
DocAntec	FechEntr	Conclusi			

**Solicitantes      Identificador: **Solicitante****

Ident	IdSolici	TipoSoli	Nombre	Domicili	Telefono	RFC
-------	----------	----------	--------	----------	----------	-----

**Valuadores      Identificador: **Valuador****

IdValuad	Nombre	TpoValua	Registro	Especiali	Eficiencia	Observac
----------	--------	----------	----------	-----------	------------	----------

**Predios      Identificador: Predio**

IdPredio	Calle	IdColoni	IdDelega	IdZona	IdUsoSue	CodPost
NomPropi	Superficie	Objeto	BoletPre	Escritura	FactForm	FactFond
FactSupe	FactFren	FactUbic	FactTopo	FactZona	ValorAva	FechReal
Calle1	Calle 2	Calle 3	Calle 4	Linderos		

**Zonas      Identificador: Zona**

IdZona	Descrip
--------	---------

**Uso de Suelo      Identificador: Suelo**

IdUsoS	Descrip	Lotemoda	DensPobl
--------	---------	----------	----------

**Delegación      Identificador: Delegación**

IdDelega	NombDele
----------	----------

**Colonia      Identificador: Colonia**

IdColoni	NombColo
----------	----------

**Zona/Suelo/Delega**

IdZona	IdUsoSue	IdDelega	InteCons	UsoSuel	AguaPota	Drenaje
Paviment	EnerElect	CableSub	RedTelef	RecoBasu	TVxcable	ComeServ
Educacio	Asistenc	Diversio	Recreaci	Trabajo	ParqPubl	Iglesias
ServTele	Contamin	Observac				

**3.3      Diccionario de datos**

El diccionario de datos es una especialidad dentro de los diccionarios de referencia que se utiliza en la vida diaria. El diccionario de datos es una referencia de "datos acerca de los datos" (esto es metadatos), recopilados por el analista de sistemas para guiarse sobre el análisis y el diseño de un sistema.

Como documento, recopila, coordina y confirma lo que un término específico significa para la gente de una organización. Los diagramas de flujos de datos son un buen inicio para la recopilación de los términos del diccionario de datos.

El diccionario de datos sirve también como el estándar consistente de los datos elementales.

Los diccionarios de datos automatizados (que también forman parte de los instrumentos CASE) son valiosos porque permiten la referencia cruzada de datos sencillos. Además estos diccionarios pueden ser relevantes para los grandes sistemas que producen varios miles de datos elementales que requieren ser catalogados y así contar con referencias cruzadas.

### **Datos que debe contener un diccionario de datos**

Una manera de saber lo que debe contener el diccionario de datos, es visualizar cómo llegará a utilizarse. Es el elemento básico de referencia para localizar los nombres y atributos de los datos utilizados en todo sistema de organización. Por esto se debe incluir todos los datos elementales.

Con el fin de ser de utilidad, los registros del diccionario de datos deben contener información referente a las categorías siguientes:

- El nombre y sinónimo del dato.
- Las descripciones del dato.
- Los datos elementales que se relacionan con el término.
- El rango permitido del dato.
- La longitud disponible en caracteres.
- Una adecuada codificación.
- Cualquier otra información pertinente de edición.

*Nombres y sinónimos (alias):*

El diccionario de datos debe contener el nombre de cada dato, estos es, la manera de denominar el dato en la mayoría de los programas, y su sinónimo. Todo esto debe quedar registrado en el diccionario de datos para facilitar la comunicación entre los departamentos y sus programas.

*Descripción:*

Debe incluir una descripción textual del dato elemental, la cual debe ser concisa (aproximadamente tres frases), pero informativa para cualquiera que lo consulte.

*Rango permitido:*

Deber incluir los distintos rangos y límites que se aplican al elemento. El rango significa el intervalo disponible de datos.

*Longitud del dato:*

Se refiere a la longitud permitida para el acceso a un dato elemental. La longitud siempre se da en función del número de caracteres impresos y no por la cantidad requerida de memoria.

*Codificación adecuada:*

Cada dato debe incorporarse al diccionario de datos junto con su código, si es que lo tiene y el significado de éste. Es indispensable que la codificación sea consistente.

*Información adicional de edición:*

La información requerida para asegurar la edición adecuada de los datos debe estar presente en el diccionario de datos. Esto incluye a cualquier orden pertinente.

Cuando el diccionario de datos se integra de manera correcta, es útil para el desarrollo del sistema, la modificación del mismo y su mantenimiento.

**Tablas que se utilizan en el sistema:**

**Avalúos:** son los elementos principales y representan la información de los trabajos realizados por la empresa para futuras referencias.

**Solicitantes:** datos de la empresa y/o de las personas que proporcionan la información y solicitan el trabajo de los predios a valuar.

**Valuadores:** almacenamiento de los datos de los peritos valuadores con que cuenta la empresa y que se encargan del estudio de mercado y de las valuaciones, los cuales pueden ser internos o externos.

**Predio:** datos generales de los terrenos que sirven:

- 1) De referencia para futuros avalúos (Datos de estudio de mercado) y
- 2) Información del predio a valuar o aquellos que fueron valuados por la empresa.

**Zona:** datos de la zona, en general conserva las características de la ubicación del predio a valuar.

**Delegación:** información de las delegaciones del D.F. para la ubicación de predios.

**Colonias:** datos de las colonias de cada una de las delegaciones que también sirve para referencia de la ubicación del predio.

**Uso de suelo:** catálogo de referencia para determinar el uso de suelo del predio a valuar.

### Estructuras de las tablas del Sistema

Avalúos      Identificador: **Avaluo**

No.	Identif.	Descripción	Tipo	Long.	Observaciones
1	IdAvaluo	Identificador de referencia secuencial de un avalúo.	Char	7	Llave primaria (PK). Está compuesto de una letra y un número consecutivo.
2	MotivAva	Motivo o asunto del avalúo.	Char	100	
3	IdSolici	Identificador del solicitante.	Char	4	Llave foránea (FK)
4	FechRece	Fecha de recepción de los datos del avalúo.	Date		
5	IdPredio	Clave para identificar el predio en estudio.	Char	3	Llave foránea (FK)
6	IdValuad	Identificador del valuator que realiza el trabajo.	Char	4	Llave foránea (FK)
7	DocAntec	Documento que antecede al avalúo (Oficio).	Char	20	
8	FechEntr	Fecha de entrega del avalúo.	Date		
9	Conclusi	Conclusiones del avalúo.	Char	150	



Solicitantes Identificar: **Solicitante**

<b>Campo</b>	<b>Ident.</b>	<b>Descripción</b>	<b>Tipo</b>	<b>Long.</b>	<b>Observaciones</b>
1	IdSolici	Identificador de la empresa o persona solicitante.	Char	4	Llave primaria (PK)
2	TipoSoli	Tipo de solicitante.	Char	1	
3	Nombre	Nombre de la empresa o persona.	Char	40	
4	Domicili	Domicilio de la empresa o de la persona.	Char	50	
5	Telefono	Números telefónicos.	Char	30	
6	RFC	RFC del solicitante.	Char	13	

Valuadores Identificador: **Valuador**

<b>Campo</b>	<b>Ident.</b>	<b>Descripción</b>	<b>Tipo</b>	<b>Long.</b>	<b>Observaciones</b>
1	IdValuad	Identificador de los peritos valuadores.	Char	4	Llave Primaria (PK)
2	Nombre	Nombre del valuador.	Char	30	
3	TpoValua	Tipo de valuador.	Char	1	
4	Registro	Registros obtenidos en otras instituciones.	Char	50	
5	Especiali	Especialidad que presenta el perito valuador.	Char	40	
6	Eficiencia	Eficiencia desarrollada en los trabajos que se le encomiendan.	Num		Porcentaje
7	Observac	Observaciones sobre el cumplimiento del valuador.	Char	100	

Predios      Identificador: Predio

Campo	Ident.	Descripción	Tipo	Long.	Observaciones
1	IdPredio	Identificar del predio.	Char	3	Campo de la llave Primaria (PK).
2	Calle	Nombre de la calle y número.	Char	30	
3	IdColoni	Clave de la colonia.	Char	3	Campo de la llave primaria (PK)
4	IdDelega	Clave de la delegación.	Char	2	Campo de la llave primaria (PK)
5	IdZona	Clave de la zona.	Char	2	Llave foránea (FK)
6	IdUsoSue	Uso de suelo.	Char	2	Llave foránea (FK)
7	CodPost	Código Postal.	Char	5	
8	NomPropi	Nombre propietario.	Char	30	
9	Superficie	Superficie que se evalúa.	Num		Con dos dígitos después del punto decimal.
10	Objeto	Objeto de la valuación.	Char	30	
11	BoletPre	Boleta predial.	Char	20	
12	Escritura	Folio de la escritura.	Char	20	
13	FactForm	Factor de forma.	Num		Dos dígitos decimales
14	FactFond	Factor de fondo.	Num		Dos dígitos decimales
15	FactSupe	Factor de superficie.	Num		Dos dígitos decimales
16	FactFren	Factor de frente.	Num		Dos dígitos decimales
17	FactUbic	Factor de ubicación.	Num		Dos dígitos decimales
18	FactTopo	Factor de Topografía.	Num		Dos dígitos decimales
19	FactZona	Factor de Zona.	Num		Dos dígitos decimales
20	ValorAva	Valor obtenido en el avalúo.	Num		Dos dígitos decimales

21	FechReal	Fecha de realización del avalúo.	Date		
22	Calle1	Calle colindancia uno.	Char	35	Rumbo y calle
23	Calle 2	Calle colindancia dos.	Char	35	Rumbo y calle
24	Calle 3	Calle colindancia tres.	Char	35	Rumbo y calle
25	Calle 4	Calle colindancia cuatro.	Char	35	Rumbo y calle
26	Linderos	Observaciones con respecto a la colindancia.	Text		

Zonas            Identificador: **Zona**

Campo	Ident.	Descripción	Tipo	Long.	Observaciones
1	IdZona	Identificador de la zona.	Char	2	Llave Primaria (PK)
2	Descrip	Descripción de la zona.	Char	30	

Uso de Suelo            Identificador: **Suelo**

Campo	Ident.	Descripción	Tipo	Long.	Observaciones
1	IdUsoS	Identificador del tipo de uso de suelo.	Char	2	Llave primaria (PK)
2	Descrip	Descripción del uso.	Char	30	
3	Lotemoda	Características del lote moda.	Char	30	
4	DensPobl	Densidad de población.	Char	20	

Delegación    Identificador: **Delegacion**

Campo	Ident.	Descripción	Tipo	Long.	Observaciones
1	IdDelega	Identificador de la delegación.	Char	2	Llave primaria (PK)
2	NombDele	Nombre de la delegación.	Char	30	

Colonia      Identificador: Colonia

Campo	Ident.	Descripción	Tipo	Long.	Observaciones
1	IdColoni	Identificador de la colonia.	Char	3	Llave primaria (PK)
2	NombColo	Nombre de la colonia.	Char	30	

**Zona/Suelo/Delega**

Campo	Ident.	Descripción	Tipo	Long.	Observaciones
1	IdZona	Identificador de la zona.	Char	2	Campo de la llave primaria (PK)
2	IdUsoSue	Clave del uso de suelo.	Char	2	Campo de la llave primaria (PK)
3	IdDelega	Clave de la delegación.	Char	2	Campo de la llave primaria (PK)
4	InteCons	Intensidad de Construcción.	Char	30	
5	UsoSuel	Clave del uso de suelo de la zona.	Char	2	Llave foránea (FK)
6	AguaPota	Tiene agua potable.	Char	1	
7	Drenaje	Tiene drenaje.	Char	1	
8	Paviment	Tiene pavimentación.	Char	1	
9	EnerElect	Tiene energía eléctrica.	Char	1	
10	CableSub	Tiene cableado subterráneo.	Char	1	
11	RedTelef	Tiene red telefónica.	Char	1	
12	RecoBasu	Tiene recolección de basura.	Char	1	
13	Vialidad	Tiene vialidades.	Char	1	
14	TVxcable	Tiene TV por cable.	Char	1	

15	ComeServ	Tiene comercios y servicios.	Char	1	
16	Educacio	Tiene educación.	Char	1	
17	Asistenc	Tiene asistencia.	Char	1	
18	Diversio	Tiene diversión.	Char	1	
19	Recreaci	Tiene recreación.	Char	1	
20	Trabajo	Tiene trabajo.	Char	1	
21	ParqPubl	Tiene parques públicos.	Char	1	
22	Iglesias	Tiene Iglesias.	Char	1	
23	ServTele	Tiene servicios de telecomunicaciones (Correos, telégrafos y teléfonos).	Char	1	
24	Contamin	Contaminación en la zona.	Char	30	
25	Observac	Observaciones.	Texto		

### 3.4 Diseño y Construcción de la base de datos

A continuación se construye la base de datos con la información del diccionario de datos, para lo cual se utilizará Access como ya se mencionó.

El sistema esta conformada por cinco tablas que se obtuvieron como entidades del presente trabajo, tres que son catálogos de referencia y una mas que resultó del análisis del modelo entidad relación.

Para poder llevar a cabo esta tarea, es necesario seguir los siguientes pasos:

1. Crear la base de datos en Access. (figura 3.4.1).

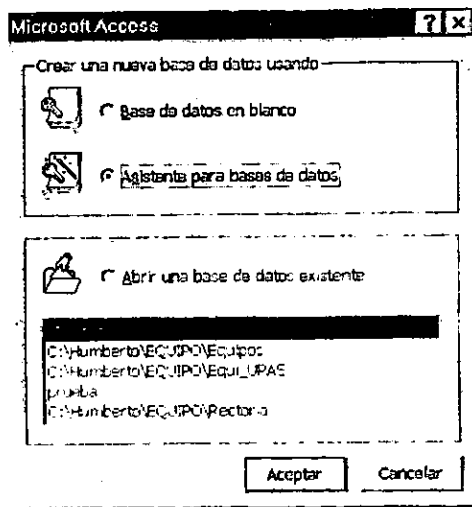


Figura 3.4.1 Forma para la creación de una base de datos Access.

Para crear la base de datos podemos utilizar dos caminos, uno crearla sin ningún elemento, sólo la estructura, o utilizar el asistente, con el cual se crea a imagen de otras bases de datos prediseñadas por Access de problemas típicos, misma que podemos modificar agregando, modificando o eliminando tablas, campos o relaciones. Se decide crearla en blanco debido a que ninguna es acorde a la necesidades de nuestro sistema, en donde podemos observar que la creación de la base de datos no refleja ningún elemento tal y como se muestra en la figura 3.4.2.

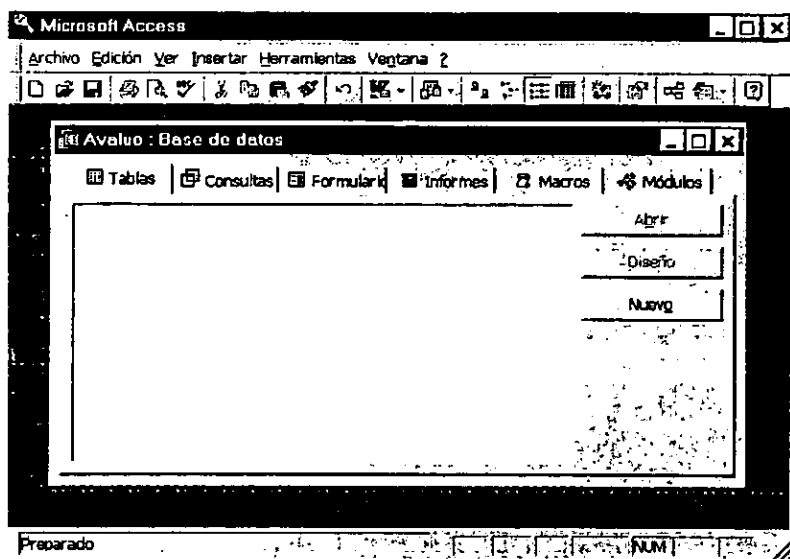


Figura 3.4.2 Creación de la base de datos sin elementos.

## 2. Crear las tablas que componen al sistema.

La creación de las tablas que conforman el sistema requiere del diccionario de datos que se realizó en pasos anteriores, cada uno de las definiciones que se dieron son reflejadas aquí para el almacenamiento de información real sobre el proceso de avalúos.

Cada tabla se crea y se va incorporando en la forma de Base de datos de Microsoft Access, también es necesario durante la creación la definición de la llave primaria de cada una de las tablas y para su creación también se cuenta con un asistente, donde nos indica paso a paso como crear la tabla, tal y como se muestra en la figura 3.4.3.

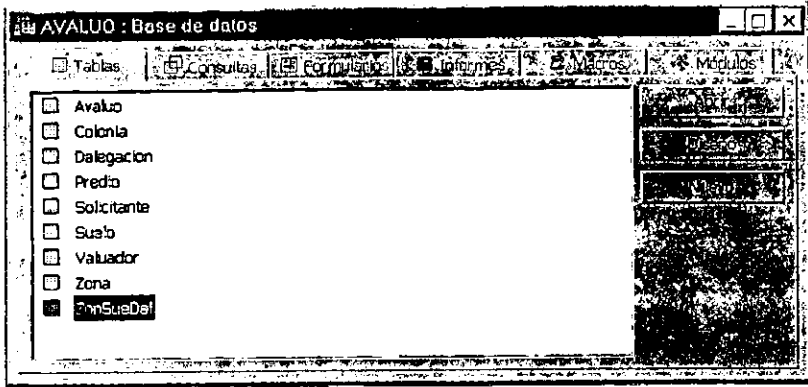


Figura 3.4.3 Presentación de la forma de base de datos de Access con las tablas del sistema creadas.

### 3. Generar la relación que existe entre ellas.

La relación entre cada una de las tablas se da por el esquema de entidad relación, es decir, cada una de las llaves primarias de cada tabla en otro tabla se refleja como una llave foránea donde exista la necesidad de una referencia hacia datos de esa tabla, tal y como se muestra en la figura 3.4.4.



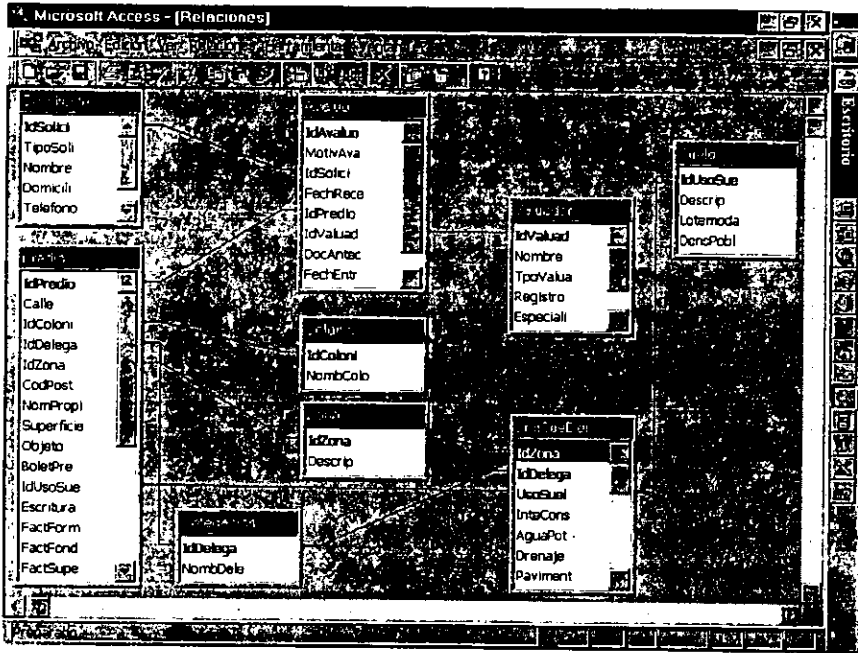


Figura 3.4.4 Relaciones entre las tablas del sistema.

### 3.5 Diseño y construcción del Front End

El uso de técnicas apropiadas de programación y el conocimiento de algún lenguaje de alto nivel nos permite el desarrollo de aplicaciones bajo ambiente Windows, Para lograr esto es básico conocer algunos conceptos de este ambiente:

**Independencia del hardware:** las aplicaciones desarrolladas solicitan la utilización de los recursos de una computadora y no los controlan.

**Gestión de memoria:** Windows es un sistema operativo que utiliza técnicas para el manejo de la memoria y no se limita a tener un solo programa residente en la misma.

**Multitarea:** a pesar de que los equipos donde se utiliza Windows están conformados por un solo procesador, se ha logrado atender varias tareas simultáneamente a gran velocidad.

**Entorno homogéneo:** se presentan todas las aplicaciones en un ambiente sencillo y estándar, fácil de aprender por parte del usuario.

**Intercambio de datos entre aplicaciones:** debido a que es un ambiente multitarea, muchas veces se requiere intercambiar o utilizar información entre las aplicaciones a través de un proceso de intercambio de datos estandarizado como el "Dynamic Data Exchange" (Intercambio dinámico de datos).

Todas estas características facilitan enormemente el desarrollo de aplicaciones, pues ahorra tiempo de programación, al no tener que controlar periféricos o manejo de memoria, ni alguna otra función que el sistema operativo le corresponde controlar.

Estas y otras características son utilizadas por Visual Basic, lenguaje de programación que se toma como Front End.

El desarrollo o diseño de aplicaciones con VB se realiza básicamente en dos etapas:

**Interfaz con el usuario:** para aplicaciones nuevas es la etapa inicial de diseño, pues se define la interfaz gráfica de la aplicación antes de escribir el código que determinará su comportamiento.

Aquí es donde se trabaja con los diferentes elementos de VB que permiten crear la parte Visual del desarrollo de la aplicación, es decir, sólo se define la parte de los controles contenidos en las ventanas (denominadas comúnmente formas) de la aplicación que se explicaran un poco más adelante.

**Programación:** Una vez definida la interfaz gráfica que interactúa con el usuario, es necesario definir las tareas que se van a realizar cada vez que el usuario realice una acción dentro del ambiente Windows (denominados comúnmente eventos).

Dependiendo de la complejidad de nuestros programas, se tendrá cierto código, que será ejecutado cuando se presente el evento al que se encuentra asociado. La programación de VB consiste en definir procedimientos llamados al presentarse los eventos producidos ya sea por el usuario, por el sistema o por otras aplicaciones; esta es la razón por la que la aplicación está conducida por eventos.

Existen elementos básicos que se utilizan dentro de la aplicación que forman parte del ambiente homogéneo de Windows, los cuales son las ventanas y los menús:

### **Ventanas**

Estos elementos regularmente llamados formas en VB, son parte del medio homogéneo que nos proporciona Windows, al igual que los menús que más adelante se describen.

Cada una de las formas que crean la interfaz gráfica, de hecho, son la base donde se colocan los demás objetos que nos dan la apariencia deseada (fig. 3.5.1). Cada una de estas ventanas nos permiten darle apariencia a nuestro sistema, ya que podemos cambiar de posición, tamaño, color o quitarle y agregarle objetos según trabaje el usuario.

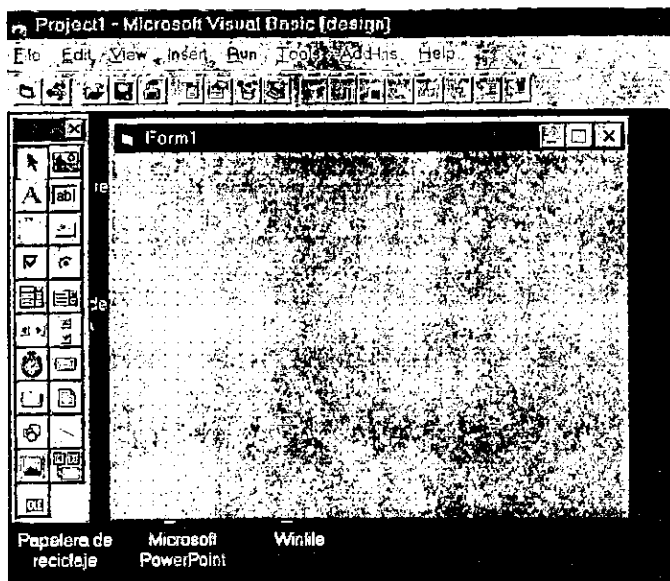


Figura 3.5.1 Presentación de una forma simple sin elementos.

Por sí sola, una ventana o forma, hablando en términos de VB también tiene propiedades, métodos y eventos que nos permiten modificar o alterar el curso del programa.

Propiedades que se utilizan:

**BackColor:** esta propiedad nos permite cambiarle el color del fondo, dando la apariencia deseada.

**Caption:** aquí se coloca el mensaje o título de nuestra ventana, indicando al usuario el lugar donde está colocado dentro del sistema.

**FontName:** nombre de la fuente que utiliza para desplegar el título de la forma o cualquier otro texto dentro de ella.

Nota: cada objeto se puede definir con tipos de letras diferentes.

**FontSize:** para indicar el tamaño de la fuente utilizada.

**MaxBotton:** permite habilitar o deshabilitar el botón de maximizar, esta opción es utilizada para que el control de las ventanas del sistema, sea sólo a través de los objetos de las formas del sistema.

**MinBotton:** esta opción al igual que la anterior deshabilita el botón de minimizar.

**Name:** propiedad que permite cambiarle el nombre a nuestras formas con un nombre más acorde a lo que se esta trabajando, con lo cual su referencia se hace de forma más sencilla dentro de la programación.

**WindowState:** esta propiedad se le asigna el valor "2" de maximizar en algunas de las formas, esto para que ocupe toda la pantalla con la información que va a desplegar.

Eventos que se emplean:

**Load:** este evento es el único que se utiliza y entre otras cosas nos permite la inicialización de variables, objetos, propiedades o el llamado a procedimientos o funciones que se requiere al comenzar a trabajar con una forma.

Métodos:

**Hide:** este método resulta muy útil para ocultar una forma mientras se manda llamar otra sin perder contacto con ella, esto nos permite no reiniciar la forma varias veces.

**Show:** para mostrar cada una de las formas desde el comienzo se tuvo que invocar y para ello se utilizó este método, excepto para la primer forma que se almacena en memoria automáticamente.

## Menús

Al igual que las ventanas, estos objetos facilitan mucho el manejo de la aplicación y es de los más utilizados para seleccionar las diferentes funciones del programa (fig. 3.5.2). Obviamente la estandarización de los nombres y distribución de las diferentes opciones muestran lo amigable que puede ser una aplicación.

A cada una de las opciones de los menús se les agregó la llamada "Tecla rápida", es decir, se le agregó una letra subrayada con la que al utilizar la tecla Alt y la letra nombrada accesa en forma inmediata a la opción que indica (fig. 3.5.3).



Figura 3.5.2 Menú creado para el sistema de valuación.

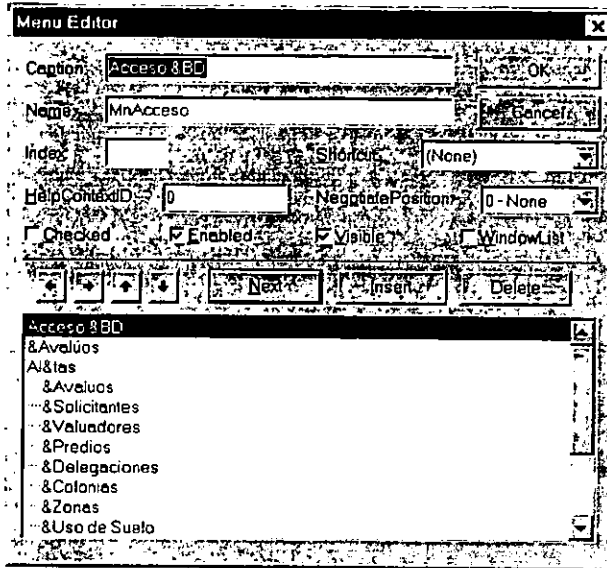
Los menús creados para el sistema presentan las siguientes características:

**Caption:** Nombre con el que el usuario lo identifica con la función que realiza. En cada nombre se le coloca un "&" para indicar cual es la tecla rápida.

**Name:** nombre con el que se identifica en la programación.

**Enabled:** Habilita el menú, el cual puede cambiar en caso de que el usuario no tenga los privilegios para esa opción.

**Visible:** Indica que opciones tendrán que mostrarse para el usuario.



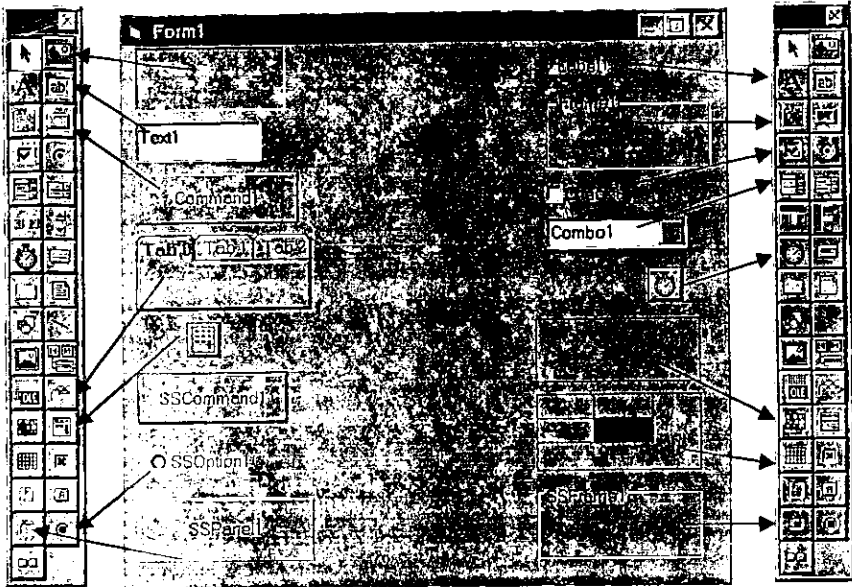
### 3.5.3 Propiedades de los menús creados para el sistema.

Además para cada una de las opciones se utilizó el evento `Clik`, el cual manda llamar a un procedimiento u otro menú.

Hasta el momento se han definido dos de los objetos de VB que se utilizan dentro del sistema y que son la base fundamental de Windows y de cualquier aplicación, pero, la principal ventaja de VB se forma a través de los controles que se agregan a cada una de las formas, cada uno de estos elementos tienen propiedades, eventos y métodos que se pueden emplear para el desarrollo del sistema.

Cada una de estas propiedades son aplicables exclusivamente a un objeto, pero otras son similares permitiendo el polimorfismo de objetos.

Estos objetos se muestran en la figura 3.5.4, además de su referencia en la ventana de herramientas de Visual Basic.



**Figura 3.5.4 Controles empleados en el sistema de valuación.**

Los elementos empleados en el sistema son:

- Picture.
- Text.
- Command.
- SSTab.
- Crystal Report.
- SSFrame.
- SSPanel.
- Label.
- Frame.
- Check.
- Combo.
- Grid.
- SSCommand.



- SSOption.
- Timer.
- Picture Clip.

## Picture

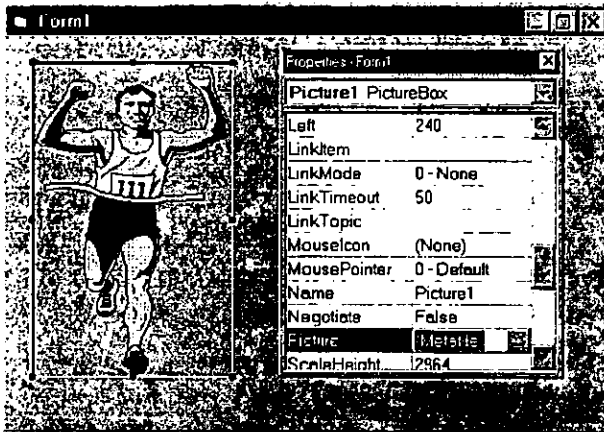


Figura 3.5.5 Selección de un objeto picture y la ventana de propiedades.

Elemento que nos permite mostrar imágenes para darle presentación al sistema, es utilizado en varias de las formas donde se muestra el logotipo del sistema o alguna imagen. Además nos permite desplegar cualquier imagen de tipo bitmap, ícono o archivo Windows Metafile (fig. 3.5.5).

Propiedades utilizadas:

**Picture:** se indica el nombre de la imagen que despliega en el momento de la ejecución de la aplicación.

Las demás propiedades se toman con los valores por omisión que no afectan la ejecución del programa.

En cuanto a los métodos y eventos de este objeto no se utiliza ninguno, ya que como se mencionó, sólo es utilizado para desplegar una imagen de presentación de logotipo u otra imagen.

## Text

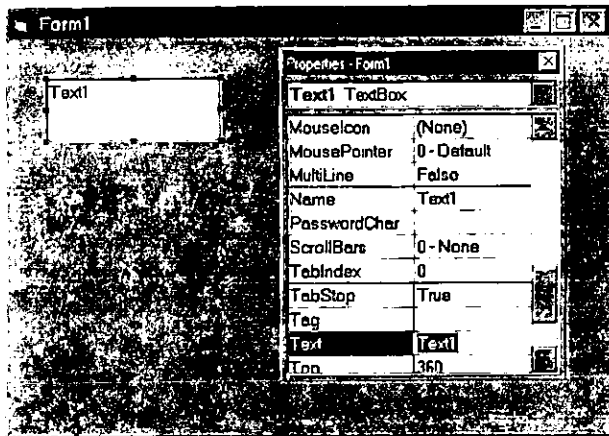


Figura 3.5.6 Presentación de un control tipo text y algunas de sus propiedades.

Objeto que nos permite introducir, actualizar o borrar información referente a una variable, campo o simplemente nos muestra la información resultante de un procedimiento. Este elemento es uno de los más útiles dentro de VB, ya que como se ha dicho es un medio muy eficaz para actualizar la información de la Base de Datos. Además de que tiene algunas propiedades que nos facilitan la programación (fig. 3.5.6).

Este es uno de los medios donde el usuario utilizando el teclado, puede introducir sus datos al sistema y que además con un click del mouse puede captar el enfoque para editar su contenido.

Cabe mencionar que los datos que aparecen dentro de una caja de texto por si solas no pueden actualizar un dato, sino que la información que contiene tiene que estar asociada con otro objeto que si pueda realizar esta operación.

Las propiedades utilizadas son:

**Datafield:** cuando la caja de texto está asociada con un objeto "Data Control", esta propiedad toma gran importancia, ya que aquí se enlaza con un campo de la tabla a la que este referenciado.

**DataSource:** Nombre del Data Control con el que está asociado.

**Enabled:** esta propiedad es bastante recurrida en la programación, ya que si el campo no debe ser modificado en alguna parte del sistema, sólo cuando ocurra cierta operación como crear un registro nuevo, actualizar, etc., y deba introducirse información se desbloquea, de lo contrario se deshabilita la posibilidad de modificar su contenido.

**FontName, FontSize, ForeColor:** Propiedad donde se indica el tipo de letra que presentara durante la ejecución del programa, tamaño y color respectivamente.

**Locket:** propiedad similar a la propiedad enabled, pero a diferencia de esta última, si se puede asignar la atención hacia el objeto, pero igualmente no permite la modificación de la información.

**Maxlength:** esta propiedad nos permite definir la cantidad máxima de caracteres que el usuario pueda introducir, por lo que ya no tenemos que estar revisando la longitud de cada texto después de haberlo actualizado.

**Name:** nombre con el que se reconoce en el programa para poder actualizar su contenido.

**PasswordChar:** se pone un carácter especial como el "\*", cuando no se desea mostrar en pantalla los caracteres introducidos, tal es el caso de la contraseña al iniciar el sistema, manteniendo un medio de seguridad.

**TabIndex:** índice que al utilizar la tecla tab hace que no se pierda la secuencia de objetos en una forma.

**Text:** esta es la propiedad más importante, ya que es el lugar donde se almacena su contenido y de donde se puede tomar también para utilizarse en otro o dentro del mismo procedimiento u objeto.

**Visible:** cuando no es necesario o no es utilizado una caja de texto se puede ocultar mediante esta propiedad.

En cuanto a los métodos empleados para este objeto, no se emplearon, ya que su principal ventaja radica en la relación de sus propiedades y sus eventos.

**Click:** este evento se utiliza para indicar las instrucciones que debe realizar cuando capta la atención del programa, si debe llamar a otros eventos, procedimientos o funciones, aunque regularmente se emplea sólo para captar la atención y realizar la edición del cuadro de texto.

**KeyPress:** este evento se utiliza para indicar que debe hacer en caso de oprimir "x" o "y" tecla, además de que se realiza la validación del texto y se puede controlar la secuencia del programa.

## Command

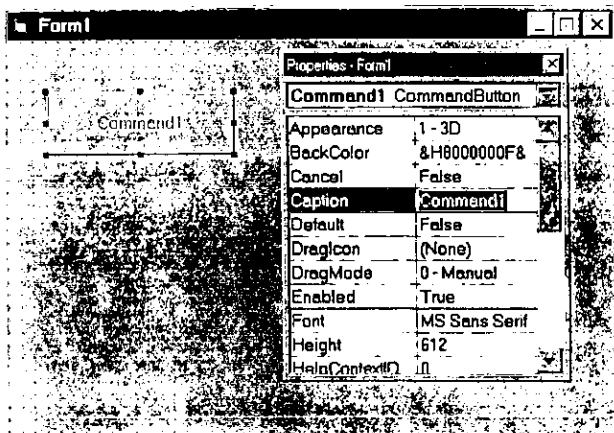


Figura 3.5.7 Objeto Command con algunas de sus propiedades.

Este objeto se utiliza para iniciar, interrumpir o finalizar algún proceso en el sistema. La cantidad empleada fue menor que el `SSCommand`, debido a que se requería un sistema más gráfico y con este no se podía dar el aspecto deseado (fig. 3.5.7).

Para realizar cualquier operación sólo se utilizó un evento, el cual en la mayoría de los objetos donde se almacena o realiza una operación se emplea, este evento es el "Click".

Propiedades empleadas:

**Caption:** nombre que despliega para el usuario final, generalmente se le pone una palabra relacionada con el proceso que realiza.

**Enabled:** habilita el botón para su uso.

**FontName, FontSize:** nombre de la fuente y tamaño empleada para desplegar el nombre del botón respectivamente.

**Name:** nombre que sirve de referencia para el desarrollo de la lógica del programa.

**TabIndex:** posición dentro de la forma que tendrá al pasar de un objeto a otro empleando la tecla tab.

**Visible:** propiedad para indicarle que estará visible en ciertas operaciones de la ejecución del programa.

Eventos y métodos empleados:

Como se mencionó anteriormente sólo se emplea el evento click, ya que con esto se indica el inicio de las instrucciones asociadas al botón, ya sea de alta, baja, modificación, impresión, llamada a otros objetos, etc.

## SSTab

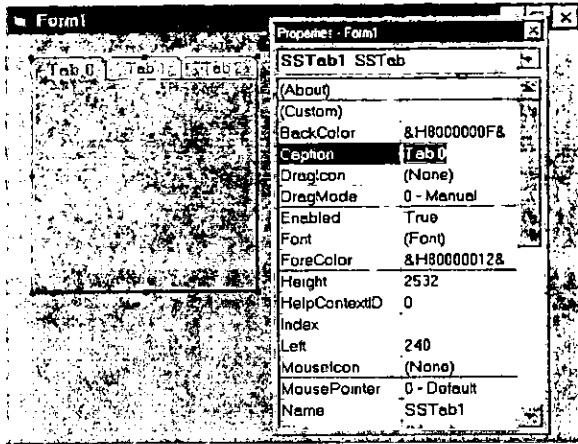


Figura 3.5.8 Objeto SSTab con algunas de sus propiedades.

Objeto que nos permite manejar de forma fácil la presentación de diferentes pantallas de información en una sola forma. Este objeto nos permite crear una serie de tabuladores (Tab's) en forma de folders, los cuales se pueden incrementar según las necesidades de la aplicación (fig. 3.5.8).

En nuestro caso se empleo este objeto para separar cada una de las pantallas de actualización de las diferentes entidades, dando además de una presentación agradable en tercera dimensión para el usuario, más sencilla que si se genera en diferentes pantallas (formas de VB).

Propiedades empleadas:

**Caption:** propiedad donde se indica el nombre de la información que presenta en cada tab. Este nombre se visualiza en cada una de las pestañas.

**Font:** Propiedad para indicar el tipo de fuente, tamaño y estilo que se emplea en la información que está dentro del objeto SSTab.

**ForeColor:** esta parte muestra un cuadro de colores donde se indica el color que se utiliza como fondo dentro de cada Tab.

**Style:** formato de presentación del SSTab, ya sea en forma de MsOffice o estilo Windows 95, los cuales no afectan mucho en la presentación.

**TabsOrientation:** sólo se emplea si se desea cambiar la orientación de nuestros folders.

**TabsperRow:** propiedad donde se indica cuantos folders por renglón debe presentar, ya que esta opción puede variar según el tipo de letra empleado, tamaño de la fuente o el mismo tamaño del título del Tab.

Eventos y métodos empleados:

**Click:** evento que permite desplazarnos de una pantalla del SSTab a otro al presionar el botón izquierdo del mouse sobre la pestaña deseada.

### Crystal Report

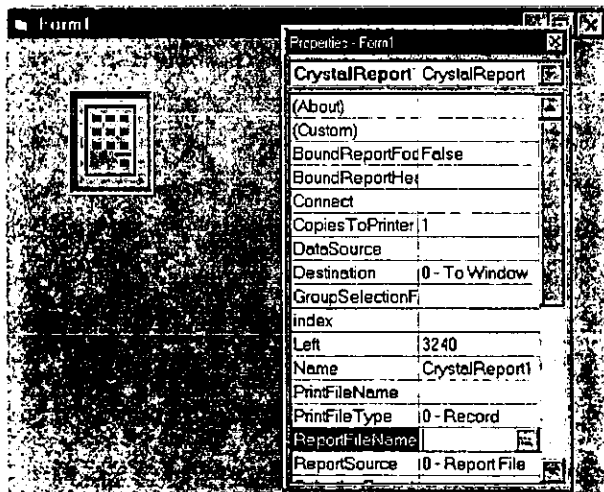


Figura 3.5.9 Crystal Report y algunas de sus propiedades.

Este tipo de control de VB se encarga de mostrar e imprimir información de la base de datos sin necesidad de realizar grandes rutinas de programación sobre impresión de reportes. Crystal Report abre la base de datos de sólo lectura, y se tiene el control sobre las características de la fuente (tipo, tamaño y estilo), alineación del texto, inclusión de gráficos, además de contar con una vista preliminar (fig. 3.5.9).

La creación de reportes se desarrolla en forma separadas de las aplicaciones y puede ser llamado desde VB.

Los reportes elaborados incluyen diferentes tipos de campos como:

- Campos de bases de datos.
- Campos de texto usados como etiquetas.
- Campos de fórmulas resultado de una expresión.
- Campos especiales (Fechas de impresión, número de página, número de registro, etc.) y
- Campos para gráficos.

Para realizar el llamado del reporte sobre la aplicación VB, primero se debe tener el archivo RPT con el reporte elaborado y se incluye en nuestra forma el icono de Crystal Report mostrado anteriormente y se configuran las siguientes propiedades:

**ReportSource:** Se indica que el origen de datos se realiza desde un archivo Report, es decir se selecciona la opción "0".

**ReportFileName:** nombre del archivo del reporte correspondiente que llamará al momento de su ejecución.

Por último se asocia a un botón Command la ejecución del reporte al asignar la propiedad Action del objeto Crystal Reports el valor "1" de ejecución.



## SSFrame, Frame y SSPanel

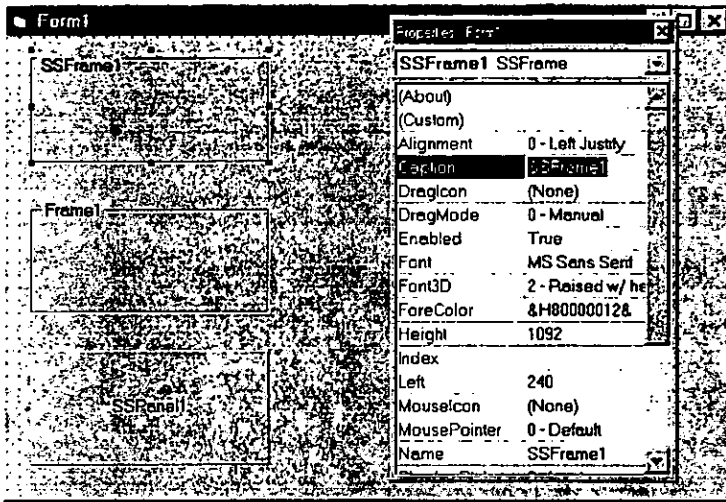


Figura 3.5.10 Frame, SSPanel y SSFrame y alguna de las propiedades de este último.

Estos objetos de VB nos permitieron agrupar otros objetos para realizar el relieve y la presentación al usuario.

El Frame y SSFrame son empleados además de la agrupación de objetos de una sola entidad para darle mayor presentación a las pantallas y en el caso del SSFrame mostrarla en 3D (Tercera dimensión) (fig. 3.5.10).

El SSPanel también nos permite agrupar objetos de una misma entidad, pero además realiza su colocación en tercera dimensión. Dentro de este objeto se colocaron los Frames para darle relieve y marco simple o en tercera dimensión según el Frame empleado.

En este caso no se emplearon propiedades excepto el Caption, donde se cambió su contenido para dar mayor referencia a la información que contenía el Frame y en otros casos se eliminó.

No se emplearon eventos y métodos ya que como se mencionó anteriormente sólo fueron empleados para darle mejor presentación a las pantallas.

## Label

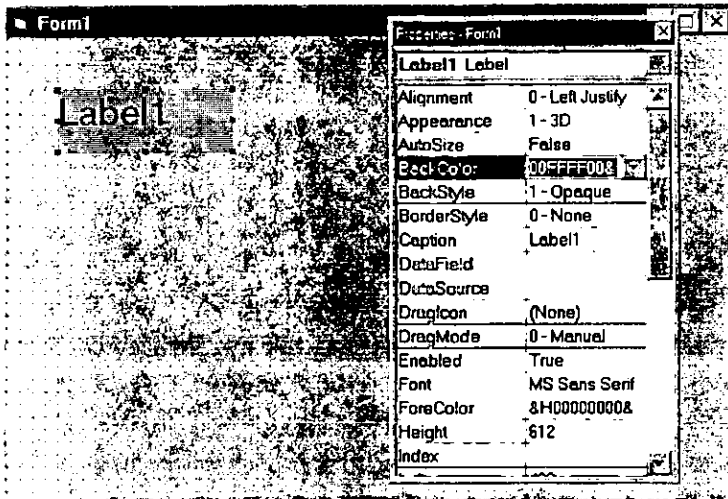


Figura 3.5.11 Objeto Label y algunas de sus propiedades.

Este objeto es empleado en todas las secciones del sistema para mostrar las etiquetas de la información que se almacena en cada caja de texto o combo (fig. 3.5.11).

Propiedades empleadas:

**Caption:** Esta propiedad es la más importante de las empleadas porque se modifica para asignarle el texto que da la información al usuario.

**Font:** Aquí se indica el tipo, tamaño y estilo de la fuente que se emplea para mostrarla en pantalla.

En el caso de los eventos y métodos, no se emplea ninguno, porque el objeto sólo se utiliza para mostrar información de referencia al usuario.

## Check

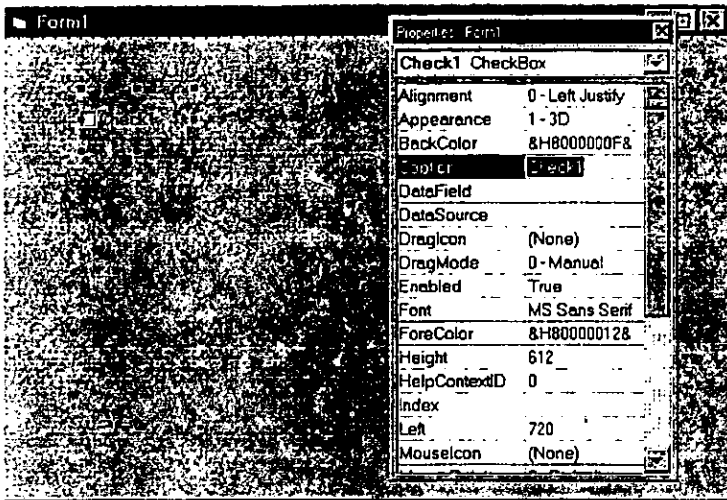


Figura 3.5.11 Objeto Check y algunas de sus propiedades.

Objeto empleado para captar opciones de verdadero o falso. Dentro del sistema existen opciones donde debe decir si existe o no algún factor y para ello es la opción más recomendable a emplear (fig. 3.5.11).

Propiedades empleadas:

**Caption:** se emplea para mostrar la leyenda de la opción que habilita o deshabilita.

**Font:** tipo, tamaño y estilo de la fuente que se emplea para la leyenda de la opción.

**Name:** propiedad que se utiliza para dar un nombre con el cual podemos emplearla en la programación para tomar su valor para otras opciones o hacer referencia en otros procedimientos.

**Value:** se emplea para asignarle el valor verdadero o falso del factor que se desea habilitar o deshabilitar.

Eventos y métodos empleados:

Sólo se emplea el evento Click para activar o desactivar el Check que será almacenado o empleado para otras operaciones al oprimir un botón de la pantalla donde aparece.

**Combo:**

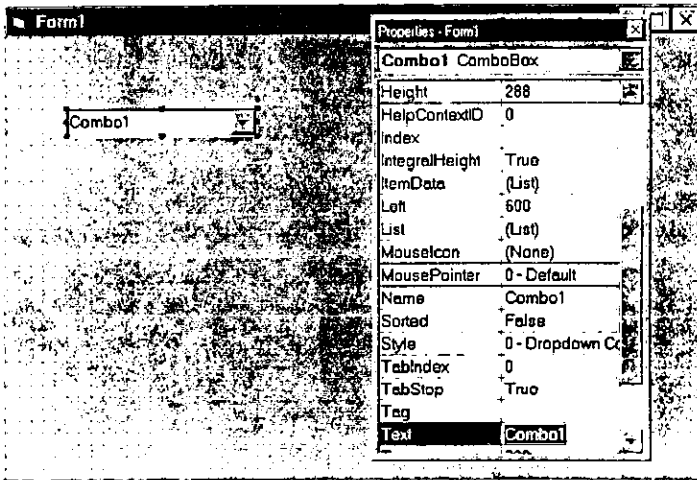


Figura 3.5.12 Objeto Combo y algunas de sus propiedades.

Objeto que se emplea de dos formas, una de ellas como caja de texto, donde se puede agregar información para una lista que se almacena y la segunda para seleccionar una de las opciones de la lista. En el sistema se emplea para seleccionar opciones dadas de alta de alguna tabla en su respectivo folder y se requiere emplear para completar la información dentro de otra sección del sistema (fig. 3.5.12).

Propiedades empleadas:

**Font:** propiedad que se emplea para asignar el tipo, tamaño y estilo de fuente que mostrará el control.

**ItemData:** característica que se emplea para almacenar valores de referencia según las opciones almacenadas.

**List:** almacenamiento de las opciones que se emplean en el combo.

**Name:** nombre de referencia que tiene dentro de la programación de VB.

**Style:** opción para indicarle cual será la presentación que mostrará en la pantalla, el estilo empleado en el sistema es DropDown List (Tipo lista donde no se permite agregar nuevas opciones por el usuario).

Eventos:

**Click:** este evento se ejecuta cuando se presiona el botón izquierdo del mouse y con ello se selecciona la opción deseada del combo y asigna la clave correspondiente en una caja de texto.

**KeyPress:** evento empleado para asignar la atención del cursor a otro control VB empleando la tecla tab o enter.

Métodos:

**Clear:** limpia la información del combo cuando se desea reasignar las opciones después de actualizar los datos a los cuales está relacionado.

## Grid

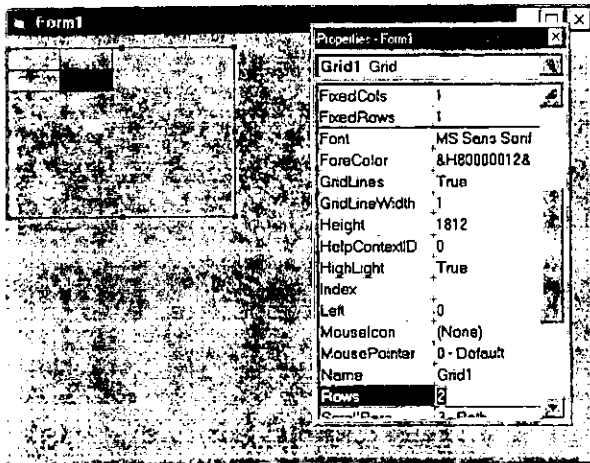


Figura 3.5.13 Objeto Grid y algunas de sus propiedades.

Este objeto nos permite mostrar la información en una serie de renglones y columnas similar a Excel, pero que para efectos de programación cada celda (intersección de una columna y un renglón) se toma en forma independiente (fig. 3.5.13).

Además este objeto permite almacenar texto o imágenes.

Las propiedades empleadas son:

**Name:** Nombre con el que se realiza una referencia durante la programación.

**ScrollBars:** Habilita las barras de desplazamiento para poder ver la información que no se puede ver en pantalla.

**Visible:** Muestra el Grid al elegir la opción en donde se realiza el cálculo automático del valor del predio.

**Cols:** Propiedad donde se indica la cantidad de columnas del grid empleado.

**Rows:** Propiedad donde se indica la cantidad de renglones del grid empleado.

**FixedCols:** Valor que indica cuantas columnas servirán de título del grid.

**FixedRows:** propiedad para indicar la cantidad de renglones que servirán de referencia al grid.

**Text:** lugar donde se almacenan los valores de cada una de las celdas, después de haber elegido el renglón y la columna correspondiente.

**ColWidth:** longitud de la columna donde se almacena determinado valor.

**CellSelected:** Indica la celda seleccionada al detectar el evento click.

Métodos:

**AddItem:** método empleado para agregar nueva información al grid.

Eventos:

**Click:** evento que se realiza sobre el grid para seleccionar el predio que servirá de referencia para el cálculo del valor unitario del predio que se está valuando.

## SSCommand

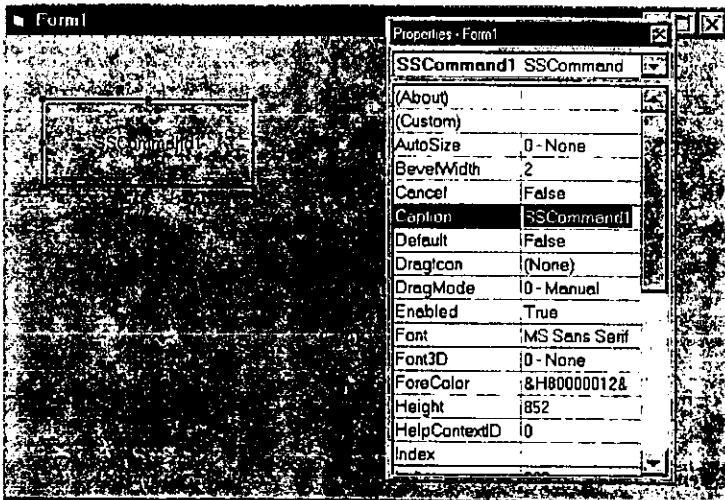


Figura 3.5.14 Objeto SSCommand y algunas de sus propiedades.

Este objeto al igual que la caja de texto es de los más empleados y a diferencia de los botones de comando sin las letras "SS" en estos se puede generar un efecto en tercera dimensión, además de que se puede incluir imágenes para la creación de menús con botones que da una mejor presentación y nos brinda más ventajas para el usuario ya que es más amigable (fig. 3.5.14).

Las propiedades empleadas son:

**Caption:** se coloca un texto para indicar la función del botón. En algunos casos se omite para dar lugar a la propiedad Picture.

**Font y Font3D:** opción para configurar el tipo, tamaño y estilo de letra y en el caso de 3D para darle efecto de tercera dimensión.

**Name:** propiedad donde se indica el nombre con que puede hacerse referencia dentro de la programación.

**Picture:** imagen que puede presentar y con ello en lugar de ser un botón con una leyenda se muestra con esta imagen que muchas veces puede ser mas representativo que un mensaje.

Evento empleado:

**Click:** Con este evento se manda llamar la función correspondiente del botón.



## SSOption

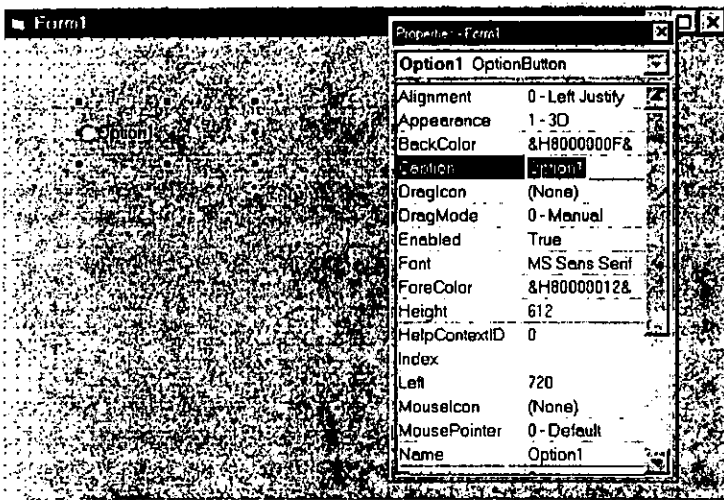


Figura 3.5.15 Objeto Option y alguna de sus propiedades.

Objeto empleado para seleccionar sólo una de varias opciones, a diferencia del Check que podían estar más de una opción habilitada. Este control es sencillo pero de gran utilidad (fig. 3.5.15).

Las propiedades que se emplean de este objeto son:

**Caption:** Leyenda que aparece del lado derecho del control y describe su función.

**Font:** propiedad para configurar el tipo, tamaño y estilo de fuente para la propiedad Caption.

**Name:** Nombre con el que se hace referencia dentro de la programación.

**Value:** valor que tiene el control para indicar que acciones deberá realizar cuando se seleccione.

Eventos:

**Click:** este evento se ejecuta cuando se presiona el botón izquierdo del mouse y con ello se selecciona la opción deseada del Option.

**KeyPress:** evento empleado para asignar la atención del cursor a otro control VB empleando la tecla tab.

## Timer

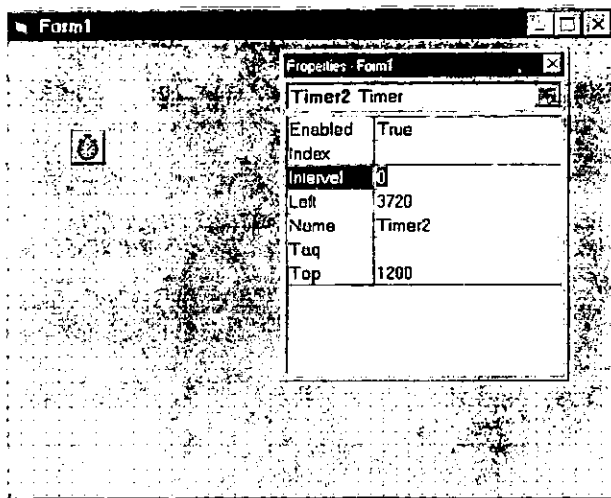


Figura 3.5.16 Objeto Timer y sus propiedades.

Este objeto y el Picture clip se utilizaron para desplegar el logotipo con movimiento. Este objeto en especial permitió darle los intervalos entre cada una de las imágenes para simular el movimiento (fig. 3.5.16).

Las propiedades empleadas de este objeto son:

**Enabled:** habilita el reloj para que inicie el conteo y al finalizar inicie de nuevo.

**Interval:** propiedad donde se le indica el intervalo que tendrá el reloj.

**Name:** nombre con el que se identifica en la programación.

Para el Timer no se emplearon ni métodos ni eventos.

## Picture Clip

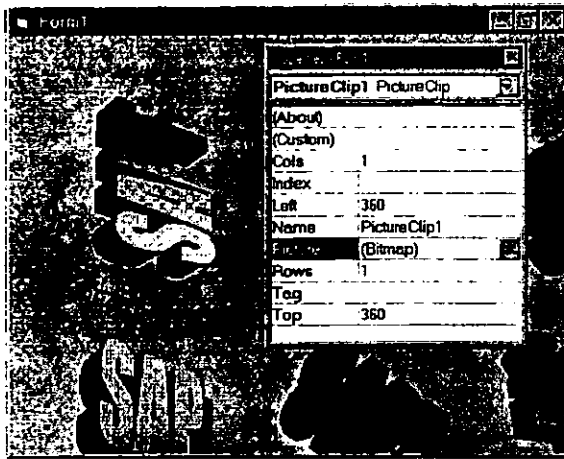


Figura 3.5.17 Objeto Picture Clip y algunas de sus propiedades.

Este objeto se emplea para mostrar por secciones una misma imagen, en este, caso en una misma imagen se colocaron varios títulos en secuencia simulando el movimiento de un logotipo y al combinarlo con el Timer se ve que en realidad tiene movimiento por sí solo (fig. 3.5.17).

Las propiedades empleadas son:

**Cols y Rows:** Número de columnas y renglones en que se desea que se divida la imagen para mostrarla.

**Name:** nombre con el que se hace referencia en la programación.

**Picture:** en esta propiedad se indica la imagen que deberá mostrarla.

No se emplearon métodos ni eventos, ya que la asignación del Picture Clip al Picture donde mostrará la sección correspondiente se hace por programación.

Hasta este momento se ha realizado la descripción de los objetos empleados en el sistema, pero como se mencionó anteriormente el desarrollo de un sistema desde el punto de vista de VB, se forma de dos partes, la parte visual y la parte de programación.

Ahora nos referiremos a la parte Visual es decir a las pantallas creadas para el sistema:

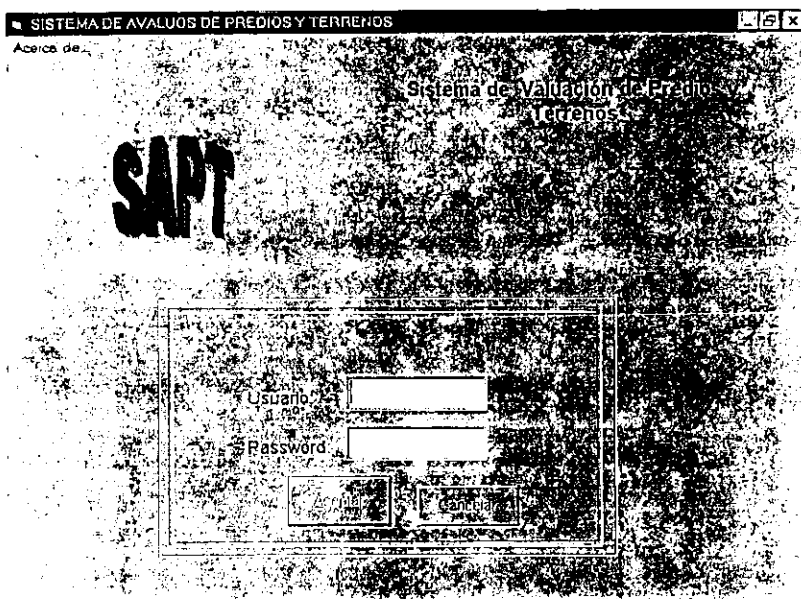


Figura 3.5.18 Pantalla de acceso al sistema.

En esta forma se crea la pantalla de acceso al sistema, se emplean objetos Text, Command, Label, SSFrame, Menú, SSPanel y los objetos Timer, Picture y el Picture Clip para mostrar el movimiento del logotipo (fig. 3.5.18).

Esta pantalla es muy sencilla, en ella se colocan los controles Text para capturar la información que el usuario deberá teclear (Nombre de usuario y Contraseña), en el caso del nombre sólo se recibe un string y se valida, para el caso de la contraseña se

coloca automáticamente la propiedad de PasswordChar para colocar un asterisco en lugar del carácter teclado, esto se realiza por seguridad.

Los controles Command se emplean para cancelar la entrada al sistema y para aceptar la información almacenada en los controles Text. Una vez aceptada la información se valida que el usuario esté dado de alta y se verifica el nivel de acceso, permitiendo entrar con privilegios "1" que representa todos los derechos, "2" todo, excepto la parte de administración del sistema y "3" sólo consultas y reportes.

También nos muestra el logotipo del sistema con movimiento, para lo cual se emplea el objeto Timer, Picture Clip y un objeto Picture.

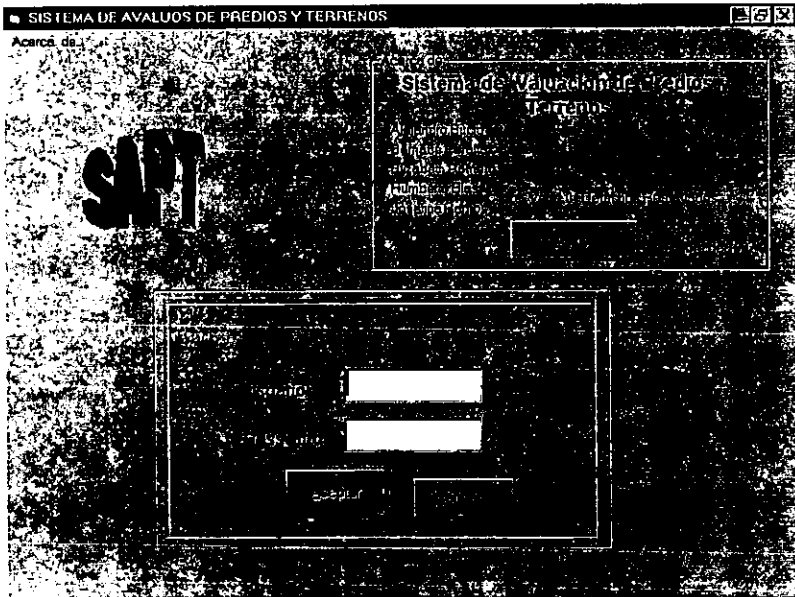


Figura 3.5.19 Pantalla de acceso al sistema con la opción habilitada de acerca de....

Además de lo anterior esta pantalla tiene un pequeño menú “Acerca de...”, el cual nos muestra información referente al sistema, versión, fecha de revisión, personas que lo realizan, etc. Ver figura 3.5.19.

Dentro de esta pantalla se colocó un control Command para ocultar el Frame donde aparecen los datos del sistema.

Las pantallas que siguen están dentro de un mismo objeto, este objeto nos ha permitido crear varias pantallas en una sola forma y obviamente es el objeto principal de la ventana, este objeto es el SSTab.

En la figura 3.5.20 se muestran los campos del solicitante, estos elementos están dentro de una de las pestañas del SSTab y cada una de las nueve pestañas nos muestran información diferente.

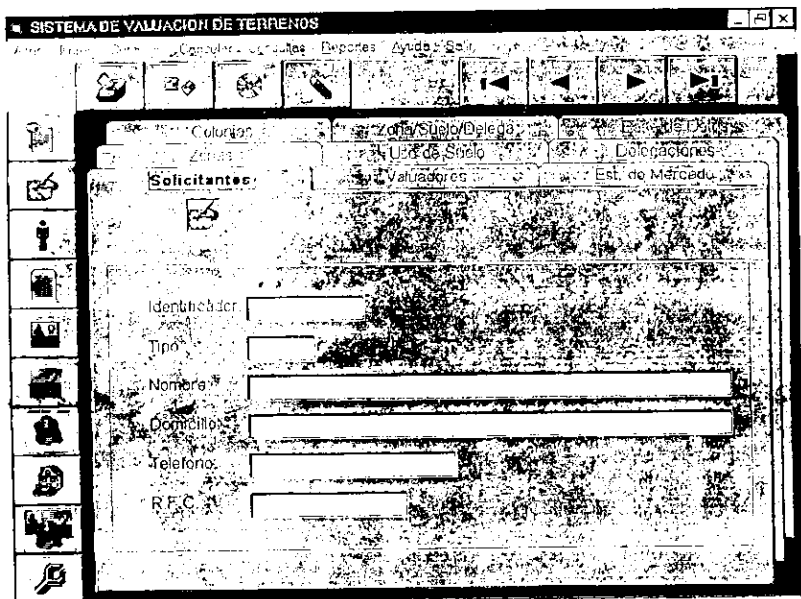


Figura 3.5.20 Pantalla con los campos del solicitante.

Esta forma está compuesta de varias secciones, en la parte superior esta el menú, dentro del cual se pueden elegir las diferentes opciones con las que cuenta el sistema. Se puede dar de alta cualquier elemento al elegir "Altas" y después la entidad donde se desea dar de alta, ya sea Solicitantes, Zonas, Colonias, etc; de igual forma para "Bajas", "Cambios", "Consultas" y "Reportes". Además de estas opciones cuenta con la opción de "Cancelar" que permite deshacer alguna operación que no esté concluida y la "Salir" del sistema.

Abajo del menú cuenta con una barra de botones, los cuales tienen una imagen de la tarea que realizan, de izquierda a derecha están: Altas, Bajas, Cambios, Cancelar Primero, Anterior, Siguiente y Ultimo, los cuales pueden ser empleadas en cualquiera de los folders del SSTab, pero sólo afectan al folder activo, los demás permanecen intactos. Estos botones, los menús y las opciones que están en la parte izquierda de la pantalla se habilitan o deshabilitan según los privilegios del usuario que haya accedido.

Al lado izquierdo de la pantalla está un menú de botones, el cual nos permite entrar a cada una de los folders con los que cuenta el SSTab. Cada uno de estos botones tiene una figura representativa de la opción que habilita.

Por último en el centro de la pantalla está el objeto principal. Este objeto almacena cada uno de los campos de las identidades que se manejan en el sistema, es decir la parte que se requiere para la realización de un avalúo.

Las partes con las que cuenta el SSTab son:

- Solicitantes.
- Zonas.
- Colonias.
- Valuadores.
- Uso de Suelo.

- Subzonas (Delegación, Uso de Suelo y Zonas).
- Estudio de Mercado (Información de predios).
- Delegaciones.
- Base de datos o mantenimiento del sistema.

En la figura 3.5.20 se muestra la pantalla con los campos del solicitante, estos campos son almacenados en campos Text, cada uno de estos campos se validan dependiendo de la información que se requiera.

Además se emplearon objetos Label para indicar el nombre del campo que se debe almacenar en cada uno de los controles Text.

Para realizar algún movimiento sobre la información, se emplean los botones de la barra de botones superiores.

Si se desea ir al folder del solicitante se puede hacer con un clic en la pestaña que dice "Solicitantes", o dando un clic al botón que tiene una imagen de una mano escribiendo en una hoja de papel, esta imagen esta también debajo de cada uno de los títulos de las pestañas para que los usuarios se familiaricen más con las opciones del sistema.



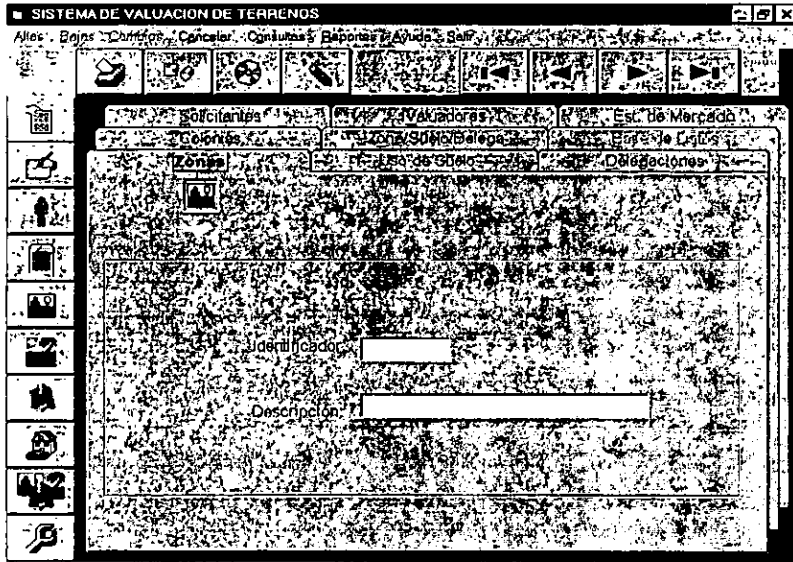


Figura 3.5.21 Pantalla de los campos de zonas

En la figura 3.5.21 se muestra la pantalla referente a la información de zonas. Este folder sólo cuenta con dos controles Text y dos Label. En cada uno de los campos Text se realiza la validación respectiva.

Para acceder a este folder se logra dando un Click sobre la pestaña que dice "Zonas" o seleccionando el botón que tiene una Zona desértica con un cactus en primer plano.

Para realizar una alta, baja, cambio o trasladarse sobre los diferentes registros se emplean los botones comunes de la barra de botones de la parte superior o seleccionando el menú respectivo y el submenú "Zonas".

En esta pantalla se capturan las diferentes zonas que puede haber dentro del Distrito Federal y que ya están determinadas por el Plan de Desarrollo General Urbano del D.F.

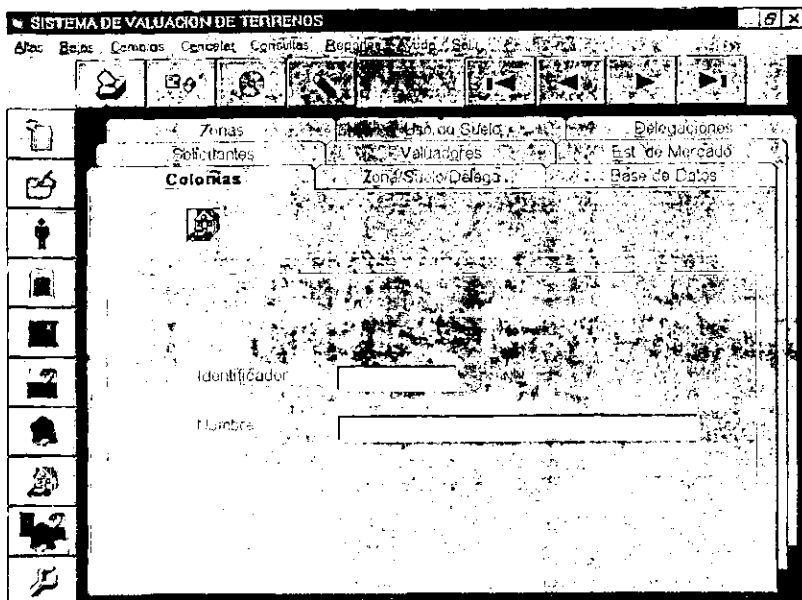


Figura 3.5.22 Pantalla con la información de colonias.

Esta pantalla al igual que la anterior tiene poca información y sólo se emplean dos controles Text y dos Label, cada control Text tiene su respectiva validación y los Label nos dan la referencia al tipo de información que permite almacenar los primeros (fig. 3.5.22).

En esta pantalla se actualiza la información referente a las colonias y también es información que sirve de referencia para la realización de los avalúos.

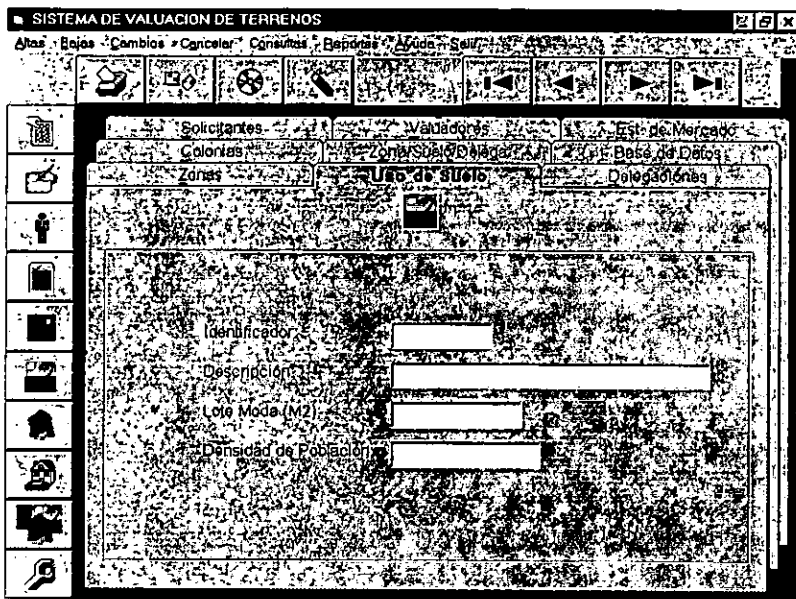
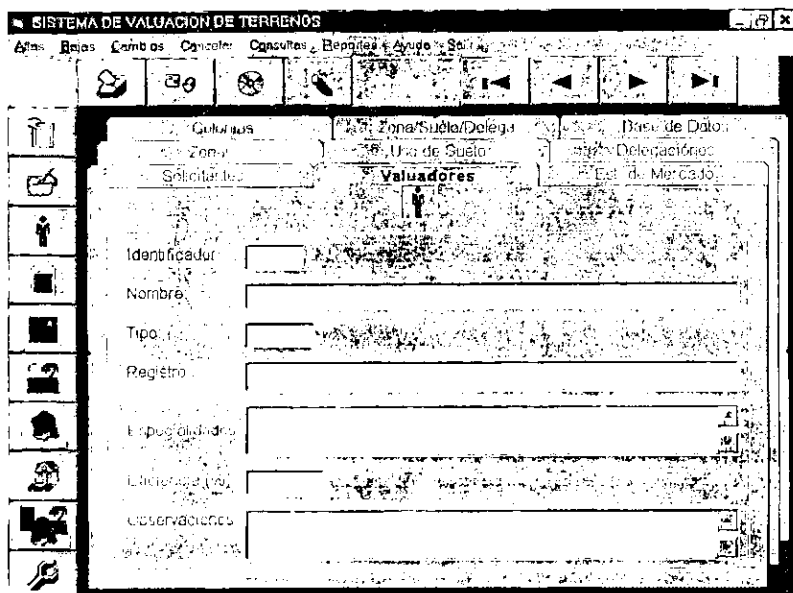


Figura 3.5.23 Pantalla con los campos referentes al uso de suelo.

Este folder nos muestra los campos de lo que se refiere a Uso de Suelo, cuenta con cuatro controles Text y cuatro Label. Para cada uno de los controles tipo Text se hace la validación correspondiente y nuevamente los Label son la referencia para cada uno de los campos (fig. 3.5.23).

Cada una de las especificaciones mencionadas de los folders también se aplican aquí, es decir, la de los botones superiores, el menú y para acceder al folder, además de la pestaña nombrada "Uso de Suelo" con el botón que tiene una isla.



**Figura 3.5.24** Pantalla de los datos de los valuadores.

La pantalla (fig. 3.5.24) referente a los valuadores contiene más elementos, debido a que se requiere mayor cantidad de datos que en las anteriores. Con los datos de los valuadores es necesario saber a detalle cierta información para poder elegir al más capaz para realizar determinado avalúo.

Como esta entidad tiene mayor cantidad de datos, se requiere también mayor cantidad de controles. Se tienen siete controles Text y los mismos Label. Cada uno de los campos tienen sus respectivas validaciones.

Las validaciones que se realizan en cada campo dependen del tipo de información del campo, por ejemplo para campos tipo texto, las validaciones que se realizan son de longitud, que sean de determinado tipo de caracteres y no numéricos o si es de tipo numérico, que se introduzcan exclusivamente números.

En esta pantalla se tienen validaciones de tipo texto y numéricos, además de que algunos de los controles Text se les agregó la posibilidad de navegar hacia arriba o hacia abajo, ya que hacen referencia a campos donde se almacena gran cantidad de información y que no se alcanza a mostrar en el tamaño del control.

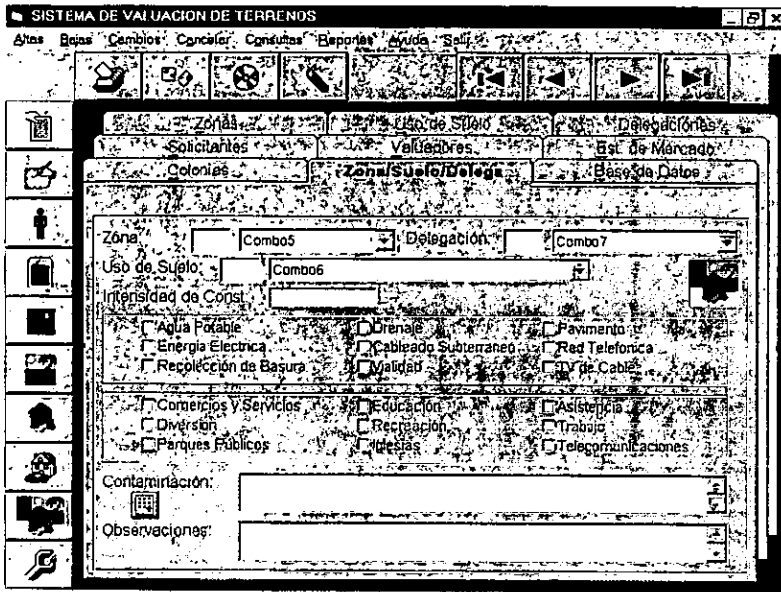


Figura 3.5.25 Pantalla con los campos de las subzonas.

Esta pantalla (fig.3.5.25) se refiere a las subzonas que es la relación que mantienen las zonas, delegaciones y el uso de suelo que puede haber dentro de estas. Esta pantalla presenta una cantidad mayor de controles empleados, entre ellos se encuentran, los text, combo, check, SSFrame, Crystal Report y Label.

Los elementos Combo son para elegir la zona, delegación y uso de suelo. Estos elementos se colocaron para facilitar al usuario en la elección de estos campos, ya que muchas veces es más fácil si vemos cuales son las opciones disponibles que el tener que memorizar las claves. Una vez seleccionada la opción de la lista del combo,

se coloca automáticamente la clave dentro de una caja de texto para futuras referencias.

Los elementos Check se emplean para que el usuario pueda indicar si la opción a la que se refiere esta presente o carece de la misma. Se emplean 18 elementos de este tipo y están divididos en dos secciones con dos SSFrame, los cuales indican los factores de Infraestructura Urbana y de Equipamiento Urbano. Los valores que puede tener son, blanco cuando carece de ese factor y una paloma cuando está presente.

Los campos Text que están al final de la pantalla se refieren a características que son variables, es decir que puede tener o no, pero que su descripción es más o menos larga cuando la tiene y con las barras de desplazamiento podemos ir hacia arriba o abajo según se requiera para textos muy largos.

Los controles Label, nos dan la referencia de la opción a la que se refiere el control Text o Combo, ya que los campos Check tienen su propia etiqueta.

En cuanto al objeto de Crystal Report, no necesariamente debe estar dentro de esta pantalla, ya que sólo nos sirve para hacer la referencia a cada uno de los diferentes reportes, podríamos decir que más bien esta asociada a la forma.

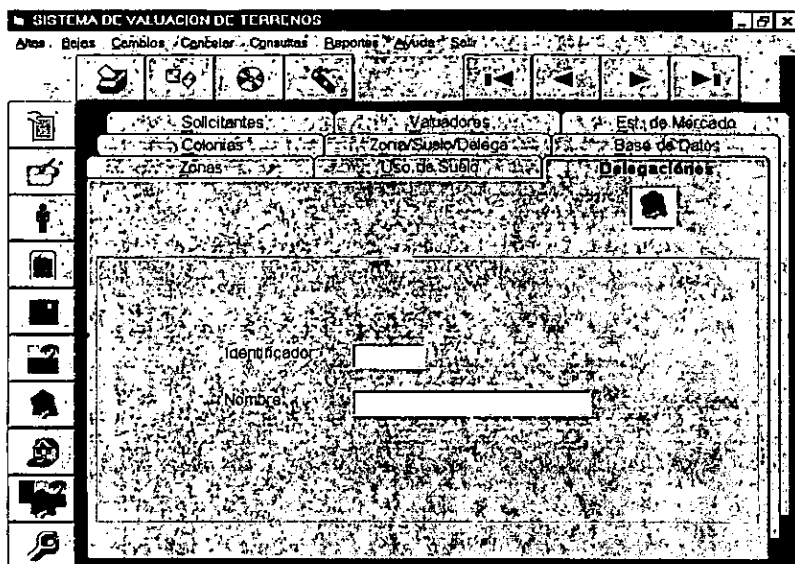
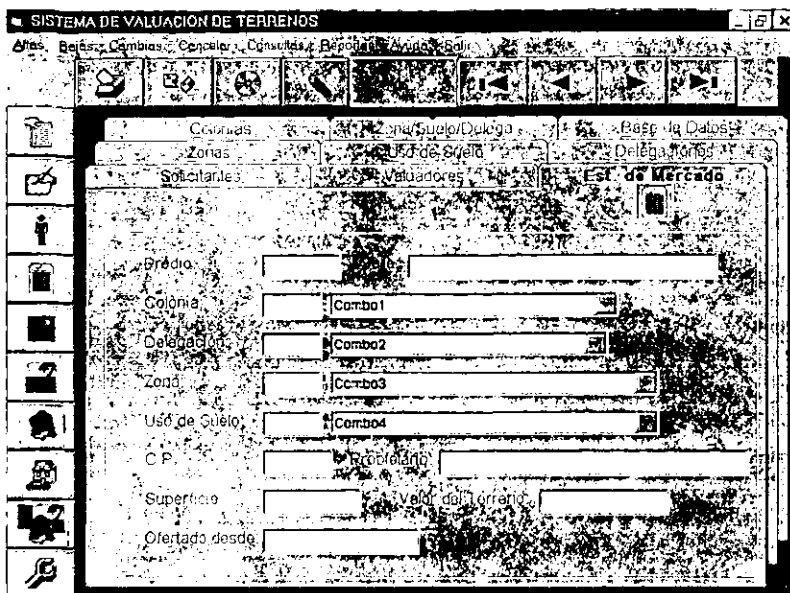


Figura 3.5.26 Pantalla con los campos de delegaciones.

Esta pantalla (figura 3.5.26) también tiene pocos elementos y se refiere a las delegaciones, sólo se manejan dos campos y por lo tanto a dos controles tipo Text y dos Label.

Para la actualización de sus campos también se emplean los controles comunes de la parte superior.

La validación es sencilla, y se toman las misma consideraciones que en los campos de las pantallas anteriores.



**Figura 3.5.27 Pantalla de estudio de mercado o datos de los predios.**

Este folder se emplea para dos aspectos importantes de lo referente a los avalúos. Uno de ellos se refiere al estudio de mercado, es decir, que los predios investigados por parte del perito valuador se actualizan aquí, se les asigna su respectivo identificador y se capturan las características propias del predio y las características de la zona, se actualizan en el folder de subzona, ya que estos últimos pueden afectar a todos los predios que se encuentren dentro de la misma subzona. Por otro lado, también se refiere a los predios que se generaron por la realización de un avalúo, pero en forma independiente, que al estar integrados forman parte del estudio de mercado para futuros avalúos (fig. 3.5.27).

Esta pantalla emplea controles tipo Text, Combo y Label. Los controles Combo se emplean para seleccionar la colonia, delegación, zona y uso de suelo de los que ya están dados de alta y automáticamente se despliega la clave en el Text a la izquierda del Combo. Algunos de los Text se emplean para editar otros campos referentes a predios. Y, al igual que los otros folders, se emplean los controles Label para mantener



una referencia de los datos que se están actualizando en cada uno de los campos Text y combo.

La validación se realiza por campo y se hace en función del tipo de dato que se almacene en cada control, generalmente son controles Text.

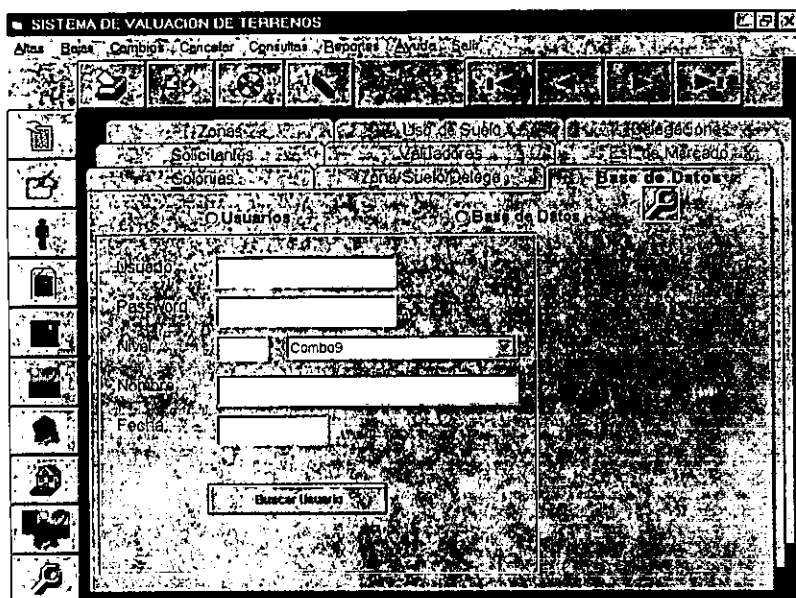


Figura 3.5.28 Pantalla de base de datos o soporte en la parte de usuarios.

En lo que se refiere al folder de Base de Datos, está dividido en dos secciones. La primera se refiere a los usuarios, para ello se emplea un control Combo para elegir el nivel de seguridad que se desea para ese usuario y los demás son controles tipo Text, Label y Command. Los controles Text almacenan los campos de los usuarios y los label mantienen la referencia del campo con el Text. El objeto Command se emplea para realizar una búsqueda rápida por clave de usuario.

Como se mencionó en el párrafo anterior, este folder está dividido en Usuarios y la segunda en Base de Datos, la primera se representa en la figura 3.5.28, la cual permite actualizar a los usuarios que accesan al sistema y los niveles de seguridad que se tiene dentro de este y el segundo se muestra en la figura 3.5.29 que nos permite darle mantenimiento a la base de datos, compactarla, encryptarla y desencryptarla.

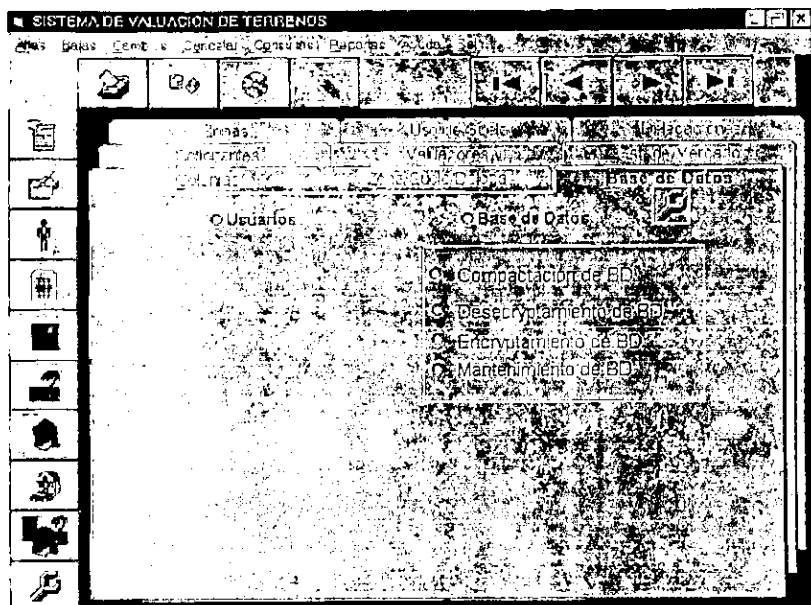


Figura 3.5.29 Pantalla de base de datos o soporte en la parte de usuarios.

Hasta el momento nos hemos enfocado a los elementos que vamos a emplear para la realización de un avalúo, pero a continuación veremos una sección que nos muestra las pantallas donde se realiza el avalúo.

Estas pantallas están más completas, ya que se toman todos los datos de los folders explicados anteriormente, a excepción del folder de usuarios, pero los demás forman parte de los datos de un predio en particular.

La información se divide en dos secciones debido a que es mucha y no cabe en una sola pantalla.

The screenshot shows a software window titled "AVALUO" with a sub-header "Datos Generales". The form contains the following fields and controls:

- No. Secuencial: Text input field.
- Fecha de Recepción: Text input field.
- Asunto: Text input field.
- Solicitante: Text input field.
- Predio: Text input field.
- Oficio Notal: Text input field.
- Valuador: Text input field.
- Fecha de Entrega: Text input field.
- Observaciones: Text area with scrollbars.
- Calle: Text input field.
- Calle 2: Text input field.
- Delegación: Text input field.
- Zona: Text input field.
- Uso de Suelo: Text input field.
- Propietario: Text input field.
- Objeto: Text input field.
- Escritura: Text input field.
- Calle 1: Text input field.
- Calle 3: Text input field.
- Linderos y Coincidencias: Text area with scrollbars.

At the bottom of the window, there are two buttons: "SIGUIENTE" and "REGRESAR".

Figura 3.5.30 Primera pantalla para la realización de un avalúo.

En la primera pantalla (figura 3.5.30) se toman los datos más relevantes del predio, datos del avalúo que se está realizando, datos generales y particulares del predio y referencia de los documentos con los que se cuentan en relación al avalúo del predio en proceso.

Dentro de esta forma se emplean varios tipos de controles, entre los de mayor frecuencia están los Text, Label y Combo, respectivamente. Al igual que en las pantallas anteriores se hace mención que los controles Text son empleados para editar los campos. Los controles Combo se emplean para elegir de una lista actualizada aquel que corresponda con el predio y los Label para darnos referencia de los datos a actualizar.

Además de los objetos antes mencionados están dos controles Command, los cuales nos permiten avanzar o a la siguiente página o cancelar el avalúo.

Las validaciones de cada uno de los campos se realiza también en función del tipo de dato que se almacene en dicho control.

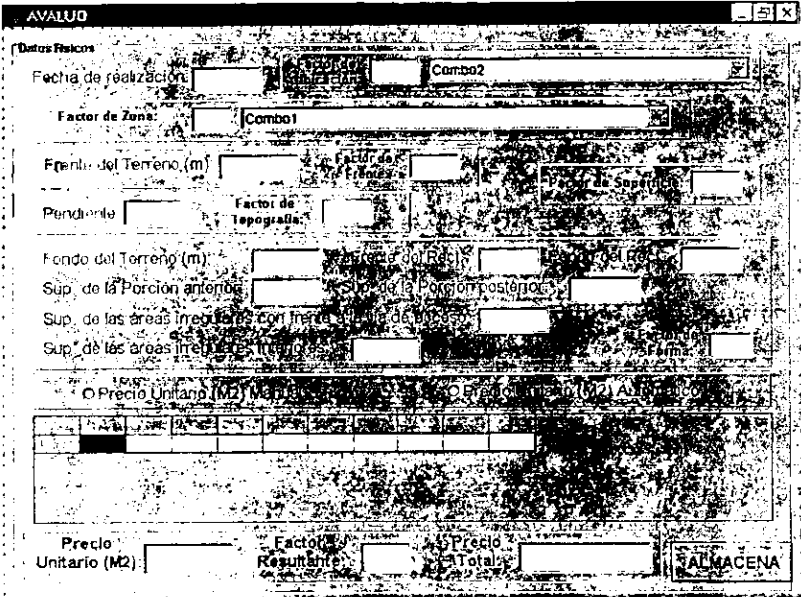


Figura 3.5.31 Segunda pantalla para la realización de un avalúo.

Esta segunda pantalla (figura 3.5.31) esta relacionada con los datos que se almacenaron en la pantalla previa, esto se debe a que se realizan algunos cálculos para la obtención del valor unitario del predio que se está valuando.

Se emplean controles Text para almacenar información que el usuario proporciona, en otros se muestra el resultado de una selección o el resultado de una operación. El cálculo de valores representa el resultado de algún factor que toma como referencia otros de la pantalla anterior o de la misma.

Además se emplearon controles SSFrame para separar cada uno de los factores y se tuvieron más claros los datos que se requieren para obtener un valor determinado, en resumen para organizar mejor los datos de la pantalla.

Se emplearon algunos controles Combo para seleccionar ciertas características y esto nos arroja el valor de algún factor.

Los objetos Label se emplean para dar una referencia de los datos a actualizar, introducir o los resultados de los mismos.

También se emplearon dos objetos Option, los cuales nos permiten seleccionar la forma de cómo obtener el resultado del valor del predio. Uno de ellos es en forma manual, que al seleccionarlo, se pasa a la parte donde se coloca el valor unitario por metro cuadrado que se desee y con el factor resultante que se obtiene automáticamente nos da el valor del predio que se está valuando. El otro es en forma automática, el cual al seleccionarlo se despliega un grid con la información de la base de datos que sean más acordes con las características del predio que se está valuando y de donde se puede elegir cualquiera, para obtener el valor del predio tomando como referencia el valor unitario por metro cuadrado del predio elegido.

Cabe mencionar que los predios que se actualizaron como estudio de mercado se agregan en esta sección, además de los ya existentes en la Base de Datos, pero que cumplen con las características del que se está valuando, con ello se amplían las opciones para obtener el valor del avalúo.

La segunda parte del desarrollo de un sistema en Visual Basic se refiere a la parte de la programación, ésta se realiza después de la parte gráfica.

Una vez que se tienen los controles de la interfaz que va a interactuar con el usuario, se requiere determinar las tareas que se van a realizar cuando el usuario efectúe determinadas acciones.

Para realizar esto se seleccionan los eventos donde se requiere que el control del programa actúe de alguna forma, por ejemplo si es un campo Text se desea realizar cierta validación, en el evento Keypress se puede realizar con las teclas que se presionan.

Para cada control, solamente los eventos que tengan código definido son los únicos que responderán de acuerdo al código escrito y por omisión ningún evento tiene código definido.

Además de los procesos anteriores que son disparados por eventos, también es posible definir procedimientos que serán llamados explícitamente en el código de la aplicación. Este tipo de procedimientos invocados por el código, generalmente son rutinas de procesos específicos (leer, calcular o grabar) sobre datos (imágenes, voz, archivos, gráficos, Texto).

**Forma de Acceso**

<i>Eventos</i>	<i>Descripción</i>
KeyPress	Verifica que cuando se presione la tecla enter la habilite como tabulador.
Load	Inicializa el reloj para presentar el logotipo con movimiento.

<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
BtnOkValidate	Procedimiento	Habilita el botón de Aceptar cuando se valide la función <b>DatalsComplete</b> .
DatalsComplete	Funcion	Verifica que el Text de Nombre de usuario y Contraseña tengan información.

Chpwd	Procedimiento	Inicializa variables, abre la Base de Datos, valida al usuario, habilita las opciones necesarias según el nivel de acceso del usuario y se registra la fecha de acceso.
IsInvalidChar	Función	Valida que se introduzcan caracteres en los text de acceso.

## Controles

### BtnExit (Botón de Cancelar)

<i>Eventos</i>	<i>Descripción</i>
Click	Cambia el puntero del mouse y descarga la forma de acceso.

### btnOK (Botón de Aceptar)

<i>Eventos</i>	<i>Descripción</i>
Click	Cambia el puntero del mouse, deshabilita el botón btnExit y btnOK y manda llamar el procedimiento chpwd.

### CommAceptar

<i>Eventos</i>	<i>Descripción</i>
Click	Oculto el frame que muestra la información del sistema.

### MnAcerca (Menu Acerca de...)

<i>Eventos</i>	<i>Descripción</i>
Click	Muestra el frame que contiene la información del sistema.

### Timer 1 (Reloj)

<i>Eventos</i>	<i>Descripción</i>
Timer	Define el ciclo que debe seguir el reloj y la sección que debe mostrar del Picture Clip sobre el objeto Picture.

**TextPassword (Text de Contraseña)**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del texto, permitiendo exclusivamente caracteres.

**TxtUserId (Text de Usuario)**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del texto, permitiendo exclusivamente caracteres.

**Forma de Folders o Principal.**

<b>Eventos</b>	<b>Descripción</b>
Load	Manda llamar procedimientos de llenado de datos.

<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
Cmb1_colonias	Procedimiento	Llena los datos del combo de colonias con la información de la tabla respectiva.
Cmb2_delegaciones Cmb7_delegaciones	Procedimiento	Llena los datos del combo de delegaciones con la información de la tabla respectiva.
Cmb3_zonas Cmb5_zonas	Procedimiento	Llena los datos del combo de zonas con la información de la tabla respectiva.
Cmb4_uso Cmb6_uso Cmb8_uso	Procedimiento	Llena los datos del combo de uso de suelo con la información de la tabla respectiva.
cmb9_permisos	Procedimiento	Llena los datos del combo de permiso de los usuarios.
Llena_colonias	Procedimiento	Abre objeto dynaset sobre la tabla de colonias.
llena_delegaciones	Procedimiento	Abre objeto dynaset sobre la tabla de delegaciones.
llena_mercado	Procedimiento	Abre objeto dynaset sobre la tabla de predios o estudio de mercado.
llena_solicitantes	Procedimiento	Abre objeto dynaset sobre la tabla de



		solicitantes.
llena_uso	Procedimiento	Abre objeto dynaset sobre la tabla de uso de suelo.
llena_valuadores	Procedimiento	Abre objeto dynaset sobre la tabla de valuadores.
llena_zonas	Procedimiento	Abre objeto dynaset sobre la tabla de zonas.
Llena_zosude	Procedimiento	Abre objeto dynaset sobre la tabla de subzonas.

### Controles

#### Altas

<b>Eventos</b>	<b>Descripción</b>
Click	Agrega un nuevo registro a la tabla según el folder que este habilitado.

#### Bajas

<b>Eventos</b>	<b>Descripción</b>
Click	Elimina un registro de la tabla según el folder que esté habilitado.

#### Cambios

<b>Eventos</b>	<b>Descripción</b>
Click	Actualiza el registro de la tabla según el folder que esté habilitado.

#### Cancelar

<b>Eventos</b>	<b>Descripción</b>
Click	Limpia la pantalla según el folder que esté habilitado.

#### Primero

<b>Eventos</b>	<b>Descripción</b>
Click	Se coloca en el primer registro de la tabla según el folder que esté habilitado.

### Anterior

<i>Eventos</i>	<i>Descripción</i>
Click	Se coloca en el registro anterior de la tabla según el folder que esté habilitado.

### Siguiente

<i>Eventos</i>	<i>Descripción</i>
Click	Se coloca en el siguiente registro de la tabla según el folder que esté habilitado.

### Último

<i>Eventos</i>	<i>Descripción</i>
Click	Se coloca en el último registro de la tabla según el folder que esté habilitado.

### Avalúo

<i>Eventos</i>	<i>Descripción</i>
Click	Manda llamar a la forma que permite capturar la primera parte del avalúo.

### Solicitante

<i>Eventos</i>	<i>Descripción</i>
Click	Activa el folder de SSTab correspondiente a Solicitantes.

### Valuadores

<i>Eventos</i>	<i>Descripción</i>
Click	Activa el folder de SSTab correspondiente a Valuadores.

### Estudio de mercado

<i>Eventos</i>	<i>Descripción</i>
Click	Activa el folder de SSTab correspondiente a Estudio de mercado o

	predios.
--	----------

### Zonas

<b>Eventos</b>	<b>Descripción</b>
Click	Activa el folder de SSTab correspondiente a Zonas.

### Uso de suelo

<b>Eventos</b>	<b>Descripción</b>
Click	Activa el folder de SSTab correspondiente a Uso de suelo.

### Delegaciones

<b>Eventos</b>	<b>Descripción</b>
Click	Activa el folder de SSTab correspondiente a Delegaciones.

### Colonias

<b>Eventos</b>	<b>Descripción</b>
Click	Activa el folder de SSTab correspondiente a Colonias.

### Subzonas

<b>Eventos</b>	<b>Descripción</b>
Click	Activa el folder de SSTab correspondiente a Subzonas (Relación entre delegación, zona y uso de suelo).

### Base de Datos

<b>Eventos</b>	<b>Descripción</b>
Click	Activa el folder de SSTab correspondiente o mantenimiento a la base de datos.

Por cada uno de los tabs:

Solicitantes:

**TxtSIdentif**

<i>Eventos</i>	<i>Descripción</i>
KeyPress	Realiza la validación del campo permitiendo que únicamente se capturen números.

**TxtSTipo**

<i>Eventos</i>	<i>Descripción</i>
KeyPress	Realiza la validación del campo permitiendo que sólo se capturen números y con una longitud máxima de 1.

**TxtSNombre**

<i>Eventos</i>	<i>Descripción</i>
KeyPress	Realiza la validación del campo.

**TxtSDomicilio**

<i>Eventos</i>	<i>Descripción</i>
KeyPress	Realiza la validación del campo.

**TxtSTelefono**

<i>Eventos</i>	<i>Descripción</i>
KeyPress	Realiza la validación del campo.

**TxtSRFC**

<i>Eventos</i>	<i>Descripción</i>
KeyPress	Realiza la validación del campo.

## Zonas

### TxtZIdentif

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo permitiendo que sólo se capturen números.

### TxtZDescrip

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

## Colonias

### TxtCIdentif

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo permitiendo que sólo se capturen números.

### TxtCDescrip

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

## Zona/Suelo/Delega

### Com5Zona

<b>Eventos</b>	<b>Descripción</b>
Click	Dependiendo de la opción seleccionada de la lista se asigna a un campo Text.

### Com6Uso

<b>Eventos</b>	<b>Descripción</b>
Click	Dependiendo de la opción seleccionada de la lista se asigna a un campo Text.

### Com7Delega

<b>Eventos</b>	<b>Descripción</b>
Click	Dependiendo de la opción seleccionada de la lista se asigna a un campo Text.

### TxtIntConst

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Valida el capo de intensidad de construcción.

Los controles Check1 al Check18 no emplearon eventos.

**TxtContamina y Txt Observa no se toma evento.**

### Valuadores

#### TxtVIdentif

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo permitiendo que sólo se capturen números.

#### TxtVNombre

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

#### TxtVTipo

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo permitiendo que sólo se capturen números.

### TxtVRegistro

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

### TxtVEspecialidad

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

### TxtVEficiencia

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

### TxtVObserva

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

### Uso de suelo

#### TxtUIdentif

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo permitiendo sólo números.

#### TxtUDescrip

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

#### TxtULoteM

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

### **TxtUDenPob**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

### **Estudio de mercado**

#### **TxtPredio**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

#### **TxtCalle**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

#### **Com1Colonia**

<b>Eventos</b>	<b>Descripción</b>
Click	Al seleccionar un elemento de la lista actualiza el control Text con la clave de la colonia elegida.

#### **Com2Delega**

<b>Eventos</b>	<b>Descripción</b>
Click	Al seleccionar un elemento de la lista actualiza el control Text con la clave de la delegación elegida.

#### **Com3Zona**

<b>Eventos</b>	<b>Descripción</b>
Click	Al seleccionar un elemento de la lista actualiza el control Text con la clave de la zona elegida.



### Com4Uso

<b>Eventos</b>	<b>Descripción</b>
Click	Al seleccionar un elemento de la lista actualiza el control Text con la clave de uso de suelo elegido.

### TxtCP

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

### TxtPropietario

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

### TxtSuperficie

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

### TxtValor

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

### TxtFechOfe

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

### Delegaciones

#### TxtDIdentif

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**TxDNombre**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**Base de datos****OptUsuario**

<b>Eventos</b>	<b>Descripción</b>
Click	Muestra el frame con la información referente a los usuarios.

**OptBaseD**

<b>Eventos</b>	<b>Descripción</b>
Click	Muestra el frame con la información referente al mantenimiento de la Base de Datos.

**TxtUsuario**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**TxtPassw**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Valida el password y cambia los caracteres tecleados por asteriscos para mantener un grado de seguridad mayor.

**ComNivel**

<b>Eventos</b>	<b>Descripción</b>
Click	Al seleccionar un elemento de la lista actualiza el control Text dependiendo el nivel de acceso del usuario.

**TxtNombre**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**TxtFecha**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación de la fecha.

**CommBusca**

<b>Eventos</b>	<b>Descripción</b>
Click	Habilita el TxtUsuario para introducir la clave de usuario que se desea buscar.

**OptCompac**

<b>Eventos</b>	<b>Descripción</b>
Click	Realiza la compactación de la Base de datos.

**OptDesencry**

<b>Eventos</b>	<b>Descripción</b>
Click	Realiza el desencryptamiento de la Base de datos.

**OptEncrypt**

<b>Eventos</b>	<b>Descripción</b>
Click	Realiza el encryptamiento de la Base de datos.

**OptManteni**

<b>Eventos</b>	<b>Descripción</b>
Click	Realiza un mantenimiento preventivo de la Base de datos de posibles fallas.

**Avalúo****TxtANume**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**TxtAFechR**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**TxtAAsunto**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**ComASolic**

<b>Eventos</b>	<b>Descripción</b>
Click	Según el solicitante seleccionado asigna la clave correspondiente al control text correspondiente

**TxtPredio**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**TxtOficio**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**ComAValuad**

<b>Eventos</b>	<b>Descripción</b>
Click	Según el valuador seleccionado asigna la clave correspondiente al control text correspondiente

**TxtFechE**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**TxtAObserv**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**TxtACalle**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**ComAColonia**

<b>Eventos</b>	<b>Descripción</b>
Click	Según la colonia seleccionada asigna la clave correspondiente al control text correspondiente.

**ComADelega**

<b>Eventos</b>	<b>Descripción</b>
Click	Según la delegación seleccionado asigna la clave correspondiente al control text correspondiente.

**ComAZona**

<b>Eventos</b>	<b>Descripción</b>
Click	Según la zona seleccionado asigna la clave correspondiente al control text correspondiente.

**ComAUsoS**

<b>Eventos</b>	<b>Descripción</b>
Click	Según el uso de suelo seleccionado asigna la clave correspondiente al control text correspondiente.

**TxtACP**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**TxtAPropiet**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**TxtASuperf**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**TxtAObjeto**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**TxtABoletaP**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**TxtAEscrit**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**TxtACalle1**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**TxtACalle2**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**TxtACalle3**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**TxtACalle4**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**TxtALinder**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**ComASig**

<b>Eventos</b>	<b>Descripción</b>
Click	Cambia a la siguiente pantalla de avalúos

**ComARegresa**

<b>Eventos</b>	<b>Descripción</b>
Click	Cancela el avalúo en proceso.

**TxtAFechR**

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación del campo.

**CombFUbi**

<b>Eventos</b>	<b>Descripción</b>
Click	Según se seleccione las opciones del combo con respecto a la ubicación del predio se obtiene en el Text contiguo, el valor correspondiente al factor de ubicación.

### CombFZona

<b>Eventos</b>	<b>Descripción</b>
Click	Según se seleccione las opciones del combo con respecto a la zona en la que se localiza el predio se obtiene en el Text contiguo, el valor correspondiente al factor de zona.

El control TxtASuperficie no emplea ningún evento ya que el valor depende del valor del Texto de superficie que se calcula automáticamente al cargar la forma.

### TxtPendi

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Se valida el valor de la pendiente y se calcula automáticamente el valor del factor de topografía el cual se almacena en un campo Text.

### TxtFon, TxtFren, TxtFonR, TxtSPA, TxtSPP TxtSARCV y TxtSAII

<b>Eventos</b>	<b>Descripción</b>
KeyPress	Realiza la validación de cada uno de los campos, los cuales generan el factor de forma del predio.

### OptPUManual

<b>Eventos</b>	<b>Descripción</b>
Click	Habilita un texto para teclear el valor unitario del predio, para que junto con el factor resultante nos proporcione el valor del avalúo.

### OptPUAutomat

<b>Eventos</b>	<b>Descripción</b>
Click	Habilita un grid para que de los predios que se almacenan dentro de éste, pueda elegir aquel que se desea manejar como el valor unitario del predio. Este evento realiza la búsqueda de los predios que sean más acordes con el predio a valuar.



## GridPredios

<b>Eventos</b>	<b>Descripción</b>
Click	Realiza el cálculo del valor del predio tomando como referencia el valor unitario del predio seleccionado y el factor resultante.

### 3.6 Integración y pruebas del sistema

Una **estrategia de prueba de software** integra las técnicas de diseño de casos de prueba en pasos bien planificados con el fin de construir correctamente el software.

Una estrategia de prueba debe planificar la prueba, diseñar casos de prueba y recolectar y evaluar los resultados. Todas las estrategias tienen las siguientes características generales:

- ✓ La prueba comienza en el nivel de 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 prueba de software es un elemento de un tema más amplio que a menudo se refiere como **verificación y validación**:

- ✓ **La verificación** se refiere al conjunto de actividades que aseguran que el software implementa correctamente una función específica.
- ✓ **La validación** se refiere a un conjunto diferente de actividades que aseguran que el software construido se ajusta a los requerimientos del cliente.

El que desarrolla el software es responsable de probar las unidades individuales (módulos) del programa, asegurándose de que cada una lleva a cabo la función para la que fue diseñada. En muchos casos también se encargara de la prueba de integración – el paso de prueba que lleva a la construcción y prueba de la estructura total del sistema.

Una vez que la arquitectura del software esté 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 construído.

Una prueba independiente elimina el conflicto de intereses que de otro modo estará presente.

### **3.6.1 Estrategia de prueba de software**

La prueba, en el contexto de la ingeniería de software, consiste en cuatro pasos:

1. Prueba de unidad.
2. Prueba de integración.
3. Prueba de validación.
4. Prueba del sistema.

Como se muestra en la figura 3.6.1.1.

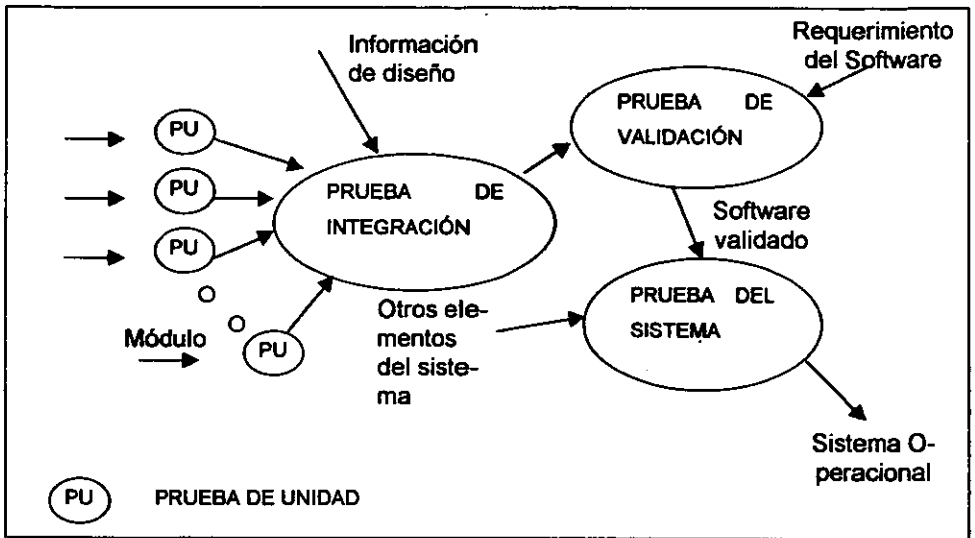


Figura 3.6.1.1 Pasos de prueba de software.

### Prueba de la unidad

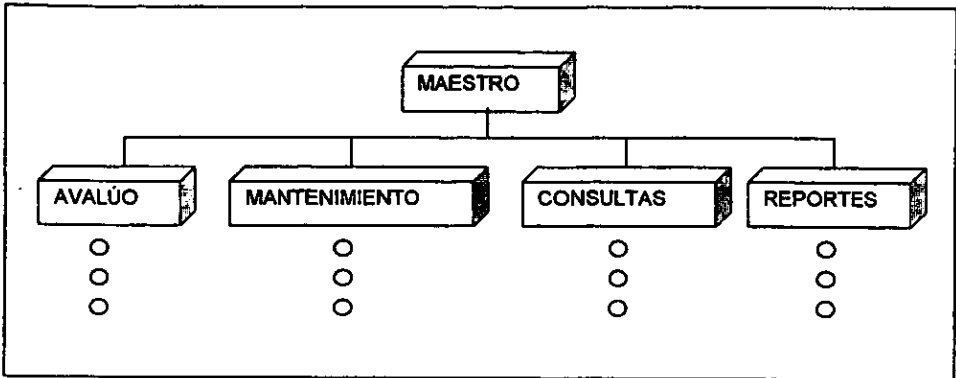
Esta prueba se centra en la menor unidad del diseño del software, es decir el módulo, en las implementaciones en código fuente. Realiza un uso intensivo de ejercicios de caminos específicos de la estructura de control del módulo para asegurar un alcance completo y una detección máxima de errores dentro del módulo.

Esta prueba se lleva a cabo en paralelo con otras pruebas del módulo, es decir mientras unos verificaban los módulos de avalúos y mantenimiento otros probaban el módulo de consultas y salida del sistema.

Posteriormente se integraron los módulos para formar el paquete del software completo.

**Pruebas para cada módulo del sistema de avalúos.**

El sistema de avalúos de predios y terrenos esta compuesto por los módulos que se muestran en la figura 3.6.1.2.



**Figura 3.6.1.2 Módulos del sistema de avalúos de predios y terrenos.**

El **Módulo central o principal** permite tener acceso al sistema mediante claves secretas de acceso (passwords), en el sistema de avalúos se manejan tres niveles de seguridad y una vez dentro, el usuario puede llamar a los módulos de avalúos, mantenimiento, consultas, reportes, altas, bajas, cambios y salir del sistema.

Se probó que cada uno de los menús de opciones y de las pantallas de captura cumpliera con los puntos que se muestran en la figura 3.6.1.3.

NOMBRE DEL SISTEMA
ORTOGRAFÍA CORRECTA
NOMBRE DEL MENÚ
OPCIONES DEL MENÚ
DESCRIPCIÓN DE LA OPCIÓN SELECCIONADA
TECLA DE TABULADOR

TECLAS DE LA PRIMERA LETRA
TECLA DE ESCAPE
BOTÓN DE ACEPTACIÓN
BOTÓN DE REGRESAR
USO DEL MOUSE

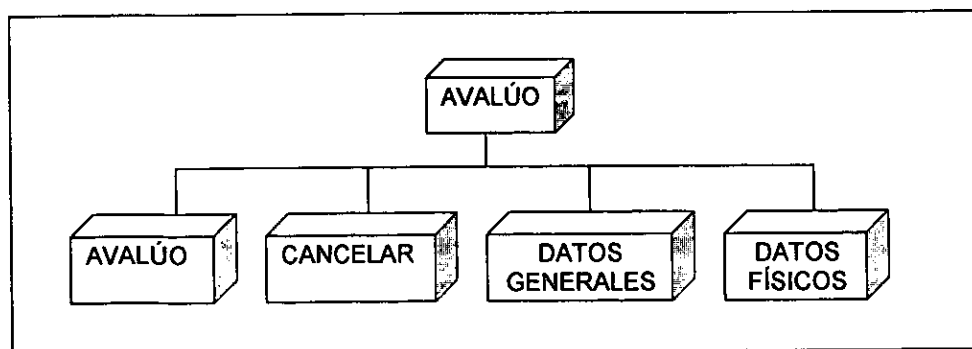
Figura 3.6.1.3 Pruebas que se aplicaron a los menús.

- ✓ **El Nombre del Sistema** siempre aparece en la parte superior de todas las pantallas como "Sistema de Valuación de Terrenos".
- ✓ **Cumplir con Ortografía correcta**, todos los mensajes y pantallas.
- ✓ **Nombre del menú.** Cuando se selecciona un menú se mantiene el nombre de éste iluminado con el fin de saber en todo momento en la opción en que nos encontramos.
- ✓ **Opciones del menú.** Se muestran todas las opciones que comprenden al menú seleccionado.
- ✓ **Descripción de la opción seleccionada.** Cuando se selecciona un folder con el mouse o con las teclas de Alt y la letra que aparece subrayada en la pestaña, ésta tiende a aparecer más resaltada.
- ✓ **Tecla de tabulador.** Se podrá hacer uso de la tecla de tabulador para moverse a través de todos los botones, opciones o campos que se desplieguen en la pantalla.
- ✓ **Tecla de entrar.** Durante la selección de una opción en los menús se podrá hacer uso de esta tecla para activar un botón de aceptación, a un botón de cancelación o una selección.

- ✓ **Teclas de la primera letra.** Se podrá seleccionar una opción mediante la primera letra del nombre de la opción.
- ✓ **Botón de aceptación.** Se podrá hacer uso de un botón para indicar que estamos de acuerdo con la selección que se escogió o con los datos introducidos que nos solicite alguna de las opciones y se ejecuta el módulo del programa que le corresponde.
- ✓ **Botón de regresar.** Se podrá hacer uso de un botón para abandonar la selección y regresarnos al menú anterior.
- ✓ **Tecla de escape.** Cuando aparece un mensaje se podrá eliminar de la pantalla con la tecla de escape.
- ✓ **Uso del mouse.** Se podrá hacer uso del mouse oprimiendo el botón izquierdo con el fin de activar a un botón de aceptación, o un botón de cancelación o una selección, o para ubicarse en un campo en específico para introducir información.

**Prueba del módulo de avalúo.**

Este módulo está formado por los submódulos que se muestran en la figura 3.6.1.4.



**Figura 3.6.1.4 Prueba del módulo de avalúo.**

Las pruebas que se aplicaron a los campos:

1. **No. Secuencial.** Este número es proporcionado por el valuator.
2. **Fecha de Recepción.** Fecha de recepción de los datos del avalúo.
3. **Solicitante.** El tipo de solicitante puede ser gubernamental o privado.
4. **No. de Oficio.** Documento que antecede al avalúo (sólo en el caso gubernamental).
5. **Valuador.** Es el perito asignado a realizar dicho avalúo.

Pruebas del submódulo de avalúo del módulo de avalúo.

En este submódulo se podrán hacer avalúos de predios y terrenos dentro de la delegación Venustiano Carranza. Los campos en los que se aplicaron las pruebas fueron:

- ✓ **No. Secuencial.**
- ✓ **Fecha de recepción.**
- ✓ **Solicitante.**
- ✓ **No. de Oficio.**
- ✓ **Valuador.**

**Pruebas de la opción cancelar del módulo de avalúo**

Si un usuario desea dar de baja un avalúo se solicitara alguno de los siguientes datos:

1. No. Secuencial.
2. Identificador del valuator.
3. Identificador del solicitante.
4. Identificador del predio.
5. Nombre del propietario.

**Pruebas de la opción datos generales del módulo de avalúo.**

Aquí se capturan todos los datos generales del predio a evaluar como se muestra en la siguiente tabla:

**AVALÚO**

<b>DATOS GENERALES</b>
No. secuencial
Fecha de recepción
Asunto
Solicitante
Predio
Oficio No.
Valuador
Fecha de entrega
Observaciones
Calle
Colonia
Delegación
Zona
Uso de suelo
Código postal
Propietario
Superficie
Objeto
Boleta predial
Escritura
Calle 1 a la 4
Linderos y colindancias



## Pruebas de la opción datos físicos del módulo de avalúo

Aquí se capturan todos los datos físicos del predio a evaluar como se muestra en la siguiente tabla:

<b>DATOS FÍSICOS</b>
Fecha de realización
Factor de zona
Factor de ubicación
Frente del terreno
Factor de frente
Pendiente
Factor de topografía
Factor de superficie
Fondo del terreno (m)
Frente del rectángulo
Fondo del rectángulo
Superficie de la porción anterior
Superficie de la porción posterior
Superficie de las áreas irregulares con frente a la vía de acceso
Superficie de las áreas irregulares interiores
Factor de forma
Precio unitario manual
Precio unitario automático

## Prueba del módulo del catálogo de solicitantes

Este módulo está formado por los submódulos que se muestran en la figura 3.6.1.5.

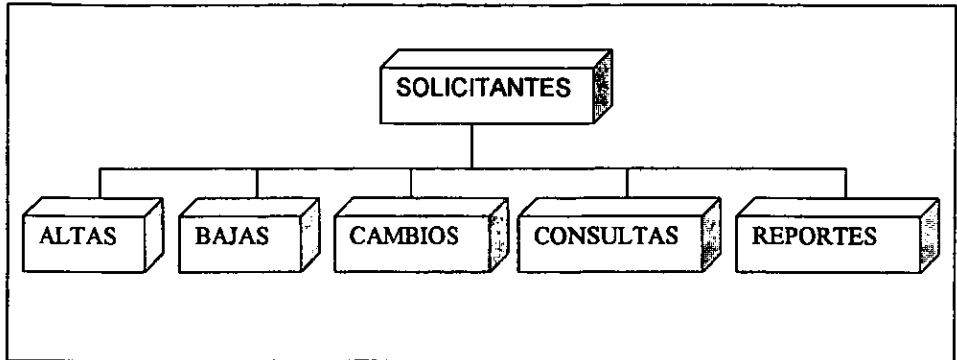


Figura 3.6.1.5. Prueba del módulo de solicitantes.

### Pruebas de la opción altas del módulo de mantenimiento del catálogo de solicitantes

En esta opción se solicitará cada uno de los campos que formarán los datos que identifican al solicitante, se probó que cada uno de los campos y botones cumpliera con:

- ✓ **Identificador.** Se proporciona el número identificador de la empresa o persona solicitante y el sistema validará si ya existe. Si ya existe el sistema lo indicará mediante un mensaje.
- ✓ **Tipo.** Aquí se pone el tipo de solicitante, el cual puede ser gubernamental o privado.
- ✓ **Nombre.** En este campo se coloca el nombre de la empresa o persona, el cual es de 40 caracteres como máximo.
- ✓ **Domicilio.** Se proporciona la dirección de la empresa o de la persona, el cual es de 50 caracteres como máximo.

- ✓ **Teléfono.** Aquí se colocan los números telefónicos, cuya longitud máxima es de 30 caracteres.
- ✓ **RFC.** Aquí va el RFC del solicitante, cuya longitud máxima es de 13 caracteres.

### **Pruebas de la opción bajas del módulo de mantenimiento del catálogo de solicitantes**

En esta opción se puede dar de baja un solicitante, se probó que cada uno de los campos y botones cumpliera con los datos vistos en la parte de altas.

### **Pruebas de la opción cambios del módulo de mantenimiento del catálogo de solicitantes**

En esta opción se podrán realizar cambios en cada uno de los campos que formarán los datos que identifican al solicitante con los datos vistos en la parte de altas.

Pruebas de la opción consultas del módulo de mantenimiento del catálogo de solicitantes.

En esta opción se podrán consultar cada uno de los campos que formarán los datos que identifican al solicitante, se probó que cada uno de los campos y botones cumpliera con los datos vistos en la parte de altas.

### **Pruebas de la opción reportes del módulo de mantenimiento del catálogo de solicitantes**

En esta opción se podrán imprimir reportes de Solicitantes, Valuadores, Estudio de mercado, Zonas, Uso de suelo, Delegaciones, Colonias y Zona/Suelo/Delegación.

## Prueba del módulo del catálogo de valuadores

Este módulo está formado por los submódulos que se muestran en la figura 3.6.1.6.

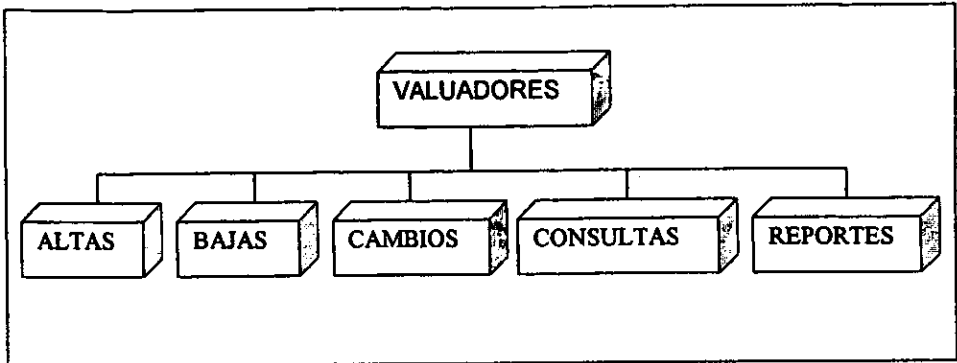


Figura 3.6.1.6. Prueba del módulo de valuadores.

### Pruebas de las opciones del módulo de mantenimiento del catálogo de valuadores

En la opción de altas se solicitará cada uno de los campos que formarán los datos que identifican a los valuadores. En el caso de las bajas, cambios, consultas y reportes se mostró la información correspondiente al tipo de valuator. En el caso de cambios se permitió modificar los datos. En el caso de los reportes se permitió la impresión. Se probó que cada uno de los campos y botones cumpliera:

- ✓ **Identificador.** Identificador de los peritos valuadores.
- ✓ **Nombre.** Nombre del perito valuator cuyo máximo de caracteres es de 30.
- ✓ **Tipo de valuator.** En este campo se colocan el tipo que puede ser interno o externo.
- ✓ **Registro.** Aquí se coloca los registros o títulos obtenidos en otras instituciones que certifiquen a dichos peritos.

- ✓ **Especialidades.** Especialidades que dominan los peritos, cuyo máximo de caracteres es de 40.
- ✓ **Eficiencia.** En este campo se coloca la capacidad desarrollada en los trabajos que se le encomiendan.
- ✓ **Observaciones.** Observaciones sobre el desempeño del perito valuator, el máximo de caracteres es de 100.

**Prueba del módulo de mantenimiento del catálogo de estudio de mercado**

Este módulo está formado por los submódulos que se muestran en la figura 3.6.1.7.

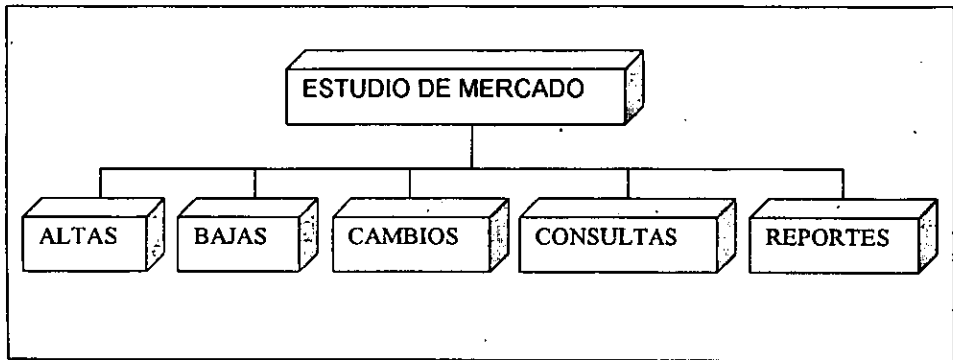


Figura 3.6.1.7 Prueba del módulo de estudio de mercado.

**Pruebas de las opciones del módulo de mantenimiento del catálogo de estudio de mercado**

En la opción de altas se solicitará cada uno de los campos que formarán los datos que identifican al estudio de mercado. En la opción de cambios podrán modificarse estos datos. En el caso de las bajas, consultas y reportes se mostró la información correspondiente. Se probó que cada uno de los campos y botones cumpliera:

- ✓ **Predio.** Predio del que se pretende realizar el estudio de mercado.

- ✓ **Calle.** Nombre de la calle y número donde se encuentra ubicado el predio.
- ✓ **Colonia.** Clave de la colonia, asignada por el usuario.
- ✓ **Delegación.** Clave de la delegación, asignada por el usuario.
- ✓ **Zona.** Clave de la zona, asignada por el usuario.
- ✓ **Uso de suelo.** Tipo de uso de suelo.
- ✓ **Código Postal.** Código postal del predio.
- ✓ **Propietario.** Nombre del propietario del predio.
- ✓ **Superficie.** Superficie que se evalúa.
- ✓ **Valor del terreno.** Valor del terreno calculado por el perito.
- ✓ **Ofertado desde.** Fecha desde cuando se esta ofertando dicho predio.

**Prueba del módulo de mantenimiento del catálogo de zonas**

Este módulo está formado por los submódulos que se muestran en la figura 3.6.1.8.

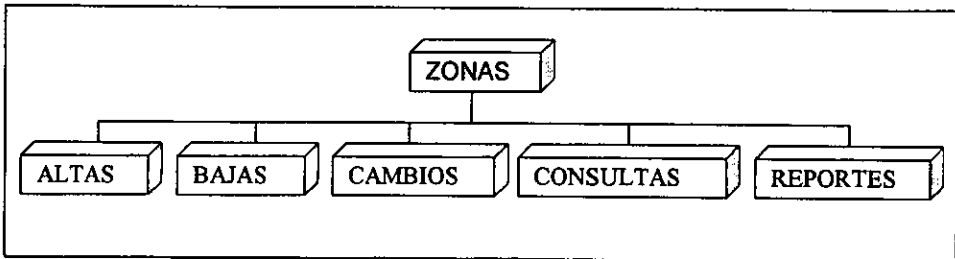


Figura. 3.6.1.8 Prueba del módulo de zonas.

**Pruebas de las opciones del módulo de mantenimiento del catálogo de zonas**

En la opción de altas se solicitará la información que formarán los datos que identifican a las diferentes zonas. En la opción de cambios podrán modificarse estos datos. En el caso de las bajas, consultas y reportes se muestra la descripción del tipo de zona. Se probó que cada uno de los campos y botones cumpliera con:

- ✓ **Identificador.** Identificador de la zona proporcionado por el usuario.

- ✓ **Descripción.** Descripción de la zona, cuya longitud de caracteres es de 30.

### Prueba del módulo de mantenimiento del catálogo de uso de suelo

Este módulo está formado por los submódulos que se muestran en la Figura 3.6.1.9.

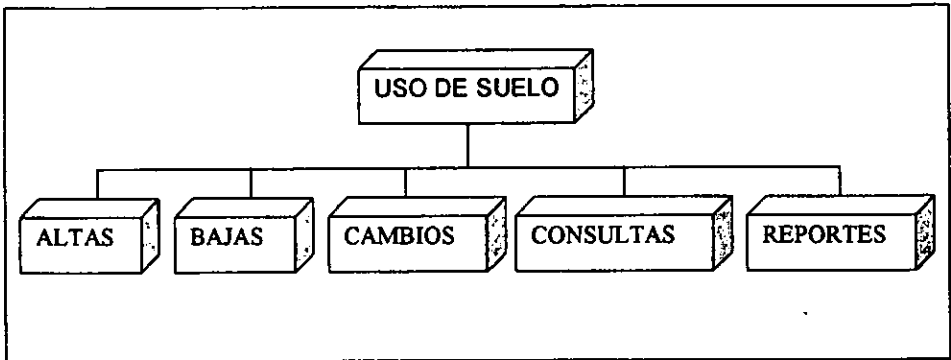


Figura 3.6.1.9 Prueba del módulo de uso de suelo.

### Pruebas de las opciones del módulo de mantenimiento del catálogo de uso de suelo

En la opción de altas se solicitará la información que formarán los datos que identifican a los tipos de uso de suelo. En la opción de cambios podrán modificarse estos datos. En el caso de las bajas, consultas y reportes se muestra la descripción del tipo de uso de suelo. Se probó que cada uno de los campos y botones cumpliera con:

- ✓ **Identificador.** Identificador del tipo de uso de suelo.
- ✓ **Descripción.** Descripción del uso de suelo.
- ✓ **Lote moda.** Características del lote moda.
- ✓ **Densidad de población.** Densidad de población.

## Prueba del módulo de mantenimiento del catálogo de delegaciones

Este módulo está formado por los submódulos que se muestran en la Figura 3.6.1.10.

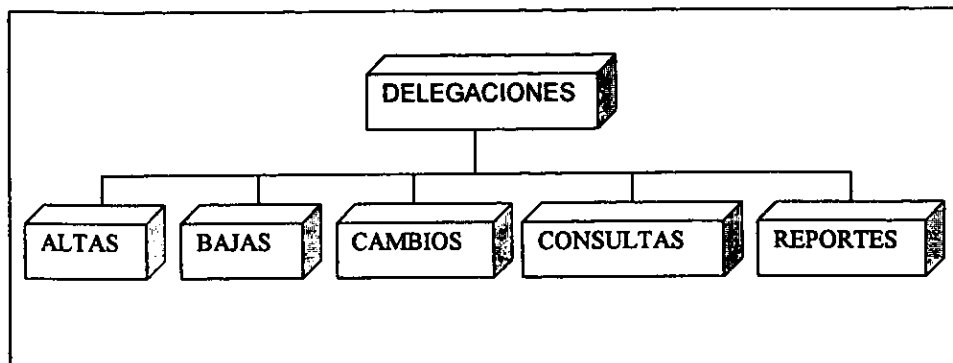


Figura 3.6.1.10 Prueba del módulo de mantenimiento del catálogo de delegaciones.

## Pruebas de las opciones del módulo de mantenimiento del catálogo de delegaciones

En la opción de altas se solicitará la información que formarán los datos que identifican a las delegaciones. En la opción de cambios podrán modificarse estos datos. En el caso de las bajas, consultas y reportes se muestra la descripción de la delegación. Se probó que cada uno de los campos y botones cumpliera con:

- ✓ **Identificador.** Identificador de la delegación, este es asignado por el usuario.
- ✓ **Nombre.** Nombre de la delegación.

## Prueba del módulo de mantenimiento del catálogo de colonias

Este módulo está formado por los submódulos que se muestran en la Figura 3.6.1.11.



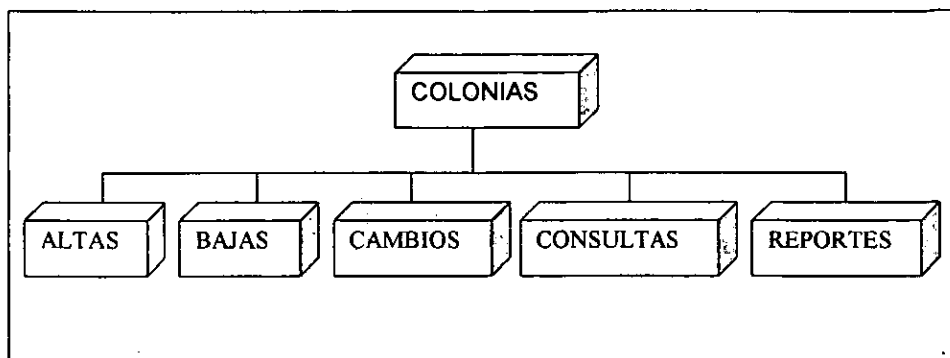


Figura 3.6.1.11 Prueba del módulo de mantenimiento del catálogo de colonias.

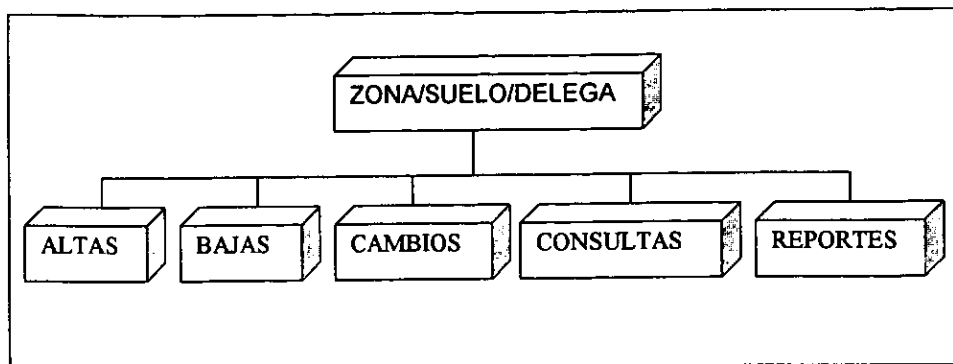
### Pruebas de las opciones del módulo de mantenimiento del catálogo de colonias

En la opción de altas se solicitará la información que formarán los datos que identifican a las colonias. En la opción de cambios podrán modificarse estos datos. En el caso de las bajas, consultas y reportes se muestra la descripción de la colonia. Se probó que cada uno de los campos y botones cumpliera con:

- ✓ **Identificador.** Identificador de la colonia, proporcionado por el usuario.
- ✓ **Nombre.** Nombre de la colonia.

### Prueba del módulo de mantenimiento del catálogo de zona/suelo/delegación

Este módulo está formado por los submódulos que se muestran en la Figura 3.6.1.12.



**Figura 3.6.1.12 Prueba del módulo de zona/uso de suelo/delegación.**

### **Pruebas de las opciones del módulo de mantenimiento del catálogo zona/suelo/delegación**

En la opción de altas se solicitará la información que formarán los datos que identifican a la zona/suelo/delega. En la opción de cambios podrán modificarse estos datos. En el caso de las bajas, consultas y reportes se muestra la descripción de la zona/suelo/delega. Se probó que cada uno de los campos y botones cumpliera con:

- ✓ **Zona.** Identificador de la zona.
- ✓ **Delegación.** Clave de la delegación.
- ✓ **Intensidad de.** Intensidad de construcción.
- ✓ **Uso de suelo.** Clave del uso de suelo de la zona.
- ✓ **Contaminación.** En este campo el perito realiza las observaciones pertinentes.
- ✓ **Observaciones.** En este campo es libre de observaciones personales del perito.

## Prueba del módulo de consultas

Este módulo está formado por los submódulos que se muestran en la figura 3.6.1.13.

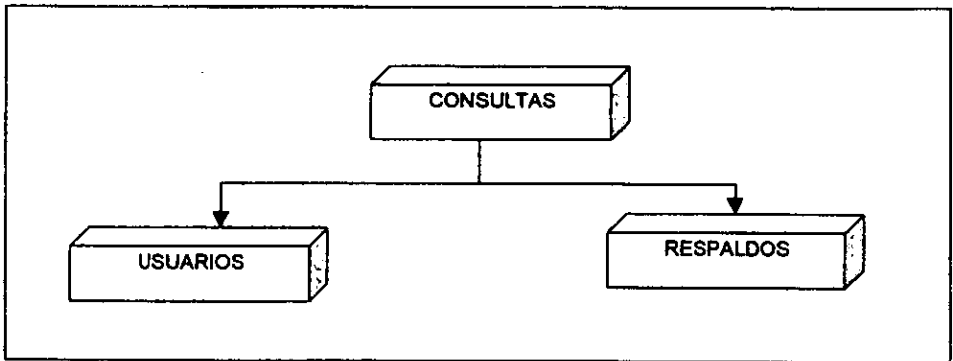


Figura 3.6.1.13 prueba del módulo de consultas.

### Pruebas de las opciones del módulo de consultas.

En la opción de consultas se solicitará información para consultar dependiendo de la cantidad de información que tenga en mi base de datos.

En este tipo de consultas contamos con cuatro botones de **Primero**, **Anterior**, **Siguiente** y **Ultimo**.

### La prueba de integración

Durante 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.

✓ **Prueba 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 (p.ej., archivos de datos). 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.** Es una prueba que demuestra que cada función es completamente operativa, conociendo la función específica para la que fue diseñado el producto. Se basa en el minucioso exámen 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 de avalúos.**

Se utilizó 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 probados en unidad y nos aseguramos que la estructura del programa estuviera de acuerdo con lo que dictaba el diseño.

Se utilizó la **integración incremental** en el cual el programa se construye y se prueban los módulos en los que los errores son más fáciles de aislar y de corregir con

el fin 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.**

Integramos los módulos moviéndolos hacia **abajo** por la jerarquía de control, comenzando con el módulo de control principal (módulo maestro). Los módulos subordinados al módulo de control principal se van incorporando en la estructura de forma primero en profundidad.

Refiriéndonos en la figura 3.6.1.14. se eligió el camino a mano izquierda, se integraron los módulos M1, M2, y M3. A continuación será integrado M4. Acto seguido se construyen los caminos de control central y derecho.

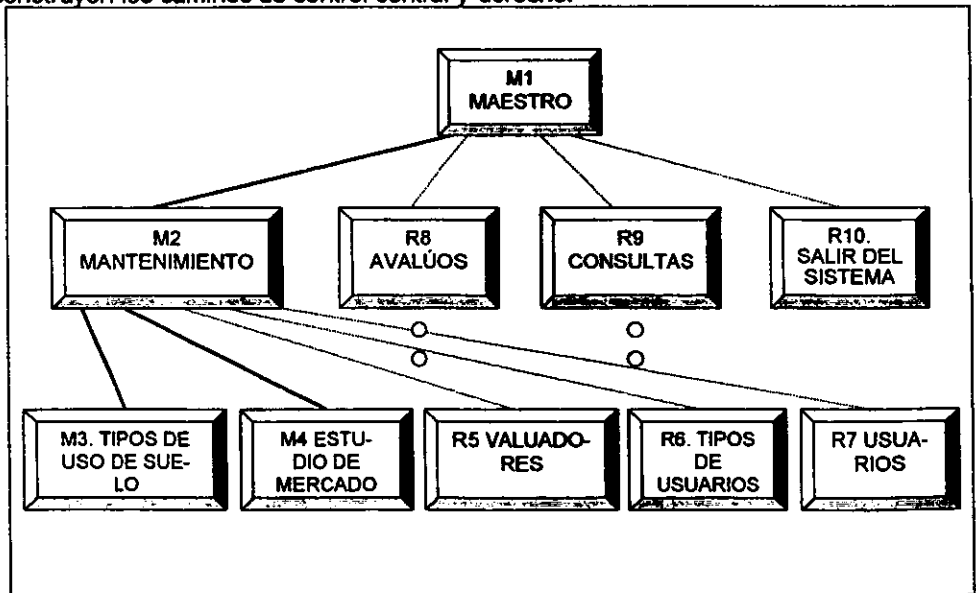


Figura 3.6.1.14 Integración descendente de forma primero en profundidad.

Llevamos a cabo el proceso de integración en cinco pasos:

Se usó 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 a el 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 reemplazamos otro resguardo con el módulo real.
5. Realizamos la prueba de regresión (o sea, todas las pruebas anteriores) para asegurarnos que no se hayan introducido nuevos errores.

El proceso continuó 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.

### **Especificación de la prueba de integración**

**Alcance de la prueba.** Se probaron características tales como:

- ☑ Fácil uso por parte del usuario.
- ☑ Desplegado de ayuda (tecla de F1).
- ☑ El fácil acceso utilizando los dispositivos de entrada: el mouse y el teclado.
- ☑ Además que el número de los módulos cumplieran con las especificaciones solicitadas.

**Plan de prueba.** La estrategia general para la integración se dividió en 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 y estadísticas

### **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.

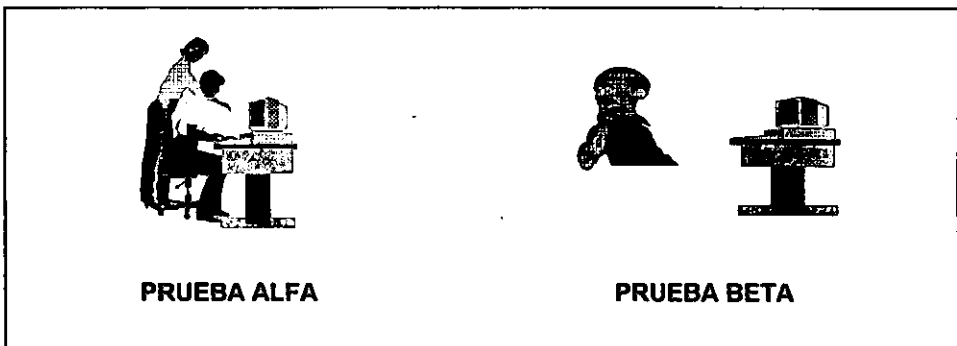
- **Contención de información.** Pruebas para descubrir errores asociados con las estructuras globales y locales.
- **Rendimiento.** Pruebas para verificar los límites de rendimiento establecidos durante el diseño del software.

Los recursos empleados fueron una "PC AT 486 con 1.6 GB en disco duro disponible".  
Procedimiento de prueba. Describimos el orden de integración y las pruebas correspondientes a cada fase de integración.

### La prueba de validación del sistema de avalúos

En una 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, Figura 3.6.1.15.



**Figura 3.6.1.15 Pruebas alfa y beta como pruebas de validación.**



- **Prueba Alfa.** Al lugar del desarrollo se invitó a operadores para probar el sistema, en presencia de los desarrolladores.
- **Prueba Beta.** La prueba se realizó con operadores en el lugar físico donde estaría funcionando el sistema.

**Prueba alfa.** Para llevar a cabo la prueba se invitó a un usuario en el lugar del desarrollo y en presencia de los desarrolladores, éste registró errores y problemas de uso.

Es decir todo esto se llevó en un entorno controlado. Al principio al utilizar el sistema, el operador tenía cierto temor y no se presentaba seguro. Los mensajes que proporcionaba el sistema algunos fueron modificados porque les faltaba claridad.

Las principales deficiencias en esta prueba fue la falta de familiaridad en el uso de menús y el entender el procedimiento para manipular los datos.

En la prueba beta se llevó a cabo en el lugar donde estaría en funcionamiento el sistema.

Sobre todo se presentaron problemas tales como la necesidad de una explicación sencilla de los errores y mensajes y además dudas en el instructivo.

### **Prueba del sistema**

Verifica que cada elemento del programa trabaja adecuadamente alcanzando la funcionalidad y el rendimiento del sistema total.

## Prueba del sistema de avalúos

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 período de tiempo de corrección de fallas Figura 3.6.1.16.

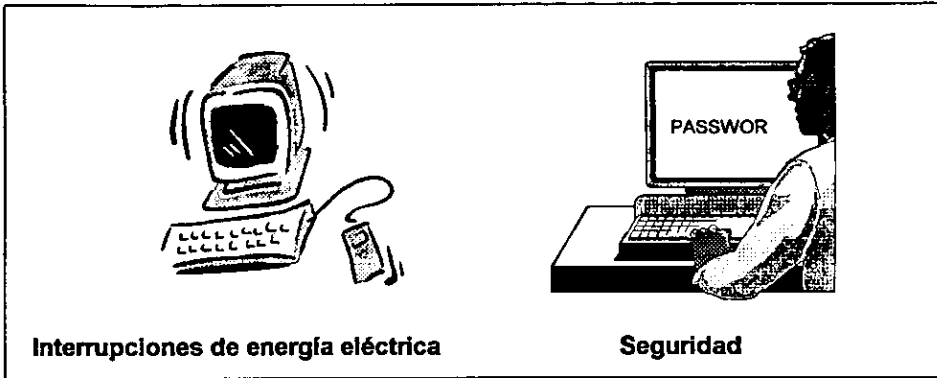


Figura 3.6.1.16 Pruebas del sistema.

## Pruebas de Recuperación

- Interrupciones de energía eléctrica.

**Prueba 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:

- Tiene acceso a todas las opciones del menú principal (jvela).
- Dos (tiscareño).
- Tres (jroa).

**Prueba de resistencia y de rendimiento.** Entre las pruebas que se sometieron fueron las siguientes:

1. Se realizaron de tal manera que se sometió al sistema a una demanda de recursos en cantidad.
2. Se probó casos que produjeran excesivas búsquedas de datos residentes en la base de datos.

### **3.7 implantación del Sistema**

En esta etapa se instalará, probará y se liberará el sistema.

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

El primer paso es la asignación de acceso y facultades de cada uno de los usuarios.

Una vez terminada la prueba del sistema, se formularon los documentos que respaldaran la prueba de aceptación del sistema. Se integró el material producido durante esta fase, llevándose a efecto la aprobación de instalación del sistema.

Después de haber cumplido con todos los requerimientos de las pruebas de aceptación, se realizó una revisión de los planes de instalación siguientes:

1. Efectuar una revisión de hardware, para asegurar que la instalación tuviera el éxito esperado.
2. Revisión de documentación final del sistema.

## **Instalación del sistema**

Se realizaron los trabajos del sistema SAPT según el plan y se supervisaron las diferentes acciones anotadas en los informes de seguimiento y sus avances para comprobar la implantación exitosa del sistema.

## **Apoyo postimplantación**

La fase de postimplantación tiene como objetivo documentar los mecanismos para cualquier necesidad de soporte y mantenimiento durante la implantación y operación del sistema SAPT, llevar a cabo un monitoreo del sistema en operación para detectar cualquier posible adecuación que permita mejorar su rendimiento y liberar formalmente el sistema.

En lo particular, esta fase tiene una importancia sumamente alta porque cuando un sistema no se encuentra bien definido y resulta ser que no se validó con el usuario final los productos que se generan durante el desarrollo, ésta fase puede ser la más cara. Esto significa que todo o gran parte del esfuerzo requerido para producir una característica o todo el software no satisfacen los requerimientos del cliente y es entonces cuando habrá que duplicar los esfuerzos para parchar el sistema o generar el producto que satisface al usuario corriendo el riesgo de ir hacia atrás en la liberación formal del sistema.

La metodología también aplica en la forma en que se hará el apoyo postimplantación del sistema, donde sugiere la elaboración de documentos que sirvieron para detectar los problemas surgidos en la operación del mismo, así como la petición de nuevas características del producto para su futuro desarrollo.

## **Aprobación**

En la última fase de apoyo postimplantación, se integra el material producido durante la fase para su revisión final, para obtener la aprobación de las actividades de la fase y así concluir con el proyecto.

Se efectúa la presentación de los productos elaborados en la fase y se obtendrá la aprobación formal por parte de los participantes, concluyendo de esta manera con las actividades de la fase.

## IV IMPACTO DEL SISTEMA

### 4.1 ¿Qué es el soporte de sistemas?

El soporte de sistemas es el mantenimiento permanente de un sistema después de que haya sido puesto en operación. Ello incluye tanto el mantenimiento estricto de los programas como las posibles mejoras que pueden añadirse al sistema.

El soporte es realizado por diseñadores y constructores de sistemas como apoyo a los usuarios de los mismos. En nuestro modelo en pirámide, están relacionados con los bloques **personas, datos, actividades y redes** implantados por medio de la **tecnología** (ver figura 4.1.1).

1. Corregir errores (también llamado *mantenimiento*).
2. Recuperar el sistema.
3. Asistir a los usuarios del sistema.
4. Adaptar el sistema ante nuevas necesidades (también llamado *reingeniería*).

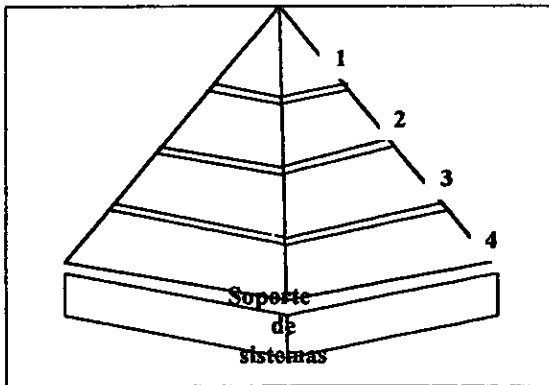


Figura 4.1.1 Modelo Pirámide.

El soporte de sistemas es a menudo ignorado en los libros sobre análisis y diseño de sistemas. Por ello, los analistas a veces se sorprenden al conocer que la mitad (o más)

de sus obligaciones estará relacionada con el soporte de los sistemas existentes. El soporte de sistemas requiere a menudo que el analista vuelva a repasar actividades típicamente desarrolladas durante el análisis, el diseño y la implantación de sistemas. Montemos primero el escenario del soporte de sistemas. Los proyectos de mantenimiento de aplicaciones se contrastan con los proyectos de planificación de sistemas y desarrollo de aplicaciones. Adviértase que las aplicaciones, una vez implantadas, se dice que están en producción. La producción es la ejecución día a día, semana a semana, mes a mes y año tras año de los programas de aplicación que procesan los datos de empresa (entradas) y generan información útil (salidas).

En las figuras siguientes, vemos también tres formas distintas de almacenamiento de datos en el sistema.

En primer lugar, tenemos nuestro **diccionario central**. Hasta ahora, sabemos que este diccionario contiene todos los modelos de sistemas esenciales y de implantación, así como las especificaciones detalladas. Se verificarán en él los subconjuntos del diccionario central que dan soporte a los diversos proyectos de planificación y desarrollo. Estos subconjuntos se almacenan como diccionarios del proyecto, y se implantan normalmente por medio de herramientas CASE (figura 4.1.2).

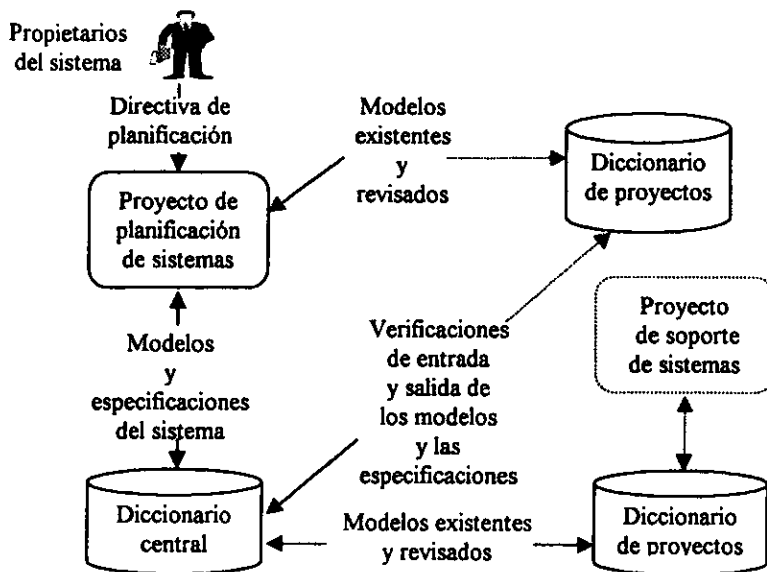


Figura 4.1.2 Contexto de los proyectos y actividades del soporte de sistemas.

En segundo lugar, podemos ver **bibliotecas de programas** donde se conservan los programas de aplicación reales que han sido puestos en producción. En la mayoría de los centros de sistemas, existe un software bibliotecario que hace un seguimiento de los cambios y mantiene algunas versiones previas del software, por si surgieran problemas en una versión determinada (figura 4.1.3).



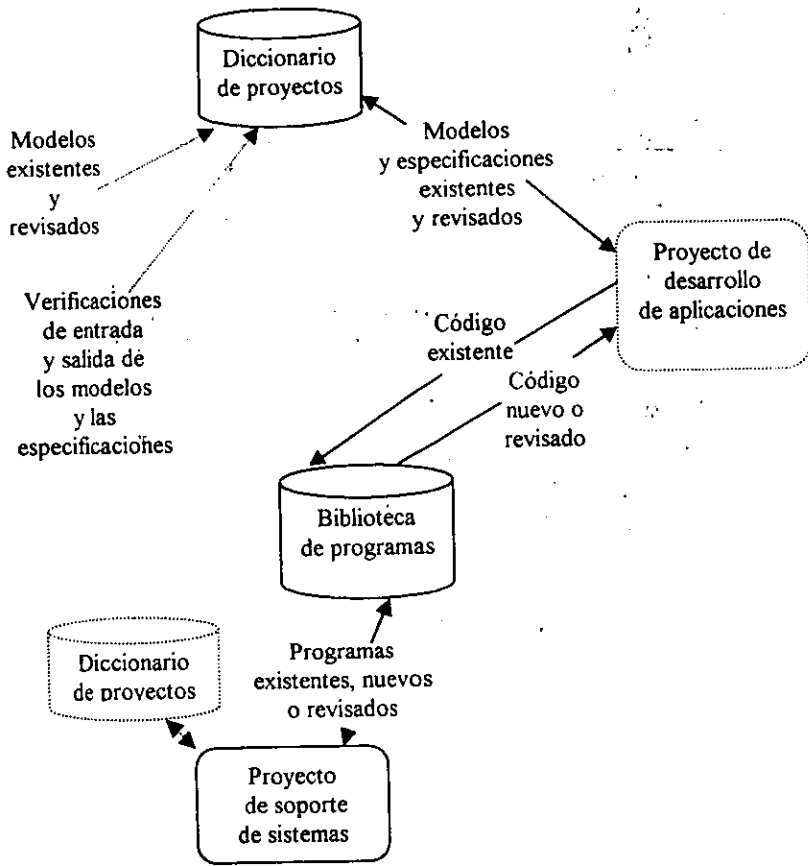


Figura 4.1.3 Contexto de los proyectos y actividades del soporte de sistemas.

En tercer lugar, vemos también los **archivos y bases de datos reales de la empresa**, donde se guardan los datos operativos creados y mantenidos por los programas de aplicaciones en producción (figura 4.1.4).

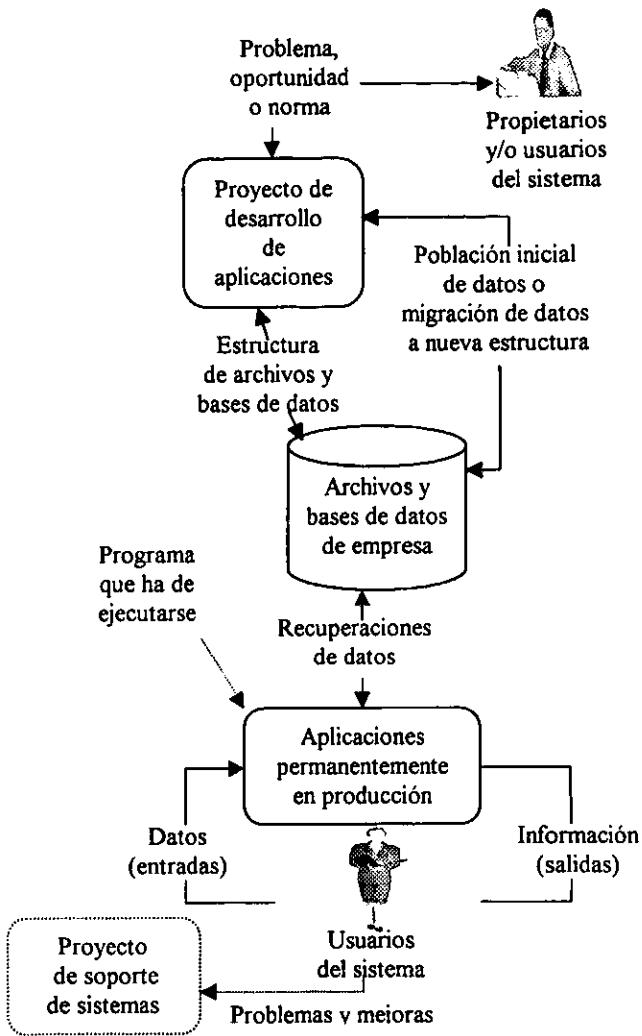
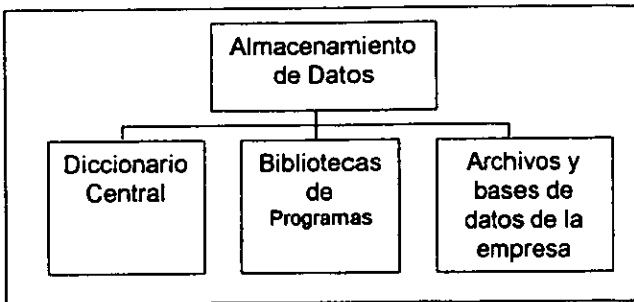


Figura 4.1.4 Contexto de los proyectos y actividades del soporte de sistemas.

Este diagrama es ilustrativo de la situación del soporte de sistemas en el contexto de otros tipos de proyectos y actividades, en particular la planificación de sistemas, el

desarrollo de sistemas, el desarrollo de sistemas y la producción de aplicaciones (figura 4.1.5).



**Figura 4.1.5 Almacenamiento de datos.**

La figura 4.1.6, 4.1.7, 4.1.8 y 4.1.9 muestra un ciclo de vida que ilustra las cuatro actividades del soporte. Los productos y resultados del trabajo se describen en el contexto de las figuras 4.1.2, 4.1.3 y 4.1.4 (mediante el empleo de almacenes de datos en vez de archivos en disco).

Este subtema examina cada una de las actividades de soporte con un cierto grado de detalle. Para transmitir la sensación de que se aplica una auténtica metodología de desarrollo de sistemas, cada fase se describirá en función de:

- (1) El propósito de la fase.
- (2) Las actividades que deberían realizarse.
- (3) Los papeles desempeñados por las diversas personas que intervienen en cada tarea.
- (4) Las entradas y las salidas de cada tarea.
- (5) Las técnicas o conocimientos que pueden utilizarse para completar cada tarea.
- (6) Tecnologías emergentes que pueden simplificar las tareas.

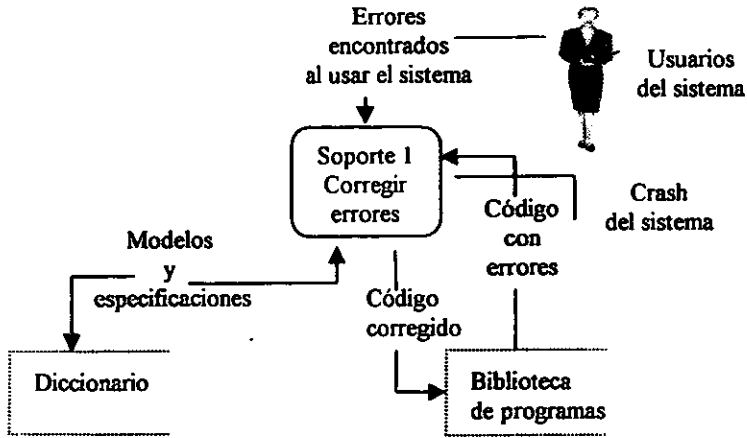


Figura 4.1.6 Actividades del soporte de sistemas.

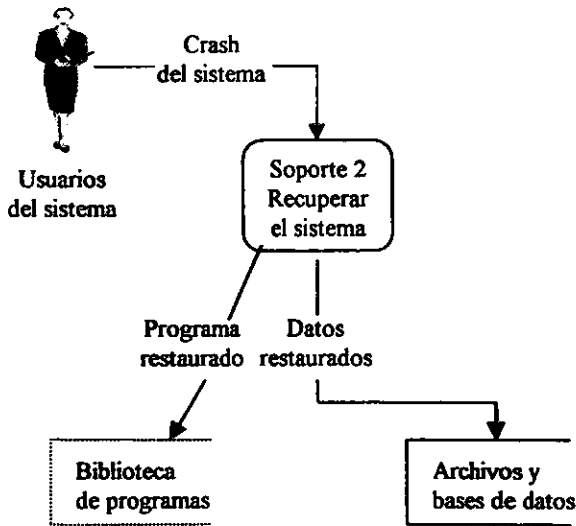


Figura 4.1.7 Actividades del soporte de sistemas.

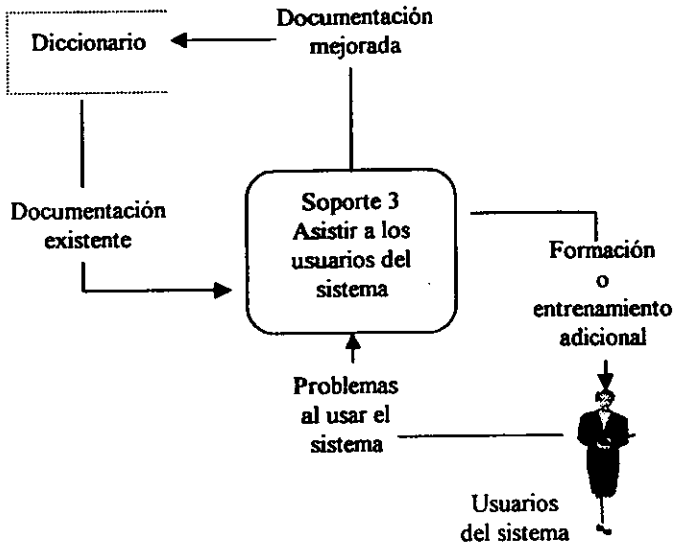


Figura 4.1.8 Actividades del soporte de sistemas.

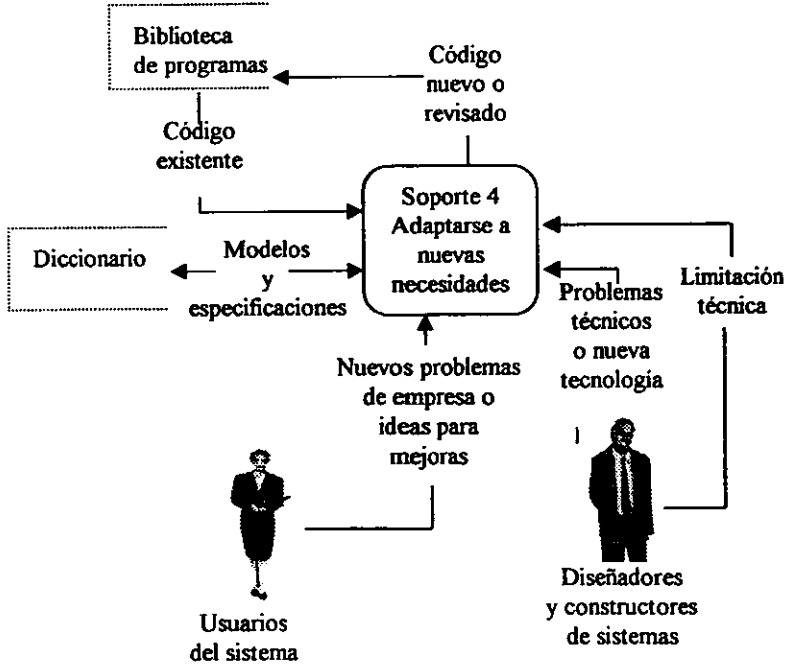
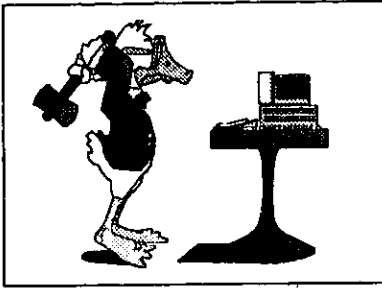


Figura 4.1.9 Actividades del soporte de sistemas.

## 4.2 Mantenimiento de Sistemas: Corrección de errores



Con la independencia de cómo esté diseñado, construido y probado un sistema o aplicación, inevitablemente aparecerán errores. Algunos de estos errores tendrán su origen en fallos en la comunicación de necesidades. Otros estarán provocados por defectos de diseño. Los habrá también originados por situaciones no previstas y, por tanto, no probadas. Y por último, los errores pueden ser causados también por un mal uso no previsto de los programas. En todas estas situaciones, deben de emprenderse acciones de corrección. A estas acciones de corrección las llamamos **mantenimiento de sistemas o mantenimiento de programas**.

### 4.2.1 Objetivos y bloques elementales del mantenimiento de sistemas

Los objetivos fundamentales del mantenimiento de sistemas son:

- Hacer cambios predecibles en los programas existentes para corregir errores que se cometieron durante el diseño y la implantación de sistemas. En consecuencia, excluimos de esta actividad las mejoras y las nuevas necesidades.
  
- Preservar aquellos aspectos de los programas que fueron ya corregidos. Al contrario, intentaremos evitar la posibilidad de que los <<arreglos>> en dichos programas originen que otros aspectos de los mismos funcionen de modo diferente.

Para alcanzar estos objetivos, ha de tenerse un conocimiento apropiado de los programas que se están arreglando y de las aplicaciones en que intervienen dichos programas. En la falta de este requisito previo está, con frecuencia, el fracaso del mantenimiento de sistemas.

¿Cómo se relaciona el mantenimiento de sistemas con los bloques elementales de los sistemas de información?

☐ PERSONAS. El mantenimiento de sistemas es normalmente iniciado por los usuarios de este sistema. El mantenimiento es llevado a cabo por lo general por constructores de sistemas con posible ayuda de los diseñadores.

☐ DATOS. El mantenimiento de sistemas rara vez influye sobre los datos, salvo por la posibilidad de que se mejore la edición.

☐ ACTIVIDADES. Los procesos de los sistemas de empresa y de información se implantan finalmente como programas de aplicaciones. El mantenimiento de sistemas consiste en arreglar los errores cometidos durante la implantación de dichos programas.

☐ REDES. El mantenimiento de sistemas rara vez tiene que ver con las redes informáticas si bien en ocasiones son las redes informáticas las fuentes de determinados errores.

☐ TECNOLOGÍA. El mantenimiento de sistemas, tal como se ha definido en esta actividad, no tiene que ver con los cambios tecnológicos.

Los elementos de un sistema de información se muestran en la figura 4.2.1.1.

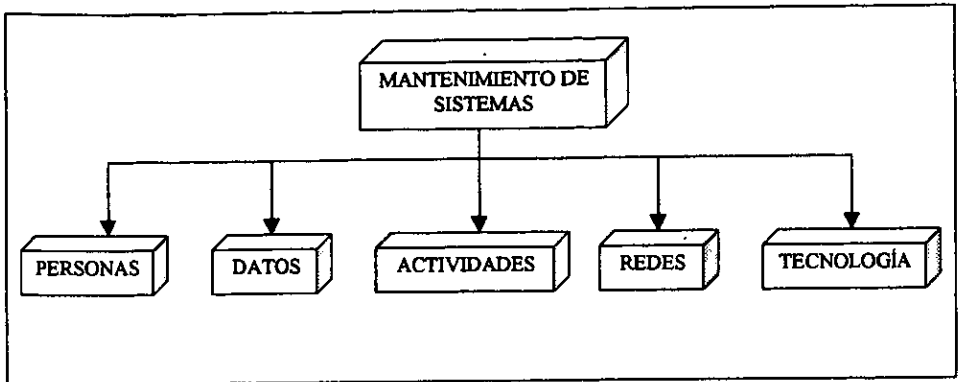


Figura 4.2.1.1 Entes que involucran al mantenimiento de sistemas.

#### 4.2.2 Tareas, participantes y técnicas del mantenimiento de sistemas

Las directrices generales para la lectura de los diagramas de actividades son:

- ☐ Las siluetas representan personas o departamentos que inician las tareas (figura 4.2.2.1).

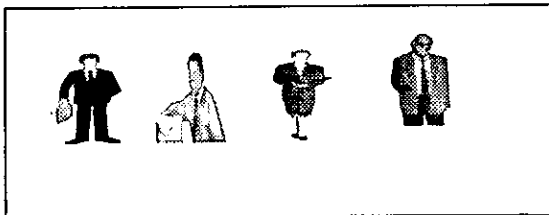


Figura 4.2.2.1 Inicio de tareas.

- ☐ Los rectángulos redondeados representan tareas. Cada tarea está numerada de forma única por cuestiones de identificación. El nombre de la tarea se imprime en la mitad superior del símbolo. Los participantes en dicha tarea se imprimen en la mitad inferior del símbolo. El primer participante es siempre la persona que dirige la tarea (figura 4.2.2.2).





Figura 4.2.2.2 Esquema de tareas.

- Las flechas reflejan las entradas y las salidas de una tarea. Todas ellas tienen un nombre. Cuando se hace referencia a una de estas entradas o salidas en el texto, aparece subrayada (figura 4.2.2.3).

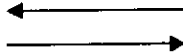


Figura 4.2.2.3 Entradas y salidas

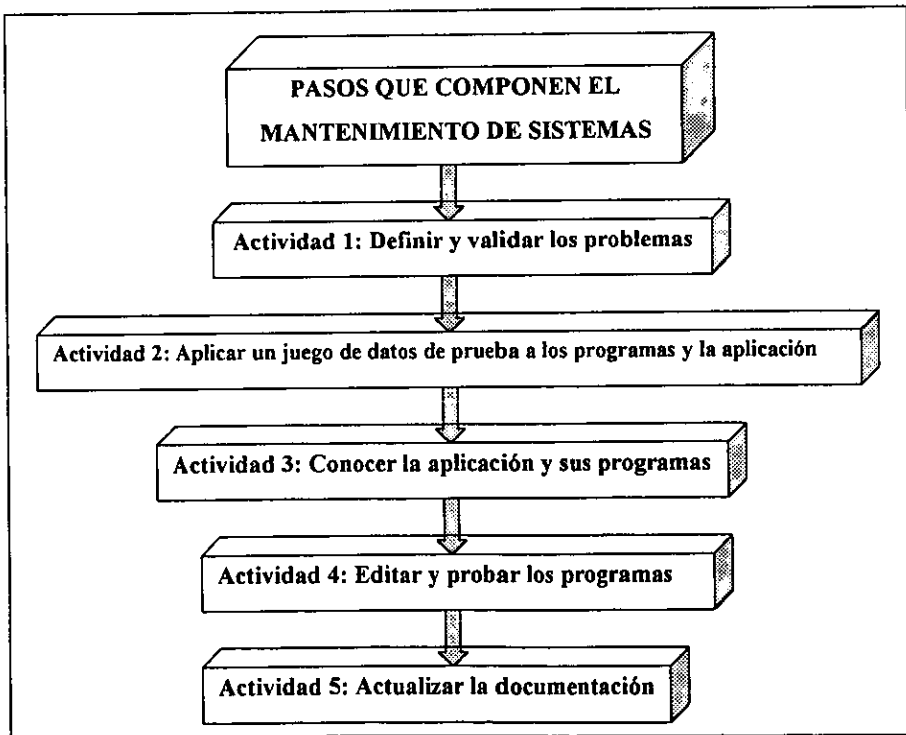


Figura 4.2.2.4 Pasos que componen el mantenimiento de sistemas.

Analicemos ahora los pasos que deben completarse durante el mantenimiento de sistemas. Como se muestra en la figura 4.2.2.4. Las figuras 4.2.2.5, 4.2.2.6, 4.2.2.7, 4.2.2.8 y 4.2.2.9 ilustran las tareas típicas del mantenimiento de sistemas.

### **Actividad 1: Definir y validar los problemas**

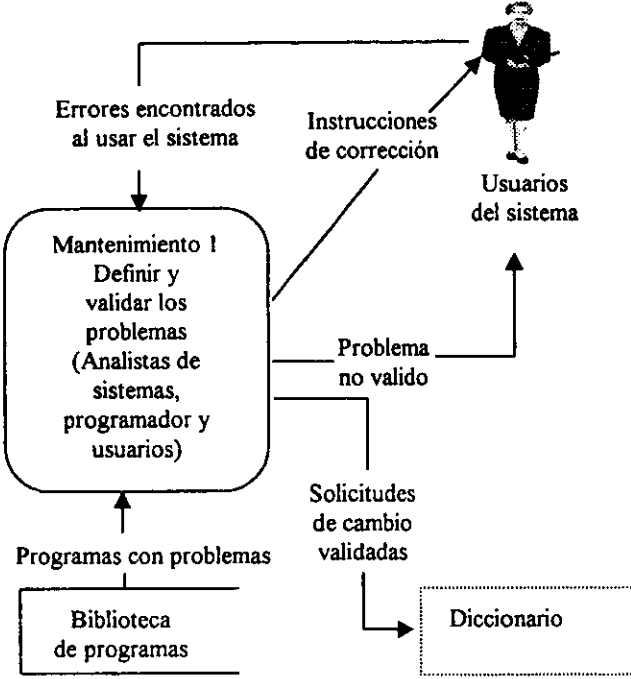
La primera actividad del equipo asignado será definir y validar los problemas. En el mejor de los casos, esta tarea será facilitada por el analista y/o el programador, pero en cualquier caso debería implicar claramente al usuario o usuarios. Los programas con problemas se recuperan de la biblioteca de programas.

Trabajando conjuntamente con el usuario o usuarios, el equipo deberá intentar validar el o los programas consiguiendo reproducirlos. Si el problema no puede reproducirse, debería suspenderse el proyecto hasta que se reprodujera el problema y el usuario pudiera explicar las circunstancias en las cuales tuvo lugar. La entrada es el conjunto de errores encontrados al usar el sistema (normalmente llamados **bugs**). Una posible salida sería las solicitudes de cambio validadas. Estas solicitudes de cambio deberían definir las expectativas de solución.

Otra posible salida es problema no válido. En el caso de que volviera a reproducirse el error, los usuarios deberían ser aleccionados para que documentaran del mejor modo posible las circunstancias que llevaron a la aparición del error y de los síntomas del problema.

En algunos casos, el error se debe a una mala comprensión o a un mal uso del sistema, y las instrucciones de corrección podrían llevar al cierre de todo el proyecto. Además, si el error ha sido validado, se pasarán los problemas y programas validados a la siguiente tarea.

**Nota:** Todo el mantenimiento posterior se llevará a cabo sobre una copia del o de los programas. El programa original permanecerá en la biblioteca de programas y podrá usarse en los sistemas en producción hasta que se subsane el error.



**Figura 4.2.2.5** Actividades y tareas del mantenimiento de sistemas.

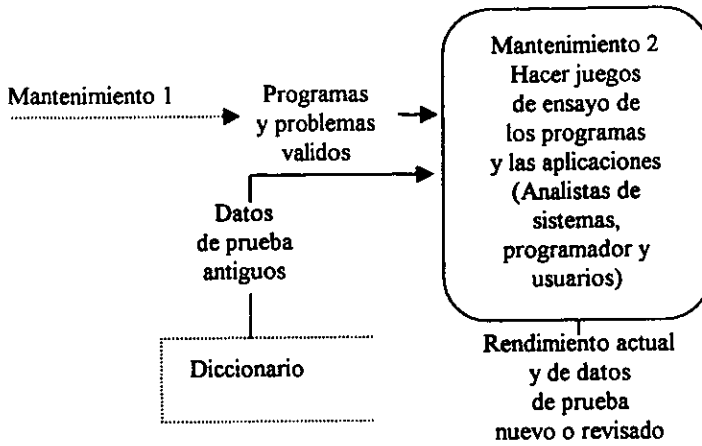


Figura 4.2.2.6 Actividades y tareas del mantenimiento de sistemas.

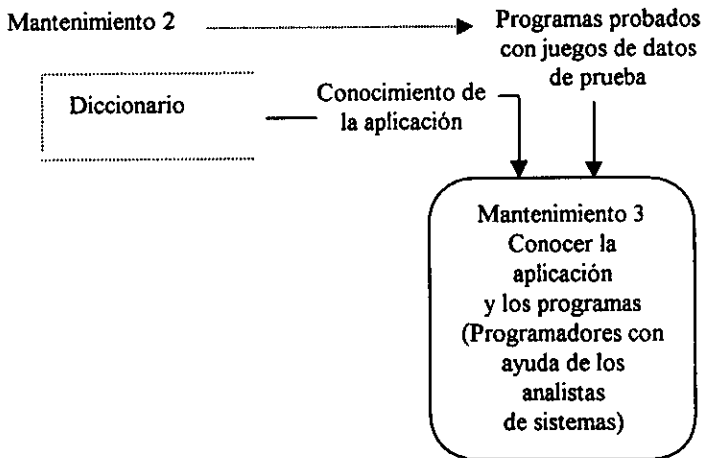


Figura 4.2.2.7 Actividades y tareas del mantenimiento de sistemas.

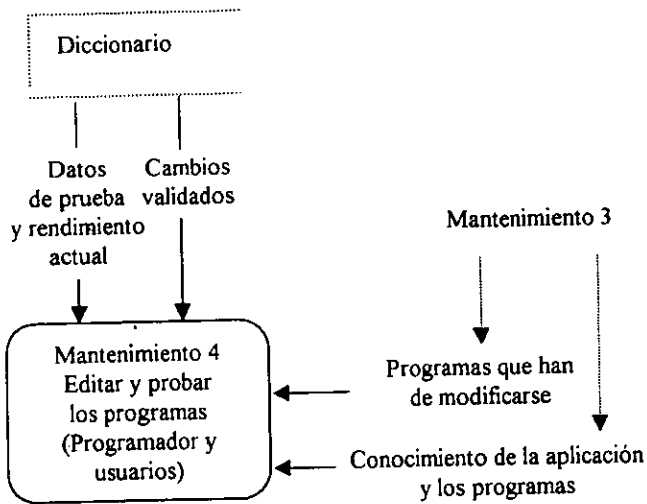


Figura 4.2.2.8 Actividades y tareas del mantenimiento de sistemas.

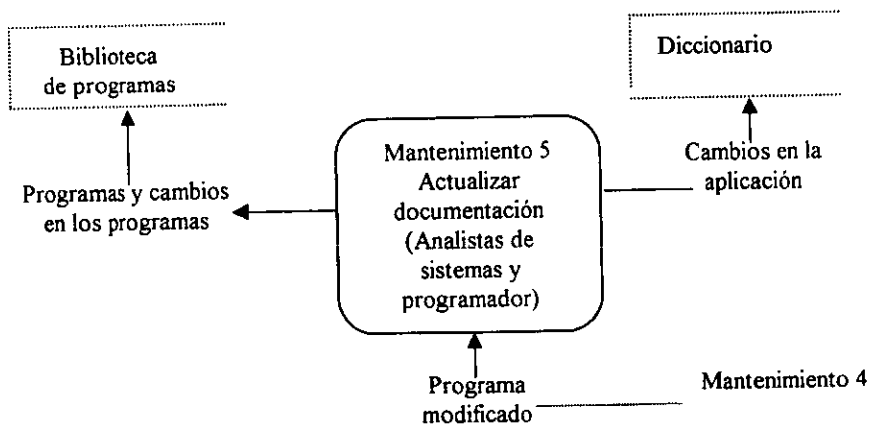


Figura 4.2.2.9 Actividades y tareas del mantenimiento de sistemas.

Estas actividades realizan el mantenimiento de los sistemas existentes. Adviértase que los programas se copian desde la biblioteca de programas y no se devuelvan a ella

hasta que finaliza el mantenimiento. Ello permite que el programa actual se mantenga en uso hasta que esté listo el programa revisado.

## **Actividad 2: Aplicar un juego de datos de prueba a los programas y la aplicación**

Los programas no son del todo malos, o no habrían sido puestos en producción en ningún momento. El equipo debería entonces aplicar un juego de datos de prueba a los programas y la aplicación. El mantenimiento de sistemas puede descubrir efectos impredecibles y no deseables que influirán sobre el funcionamiento y el rendimiento global de los programas o la aplicación. Por este motivo, recomendamos encarecidamente que, antes de realizar algún cambio en los programas, se ejecuten y se prueben para definir una línea de partida con respecto a la cual puedan compararse los programas y aplicaciones modificados.

Este paso es llevado a cabo por el analista y/o el programador de sistemas. Los usuarios también pueden participar en él. Las entradas principales al mismo son los problemas y programas validados.

Los casos del juego de datos de prueba pueden definirse de dos maneras posibles. La primera consistirá en buscar datos de prueba antiguos en el diccionario. Si existieran, deberían de servir como juego de datos de prueba para la ejecución del programa. También debería analizarse si son suficientemente complejos y, si fuera necesario, habría que revisarlos. Las respuestas correctas, incluidos los errores, deberían grabarse en el diccionario.

Alternativamente, es posible capturar automáticamente los datos de pruebas por medio del empleo de una herramienta de prueba. Por ejemplo, el producto de IBM llamado Workstation Interactive Test Tool (WITT) registra pulsaciones de teclado, movimientos de ratón e instrucciones, así como las respuestas del sistema, mientras un usuario muestra el uso típico del programa. (Nota: El usuario nunca debe saber que dicha

herramienta está activada) La ventaja de usar estas herramientas radica en que es posible grabar en el diccionario los ejemplos y las respuestas para reproducirlos más adelante. Las herramientas de pruebas también miden el tiempo de respuesta y la productividad obtenidos en los casos de prueba.

En cualquier caso, las salidas serán un rendimiento actual y de datos de prueba nuevo o revisado y programas probados con juegos de datos de prueba (que se pasan a la siguiente tarea).

El analista o programador debe disponer de buenos conocimientos en la realización de pruebas informáticas y puede requerir formación en las herramientas de prueba.

### **Actividad 3: Conocer la aplicación y sus programas**

Frecuentemente, el mantenimiento de sistemas no es realizado por las mismas personas que escribieron el programa. De hecho, tal vez varias personas hayan escrito partes de un programa o aplicación y ya no estén disponibles para posibles aclaraciones. Por esta razón, es preciso obtener un conocimiento de la aplicación y de los programas problemáticos. Tal vez sorprenda saber que, en su mayoría, los analistas y los programadores intervienen más tiempo en esta tarea que en ninguna otra.

En términos ideales, el conocimiento de la aplicación se obtiene del diccionario. Ello supone, claro está, que el conocimiento de la aplicación ha sido mantenido a lo largo de toda la vida media de la aplicación. Pero con mucha frecuencia esto no es verdad, en especial en los sistemas más antiguos. En centros de sistemas de información no basados en diccionarios, puede conseguirse el conocimiento de la aplicación de los programadores y analistas anteriores, pero dicho conocimiento puede no estar al día. Sin embargo, puede ser de utilidad para alcanzar un nivel suficiente de comprensión de

la aplicación y de donde encajan los programas problemáticos dentro de dicha aplicación.

El conocimiento de la aplicación y los programas proviene por lo general del estudio del código fuente de los programas probados con juegos de datos de prueba. Por desgracia, la comprensión de los programas puede llevar un tiempo considerable. Esta actividad se realiza por la combinación de una serie de limitaciones:

- ☒ Una deficiente estructura modular.
- ☒ Uso de lógica no estructurada (procedente de la era anterior a la estructuración).
- ☒ Mantenimientos previos (arreglos rápidos y ampliaciones deficientemente diseñadas).
- ☒ Código muerto (instrucciones que nunca se ejecutan o a las que nunca se llega, a menudo olvidadas durante fases anteriores de prueba y depuración).
- ☒ Documentación deficiente o inadecuada.

El propósito de conocer la aplicación es llegar a tener una visión general del asunto, es decir, saber cómo encajan los programas en una aplicación total y cómo interaccionan con otros programas. También tiene como objetivo la comprensión de los programas conseguir suficiente información sobre cómo funciona el programa, y sobre lo que no funciona. Para ello, hay que conocer los campos en el conjunto de los programas. Conocer los programas puede llevar también a mejores estimaciones del tiempo y los recursos que se requerirán para arreglar los errores.

No suele haber atajos posibles en la comprensión de los programas. En la actualidad, la tecnología CASE puede servir de ayuda para el mantenimiento. **Por ejemplo VIA/Insight de VIASOFT** proporciona medios para estudiar y analizar los programas para ofrecer al programador una amplia perspectiva e información sobre el código que no conoce. Pone de relieve la estructura del programa, marca o traza el código relacionado, incluso aunque esté presente en párrafos y subrutinas, aísla el código



muerto (código que no puede ejecutarse), hace un seguimiento de los usos de los campos (y sus no usos) y las relaciones entre los diferentes nombres de campos y estructuras y, finalmente, ofrece numerosas referencias cruzadas. Esta información puede reducir el tiempo necesario para conocer los programas en varias horas, días o incluso semanas.

El producto de IBM llamado COBOL Structuring Facility reestructura programas con deficiente estructura para adecuarlos a las prácticas generalmente aceptadas de programación estructurada. El producto VIA/SmartDoc de VIASOFT puede generar una gran riqueza de información, gráfica y de texto, para dichos programas.

#### **Actividad 4: Editar y probar los programas**

Dado el conocimiento de la aplicación y los programas y los cambios válidos, pueden entonces realizarse los cambios en los programas que han de modificarse.

Existe una gran diferencia entre editar un nuevo programa y editar un programa existente. Como diseñador y creador de un nuevo programa probablemente se estará muy familiarizado con la estructura y la lógica del programa. Por el contrario, como editor del programa existente, la familiaridad no será tan acusada con dicho programa. Los cambios que se introducen pueden tener un efecto de bucle no deseado que afecte a otras partes del programa o, lo que es a un peor, a otros programas de la aplicación.

Es de esperar que los cambios en el código sean mejores si se conocen (o se revisan) la aplicación y el programa. Pero más importantes aún en el mantenimiento de sistemas son las pruebas. Se consideran esenciales las siguientes pruebas:

☐ **Prueba de unidades** (esencial), que asegura que el programa considerado en solitario arregla el error sin efectos colaterales. En este punto se usarán los datos

de prueba y rendimiento actual que se recuperaron, crearon, editaron o generaron en la tarea 2.

- ☒ **Prueba del sistema** (esencial), que asegura que la aplicación en conjunto, de la que forma parte el programa modificado, aún funciona. De nuevo, se usarán aquí los datos de prueba y rendimiento actual de la tarea 2.
- ☒ **Prueba de regresión** (recomendada), que extrapola el impacto de los cambios en la productividad y el tiempo de respuesta del programa y la aplicación antes y después, usando para ello los datos de prueba y rendimiento actual.

Los programas de pruebas, como WITT de IBM (antes descrito), tan sólo simplifican las pruebas reproduciendo los datos registrados en los programas modificados. Los programas probados volverán a pasar a producción. En términos generales, puede decirse que cuando se programa se regresa a la biblioteca de programas y este queda sujeto al control de versiones.

El **control de versiones** es un proceso por el cual el bibliotecario (normalmente, un software) hace un seguimiento de los cambios realizados en los programas. Ello permite recuperar versiones anteriores de los mismos en caso de que en las nuevas versiones surjan problemas inesperados. En otras palabras, el control de versiones permite a los usuarios recuperar una versión del sistema aceptada con anterioridad.

**Ejemplos** de software de control de versiones son SCLM de IBM (para software basado en ordenadores centralizados) Y PVCS de INTERSOLV (para redes de área local).

#### **Actividad 5: Actualizar la documentación**

El alto costo del mantenimiento de sistemas se debe, en gran parte, a fallos en la actualización de la documentación de una aplicación, debe modificarse en el diccionario y en la biblioteca de programas. La documentación de la aplicación es, por

lo general, responsabilidad del analista de sistemas que da soporte a dicha aplicación. La documentación de los programas suele ser responsabilidad del programador que realiza los cambios en los programas.

El programador es el responsable de esta actividad. Su entrada es el programa modificado obtenido de la tarea 4. Los cambios en la aplicación (cambios en los modelos y las especificaciones) se guardan en el diccionario. Los nuevos programas y cambios en los programas se guardan en la biblioteca de programas. Una vez devueltos a la biblioteca, quedan disponibles para producción.

Grabar los cambios de las aplicaciones y los programas en el diccionario y la biblioteca de programas ayudará a los futuros programadores y analistas (tal vez los mismos) a reducir el tiempo dedicado al aprendizaje de la aplicación durante futuras tareas de mantenimiento. Los cambios realizados se olvidarán, por pequeños que sean, a menos que se guarde un registro apropiado de ellos. Las ventajas obtenidas a largo plazo de este trabajo llegarán cuando tengan que hacerse nuevos desarrollos importantes en la aplicación. La fase de estudio del análisis de sistemas se hará más rápidamente si existe una documentación actualizada.

#### **4.3 Recuperación del sistema: Superar los fallos generales del Sistema**

De cuando en cuando, es inevitable que un sistema falle. Este fallo se traduce generalmente en lo que se llama un programa <<abortado>> (también llamado <<ABEND>> o <<crash>>) y en posible pérdida de datos. Entonces, es a menudo el analista de sistemas el encargado de arreglar el sistema o de actuar como intermediario entre los usuarios y quienes deben recuperar el sistema. El propósito de esta sección es resumir brevemente el papel del analista en la recuperación de los sistemas.

Esta actividad no requiere un diagrama de flujo multitarea para detallar sus pasos, y puede resumirse del modo siguiente:

1. En muchos casos, el analista puede sentarse ante el terminal de usuario y recuperar el sistema. A veces, puede ser sencillo como pulsar una tecla específica o volver a arrancar el ordenador personal. Es posible que se requieran instrucciones de corrección para evitar que vuelva a producirse el fallo generalizado. En algunos casos, el analista puede quedarse a observar al usuario durante su siguiente uso del programa o la aplicación.
2. En ciertos casos, el analista debe ponerse en contacto con el servicio de explotación de los sistemas para corregir el problema. Las acciones realizadas por el servicio de explotación suelen consistir en dar fin a la sesión on-line y reinicializar la aplicación y sus programas.
3. En algunos casos, el analista puede tener que recurrir a la administración de datos para recuperar archivos o bases de datos perdidos o deteriorados.
4. En algunos casos, el analista puede tener que recurrir a la administración de redes para resolver un problema de redes locales o extendidas, o de interconexión de redes. Los profesionales de redes suelen desconectar al usuario y reinicializar los programas.
5. En algunos casos, el analista puede tener que recurrir a los técnicos o los representantes de los vendedores para arreglar un problema de hardware.
6. En algunos casos, el analista tal vez descubra el error que ha provocado el fallo. El analista intentará aislar dicho error rápidamente y bloquearlo (automáticamente o aleccionando a los usuarios para que lo eviten manualmente) para evitar dar lugar a otro fallo.

#### **4.4 Asistencia al usuario final**

Otra actividad permanente y relativamente rutinaria en el soporte de sistemas es la asistencia rutinaria al usuario final. Independientemente de cómo haya sido la formación de usuarios o de la calidad de la documentación, los usuarios requerirán asistencia adicional. El analista de sistemas está, por lo general, a disposición de los usuarios para ofrecerles ayuda en el uso diario de aplicaciones específicas. En aplicaciones de máxima importancia, el analista debe estar disponible día y noche.

Una vez más, es innecesario ofrecer un diagrama de flujo detallado para esta actividad.

Las tareas más características comprenden:

- Observación rutinaria del uso del sistema.
- Realización de estudios y reuniones para conocer el grado de satisfacción del usuario.
- Cambiar los procedimientos de empresa para que sean más claros (se escribirán y se grabarán en el diccionario).
- Ofrecer formación adicional.
- Anotar en el diccionario las ideas y las solicitudes sobre posibles mejoras.

#### **4.5 Mejoras y Reingeniería de Sistemas**

La adaptación de un sistema existente a las nuevas necesidades es una posibilidad siempre abierta en todos los sistemas de nueva implantación. El mantenimiento ligado a estas adaptaciones obliga al analista a analizar las nuevas necesidades y volver a las fases adecuadas del análisis, del diseño y la implantación de sistemas.

## 4.5.1 Tipos de Mantenimiento

En esta sección, examinaremos cuatro tipos de mantenimiento: correctivo, preventivo, perfectivo y adaptativo en el cual se encuentra él (aumentativo y tecnológico) (figura 4.5.1.1).

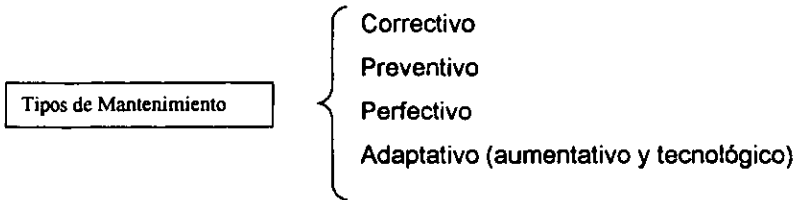


Figura 4.5.1.1 Tipos de mantenimiento.

### 4.5.1.1 Correctivo

La primera actividad del mantenimiento se da ocasionalmente cuando la prueba del software no haya descubierto todos los errores latentes de un sistema. Durante el uso del sistema se encontrarán errores, los cuales deben ser informados al equipo de desarrollo. El proceso que incluye el diagnóstico y corrección de uno o más errores se denomina mantenimiento correctivo.

### 4.5.1.2 Preventivo

En este tipo de mantenimiento es para prevenir errores. Se da cuando cambia el software para mejorar una futura facilidad de mantenimiento. También se puede considerar el mantenimiento a la información que se maneja para actualizarla y los resultados dados por el sistema sean los correctos.

#### **4.5.1.3 Perfectivo**

Esta actividad de mantenimiento se da cuando un paquete de software tiene éxito. A medida que se usa el software, se reciben de los usuarios recomendaciones sobre nuevas posibilidades acerca de modificaciones a funciones ya existentes. Para satisfacer estas peticiones se lleva a cabo el mantenimiento perfectivo.

El mantenimiento perfectivo comprende también los cambios solicitados al programador del sistema.

#### **4.5.1.4 Adaptativo (aumentativo y tecnológico)**

En este tipo de mantenimiento se encuentran implícitos el aumentativo y el tecnológico.

La vida útil estimada del software de aplicación puede fácilmente sobrepasar los diez años, pero considerando la evolución del ambiente, en la práctica éste puede volverse obsoleto. Por lo tanto, el mantenimiento adaptativo es una actividad que modifica al software para que las interacciones sean adecuadamente con su entorno cambiante.

El mantenimiento adaptativo se debe a cambios en el ambiente del programa y a la adaptación de nuevas unidades o módulos.

Un estudio echo por Lientz y Swanson (1980) descubrió que alrededor de 65% del mantenimiento era perfectivo, el 18% adaptativo, y el 17% correctivo.

#### **Aumentativo**

Este tipo de mantenimiento se da cuando se incluyen nuevas funciones que no se contemplaron al inicio del desarrollo del sistema y surgen como una necesidad del usuario.

## Tecnológico

Esta actividad que contribuye al mantenimiento se da debido a todo cambio importante en la informática. Si en un ciclo de 36 meses surgen nuevas generaciones de hardware, regularmente aparecen nuevos sistemas operativos o nuevas versiones de los antiguos; y frecuentemente se mejoran o modifican los equipos periféricos y otros elementos de sistemas.

### 4.5.2 Objetivos y bloques elementales de las mejoras y la reingeniería de sistemas

La mayor parte del mantenimiento de adaptaciones se hace como respuesta a la aparición de nuevos problemas de empresa, nuevas necesidades de información o nuevas ideas de mejoras. Por naturaleza, actúa como reacción, cuando algo se estropea o cuando los usuarios transmiten una solicitud. Estas actividades reciben el nombre de **mejoras del sistema**. El objetivo de las mejoras al sistema es modificar o ampliar el sistema de aplicaciones como respuestas a las necesidades cambiantes de empresa. Este objetivo puede relacionarse con los bloques elementales de los sistemas de información del modo siguiente (figura 4.5.2.1).

#### DIAGRAMA DE MEJORAS DEL SISTEMA

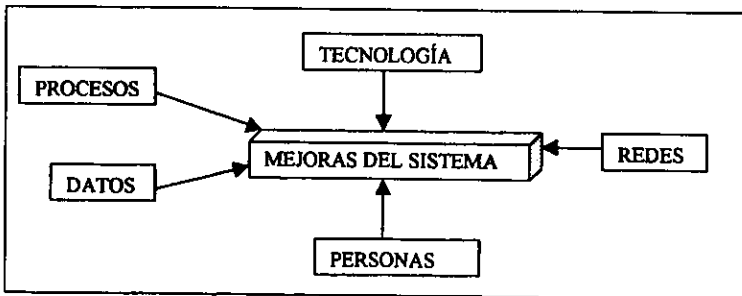


Figura 4.5.2.1 En esta figura se observa como las personas, datos, tecnología, redes y procesos, contribuyen a la mejora del sistema.



☐ PERSONAS. En su mayoría, las mejoras a los sistemas son propuestas por los usuarios de los sistemas, si bien los analistas, diseñadores y constructores de sistemas también pueden detectar posibles problemas técnicos relativos al rendimiento, la seguridad y los controles internos.

☐ DATOS. Muchas mejoras de los sistemas son demandadas de nueva información que pueden derivarse de datos almacenados existentes. Algunas mejoras de datos pueden requerir la ampliación del almacenamiento de datos.

☐ PROCESOS. En su mayoría, las mejoras a los sistemas requieren la modificación de programas existentes o la creación de nuevos programas para ampliar al ámbito general del sistema de aplicaciones.

☐ REDES. En su mayoría, las mejoras a los sistemas no tienen que ver con las redes.

☐ TECNOLOGÍA. En su mayoría, las mejoras a los sistemas se basan en la tecnología.

Otro tipo de mantenimiento que actúa como reacción es el asociado a los cambios tecnológicos. Los equipos de los sistemas de información se resisten cada vez más a esperar a que un sistema falle. En su lugar, optan por analizar sus bibliotecas de programas para determinar qué aplicaciones y programas son los más costosos de mantener o en cuáles el mantenimiento resulta más difícil. Estos sistemas pueden ser adaptados para reducir los costos de mantenimiento.

Los anteriores ejemplos de mantenimiento de adaptaciones se encuadran dentro de la categoría de **reingeniería**. Los objetivos de la reingeniería son o bien adaptar el sistema ante un cambio tecnológico importante y arreglar el sistema antes de que falle o bien hacer el sistema más sencillo de manejar para cuando falle o tenga que ser

adaptado. Estos objetivos pueden relacionarse con los bloques elementales de los sistemas de información del modo siguiente (figura 4.5.2.2).

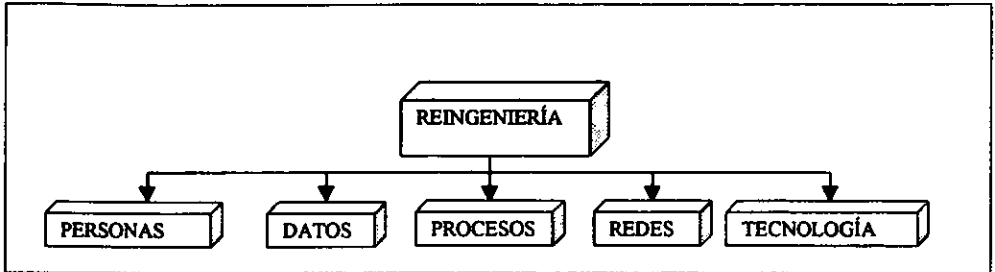


Figura 4.5.2.2 Entes que participan en la reingeniería.

- ☐ **PERSONAS.** En su mayor parte, la reingeniería es llevada a cabo por personal técnico y de sistemas de información.
  
- ☐ **DATOS.** Muchos proyectos de reingeniería son debidos a la necesidad de reestructurar los datos almacenados, ya sea para hacerlos más flexibles y fáciles de adaptar o para convertirlos a un nuevo entorno tecnológico.
  
- ☐ **PROCESOS.** Muchos proyectos de reingeniería intentan reestructurar o reorganizar programas de aplicación para hacerlos más fáciles de mantener o convertirlos a un nuevo entorno tecnológico (por ejemplo, el lenguaje). Muchos otros modifican los métodos de entradas y salidas en los programas (por ejemplo, de modo **batch** a modo **on-line**, o de modo **on-line** a interfaces gráficas de usuario).
  
- ☐ **REDES.** Algunos proyectos de aplicación buscan modificar las aplicaciones para adaptarlas a una nueva tecnología de redes.
  
- ☐ **TECNOLOGÍA.** En su mayoría, los proyectos de reingeniería se deben a cambios en la tecnología o a la necesidad de aprovechar mejor la tecnología existente.

Antes de seguir adelante, hemos de mencionar la existencia de otra tendencia en el soporte de sistemas. Cuando la vida útil de un sistema se acerca a su fin, los analistas y los diseñadores de sistemas pueden recurrir a tecnología de recuperación del análisis y el diseño para descubrir automáticamente los modelos y las especificaciones ocultos en los antiguos programas COBOL. (Véase el cuadro La nueva generación.)

### 4.5.3 Actividades, participantes y técnicas de las mejoras y la reingeniería de sistemas

La figura 4.5.3.1, 4.5.3.2, 4.5.3.3, 4.5.3.4, 4.5.3.5 y 4.5.3.6 recoge en detalle las actividades de las mejoras y la reingeniería de sistemas.

En esta sección, describiremos brevemente cada una de las actividades, participantes y funciones, entradas y salidas, y técnicas asociadas.

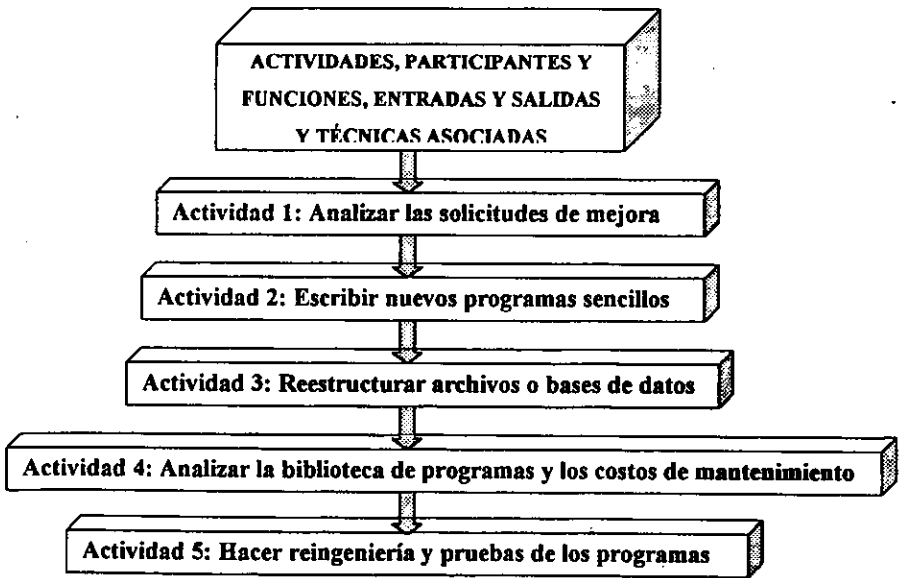


Figura 4.5.3.1 Actividades, participantes y funciones.

## **Actividad 1: Analizar las solicitudes de mejora**

El propósito de esta actividad es determinar el curso apropiado de acciones para tratar nuevos problemas de empresa o ideas de mejoras, problemas o limitaciones técnicas o ideas de mejora (resultantes de otras actividades de soporte). Recuérdese que esta fase de soporte, en general, no sirve en realidad para mejorar el sistema. En vez de ello, estudia la documentación existente para determinar el curso apropiado de acciones que han de emprenderse. Sobre la base del análisis de los modelos del sistema actual, estas acciones pueden incluir:

- ☒ Definir nuevas necesidades de empresa y volver al análisis de sistemas.
- ☒ Definir nuevas necesidades técnicas y volver al diseño de sistemas.
- ☒ Definir nuevas necesidades de programas y proceder a la tarea 2.

En el último caso, los nuevos programas se limitan generalmente a aquellos que generan nueva información a partir de los almacenes de datos existentes. Los programas más complejos deberían tratarse a través del análisis y el diseño de sistemas.

El analista de sistemas debería poseer experiencia en planificación de proyectos, investigación de hechos y análisis de costos/beneficios. Esta última técnica puede ser necesaria para justificar el proyecto de mejora.

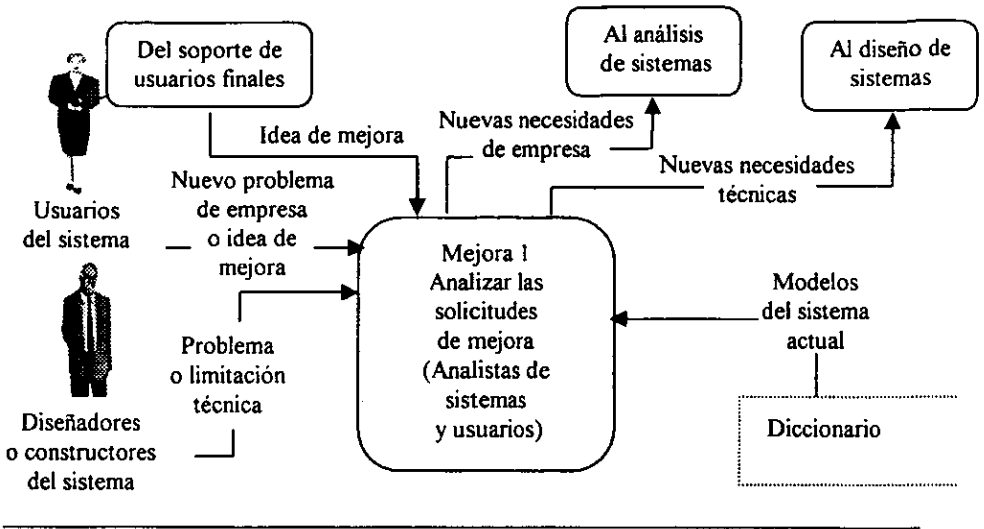


Figura 4.5.3.2 Actividades de mejora y de reingeniería de sistemas.

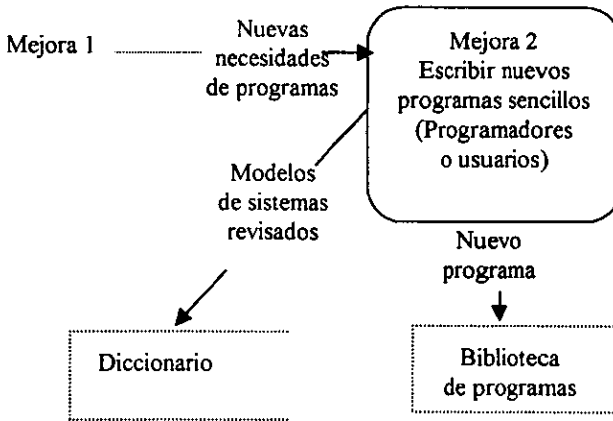


Figura 4.5.3.3 Actividades de mejora y de reingeniería de sistemas.

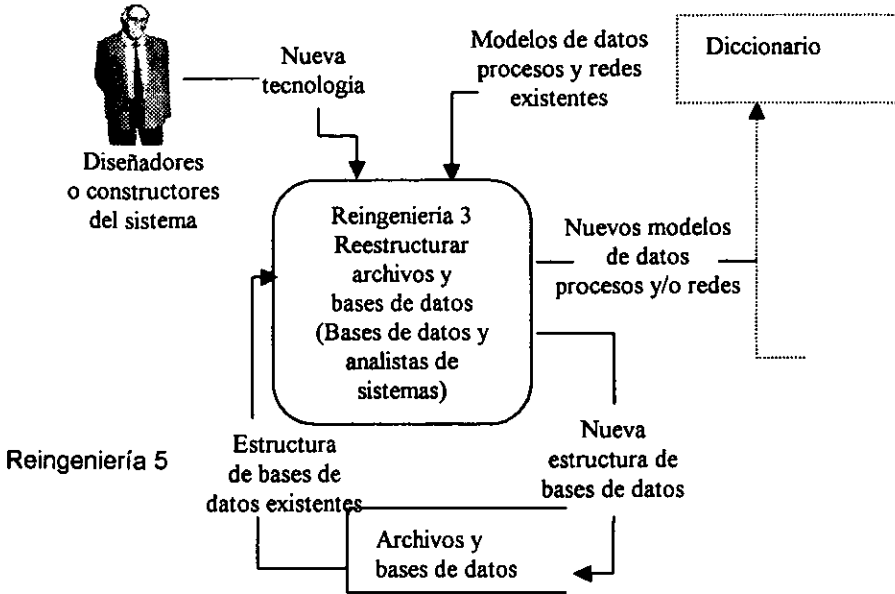


Figura 4.5.3.4 Actividades de mejora y de reingeniería de sistemas.

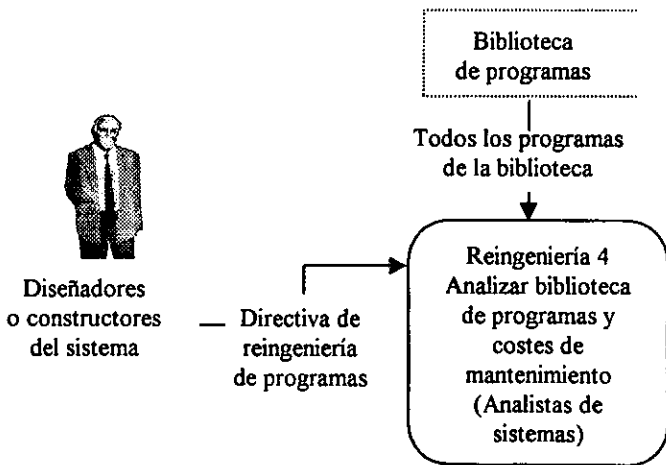
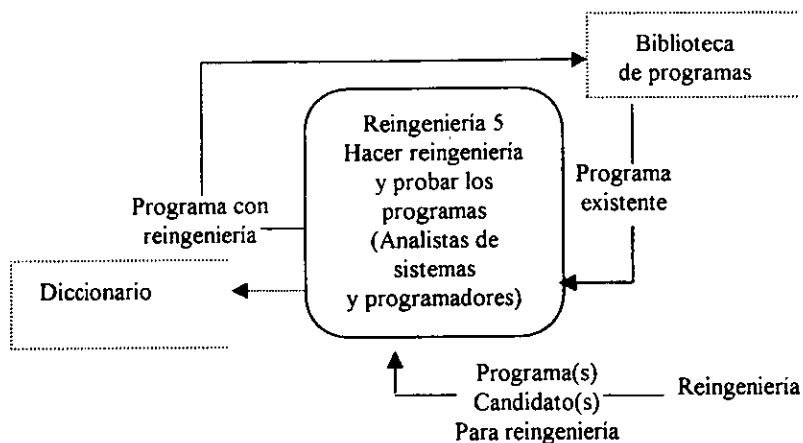


Figura 4.5.3.5 Actividades de mejora y de reingeniería de sistemas.



**Figura 4.5.3.6 Actividades de mejora y de reingeniería de sistemas.**

Estas tareas apoyan las mejoras y la reingeniería de los sistemas existentes. Adviértase que muchos proyectos de mejora y reingeniería devuelven el control al análisis o al diseño de sistemas para que sean consideradas con más profundidad las aplicaciones del cambio.

### **Actividad 2: Escribir nuevos programas sencillos**

Muchas mejoras pueden conseguirse rápidamente mediante la escritura de nuevos programas sencillos. ¿Qué quiere decir que los programas sean <<sencillos>>? Los programas sencillos son aquellos que utilizan datos existentes, no actualizan los datos existentes y no introducen nuevos datos (por motivos de almacenamiento de datos). En otras palabras, dichos programas generan nuevos informes y responden a nuevas consultas. Las necesidades de nuevos programas conforman la mayoría de las mejoras que se requieren hoy en día.

**Nota:** En nuestra opinión, toda nueva necesidad de programa que vaya más allá de nuestra definición de <<sencillo>> debería tratarse como una nueva necesidad de empresa y someterse al análisis y diseño de sistemas para considerar más en profundidad las posibles implicaciones que tendrá sobre la estructura completa del sistema de aplicaciones.

Muchos de estos programas pueden ser escritos fácilmente por usuarios finales con un mínimo conocimiento de lenguajes de cuarta generación (como Focus o SAS) o un lenguaje de recuperación de bases de datos desde una PC a un ordenador central (como Q & E), pero también están disponibles cada vez más en muchos paquetes de bases de datos del entorno de las PC'S (como Access, Approach y Paradox). Los programadores y los analistas son también capaces de escribir estos programas, pero algunos centros de sistemas de información ponen en duda que hagan así un uso valioso de su tiempo.

Con los lenguajes y herramientas de cuarta generación y de bases de datos disponibles hoy en día, estos programas pueden terminarse en cuestión de horas. Dado que, por lo general, no introducen o actualizan almacenes de datos, los requisitos de pruebas no son ni con mucho rigurosos. Una vez implantado, el nuevo programa puede almacenarse localmente (en una PC o un servidor LAN) o añadirse a la biblioteca de programas (si fueran muchas las personas que pudieran beneficiarse de su uso). Como los programas <<locales>> se crean para usuarios específicos, por lo general no se tienen en cuenta en el soporte dado por los centros de sistemas de información; sin embargo, en la mayoría de los casos el analista de sistemas del área o el usuario final local más experto ofrecen algún grado de soporte.

Opcionalmente, pueden actualizarse en el diccionario los modelos de sistemas revisados, con el fin de reflejar la existencia de los nuevos procesos (programas) que han sido añadidos al sistema.



Obviamente, la técnica principal necesaria para esta tarea es la programación en un lenguaje de cuarta generación (LAG). Los cursos sobre L4G se están haciendo cada vez más presentes en las instituciones académicas, ante la declinante popularidad y el uso decreciente de lenguajes de tercera generación como COBOL.

### **Recuperación del diseño y el análisis**

La reingeniería de sistemas se presenta en este capítulo como un medio moderno para ampliar la vida útil del patrimonio de aplicaciones y programas de su empresa. Sin embargo, todos los sistemas terminarán por quedar obsoletos, y se deberá volver a empezar la planificación o el análisis de sistemas y a construir completamente nuevas aplicaciones ¿O no es así?

En este subtema, hemos visto diversas técnicas de modelización de sistemas. Los detalles de los modelos se han registrado en un diccionario. Pero, ¿tendría que empezar siempre la planificación y el análisis de sistemas a partir de una hoja de papel en blanco? Algunos expertos creen que no, y están intentando desarrollar nuevas herramientas CASE que usen tecnologías de ingeniería inversa para recuperar los modelos y los detalles de análisis y diseño a partir de los programas existentes.

El ciclo de vida ha sido presentando en este subtema desde el punto de vista de la ingeniería directa. Primero, se analiza el problema y las necesidades, y después se diseña e implanta una solución. Pero también es posible hacer las cosas al revés; es lo que se llama ingeniería inversa.

Dentro de cada programa existente, no importa lo nuevo o antiguo que pueda ser; hay almacenados conocimientos de diseño de todas las personas que han trabajado alguna vez en el programa (tanto en su creación como en su mantenimiento). Si pudiéramos extraer estos conocimientos con facilidad, podríamos guardarlos en el diccionario y hacerlos disponibles para reducir los esfuerzos de planificación, análisis

y diseño de sistemas. Ello rebajaría enormemente la cantidad de tiempo que habría de invertirse en estas fases. Sobre estas cuestiones trata la recuperación del análisis y del diseño.

La **recuperación del diseño** es la captura de modelos y detalles de la implantación a partir de un programa. Pensemos en los beneficios obtenidos si una herramienta CASE pudiera leer un programas COBOL y generar sus organigramas, sus gráficos de estructuras y sus diagramas de flujo de datos automáticamente. El programador o analista podría entonces realizar los oportunos cambios técnicos y hacer la ingeniería directa del programa para construir uno mejor. Esta es la pretensión de herramientas en auge como Excelerator para Recuperación de Diseño, de INTERSOLV. Mediante el empleo de esta tecnología, deberíamos ser capaces de abordar objetivos de reingeniería más complejos y ambiciosos.

La recuperación del análisis es el último objetivo de este propósito. La **recuperación del análisis** es la captura de los modelos esenciales y los detalles asociados a partir de un programa implantado. Ello nos ofrecería un punto de partida rápido desde el que abordar las necesidades de empresa, siempre cambiantes y en expansión. Existe un punto de gran importancia que impide el desarrollo rápido de esta tecnología. Un programa es, por naturaleza, la implantación de un modelo esencial. La esencia está en el programa, pero no se ha registrado en él de forma explícita. En su mayoría, los investigadores coinciden en que lo más probable es que se usen soluciones basadas en sistemas expertos y de ayuda a la decisión. La herramienta de recuperación del análisis identificaría los componentes de los modelos de implantación e interaccionaría con los analistas y los usuarios para determinar y registrar su equivalencia esencial. Sin embargo, el gran potencial que se asociaría a la reducción del tiempo de arranque de los proyectos hace de la tecnología de recuperación del análisis un astro brillante en el horizonte informático.

### **Actividad 3: Reestructurar archivos o bases de datos**

De vez en cuando, los analistas de sistemas colaboran en la reingeniería de archivos y bases de datos. Muchos de los almacenes de datos de la actualidad se implantan con estructuras de archivos tradicionales (por ejemplo, estructuras jerárquicas IMS y estructuras en red). La tecnología actual de bases de datos más idónea es la de bases de datos relacionales son SQL (que almacenan los datos en tablas integradas por medio de campos redundantes que actúan con punteros). En el mañana, la tecnología de bases de datos puede tal vez presentar otro vuelco en su tendencia más popular.

La migración de las estructuras de datos de una tecnología a otra constituye un empeño de primer orden, más complicado aún por la posibilidad de deteriorar programas y datos de empresa esenciales. Así pues, la reingeniería de estructuras de archivos y bases de datos se ha convertido en una tarea muy importante.

El mediador clave en la reestructuración de bases de datos es el analista de bases de datos (o administrador de bases de datos). El analista de sistemas desempeña un papel de importancia, debido al impacto potencial en las aplicaciones existentes. Los analistas de redes pueden también verse involucrados en esta tarea si las bases de datos están, o han de estar, distribuidas en redes informáticas.

Las entradas clave a esta acción es la estructura de bases de datos existentes (que puede obtenerse del diccionario del sistema de gestión de bases de datos o archivos que se incluye en la mayoría de los almacenes de datos) y los datos, procesos y redes existentes también almacenados en el diccionario. Las salidas son una nueva estructura de bases de datos y un nuevo modelo de datos, procesos y redes.

En muchas empresas, la reestructuración de datos se ha hecho tradicionalmente a mano. Sin embargo, cada vez se utilizan más las herramientas CASE de reingeniería de bases de datos para leer la estructura de datos, producir el modelo de implantación, realizar análisis de datos para mejorar el modelo y regenerar la nueva estructura de

bases de datos. Sin discusión, líder en este campo ha sido Bachman Information Systems, cuyo producto Bachman/DBA ha ayudado a muchas empresas a convertir sus antiguos archivos VSAM y bases de datos IMS en bases de datos relacionales DB2 (y hoy día, en bases de datos relacionales en servidores Sybase). En su mayoría, las herramientas de reingeniería de bases de datos sólo convierten los almacenes de datos. Los procesos deben también ser convertidos para ejecutar los nuevos mandatos de recuperación y actualización de las bases de datos ante las nuevas estructuras de datos. Existe la tecnología para realizar tales conversiones, pero en general es muy cara.

Como ya hemos dicho con anterioridad, los analistas de sistemas deben tener formación tanto en bases de datos como en gestión de datos. En la mayoría de los programas educativos sobre sistemas de información existen cursos que proporcionan conocimientos en este campo.

#### **Actividad 4: Analizar la biblioteca de programas y los costos de mantenimiento**

Como se ha dicho antes, muchas empresas están cuestionándose las ventajas de invertir en tareas de mantenimiento de correcciones y adaptaciones. Se han dado cuenta de que si pudiera identificarse el software más complejo y costoso, podría ser preferible hacer una reingeniería para reducir la complejidad y los costos de mantenimiento. La primera actividad requerida para lograr este objetivo es analizar la biblioteca de programas y los costos de mantenimiento. Esta actividad casi siempre requiere software capaz de llevar a cabo el análisis. Los analistas de sistemas suelen ser quienes interpretan los resultados.

Herramientas de software como VIA/Recap de VIASOFT recorren la biblioteca de software utilizando una amplia variedad de métricas de software de aceptación general.

La **métrica de software** es un conjunto de medidas matemáticamente probadas sobre la calidad y la productividad del software.

Ejemplos de métrica de software aplicables al mantenimiento son:

☐ **Nudos de flujo de control**, o número de veces que se cruzan entre sí los caminos lógicos. En términos ideales, un programa debería tener cero nudos de flujo de control. (Hemos visto nudos que se cuentan por miles en algunos programas antiguos y deficientemente estructurados.)

☐ **Complejidad de los ciclos**, o número de caminos únicos a través de un programa. En términos ideales, cuantos menos sean mejor.

La métrica de software, en combinación con la contabilidad de costos (en las labores de mantenimiento), puede ayudar a identificar los programas cuya reestructuración produciría mayores beneficios.

Entradas a esta tarea son todos los programas de la biblioteca (o la mayoría de ellos). Su salida es un programa o programas candidatos para reingeniería.

### **Actividad 5: Hacer reingeniería y pruebas de los programas**

Dado un programa o programas candidatos para reingeniería, existen tres tipos de reingeniería que pueden aplicarse sobre dicho programa: reorganización de código, conversión de código y fragmentación de código.

☐ La **reorganización de código** reestructura la organización modular y/o la lógica del programa. Por ejemplo, los módulos pueden combinarse o separarse para reducir el acoplamiento o aumentar la cohesión. La lógica puede reestructurarse para eliminar nudos de flujo de control y reducir la complejidad de los ciclos. La herramienta de

IBM llamada COBOL Structuring Facility automatiza la reorganización de código en programas COBOL dotados de una estructura deficiente.

☐ La **conversión de código** traduce el código de un lenguaje a otro. Típicamente, esta traducción se realiza de una a otra versión de un mismo lenguaje. Por ejemplo, el producto de IBM COBOL and CICS Conversion Aid (CCCA) convierte COBOL basado en estándares anteriores a COBOL 2. Existe una cierta controversia sobre la utilidad de los traductores entre diferentes lenguajes. Si los lenguajes son suficientemente diferentes, la traducción puede ser muy difícil. Si la traducción es sencilla, podría plantearse la pregunta: <<¿Por qué cambiar? >> Por otra parte, podrían existir argumentos convincentes para traducir las llamadas COBOL basadas en tecnologías antiguas (por ejemplo, IMS) a llamadas basadas en nuevas tecnologías de bases de datos. Aunque no pudiera disponerse de conversión automática, la traducción de las instrucciones de llamada seguiría siendo esencial en caso de que cambiara la estructura de base de datos.

☐ La **fragmentación de código** constituye la opción de reingeniería de programas más interesante de todas. Muchos programas contienen componentes que deberían definirse como subprogramas. Si se hace tal descomposición, se obtendrían ventajas de mantenimiento. Y lo que es más importante, si se divide el programa, podría ser realizado en labores posteriores. La fragmentación de código extrae un fragmento de un programa para crear un programa o un subprograma independiente. Aunque pueda parecer fácil, no lo es en absoluto. Piénsese en un programa COBOL normal. El código que quiere fragmentarse puede estar situado en muchos párrafos y tener una lógica que dependa de muchos otros párrafos. Más aún, también se tendrá que extraer un subconjunto de la Data División con destino al nuevo programa o subprograma.

La fragmentación de código se simplifica enormemente con el uso de reingeniería de software. El producto VIA/Renaissance, de VIASOFT, es una herramienta de

fragmentación de código que hace un seguimiento del código para detectar el código y las estructuras de datos requeridos para la fragmentación. De esta forma, puede modificar el programa original para llamar al nuevo programa (borrando el código fragmentado del original) o mantener el programa original sin cambios.

El programa candidato para reingeniería se copia desde la biblioteca de programas. La reingeniería se hace por medio del empleo de uno o más de los métodos recién enunciados, se prueba minuciosamente (tal como se ha descrito anteriormente en el capítulo) y se devuelve el programa con reingeniería a la biblioteca de programas para dejarlo disponible para producción. Los nuevos modelos de datos, procesos y/o redes se actualizan en el diccionario.

## CONCLUSIONES

El trabajo realizado a lo largo del desarrollo del Sistema de Avalúos de Predios y Terrenos, así como la observación de su funcionamiento y aplicación en el trabajo cotidiano en la empresa, ha permitido a los autores obtener una serie de experiencias tanto a nivel personal como profesional, mismas que generan las siguientes conclusiones:

- ◆ La puesta en marcha del Sistema mejoró significativamente los tiempos requeridos para la captura, proceso y entrega de resultados a los usuarios, aplicando la tecnología avanzada a procesos cotidianos de avalúo de predios y terrenos.
- ◆ La operación sistematizada de los datos favoreció la confiabilidad de la información reportada, disminuyendo considerablemente los errores de cálculo y comparación debidos a fatiga del personal y al proceso de operaciones en forma manual.
- ◆ La integración de datos generada por la puesta en funcionamiento del Sistema, facilitó los cálculos del valor de los predios analizados, posibilitando la conciliación de datos y valuaciones previas con los valores obtenidos para cada caso en particular, más la ponderación de criterios especiales que los valuadores consideren aplicar.
- ◆ Técnicamente, el desarrollo del Sistema permitió comprobar la compatibilidad del lenguaje Visual Basic con el manejador de bases de datos Access 97, generando un ambiente agradable y de fácil manejo para el usuario, a través de ventanas con apoyos gráficos y menús de operación.
- ◆ Por otra parte, esta conjunción de lenguajes, favoreció las actividades de programación y diseño, permitiendo aplicar versatilidad al desarrollo, minimizando el código de programación y favoreciendo el mantenimiento del sistema.



- ◆ La base de datos del sistema permite la integración de información a partir de la cual se pueden identificar zonas con diferentes niveles de equipamiento urbano o infraestructura urbana, relacionadas con el Plan de Desarrollo de Urbanización del Distrito Federal, para la realización de estudios posteriores.
- ◆ El banco de datos que va generando el sistema es lo suficientemente flexible y dinámico para poder tener la información al día y para ampliarla de acuerdo a las necesidades crecientes.
- ◆ La recuperación de la totalidad o parte de la información se realiza en forma expedita, mediante listados impresos o directamente en el monitor del equipo, presentada a detalle, estratificada o resumida de acuerdo a los requerimientos especificados por el usuario.
- ◆ A pesar de que la programación desarrollada hasta el momento conforma un sistema completo, el diseño modular con el que fue creado, permite la incorporación gradual de nuevas aplicaciones que habiliten operaciones relacionadas, lo cual asegura que el rango de operación del sistema no sólo permanecerá vigente, sino que se tendrá la capacidad de ampliar progresivamente sus alcances.
- ◆ Por todo lo anterior, el grupo de trabajo concluye que el Sistema cumple con los objetivos planteados al inicio del desarrollo, cubriendo los requerimientos de información de los usuarios y ofreciendo la sistematización adecuada para la reducción del tiempo de respuesta en la entrega de resultados con un alto grado de confiabilidad en la información.
- ◆ En el aspecto de desarrollo profesional, el Sistema permitió a los autores obtener experiencia en el manejo de conocimientos adquiridos durante la formación educativa en el área de Ingeniería en Computación aplicados a la solución de problemas reales con requerimientos específicos.

- ◆ Además, la experiencia vivida, permitió constatar que la formación profesional en Ingeniería, amplía la perspectiva del individuo hacia áreas de conocimiento que no están directamente relacionadas con ella, pero que, sin embargo, requieren de la materia de Sistemas como herramienta para la automatización de funciones y procesos.
- ◆ Dadas las características y ventajas que presenta el sistema, su aplicación no está restringida para una empresa en particular, pudiéndose adaptar a las necesidades de la esfera tanto pública como privada.

## **BIBLIOGRAFÍA**

Apuntes de edificación Tomo I. Avalúos.

Francisco Solares Alemán.

UNAM, Facultad de Ingeniería.

Noviembre de 1984.

Criterios de Valuación I.

Luis González Orea, Jorge Aguilar Herrera.

1995.

Curso práctico de valuación inmobiliaria.

Ing. Raúl J. Pulido Sánchez.

Colegio de Ingenieros Civiles de México.

Material del curso "Catastro".

Ing. Manuel Madrigal Morales.

División de Educación Continua, Facultad de Ingeniería. UNAM.

Abril de 1988.

Introducción de la valuación de Inmuebles Urbanos.

Luis F. Cuellar Ulloa.

Mayo de 1986.

Análisis y Diseño de Sistemas de Información.

Jeffrey I. Whitten, Lonnie D. Bentley, Victor M. Barlow.

3ª edición Madrid 1996.

Traducida y producida por editorial Diorki.

Curso manual de Visual Basic 4.0.  
InforCap, Informática y Capacitación.  
1997.

Access para Windows 95.  
Jeffrey Byrne.  
Prentice Hall.  
1996.

Ingeniería de Software.  
Richard Fairley.  
Editorial Mc Graw Hill.

Catastro (conceptos, técnicas, avances, sistemas y aplicaciones).  
Horst Karl Dobner Ebert.

Fundamentos de los sistemas de bases de datos.  
S. M. Deen.  
Gustavo Gili.

Organización de las bases de datos.  
James Martín.  
Prentice Hall.

Metodologías de desarrollo.  
Antonio López.  
Macrobit-Rama.

## **Glosario**

### **A**

**Almacenadores:** Se representan por medio de dos líneas paralelas, y se utilizan para almacenar datos que serán utilizados por uno o más procesos.

**Avalúo:** Es el documento en el cual se ha reflejado la estimación del valor de un bien raíz en función de una unidad monetaria, para un mercado dado y una fecha determinante.

**Avalúo Bancario:** Este tipo de Avalúo se utiliza principalmente con objeto de fijar las bases para realizar transacciones, compraventa, hipotecas, etc.; y es la CNB (Comisión Nacional Bancaria) la que establece las Normas para llevarlos a cabo.

**Avalúo Catastral:** Tiene por objeto determinar el valor del bien para efectos fiscales, impuesto predial ó indemnizaciones gubernamentales.

### **B**

**Barra de Menús:** Despliega los comandos que se usarán para construir alguna aplicación.

**Bases de Datos:** Es un conjunto de información relacionada con un asunto o con una finalidad.

**Biblioteca de Programas:** Se conservan los programas de aplicación reales que han sido puestos en producción.

**Bien raíz:** Son las propiedades que no pueden transportarse de un lugar a otro.

## C

**Comercialización:** Es el hecho de negociar el predio del terreno, es decir dar quizás ciertos descuentos al valor real del terreno, dependiendo de los factores que afecten el terreno.

**Contexto-Sensitivo:** Significa que se puede obtener ayuda sobre ciertas partes sin tener que ir al menú de Ayuda.

## D

**Demanda:** Depende del deseo o necesidad que se tenga del producto y de la capacidad de compra que exista.

**Densidad Habitacional:** Es la relación que existe entre el número de viviendas y la superficie donde éstas se encuentran, considerando como promedio 5 integrantes por familia, esta se expresa en habitantes por hectárea (hab/ha).

**Descripción de Procesos:** Describe todos los procesos del modelo funcional que aparecen en el nivel final de refinamiento, con el propósito de definir lo que debe hacerse para transformar entradas en salidas.

**Diagrama Entidad-Relación:** El modelo Entidad-Relación surge con la idea de separar el diseño conceptual de datos del diseño físico, es decir, con el fin de plasmar en primer lugar mediante técnicas de diagramación en el mundo real, con sus entidades o agrupaciones de datos con significado dentro del sistema y las relaciones entre ellas.

**Diagrama de Contexto:** El primer nivel de un DFD (Diagrama de Flujo de Datos) es el diagrama de Contexto, en donde a través de una sola burbuja se representa al sistema completo, con datos de entrada y de salida respectivamente.

**Diagrama de Funcionalidad:** Son diagramas que representan en forma gráfica la secuencia en que se realizan las operaciones de un determinado procedimiento y/o el recorrido de las formas materiales.

**Diagrama de Flujo de Datos (DFD):** Es una técnica gráfica que representa el flujo de la información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida.

**Diagrama de Nivel 0:** Se representan los procesos principales de la aplicación que se está modelando, los cuales al llevarse a cabo el refinamiento sucesivo se irán explotando en diagramas de nivel inferior.

**Diagrama de Nivel n:** Son el resultado del refinamiento de cada una de las burbujas en distintos niveles para mostrar un mayor detalle.

**Diccionario Central:** Contiene todos los modelos de sistemas esenciales y de implantación, así como las especificaciones detalladas.

## **E**

**Entidad:** La componen una serie de datos que, agrupados, tienen un cierto significado para la empresa.

**Entidad Externa:** Son las entidades con las que el sistema se comunica, es por ello que residen fuera de los límites del sistema a ser modelado. Generalmente se trata de personas, organizaciones, documentos, sistemas de cómputo externos u otros objetos que proporcionen o consuman información del sistema.

## F

**Factor:** Es una variable que incide directamente en el valor de un predio. Con la obtención de los factores se puede aplicar un incremento o decremento al importe del bien evaluado.

**Factor por comercialización:** En este factor se involucra una investigación exhaustiva del mercado, que permite identificar los precios en que se están vendiendo los lotes, se toma en cuenta los antecedentes del predio, condiciones actuales y perspectivas físicas, políticas, sociales y jurídicas del mismo para que un determinado precio se disminuya o se eleve.

**Factor de Forma:** Este factor considera la geometría del predio diferenciando zonas habitacionales de otras zonas.

**Factor de Frente:** Este factor se determina tomando en cuenta la longitud de lo que se considera como parte frontal del terreno.

**Factor de homologación por servicios públicos:** Este factor es determinado en base a los servicios públicos con los que cuenta la zona en que se ubica el predio. Este factor algunas veces se determina en base al criterio del perito valuador que esté realizando el avalúo del predio.

**Factor de homologación por uso de suelo:** Este factor es determinado en base al uso de suelo que le corresponda a la zona en donde esté ubicado el predio. Este factor algunas veces se determina en base al criterio del perito valuador que esté realizando el avalúo del predio.

**Factor de Topografía:** Este factor toma en cuenta principalmente la pendiente del terreno.



**Factor de Ubicación:** Este factor toma en cuenta la ubicación del lote respecto a la manzana.

**Factor de Zona:** Este factor toma en cuenta donde está situado el predio con respecto a las calles y avenidas principales que se encuentran en esa zona.

**Flujo de Datos:** Representan la información de procesos, se muestran por medio de flechas curvas; la cabeza de la flecha indica la dirección del flujo de datos.

**Forma:** Funciona como una ventana, la cual puedes personalizar como una interface para tu aplicación. Pueden sumarse a esta forma controles, gráficos e imágenes para darle realce a la forma que se desea.

## **G**

**GUI:** Interface gráfica de usuario.

## **I**

**Inmueble:** Véase Bien raíz.

**Intensidad de Construcción:** Es la relación que existe entre el número de metros cuadrados construidos y la superficie del predio en el cual se encuentran estos, que se expresan en "n" veces el área del terreno.

**Investigación de Mercado:** Punto inicial para un Avalúo, análisis para obtener el valor de un inmueble en base a su ubicación, edad, calidad, etc.

## **Ll**

**Llave Primaria:** Consiste en uno o más campos que identifican a cada registro almacenado en la tabala.

## **M**

**Mantenimiento Adaptativo:** Es una actividad que modifica al software para que las interacciones sean adecuadamente con su entorno cambiante.

**Mantenimiento Correctivo:** Es el proceso que incluye el diagnóstico y corrección de uno o más errores se denomina mantenimiento correctivo.

**Mantenimiento Perfectivo:** Se da cuando un paquete de software tiene éxito.

**Mantenimiento Preventivo:** Se da cuando se cambia el software para mejorar una futura facilidad de mantenimiento.

**Menús de Contexto:** El ambiente de desarrollo de Visual Basic provee menús de contexto que contienen teclas rápidas para ejecutar más rápidamente las acciones más frecuentes.

**Mercadotecnia:** Cualquier relación entre individuos u organizaciones que implique un intercambio (una transacción), es mercadotecnia.

**Método del costo:** Basa toda su estructura en el costo de reposición menos demérito y en algunos casos en el costo de reproducción menos demérito.

**Método del ingreso:** Este método se funda en el valor futuro de ingresos presentes, es mucho más ágil, pero menos inexacto.

**Método del mercado:** Este método consiste en la comparación del inmueble valuado con otro u otros semejantes, cuyo valor real sea plenamente conocido y confiable.

**Metodología:** Sistema ordenado de proceder para la obtención de un fin.

**Métrica de Software:** Es un conjunto de medidas matemáticamente probadas sobre la calidad y la productividad del software.

**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.

**Modelos de Bases de Datos:** Los modelos de datos son herramientas conceptuales para describir los datos, sus relaciones, su semántica y sus limitantes.

**Modelo Jerárquico:** En un modelo jerárquico los datos están estructurados en forma arborescente y las relaciones entre los diferentes tipos de registros se resuelven mediante punteros o enlaces entre ellos.

**Modelo Relacional:** El modelo relacional de datos es definido por medio de una serie de reglas cuyo objetivo es lograr la independencia de la representación lógica de los datos de su almacenamiento físico.

**Modelo de Red:** Si se permite la relación entre "hermanos", es decir, entre registros de un mismo padre lógico, se tendrá una base de datos en Red.

**N**

**Normalización:** Es el proceso de agrupar a los campos de datos en tablas que representan a las entidades y sus relaciones.

**Normalización Primera Forma:** Cuando no contiene grupos que se repiten.

**Normalización Segunda Forma:** Los campos clave dependen de la totalidad de la clave.

**Normalización Tercera Forma:** Los campos no clave son independientes entre sí, es decir, no hay ningún campo no clave que depende de otro no clave.

## O

**Object Browser:** Lista los objetos disponibles para usarse en el proyecto y proporcionar una forma más rápida de navegar dentro del código.

**Oferta:** Depende del número de productos que existen en el mercado en un momento dado.

## P

**Poblaciones primarias:** Son aquellas que tienen una base económica propia.

**Poblaciones secundarias:** Son aquellas que dependen económicamente de otras, como por ejemplo ciudades satélites, ciudades residenciales, etc.

**Precio:** Puede entenderse como la suma de dinero pedida y algunas veces recibida por un producto.

**Predios irregulares:** Son los que tienen un solo frente y sus ángulos internos difieren en más de 10 grados del de 90 grados, los triángulos con lado a la calle, y los polígonos con más de 4 lados que no sean divisibles totalmente en cuadriláteros regulares.

**Predios regulares:** Se consideran regulares aquellos cuadriláteros con un sólo frente cuyos ángulos no difieren en más ó menos 10 grados del ángulo recto, los triángulos con dos o tres frentes, los polígonos que sean divisibles totalmente con frentes a la calle que reúnan las condiciones de los predios regulares.

**Procesos:** Se representan por medio de círculos, o burbujas en el diagrama. Representan las diversas funciones individuales que el sistema lleva a cabo.

**Programación Visual:** Se entiende el uso de expresiones visuales (tales como gráficas, dibujos, íconos, barra de menús, etc. ) en el proceso de la programación de aplicaciones.

## **R**

**Reingeniería:** Adaptar el sistema ante un cambio tecnológico importante y arreglar el sistema antes de que falle o bien hacer el sistema más sencillo de manejar para cuando falle o tenga que ser adaptado.

**Relaciones:** Constituyen el enlace entre las diferentes entidades del sistema.

**Relación uno a uno:** Una entidad en A está relacionada únicamente con una entidad de B y una entidad de B con sólo una de A.

**Relación uno a muchos:** Una entidad de A está relacionada con cualquier número de entidades B, pero una entidad de B sólo está relacionada con una de A.

**Relación muchos a uno:** Una entidad de A está relacionada únicamente con una entidad de B, pero una entidad de B está relacionada con cualquier número de entidades en A.

**Relación muchos a muchos:** Una entidad A está asociada con cualquier número de entidades B, y una entidad de B está vinculada con cualquier número de entidades de A.

## S

**Soporte de Sistemas:** Es el mantenimiento permanente de un sistema después de que haya sido puesto en operación.

## T

**Tablas:** Es un conjunto de datos sobre un tema específico. Las tablas organizan los datos en columnas (denominados campos) y filas (denominadas registros).

**Terreno:** Es la porción o porciones de Tierra, incluyendo, en su caso, las construcciones ó mejoras, cuyos linderos con propiedades ajenas se encuentran totalmente definidos y determinen un perímetro sin solución de continuidad.

**Terrenos Industriales:** Son los terrenos destinados a construcción de industrias ya sea dentro o fuera de las zonas industriales.

**Terrenos Rústicos:** Son aquellos que se ubican fuera de las poblaciones y que carecen de los servicios fundamentales como pavimentos, alumbrado, agua, drenaje, etc. Son por lo general terrenos agrícolas, predios destinados a habitación campesina (pueblos, rancherías, villas, etc.).

**Terrenos Suburbanos:** Son aquellos que cuentan con los más indispensables servicios, aunque en algunos casos estos sean colectivos, caso de hidrantes públicos, energía eléctrica, etc.

**Terrenos Urbanos:** Son aquellos que se encuentran ubicados dentro de la zona urbana, fijada por las autoridades.

**Toolbar (Barra de Herramientas):** Proporciona un acceso rápido a los comandos comúnmente usados en el ambiente de Programación.

**Toolbox (Caja de Herramientas):** Provee un conjunto de herramientas que se utilizan en el modo de diseño de controles sobre una forma.

## U

**Uso comercial:** Se caracteriza por la densidad de comercios, negocios, oficinas, hoteles, etc.

**Uso de Suelo:** Se refieren a aquella actividad específica a la que se encuentra dedicado o se pretende dedicar un predio en particular, los cuales pueden ser habitacional, comercial, oficinas, clínica, etc.

**Uso industrial:** Zonas usadas principalmente para el establecimiento de fábricas, oficinas, almacenes, plantas de energía, depósitos de abastecimientos.

**Uso institucional:** Areas con edificios públicos gubernamentales, escuelas, hospitales, iglesias, etc.

**Uso residencial:** Predominan viviendas propias o rentadas, con casas separadas o semiseparadas, casas construidas espalda con espalda, bungalows.

**Uso residencial departamental:** Se encuentran principalmente edificios de condominios o departamentos alquilados.

**Uso para transporte:** Donde se encuentran vías de carreteras, ferrocarriles o fluviales.

## V

**Valor Catastral:** Es aquel que se establece tomando como base los valores dados por la oficina de catastro de la localidad que sirven generalmente para fijar los impuestos prediales.

**Valor Comercial:** Para los efectos del avalúo comercial de un bien raíz, es la cantidad expresada en moneda, justa y probable que se podrá obtener por una propiedad en un mercado abierto y en una fecha determinada.

**Valor de Capitalización:** Se determina con base en la rentabilidad del edificio que se supondría el ideal para el máximo aprovechamiento del terreno.

**Valor de Garantía:** Es el valor que tiene un inmueble estimado para cubrir riesgos futuros previsibles. Este valor se expresa como resultado de un reajuste del valor comercial, el cual fue determinado en base al valor en plaza.

**Valor de Mercado:** Es el precio que un comprador pudiera pagar y un vendedor pudiera aceptar, si cada uno está bien informado y advertido de las ventajas y desventajas de un inmueble, y ambos son guiados en este acto por motivos de un inversionista normal, libre de presiones, con conocimiento de la vocación o uso autorizado del suelo.

**Valor de Rentabilidad:** Se obtiene por la capitalización adecuada de las rentas anuales recibidas de una propiedad, ya afectadas por las deducciones propias correspondientes.

**Valor de Reposición:** Es el importe que será necesario invertir para reemplazar un inmueble, con las mismas características y amenidades que posee.



**Valor de Reproducción:** Es el importe que será necesario invertir para rehacer un inmueble por otro exactamente igual.

**Valor en Plaza:** Es aquel establecido en un mercado abierto por un número de transacciones reales efectuadas entre vendedores y compradores, sobre inmuebles de semejantes características.

**Valor Físico:** Es el que tiene una propiedad tomando en cuenta solo su costo de reposición menos su depreciación.

**Valor objetivo:** Es el costo de crear o reproducir un bien. Se afirma que ese valor es intrínseco al bien valuado, es decir, que depende del costo de fabricación o reproducción.

**Valor subjetivo:** Es el que existe en la mente de los individuos y refleja la estimación o deseo de poseer un bien.

**Valuación:** Es el procedimiento de estimar el precio justo de una casa, un producto o un servicio.

**Valuación de inmueble:** Es el procedimiento de estimar el precio justo de un bien raíz.

**Valuador:** Persona que ayudará al vendedor a decidir qué tanto pedirá por una propiedad, o a un comprador a decidir que tanto ofrecerá pero el valuador nunca tomará la decisión final.

**Ventana de Código:** Es el lugar donde se escribe el código en Visual Basic para la aplicación.

**Ventana de Propiedades:** Lista el conjunto de propiedades para la forma o el control seleccionado.

**Ventana del Proyecto:** Lista las formas y módulos en el proyecto actual, es decir, el proyecto con el que se está trabajando.

**Visual Basic:** Herramienta (Software) de desarrollo de Aplicaciones generadas mediante lo que se denomina programación visual.

## **APÉNDICE A**

### **Manual de usuario**

#### **¿A quién está dirigido el manual?**

El manual está dirigido al usuario final del sistema, el cual debe tener conocimientos previos en Microsoft Windows, asegurando así la completa comprensión del presente documento.

En el punto Conceptos Básicos apartado de Microsoft Windows, se presentan en general algunos temas, con la finalidad de manejar la misma terminología sobre los objetos que el ambiente maneja. Si es un usuario experto en Microsoft Windows puede no consultarlo.

## 1. CONCEPTOS BÁSICOS

Microsoft Windows es un Subsistema Operativo con una interfaz gráfica, su principal área de trabajo es un escritorio, donde usted puede añadir o eliminar elementos de acuerdo a sus necesidades.

Al iniciar Windows, aparecerá en su pantalla el menú de inicio, con el cual podrá organizar sus aplicaciones y accederlas.

Una vez que inicie una aplicación, ésta se ejecutará en una ventana de aplicación que aparecerá en el escritorio.



Para mayor información acerca del tema, refiérase al *Manual del usuario de Microsoft Windows*, en el capítulo *Fundamentos de Windows*.

## 2. ESTÁNDARES DE MICROSOFT WINDOWS.

### 2.1 BARRA DE MENÚ.

Contiene los nombres de los menús que están situados cerca del borde superior de la ventana, los cuales incluyen la lista de menús disponibles en la aplicación. Para acceder un menú hay que seleccionar su nombre utilizando el mouse o presionando la combinación de teclas Alt + la tecla subrayada que muestre el menú.



Fig. 2.1.1 Barra de menú típica

Por ejemplo, para acceder mediante el teclado el menú de Archivo se deberá presionar la tecla ALT+ A.

## 2.2 TECLAS DE MENÚ

Use las siguientes teclas para seleccionar menú y elegir opciones de éstos.

PRESIONE LA TECLA	PARA
ALT o F10	Para posicionarse en la barra de menús.
IZQUIERDA o DERECHA (← →)	Desplazarse horizontalmente a través de la barra de menús.
ARRIBA o ABAJO (↑ ↓)	Desplazarse verticalmente a través de las opciones de un menú seleccionado.
ENTRAR	Desplegar las opciones del menú seleccionado, en caso de tratarse de una opción la ejecución de ésta.
ESC	Cancelar el nombre del menú seleccionado o cerrar el menú abierto.

## 2.3 TECLAS DE NAVEGACIÓN ENTRE OBJETOS

La tabla siguiente nos indica las teclas que pueden ser utilizadas cuando este en una ventana de aplicación.

PRESIONE LA TECLA	PARA
TAB	Mover de control en control (de izquierda a derecha).
SHIFT+TAB	Mover de control en control en sentido inverso.

PRESIONE LA TECLA	PARA
ALT+TECLA DE SUBRAYADA	Mover a la opción o grupo cuya letra o nombre subrayado es igual al que se ha escrito.
TECLA DE DIRECCIÓN	Mover el cursor de selección de opción en opción, dentro de un grupo de opciones. O mover el cursor hacia la izquierda, derecha, arriba o abajo, dentro de una lista o cuadro de texto.
INICIO	Mover al primer elemento o carácter en una lista o cuadro de texto.
FIN	Mover al último elemento o carácter en una lista o cuadro de texto.
AV PÁG o RE PÁG	Desplazar hacia arriba o hacia abajo en un control de lista.
ALT+FLECHA ABAJO	Desplegar una lista.
BARRA ESPACIADORA	Seleccionar un elemento o cancelar una selección en una lista.
SHIFT+TECLA DE DIRECCIÓN	Ampliar o cancelar la selección en un cuadro de texto, un carácter por vez.
SHIFT+INICIO	Ampliar o cancelar la selección hacia el primer carácter en un cuadro de texto.
SHIFT+FIN	Ampliar o cancelar la selección hacia el último carácter en un cuadro de texto.

PRESIONE LA TECLA

PARA

ENTRAR

Ejecutar un comando o elegir el elemento seleccionado en una lista y después, ejecutar un comando.

ESC o ALT+F4

Cerrar una ventana de aplicación o cuadro de diálogo sin realizar el comando.

Las figuras siguientes nos muestran ejemplos de una ventana de aplicación y un cuadro de diálogo.

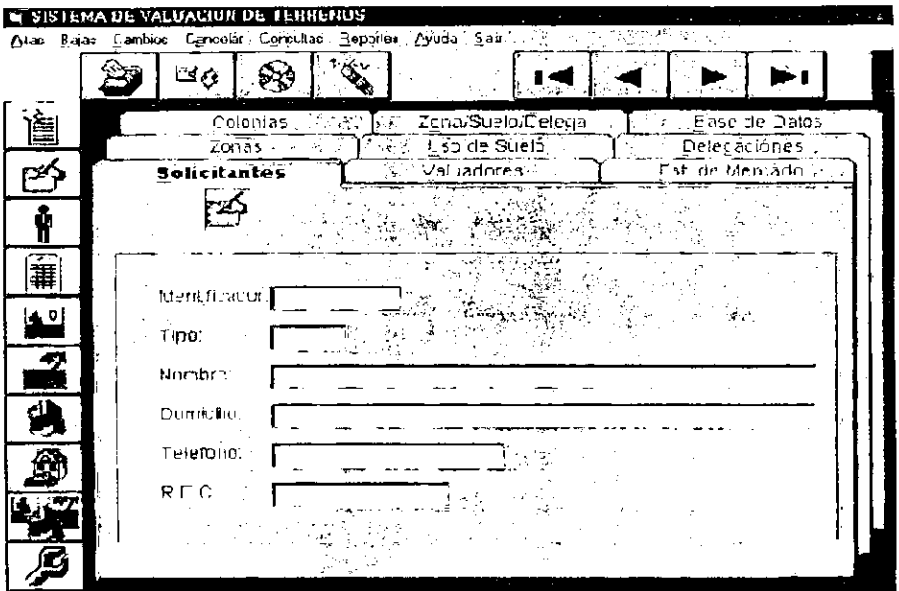


Fig. 2.3.1 Ventana de Aplicación



Fig. 2.3.2 Cuadro de diálogo

## 2.4 TECLAS DEL SISTEMA

Las siguientes teclas pueden ser usadas desde una ventana o una ventana completa, sin considerar la aplicación que se esté usando.

PRESIONE LA TECLA	PARA
CTRL+ESC	Conmutar a lista de tareas.
ALT+ESC	Conmutar a la siguiente aplicación, tanto si se está ejecutando en una ventana o como un icono.
ALT+TAB	Conmutar a la última aplicación usada, o bien, cambiar a la próxima aplicación, restaurando las que se están ejecutando como iconos. Se debe mantener presionada la tecla ALT mientras se presiona repetidamente TAB.
ALT+ESPACIO	Abrir el menú Control para una aplicación de ventana.
ALT+GUIÓN	Abrir el menú Control para una ventana de documento.



ALT+F4	Salir de una aplicación o cerrar una ventana
CTRL+F4	Cerrar la ventana de grupo activa o la ventana de documento.
ALT+ENTRAR	Cambiar una aplicación No-Windows entre ejecutarse en una ventana y ejecutarse en una pantalla completa.

## 2.5 TECLAS PARA MOVIMIENTO DEL CURSOR

Use las teclas siguientes para mover el cursor o insertar el puntero en cuadros de texto y en otros controles en los que se puede escribir texto.

PRESIONE LA TECLA	PARA MOVER PUNTO DE INSERCIÓN.
FLECHA ARRIBA	Una línea hacia arriba.
FLECHA ABAJO	Una línea hacia abajo.
FLECHA DERECHA	Un carácter hacia la derecha.
FLECHA IZQUIERDA	Un carácter hacia la izquierda.
CTRL+FLECHA DER.	Una palabra hacia la derecha.
CTRL+FLECHA IZQ.	Una palabra hacia la izquierda.
INICIO	Hacia el comienzo de la línea.
FIN	Hacia el fin de la línea.
CTRL+INICIO	Hacia el comienzo del documento.

CTRL+FIN

Hacia el final del documento.

## 2.6 TECLAS PARA LA SELECCIÓN DE TEXTO

Las siguientes teclas pueden ser usadas con la mayoría de las aplicaciones de Windows. Sin embargo, puede que alguna de las que están enlistadas aquí, no se puedan utilizar en todas las ocasiones en que se selecciona texto o para todas las aplicaciones. Las siguientes selecciones empiezan en el lugar del punto de inserción. Si el texto está seleccionado, presione las teclas para cancelar esa selección.

PRESIONE	PARA SELECCIONAR O CANCELAR LA SELECCIÓN DE TEXTO
MAY+IZQUIERDA o DERECHA	Un carácter por vez a la derecha o a la izquierda.
MAY+ARRIBA o ABAJO	Una línea de texto, arriba o abajo.
MAY+RE PÁG	Todo el texto una pantalla hacia arriba.
MAY+AV PÁG	Todo el texto una pantalla hacia abajo.
MAY+INICIO	Texto hacia el comienzo de la línea.
MAY+FIN	Texto hacia el fin de la línea.
CTRL+MAY+IZQUIERDA	La palabra anterior.
CTRL+MAY+DERECHA	La palabra siguiente.
CTRL+MAY+INICIO	Texto hacia el comienzo del documento.
CTRL+MAY+FIN	Texto hacia el final del documento.

## 2.7 BARRAS DE DESPLAZAMIENTO

Se utilizan para moverse horizontal y verticalmente en controles de tipo tabla, lista y combo box mediante el uso del mouse. Se encuentran en el borde inferior y derecho de la ventana de la aplicación, conteniendo marcadores, denominados cuadros de desplazamiento, que indican la posición vertical y horizontal del punto de inserción en la ventana.

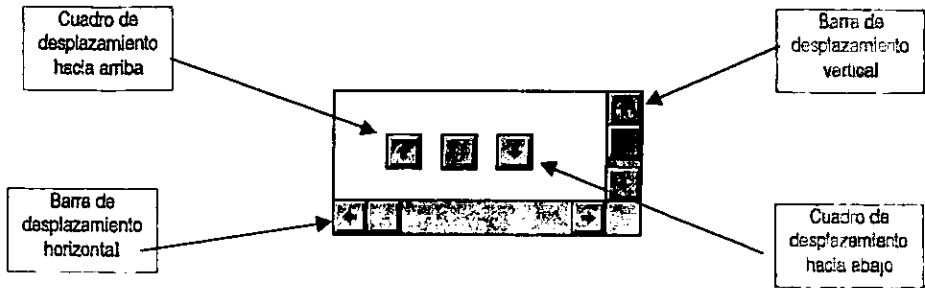


Fig. 2.7.1 Barras de desplazamiento

### Manejo de las barras de desplazamiento

PARA	HAGA ESTO
Desplazarse una línea hacia arriba	Haga click en el botón flecha arriba.
Desplazarse una línea hacia abajo	Haga click en el botón flecha abajo.
Desplazarse una tabla, lista o combo completamente hacia arriba	Haga click encima del cuadro de desplazamiento.
Desplazarse una tabla, lista o combo completamente hacia abajo	Haga click debajo del cuadro de desplazamiento.

Moveirse a una posición aproximada de la tabla, lista o combo.	Arrastre el cuadro de desplazamiento.
--	---------------------------------------

La barra horizontal también contiene botones análogos que puede utilizar para cambiar rápidamente de posición en la ventana.

## 2.8 TIPOS DE CONTROLES U OBJETOS

### 2.8.1 LISTA DE DESPLEGADO

En principio las listas desplegables se presentan como un rectángulo con un cursor de selección. Al seleccionar el botón flecha abajo que aparece a la derecha del cuadro, aparecerá una lista de opciones disponibles. Si existen más elementos de los que pueden ser mostrados en el cuadro, éste tendrá barras de desplazamiento.

Este objeto se puede encontrar en datos como delegación, colonia, tipo de suelo, zona, etc.

El procedimiento de selección de un dato de este objeto se ejemplificará con el campo delegación.

1. Colocar el puntero del mouse sobre la flecha o cursor de selección, que se encuentra en el lado derecho del objeto.



Fig. 2.8.1.1 Lista desplegable que contiene el catálogo de delegaciones

2. Al dar un click sobre el mismo se despliega la información contenida en la lista desplegable.

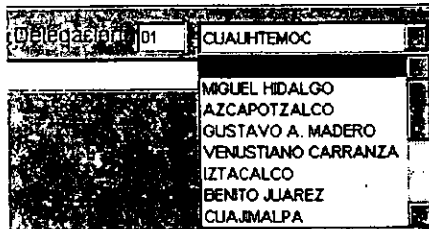


Fig. 2.8.1.2 Lista desplegable que muestra su contenido

3. El dato en el que este el puntero del mouse en ese momento se encuentra iluminado, lo cual indica que es el dato que esta seleccionado en ese momento.
4. Ahora se puede seleccionar cualquier dato al mover el puntero del mouse sobre la lista mostrada y dar click sobre el dato que se elija. Si se desea ver más datos de la lista basta con navegar a través de la lista mostrada mediante las barras de desplazamiento.
5. Ya que se tiene seleccionado el dato de interés, solo basta con dar click sobre el mismo para que quede seleccionado.

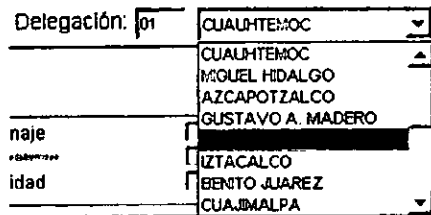


Fig. 2.8.1.3 Lista desplegable mostrando la delegación seleccionada

Otra forma de seleccionar datos contenidos en controles de listas desplegables es utilizando el teclado.

1. Presionar la tecla F4 una vez que se encuentra en la lista desplegable (ver navegación de objetos) para desplegar la información del mismo.
2. Presionar las teclas flecha arriba(↑) o abajo(↓) para desplazarse por cada uno de los elementos de la lista.
3. Para seleccionar el dato deseado basta con presionar la tecla ENTER sobre el mismo para que el dato quede seleccionado.

Existe una tercera alternativa para la selección de un elemento dentro de la lista desplegable, esta consiste en seleccionar la lista desplegable y teclear la primer letra del elemento buscado, si existe más de un elemento cuyo nombre inicie con la letra proporcionada se mostrará el primer elemento encontrado.

## 2.8.2 CAJA DE TEXTO

La información que usted escribe se captura en un cuadro de texto. Al desplazarse a un cuadro de texto vacío, aparecerá un punto de inserción (barra vertical intermitente). El texto que escriba se muestra en un punto de inserción. Si el cuadro al cual se desplace contiene texto y se encuentra resaltado, cualquier texto que escriba lo reemplazará.

Este objeto se puede ubicar en campos como propietario, superficie, valor del terreno, etc. Para ilustrar su funcionalidad ver el siguiente ejemplo en el que se muestran para el caso de la superficie.

Para indicar la superficie del terreno se realiza colocando el puntero del mouse sobre la caja de texto de superficie y teclear la cantidad en metros cuadrados.

Superficie:

Fig. 2.8.2.1 Cuadro de texto para la superficie del predio

### 2.8.3 BOTONES DE COMANDO

Se eligen para ejecutar una acción, los botones de "Aceptar" y "Cancelar" son botones de comando comunes. A menudo se encuentran a lo largo de la parte inferior o a la derecha de un cuadro de diálogo.

### 2.8.4 BOTONES DE OPCIÓN

Los botones de opción representan grupos de opciones mutuamente excluyentes. Solamente puede seleccionar una opción a la vez. La opción seleccionada contiene un punto negro.

Precio Unitario (M2) Manual       Precio Unitario (M2) Automático

Fig. 2.8.4.1 Botones de opción

### 2.8.5 CASILLAS DE VERIFICACIÓN

Una casilla de verificación presenta opciones no excluyentes de las cuales puede seleccionar tantas como necesite. Cuando se selecciona una casilla de verificación, ésta contiene una x.

<input type="checkbox"/> Agua Potable	<input type="checkbox"/> Drenaje	<input type="checkbox"/> Pavimento
<input type="checkbox"/> Energía Eléctrica	<input type="checkbox"/> Estabilidad	<input type="checkbox"/> Red Telefónica
<input type="checkbox"/> Red de Agua	<input type="checkbox"/> Vialidad	<input type="checkbox"/> TV de Cable

Fig. 2.8.5.1 Casillas de verificación

### **3. ASPECTOS GENERALES DEL SISTEMA DE VALUACIÓN DE PREDIOS**

#### **3.1 ESTRUCTURA DEL SISTEMA SAPT**

El Sistema de Valuación de Predios está formado por:

##### **Seguridad y Acceso al Sistema**

El sistema de seguridad, tiene por objeto realizar la administración de los usuarios que tendrán acceso al sistema, la asignación de permisos a los usuarios y los niveles de seguridad para la ejecución de transacciones.

##### **Catálogos**

Los catálogos tienen por objeto, realizar el mantenimiento a la información básica para el funcionamiento del sistema, como es el registro y actualización.

##### **Avalúo de Predios**

Tiene por objeto realizar el cálculo del valor unitario de un predio

##### **Mantenimiento de la Base de Datos**

Tiene por objeto administrar la base de datos y está a cargo de una persona con conocimientos para realizar operaciones tales como compactación de la Base de Datos.

#### **3.2 ACCESO AL SISTEMA**

El sistema, presenta la siguiente pantalla de acceso al sistema:

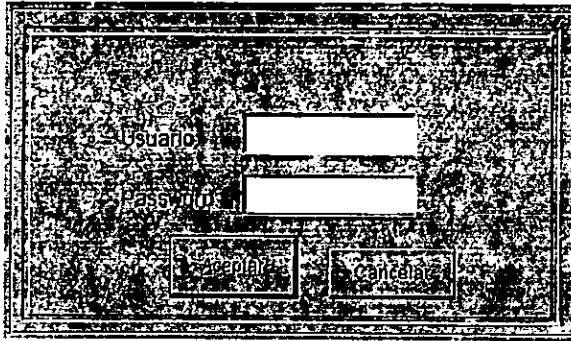


Fig.3.2.1 Ventana de acceso al sistema.

Los datos de entrada, para el ingreso al sistema son:

- Nombre de Usuario (Máximo 60 caracteres).
- Contraseña (Máximo 8 caracteres).

Presionar **ACEPTAR** para ingresar al sistema.

Presionar **CANCELAR** para abandonar el Sistema y regresar a Windows.

Si el usuario o contraseña es incorrecto, está mal escrito o el usuario no tiene acceso al sistema se despliega un mensaje de error.

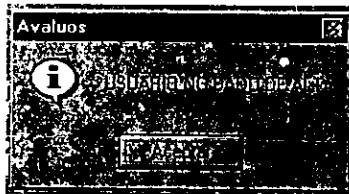


Fig. 3.2.2 Mensaje de Seguridad.

Presionar **Aceptar** y volver a intentar la entrada de los datos hasta que sean correctos para el sistema.

- Si no puede tener acceso al sistema verificar la configuración del mismo para su instalación.

Si el usuario se encuentra registrado en la Base de Datos y sus datos de acceso son correctos, entonces se muestra la siguiente ventana:



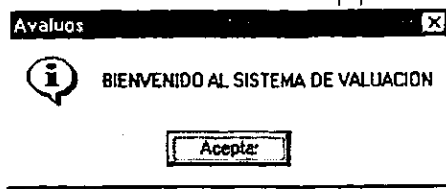











Fig.3.2.3 Mensaje de Bienvenida

### 3.3 Barra de Herramientas

La siguiente tabla muestra los botones contenidos en la barra de herramientas, así como una breve descripción de la operación asociada a cada uno de los botones.

BOTÓN	OPERACIÓN ASOCIADA AL BOTÓN
	Almacena, guarda, salva o da de alta un nuevo elemento al catálogo específico en que se este trabajando.
	Actualiza o modifica los datos de un elemento ya existente en un catálogo específico.
	Elimina o borra de forma permanente un elemento específico de un catálogo.
	Limpia el contenido de los objetos mostrados en el folder del catálogo actual o activo.
	Envía a impresión los datos contenidos en una tabla de un catálogo específico.
	Permite ir al primer elemento en forma directa dentro de una tabla de un catálogo específico.
	Permite ir al elemento anterior dentro de una tabla de un catálogo específico.
	Permite ir al siguiente elemento dentro de una tabla de un catálogo específico.
	Permite ir al último elemento en forma directa dentro de una tabla de un catálogo específico.



Activa la ayuda en línea del submódulo.

### 3.4 CATÁLOGOS

**Solicitante.** En este catálogo se realiza el registro, modificación, eliminación o consulta de los datos generales de los solicitantes, los cuales requieren sea valuado algún bien inmueble (predios para este caso).

**Valuador.** En este catálogo se realiza el registro, modificación, eliminación o consulta de los datos generales de los valuadores, los cuales son asignados para llevar a cabo el avalúo del predio correspondiente.

**Colonia.** En este catálogo se crean, actualizan, eliminan o consultan las colonias que pertenecen a las diferentes delegaciones del Distrito Federal.

**Delegación.** En este catálogo se crean, actualizan o eliminan las delegaciones del Distrito Federal.

**Zona.** En este catálogo se realiza el registro de la clasificación de las zonas en nuestro país dependiendo del desarrollo urbano que se tenga.

**Uso de Suelo.** Este catálogo es muy importante, pues es una de las características más importantes de un predio para su valuación; en este se lleva a cabo la actualización, y consulta de los diferentes usos de suelo que se le puede asignar a un predio determinado.

Los catálogos anteriores se encuentran organizados en folders y para acceder a ellos basta con posicionarse en el folder deseado y dar click sobre este.

### 3.4.1 CATÁLOGO DE SOLICITANTE

**Objetivo:**

En este catálogo se realiza el registro, modificación, eliminación y consulta de los datos generales de los solicitantes existentes.

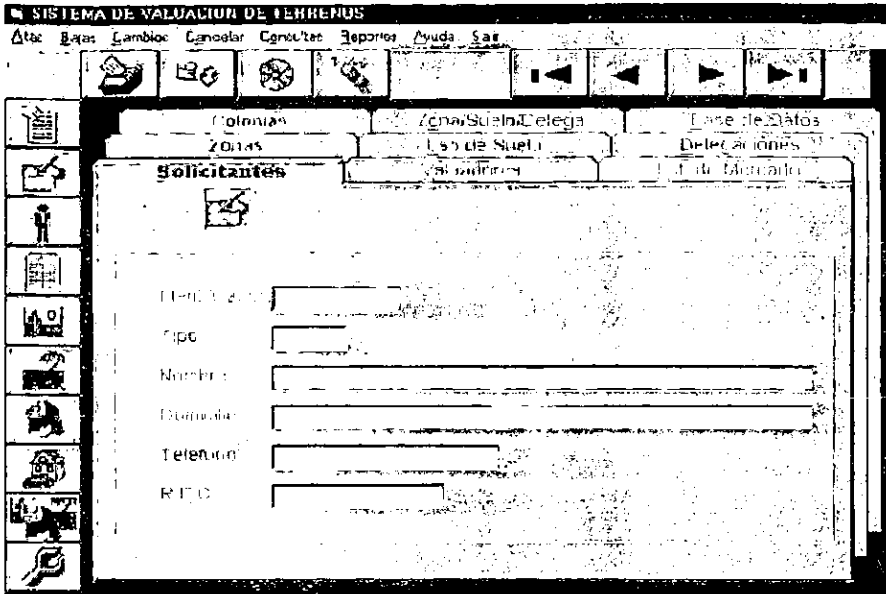


Fig. 3.4.1.1 Folder del catálogo de Solicitante

En este folder se pueden realizar las siguientes operaciones:

- Dar de Alta un nuevo solicitante.
- Actualizar los datos de un solicitante existente.
- Eliminar los datos de un solicitante existente.
- Consultar los solicitantes existentes.
- Ir al primer solicitante registrado.
- Ir al anterior.
- Ir al siguiente.
- Ir al último solicitante registrado.

Datos requeridos para el folder de solicitantes:

- Tipo de solicitante. En este campo se captura el tipo de solicitante que requiere el avalúo, y se tienen dos tipos de solicitantes: internos (I) y externos(E).
- Nombre. En este campo se captura el nombre completo del solicitante.
- Domicilio. Aquí se captura el domicilio del solicitante.
- Teléfono. Se captura el teléfono del solicitante.
- RFC. Se captura el Registro Federal de Causantes del solicitante.

#### **Procedimiento para el registro o alta de un solicitante**

1. Seleccionar de la lista de tipo de solicitante, el tipo de solicitante a dar de alta.
2. Teclear el nombre completo del solicitante.
3. Teclear el domicilio del solicitante.
4. Teclear el teléfono de la oficina en la que se localiza el solicitante.
5. Teclear el RFC del solicitante (no es indispensable).
6. Al proporcionar todos los datos requeridos del solicitante, presionar el botón de guardar con el cual se salvarán los datos proporcionados.

#### **Procedimiento para la actualización o modificación de los datos de un solicitante**

1. Seleccionar el solicitante del que se desea modificar, con los botones de ver primero, último, siguiente o anterior.
2. Modificar los datos deseados.
3. Presionar el botón de modificar, para almacenar los cambios realizados a los datos del solicitante existente.

#### **Procedimiento para la eliminación o borrado de los datos de un solicitante**

1. Seleccionar el solicitante que se busca eliminar con los botones de consulta.
2. Presionar el botón de borrar, para borrar los datos del funcionario, al realizar esto se desplegará un mensaje de advertencia en el que se indica si está seguro de que desea borrar los datos del solicitante seleccionado, si está seguro de borrar el solicitante, confirme afirmativamente el mensaje, en caso contrario confirme negativamente el mensaje.

### **3.4.2 CATÁLOGO DE VALUADORES**

#### **Objetivo:**

En este catálogo se realiza el registro, modificación, eliminación y consulta de los datos generales de los valuadores existentes.

### 3.4.2.1 Folder del catálogo del Valuador

En este folder se pueden realizar las siguientes operaciones:

- Dar de Alta un nuevo valuador.
- Actualizar los datos de un valuador existente.
- Eliminar los datos de un valuador existente.
- Consultar los valuadores existentes.
- Ir al primer valuador registrado.
- Ir al anterior.
- Ir al siguiente.
- Ir al último valuador registrado.

Datos requeridos para el folder de valuadores:

- Tipo de valuador. Se selecciona el tipo de valuador que ha sido asignado, y el tipo puede ser interno (I) o externo (E).
- Nombre. En este campo se captura el nombre completo del valuador.
- Domicilio. Se captura el domicilio del valuador.
- Registro. En este campo se capturan el número de registro obtenido por el valuador asignado.
- Especialidad. En este campo se captura la o las especialidades en avalúos del valuador asignado.
- Eficiencia. Aquí se captura el grado de eficiencia que posee el valuador asignado.
- Observaciones. En este campo se capturan las observaciones que se quieran hacer a cerca del valuador asignado.

#### **Procedimiento para el registro o alta de un valuador**

- 1  Teclar el nombre completo del valuador.
- 2  Seleccionar de la lista de tipo de valuador, el tipo de valuador a dar de alta.
- 3  Teclar el registro del valuador.
- 4  Teclar la eficiencia del valuador.
- 5  Teclar las observaciones a cerca del valuador.

El máximo de las observaciones es de 255 caracteres, en caso de que la nota exceda este máximo se truncará el texto.

- 1  Al proporcionar todos los datos requeridos del valuador, presionar el botón de guardar con el cual se salvarán los datos proporcionados.

**Procedimiento para la actualización o modificación de los datos de un valuador**

1. Seleccionar el valuador del que se desea modificar sus datos con los botones de consulta.
2. Modificar los datos específicos que se requiere sean actualizados o modificados.
3. Presionar el botón de modificar, para almacenar los cambios realizados a los datos del valuador existente.

**Procedimiento para la eliminación o borrado de los datos de un valuador**

1. Seleccionar el valuador que se busca eliminar con los botones de consulta.
2. Presionar el botón de borrar, para borrar los datos del valuador, al realizar esto se desplegará un mensaje de advertencia en el que se indica si está seguro de que desea borrar los datos del valuador seleccionado, si está seguro de borrar el valuador confirme afirmativamente el mensaje, en caso contrario, confirme negativamente el mensaje.

**3.4.3 CATÁLOGO DE DELEGACIONES**

**Objetivo:**

En este catálogo se realiza el registro, modificación, eliminación y consulta de las delegaciones existentes.

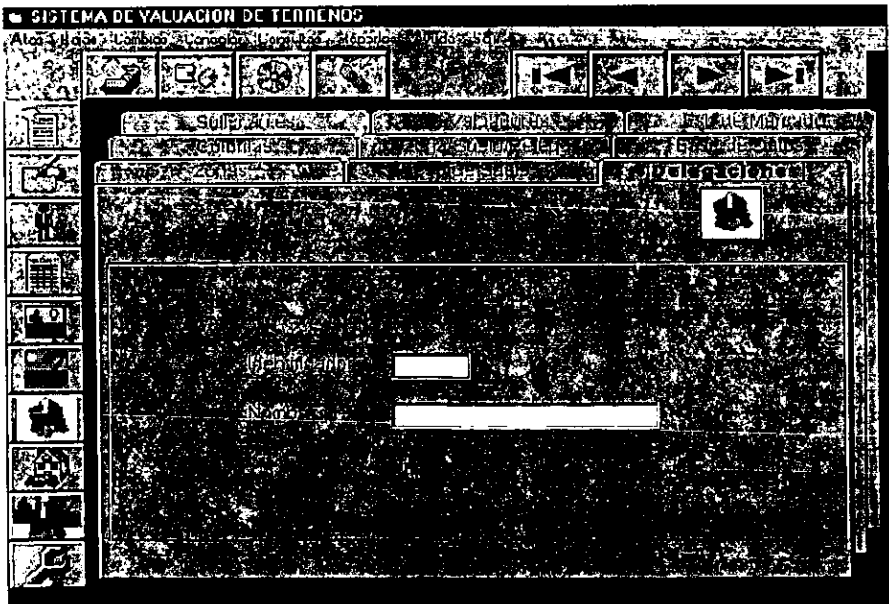


Fig. 3.4.3.1 Folder del catálogo de Delegaciones

En este folder se pueden realizar las siguientes operaciones:

- Dar de Alta una nueva delegación.
- Actualizar los datos de una delegación existente.
- Eliminar los datos de una delegación existente.
- Consultar las delegación existentes.
- Ir a la primera delegación registrada.
- Ir a la anterior.
- Ir a la siguiente.
- Ir a la último delegación registrada.

Datos requeridos para el folder de delegaciones:

- Nombre de la delegación. Se captura el nombre de la delegación existente en el Distrito Federal.

#### **Procedimiento para el registro o alta de una Delegación**

- 1  Teclear el nombre de la delegación.
- 2  Al proporcionar todos los datos requeridos de la delegación, presionar el botón de guardar con el cual se salvarán los datos proporcionados.

#### **Procedimiento para la actualización o modificación de los datos de una Delegación**

1. Seleccionar la delegación de la que se desea modificar sus datos con los botones de consulta.
2. Modificar los datos específicos que se requiere sean actualizados o modificados.
3. Presionar el botón de modificar, para almacenar los cambios realizados a los datos de la delegación existente.

#### **Procedimiento para la eliminación o borrado de los datos de una Delegación**

1. Seleccionar la delegación de la que se desea modificar sus datos con los botones de consulta.
2. Presionar el botón de borrar, para borrar los datos de la delegación, al realizar esto se desplegará un mensaje de advertencia en el que se indica si está seguro de que desea borrar los datos de la delegación seleccionada, si está seguro de borrarla confirme afirmativamente el mensaje, en caso contrario, confirme negativamente el mensaje.

### 3.4.4 CATÁLOGO DE COLONIAS

#### Objetivo:

En este catálogo se realiza el registro, modificación, eliminación y consulta de las colonias existentes.

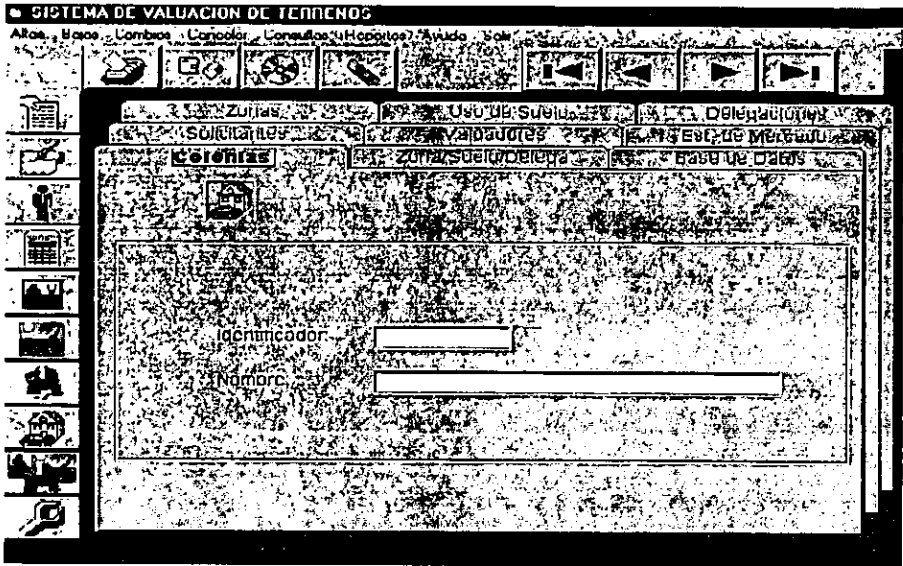


Fig. 3.4.4.1 Folder del catálogo de Colonias

En este folder se pueden realizar las siguientes operaciones:

- Dar de Alta una nueva colonia.
- Actualizar los datos de una colonia existente.
- Eliminar los datos de una colonia existente.
- Consultar las colonias existentes.
- Ir a la primera colonia registrada.
- Ir a la anterior.
- Ir a la siguiente.
- Ir a la última colonia registrada.

Datos requeridos para el folder de colonias:

- Delegación a la que pertenece la colonia.
- Nombre de la colonia.



#### **Procedimiento para el registro o alta de una Colonia**

- 1  Seleccionar de la lista de delegaciones, el nombre de la delegación a la que pertenece la colonia.
- 2  Teclear el nombre de la colonia.
- 3  Al proporcionar todos los datos requeridos de la delegación, presionar el botón de guardar, con el cual se salvarán los datos proporcionados.

#### **Procedimiento para la actualización o modificación de los datos de una Colonia**

- 1  Seleccionar la delegación de la lista de delegaciones, la delegación a la que pertenece la colonia que se desea modificar.
- 2  Seleccionar la colonia de la lista de colonias, la colonia que desea modificar.
- 3  Modificar los datos específicos que se requiere sean actualizados o modificados.
- 4  Presionar el botón de modificar, para almacenar los cambios realizados a los datos de la colonia existente.

#### **Procedimiento para la eliminación o borrado de los datos de una Colonia**

- 1  Seleccionar la delegación de la lista de delegaciones, la delegación a la que pertenece la colonia que se desea eliminar.
- 2  Seleccionar la colonia de la lista de colonias, la colonia que desea eliminar.
- 3  Presionar el botón de borrar, para borrar los datos de la colonia, al realizar esto se desplegará un mensaje de advertencia en el que se indica si está seguro de que desea borrar los datos de la colonia seleccionada, si está seguro de borrarla confirme afirmativamente el mensaje, en caso contrario, confirme negativamente el mensaje.

### 3.4.5 CATÁLOGO DE ZONAS

#### Objetivo:

En este catálogo se realiza el registro, modificación, eliminación y consulta de las zona existentes.



Fig. 3.4.5.1 Folder del catálogo de Zonas

En este folder se pueden realizar las siguientes operaciones:

- Dar de Alta una nueva zona.
- Actualizar los datos de una zona existente.
- Eliminar los datos de una zona existente.
- Consultar las zonas existentes.
- Ir a la primera zona registrada.
- Ir a la anterior.
- Ir a la siguiente.
- Ir a la última zona registrada.

Datos requeridos para el folder de zonas:

- Nombre de la zona. Aquí se captura el nombre o descripción de la zona.

#### **Procedimiento para el registro o alta de una Zona**

- 1  Teclear el nombre de la zona.
- 2  Al proporcionar todos los datos requeridos de la zona, presionar el botón de guardar con el cual se salvarán los datos proporcionados.

#### **Procedimiento para la actualización o modificación de los datos de una Zona**

- 1  Seleccionar la zona de la que se desea modificar sus datos con los botones de consulta.
- 2  Modificar los datos específicos que se requiere sean actualizados o modificados.
- 3  Presionar el botón de modificar, para almacenar los cambios realizados a los datos de la zona existente.

#### **Procedimiento para la eliminación o borrado de los datos de una Colonia**

1. Seleccionar la zona de la que se desea modificar sus datos con los botones de consulta.
2. Presionar el botón de borrar, para borrar los datos de la zona, al realizar esto se desplegará un mensaje de advertencia en el que se indica si está seguro de que desea borrar los datos de la zona seleccionada, si está seguro de borrarla confirme afirmativamente el mensaje, en caso contrario, confirme negativamente el mensaje.

### 3.4.6 CATÁLOGO DE USO DE SUELO

**Objetivo:**

En este catálogo se realiza el registro, modificación, eliminación y consulta de los usos de suelo.

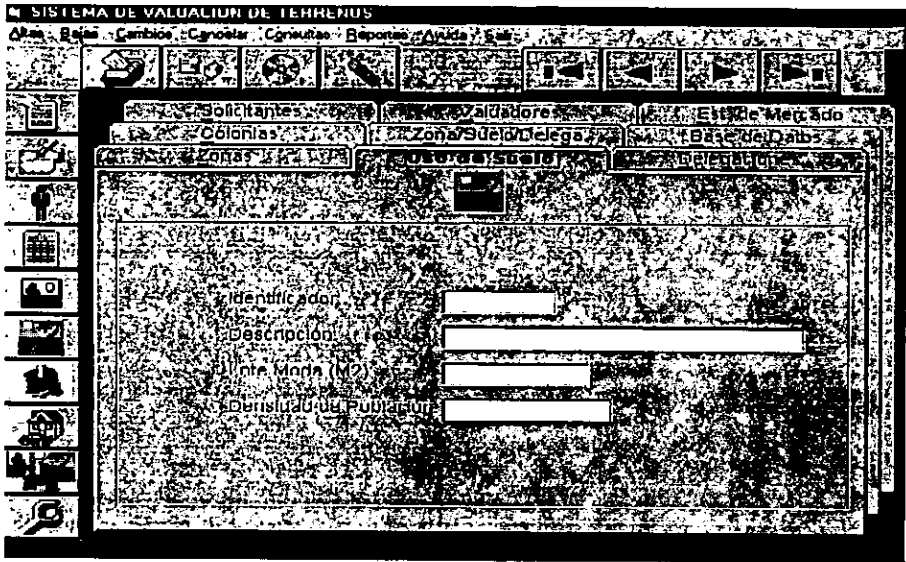


Fig. 3.4.6.1 Folder del catálogo de Uso de Suelo

En este folder se pueden realizar las siguientes operaciones:

- Dar de Alta un nuevo tipo de uso de suelo.
- Actualizar los datos de un tipo de uso de suelo existente.
- Eliminar los datos de un tipo de uso de suelo existente.
- Consultar los tipos de uso de suelo existentes.
- Ir al primer tipo de uso de suelo registrado.
- Ir al anterior.
- Ir al siguiente.
- Ir al último tipo de uso de suelo registrado.

Datos requeridos para el folder de tipos de uso de suelo:

- Clave del tipo de uso de suelo. En este campo se captura la clave del uso de suelo ( H5, H1, etc.).
- Descripción del tipo de uso de suelo. Aquí se captura la descripción de la clave que se capturó anteriormente y que nos dice cual es el uso de suelo (habitacional, industrial, etc.).

- El lote moda. En este campo se captura la cantidad en metros cuadrados promedio del lote representativo para el uso de suelo que se está dando de alta.
- Densidad de población. Aquí se captura el grado de concentración de población para el uso de suelo que se está dando de alta.

#### **Procedimiento para el registro o alta de un Uso de Suelo**

- 1  Teclear la clave o identificador del tipo de uso de suelo.
- 2  Teclear la descripción del tipo de uso de suelo.
- 3  Teclear el lote moda.
- 4  Teclear la densidad de población.
- 5  Al proporcionar todos los datos requeridos del tipo de uso de suelo, presionar el botón de guardar con el cual se salvarán los datos proporcionados.

#### **Procedimiento para la actualización o modificación de los datos del tipo de uso de suelo**

1. Seleccionar el tipo de uso de suelo del que se desea modificar sus datos con los botones de consulta.
2. Modificar los datos específicos que se requiere sean actualizados o modificados.
3. Presionar el botón de modificar, para almacenar los cambios realizados a los datos del tipo de uso de suelo existente.

#### **Procedimiento para la eliminación o borrado de los datos de un tipo de uso de suelo**

1. Seleccionar el tipo de uso de suelo del que se desea modificar sus datos con los botones de consulta.
2. Presionar el botón de borrar, para borrar los datos del tipo de uso de suelo, al realizar esto se desplegará un mensaje de advertencia en el que se indica si está seguro de que desea borrar los datos del tipo de uso de suelo seleccionado, si está seguro de borrar el tipo de uso de suelo confirme afirmativamente el mensaje, en caso contrario, confirme negativamente el mensaje.

### 3.5 ZONA- USO DE SUELO

#### Objetivo:

En este folder se definen las características de la zona a la que pertenece el predio que se está evaluando. Las características están relacionadas con el tipo de zona según el grado de urbanización que se tenga, la delegación a la que pertenece y el tipo de suelo que se tiene. Tales características son el tipo de zona, el uso de suelo, la intensidad de construcción, los servicios públicos, contaminación, etc.

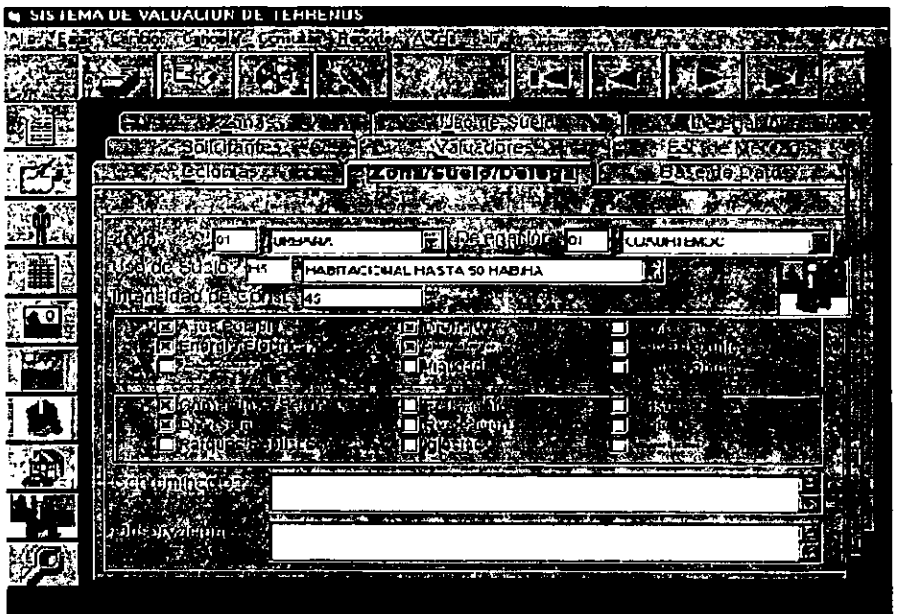


Fig. 3.5.1 Folder de las subzonas.

En este folder se pueden realizar las siguientes operaciones:

- Actualizar las características de una zona.

Datos requeridos para el folder de características de zona-uso de suelo-delegación:

- Delegación a la que pertenece la zona.
- Clasificación de la zona según su grado de urbanización.
- El uso de suelo que posee el predio.
- La intensidad de construcción de la zona a la que pertenece el predio.
- Los servicios públicos con que cuenta la zona a la que pertenece el predio.
- El grado de contaminación de la zona a la que pertenece el predio.
- Observaciones que se quieran hacer a cerca de las características de la zona.

#### **Procedimiento para el registro de las características de la zona**

- 1  Seleccionar la clasificación de la zona de la lista de zonas.
- 2  Seleccionar la delegación de la lista de delegaciones.
- 3  Seleccionar de la lista de tipo de uso de suelo, el tipo de suelo para esta zona.
- 4  Seleccionar de la lista de intensidades de construcción, la intensidad de construcción para esta subzona.
- 5  Marcar en las casillas de verificación, los servicios públicos con que cuente la subzona.
- 6  Teclar los tipos principales de contaminación que existan en la subzona.
- 7  Teclar las observaciones a cerca de la subzona.

El máximo de las observaciones es de 255 caracteres, en caso de que la nota exceda este máximo se truncará el texto.

Al seleccionar la clasificación de la zona, el uso de suelo y la intensidad de construcción aparece junto a cada dato un valor que es el que el sistema le asigna automáticamente, estos valores son tomados por el sistema para el cálculo de precio unitario del predio que pertenece a esta subzona.

### 3.6 ESTUDIO DE MERCADO

#### Objetivo:

En este folder se realiza el registro, modificación, eliminación y consulta de la información que se obtiene a través del estudio de mercado que se realiza. Esta información sirve como referencia para la obtención del precio unitario del predio en cuestión.

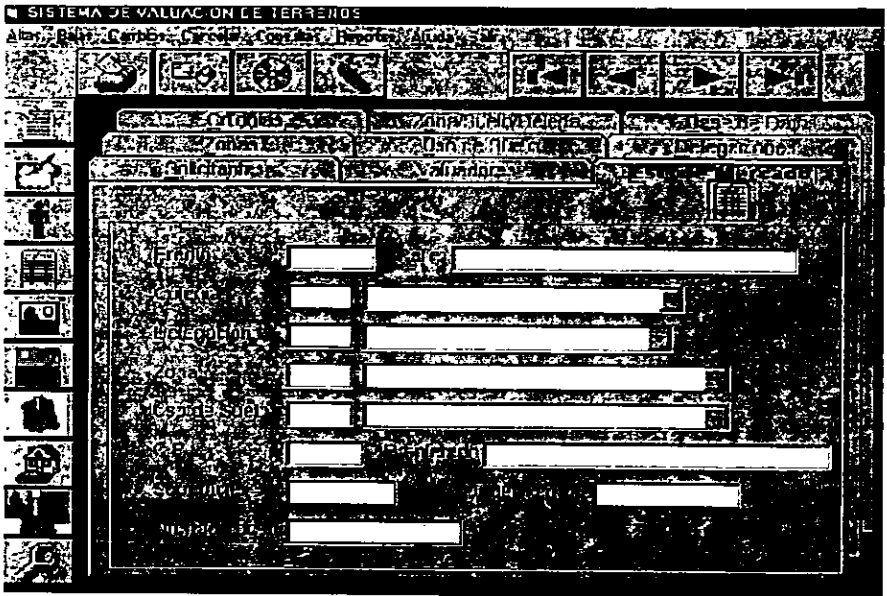


Fig. 3.6.1 Folder del Estudio de Mercado

En este folder se pueden realizar las siguientes operaciones:

- Dar de Alta un nuevo estudio de mercado.
- Actualizar los datos de un estudio de mercado existente.
- Eliminar los datos de un estudio de mercado existente.
- Consultar los estudios de mercado existentes.
- Ir al primer estudio de mercado registrado.
- Ir al anterior.



- Ir al siguiente.
- Ir al último estudio de mercado registrado.

Datos requeridos para el folder de estudio de mercado:

- Clave o identificador del predio al que le corresponde dicho estudio de mercado.
- Calle en que se ubica el predio al que le corresponde dicho estudio de mercado.
- Colonia en que se ubica el predio al que le corresponde dicho estudio de mercado.
- Delegación a la que pertenece el predio al que le corresponde dicho estudio de mercado.
- Código Postal.
- Clasificación del a zona según su grado de urbanización a la que pertenece el predio al que le corresponde dicho estudio de mercado.
- El uso de suelo que posee el predio al que le corresponde dicho estudio de mercado.
- Nombre del propietario del predio al que le corresponde dicho estudio de mercado.
- La superficie del predio el predio al que le corresponde dicho estudio de mercado.
- El valor del precio unitario en que se está vendiendo el predio al que le corresponde dicho estudio de mercado.
- La fecha desde que se ha pretendido vender el predio al que le corresponde dicho estudio de mercado.

#### Procedimiento para el registro de la información de estudio de mercado

- 1  Teclear la clave o identificador del predio al que le corresponde dicho estudio de mercado.
- 2  Teclear calle en que se ubica el predio al que le corresponde dicho estudio de mercado.
- 3  Seleccionar la colonia de la lista de delegaciones, la delegación en que se ubica el predio al que le corresponde dicho estudio de mercado.
- 4  Seleccionar de la lista de colonias, la colonia a la que pertenece el predio al que le corresponde dicho estudio de mercado.
- 5  Teclear el código postal.
- 6  Seleccionar de la lista de zonas la clasificación del a zona según su grado de urbanización a la que pertenece el predio al que le corresponde dicho estudio de mercado.
- 7  Seleccionar de la lista de tipos de usos de suelo, el uso de suelo que posee el predio al que le corresponde dicho estudio de mercado.
- 8  Teclear el nombre del propietario del predio al que le corresponde dicho estudio de mercado.
- 9  Capturar la superficie del predio el predio al que le corresponde dicho estudio de mercado.
- 10  Capturar el valor del precio unitario en que se está vendiendo el predio al que le corresponde dicho estudio de mercado.
- 11  Capturar la fecha desde que se ha pretendido vender el predio al que le corresponde dicho estudio de mercado.

12]Al proporcionar todos los datos del estudio de mercado, presionar el botón de guardar con el cual se salvarán los datos proporcionados.

**Procedimiento para la actualización o modificación de la información de estudio de mercado**

1. Seleccionar la clave del predio que tiene ligado el estudio de mercado del que se desea modificar sus datos con los botones de consulta.
2. Modificar los datos específicos que se requiere sean actualizados o modificados.
3. Presionar el botón de modificar, para almacenar los cambios realizados a los datos del estudio de mercado existente.

**Procedimiento para la eliminación o borrado de los datos de un tipo de uso de suelo**

1. Seleccionar el estudio de mercado del que se desea borrar sus datos con los botones de consulta.
2. Presionar el botón de borrar, para borrar los datos del tipo del estudio de mercado, al realizar esto se desplegará un mensaje de advertencia en el que se indica si está seguro de que desea borrar los datos del estudio de mercado seleccionado, si está seguro de borrarlo, confirme afirmativamente el mensaje en caso contrario, confirme negativamente el mensaje.

### 3.7 VALUACIÓN DE PREDIOS

**Objetivo:**

En la primer pantalla, se lleva a cabo el registro de la información general del predio(propietario, colindancias del predio, nombre del solicitante, valuador asignado).

The image shows a screenshot of a software interface titled "AVALUO". The interface contains a list of labels on the left side, each followed by a corresponding input field. The labels are: "No.", "Asunto", "Predio", "Valuador", "Objeto", "Calle", "Delet.", "Uso de Suelo", "Propietario", "Objeto", "Escritura", "Calle", "Calle", and "Colindancias". The input fields are mostly empty, with some containing faint text. At the bottom of the form, there are two buttons labeled "SIGUIENTE" and "PREVIA".

Fig. 3.7.1 Primera pantalla de avalúos.

En la segunda pantalla, es donde se lleva a cabo el cálculo del valor unitario del predio, por lo que se captura la información necesaria para calcular los factores que influyen en el valor del predio.

The screenshot shows a software interface for property valuation. The title bar reads 'AVAUO'. The interface includes several input fields and buttons. At the top, there are fields for 'CANTIDAD' and 'VALOR'. Below that, there are fields for 'CANTIDAD' and 'VALOR' with a '1.00' value. The main area contains several rows of input fields, some with labels like 'CANTIDAD' and 'VALOR'. At the bottom, there are buttons for 'CANTIDAD', 'VALOR', and 'REGISTRAR'.

### 3.7.1 Segunda pantalla de avalúos.

Aquí podemos realizar las siguientes operaciones:

- Dar de Alta un nuevo avalúo.
- Actualizar los datos de un avalúo existente.
- Eliminar los datos de un avalúo existente.
- Consultar los avalúos existentes.
- Ir al primer avalúo registrado.
- Ir al anterior.
- Ir al siguiente.
- Ir al último avalúo registrado.

Datos requeridos para el avalúo :

- Objeto del avalúo.
- Clave o identificador del predio al que le corresponde dicho estudio de mercado.
- Valuador asignado para ese avalúo.
- Fecha en que fue hecha la solicitud del avalúo.
- Número de oficio.
- Fecha de entrega del avalúo.
- Calle en que se ubica el predio a valuar.
- Colonia en que se ubica el predio a valuar.
- Delegación a la que pertenece el predio.
- Código Postal.
- Clasificación de la zona según su grado de urbanización a la que pertenece el predio al que le corresponde dicho estudio de mercado.
- El uso de suelo que posee el predio al que le corresponde dicho estudio de mercado.
- Nombre del propietario del predio.
- Escritura del predio.
- Boleta predial del terreno.
- Línderos y colindancias del predio.
- Calles entre las que se encuentra el predio (calle 1,calle2, calle3,calle 4).
- La superficie del predio.
- Observaciones.
- Frente del predio.
- Pendiente del terreno.
- Fondo del terreno.
- Superficie de la porción anterior.
- Superficie de la porción posterior.
- Frente del rectángulo.
- Fondo del rectángulo.
- Superficie de las áreas irregulares con frente a las vías de acceso.
- Superficie de las áreas irregulares interiores.

**Procedimiento para la realización de un avalúo**

- 1  Capturar el Objeto del avalúo. Se captura la finalidad del avalúo.
- 2  Capturar la Clave o identificador del predio al que le corresponda dicho estudio de mercado.
- 3  Seleccionar al Valuador asignado para ese avalúo.
- 4  Capturar la Fecha en que fue hecha la solicitud del avalúo.
- 5  Capturar el Número de oficio.
- 6  Capturar la Fecha de entrega del avalúo.
- 7  Capturar la Calle en que se ubica el predio a valuar.
- 8  Seleccionar la Colonia en que se ubica el predio a valuar.
- 9  Seleccionar la Delegación a la que pertenece el predio.
- 10  Capturar el Código Postal.

- 11  Seleccionar la Clasificación de la zona según su grado de urbanización a la que pertenece el predio al que le corresponde dicho estudio de mercado.
- 12  Seleccionar el uso de suelo que posee el predio al que le corresponde dicho avalúo.
- 13  Capturar el nombre del propietario del predio.
- 14  Capturar el número de escritura del predio.
- 15  Capturar el número de Boleta predial del terreno.
- 16  Capturar los linderos y colindancias del predio.
- 17  Capturar las Calles entre las que se encuentra el predio (calle 1, calle2, calle3, calle 4).
- 18  Capturar la superficie del predio.
- 19  Capturar las Observaciones.
- 20  Capturar la fecha de realización del avalúo.
- 21  Seleccionar el factor de ubicación de la lista de factores de ubicación.
- 22  Capturar el Frente del predio.
- 23  Capturar la Pendiente del terreno.
- 24  Capturar el Fondo del terreno.
- 25  Capturar la Superficie de la porción anterior.
- 26  Capturar la Superficie de la porción posterior.
- 27  Capturar el Frente del rectángulo.
- 28  Capturar el Fondo del rectángulo.
- 29  Capturar la Superficie de las áreas irregulares con frente a las vías de acceso.
- 30  Capturar la Superficie de las áreas irregulares interiores.

Al seleccionar la ubicación del predio, el sistema proporciona automáticamente el factor de ubicación, dependiendo de los frentes que el predio posea con respecto a las vías de circulación.

Al seleccionar la zona del predio, el sistema proporciona automáticamente el factor de zona, dependiendo de los frentes que el predio posea con respecto a la calle moda de la zona.

Al capturar el frente del terreno del predio, el sistema proporciona automáticamente el factor de frente, dependiendo de la longitud del frente que el predio posea.

Al capturar la pendiente del predio, el sistema proporciona automáticamente el factor de topografía, dependiendo de que tan pronunciada sea la pendiente que el predio posea.

Al capturar la superficie del predio, el sistema proporciona automáticamente el factor de superficie, dependiendo del área del predio.

Una vez calculados estos factores, el sistema calcula el factor resultante, tomando en cuenta todos los factores calculados anteriormente.

Por último calcula el precio unitario del predio.

Para calcular el precio total del terreno se toma en cuenta el precio unitario y la superficie del predio, además del factor resultante. Esto se hace de forma automática, pero también es posible ingresar de forma manual el precio del terreno de forma manual, seleccionando la opción de Precio Unitario manual.

Al llevar a cabo el cálculo del avalúo para guardar dicho avalúo, se debe oprimir el botón de almacenar.

#### **Procedimiento para la actualización o modificación de un avalúo**

- 1  Seleccionar la clave del avalúo del que se desea modificar sus datos con los botones de consulta.
- 2  Modificar los datos específicos que se requiere sean actualizados o modificados.
- 3  Presionar el botón de modificar, para almacenar los cambios realizados a los datos del estudio de mercado existente.

#### **Procedimiento para la eliminación o borrado de un avalúo**

1. Seleccionar el avalúo que se desea borrar sus datos con los botones de consulta.
2. Presionar el botón de borrar, para borrar los datos del avalúo, al realizar esto se desplegará un mensaje de advertencia en el que se indica si está seguro de que desea borrar los datos del avalúo seleccionado, si está seguro de borrarlo, confirme afirmativamente el mensaje, en caso contrario confirme negativamente el mensaje.

### **3.8 REPORTES**

Para poder emitir un reporte lo hacemos dando click en el menú del sistema en la opción de reportes y seleccionando el que queramos enviar.

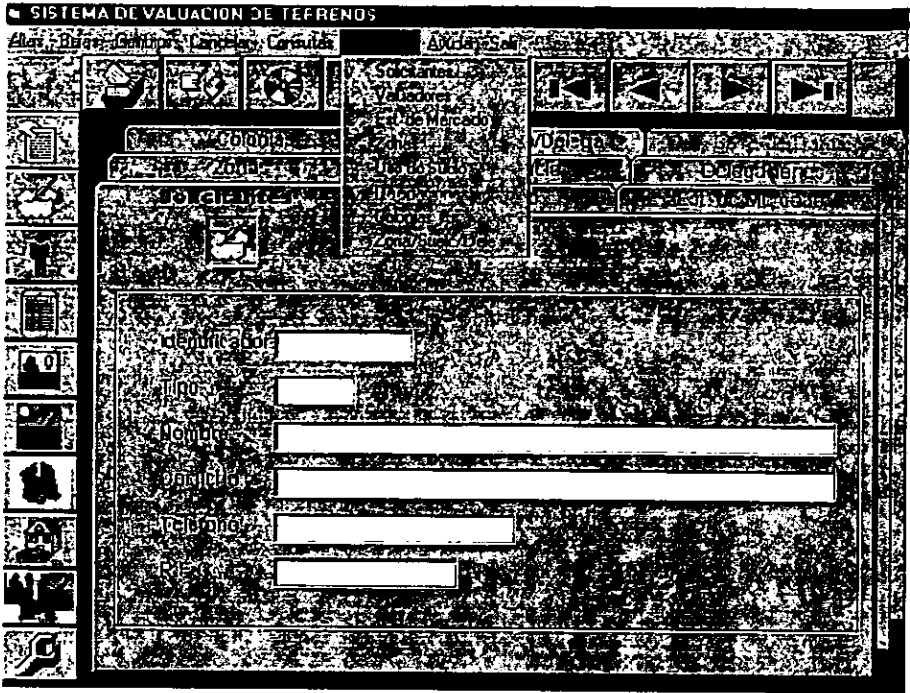


Fig. 3.8.1 Menú de Reportes.

Una vez seleccionado el reporte que queremos, damos click:



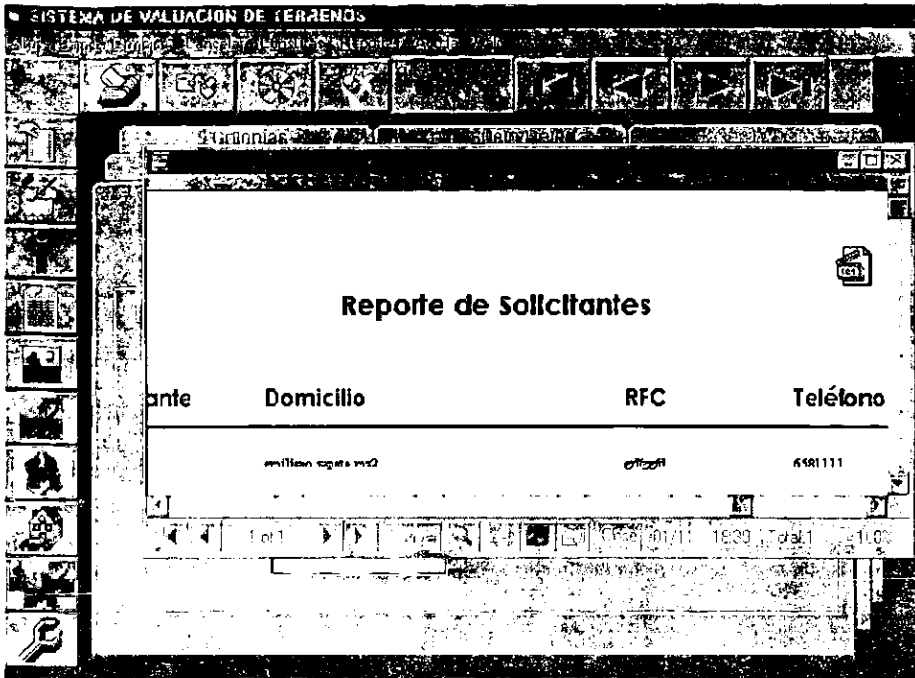


Fig. 3.8.2 Emisión de un reporte.

### 3.9 AYUDA DEL SISTEMA

Para acceder la ayuda del sistema basta con posicionarse en el control que hace referencia al concepto del que se desea saber y presionar F1 con lo que se mostrará la pantalla de la ayuda.

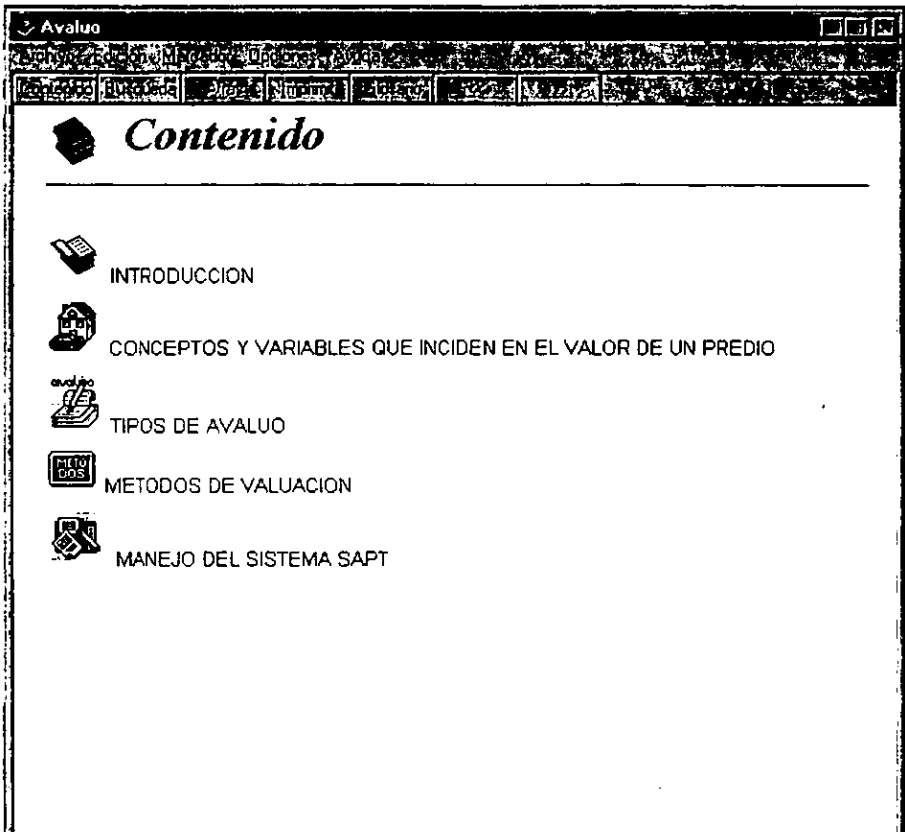


Fig. 3.9.1 Ayuda del Sistema.

### 3.10 MANTENIMIENTO DE LA BASE DE DATOS

#### 3.10.1 Usuarios

**Objetivo:**

En este folder se realiza el registro, modificación, eliminación y consulta de los usuarios que tendrán acceso al sistema, así como la asignación de su nivel de acceso, pues de ello dependen las opciones del sistema que podrán utilizar.

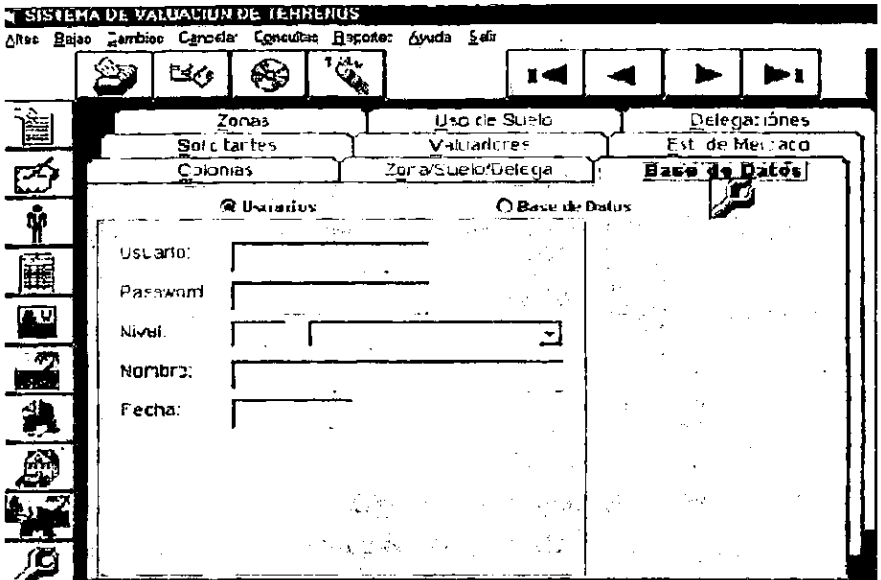


Fig. 3.10.1.1 Mantenimiento a usuarios.

Aquí se pueden realizar las siguientes operaciones:

- Dar de Alta un nuevo usuario.
- Actualizar los datos de un usuario existente.
- Eliminar un usuario existente.

Los datos del usuario que se registran en la base de datos son:

- Clave del usuario.
- Nombre del usuario.
- Nivel de acceso del usuario.
- Password.
- Fecha en que se registró en la base de datos.

### 3.10.2 Base de Datos

**Objetivo:**

Aquí la persona encargada de administrar la base de datos le dará mantenimiento.

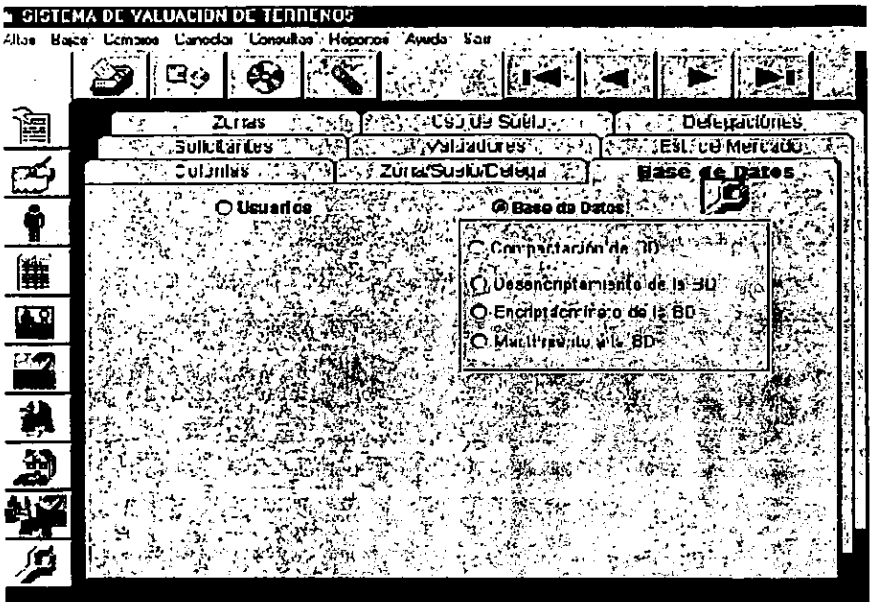


Fig. 3.10.2.1 Mantenimiento a la Base de Datos

Aquí se pueden realizar las siguientes operaciones:

- Compactar la BD con el fin de reducir el espacio que ocupa.
- Encriptar la BD con el fin de proteger la BD.
- Desencriptarla para volverla a su estado normal y poder consultarla.

## APÉNDICE B

### Manual Técnico

#### Objetivo del Manual

El presente manual tiene como función dar soporte al administrador del sistema, para la adecuada operación del sistema SAPT; tomando en cuenta las consideraciones técnicas del sistema.

#### Descripción

En este manual se encuentra toda la información técnica detallada que se requiere para tener una mejor comprensión del sistema SAPT, en términos de cómo esta construido el sistema. Además se contemplan los aspectos necesarios para poder ejecutar el sistema. Finalmente se cuenta con una lista de los programas y rutinas que conforman y hacen posible el funcionamiento del sistema SAPT.

#### Ambiente de Operación

Para la instalación y operación del sistema SAPT se debe contar con el ambiente computacional mencionado a continuación:

- Disponer de un equipo de computo con un Microprocesador 80486 o superior.
- Un disco duro con un espacio mínimo de 8 Mb para poder realizar la instalación del sistema.
- Una unidad de disquete de 3 ½" de alta densidad.
- Un ratón.
- Una tarjeta de vídeo soportada por Windows 3.1.
- 4 Mb de memoria o más. Con Windows 95 se requieren 8 Mb mínimo.

- Windows 3.1 (en modo extendido) o superior

Aparte de los requisitos formales es mejor trabajar con la mayor cantidad de memoria posible y un microprocesador con la mayor velocidad posible.

También es recomendable que el administrador del sistema tenga instalada la versión 4.0 de Microsoft Visual Basic y Microsoft Access 7.0 para que en caso de encontrarse con alguna falla, ó desee modificar y/o actualizar alguna propiedad del sistema, tenga bases suficientes para poder hacerlo.

### **Infraestructura de Desarrollo**

La base del desarrollo del Sistema SAPT está soportada en dos herramientas que son: Visual Basic y Access. Estas dos herramientas conforman la infraestructura que hace posible la construcción del Sistema.

En lo que se refiere al desarrollo del front-end gráfico se tiene que hacer uso de Visual Basic. Para esto es necesario que el administrador del sistema ejecute Visual Basic, como se muestra en la siguiente figura.

## APÉNDICE B

### Manual Técnico

#### Objetivo del Manual

El presente manual tiene como función dar soporte al administrador del sistema, para la adecuada operación del sistema SAPT; tomando en cuenta las consideraciones técnicas del sistema.

#### Descripción

En este manual se encuentra toda la información técnica detallada que se requiere para tener una mejor comprensión del sistema SAPT, en términos de cómo esta construido el sistema. Además se contemplan los aspectos necesarios para poder ejecutar el sistema. Finalmente se cuenta con una lista de los programas y rutinas que conforman y hacen posible el funcionamiento del sistema SAPT.

#### Ambiente de Operación

Para la instalación y operación del sistema SAPT se debe contar con el ambiente computacional mencionado a continuación:

- Disponer de un equipo de computo con un Microprocesador 80486 o superior.
- Un disco duro con un espacio mínimo de 8 Mb para poder realizar la instalación del sistema.
- Una unidad de disquete de 3 ½" de alta densidad.
- Un ratón.
- Una tarjeta de vídeo soportada por Windows 3.1.
- 4 Mb de memoria o más. Con Windows 95 se requieren 8 Mb mínimo.

- Windows 3.1 (en modo extendido) o superior

Aparte de los requisitos formales es mejor trabajar con la mayor cantidad de memoria posible y un microprocesador con la mayor velocidad posible.

También es recomendable que el administrador del sistema tenga instalada la versión 4.0 de Microsoft Visual Basic y Microsoft Access 7.0 para que en caso de encontrarse con alguna falla, ó desee modificar y/o actualizar alguna propiedad del sistema, tenga bases suficientes para poder hacerlo.

### **Infraestructura de Desarrollo**

La base del desarrollo del Sistema SAPT está soportada en dos herramientas que son: Visual Basic y Access. Estas dos herramientas conforman la infraestructura que hace posible la construcción del Sistema.

En lo que se refiere al desarrollo del front-end gráfico se tiene que hacer uso de Visual Basic. Para esto es necesario que el administrador del sistema ejecute Visual Basic, como se muestra en la siguiente figura.



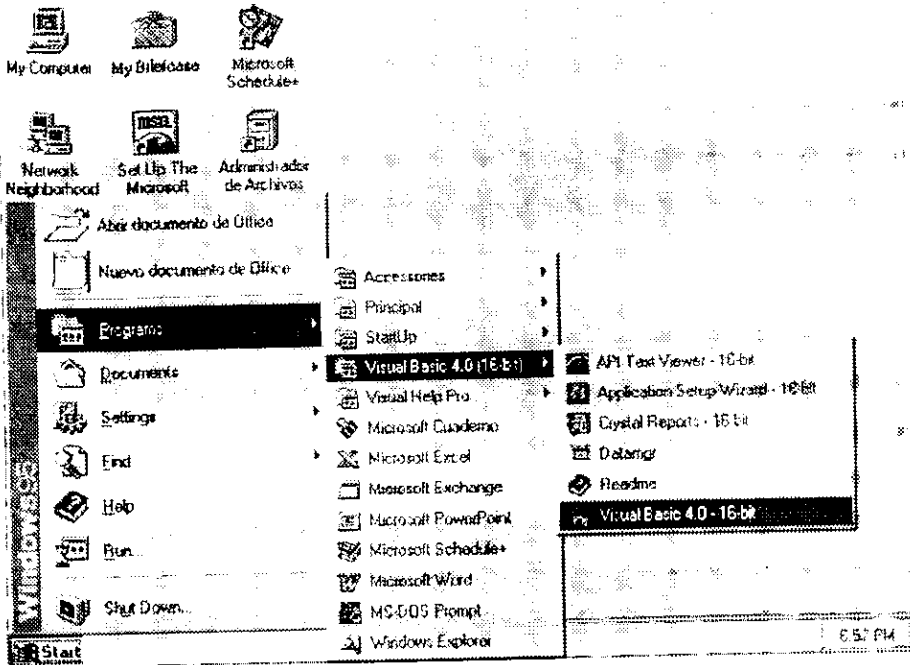


Figura. I. Ejecución de Visual Basic ( Windows 95).

Una vez que se ejecuto Visual Basic hay que seleccionar el proyecto como se muestra en la siguiente figura.

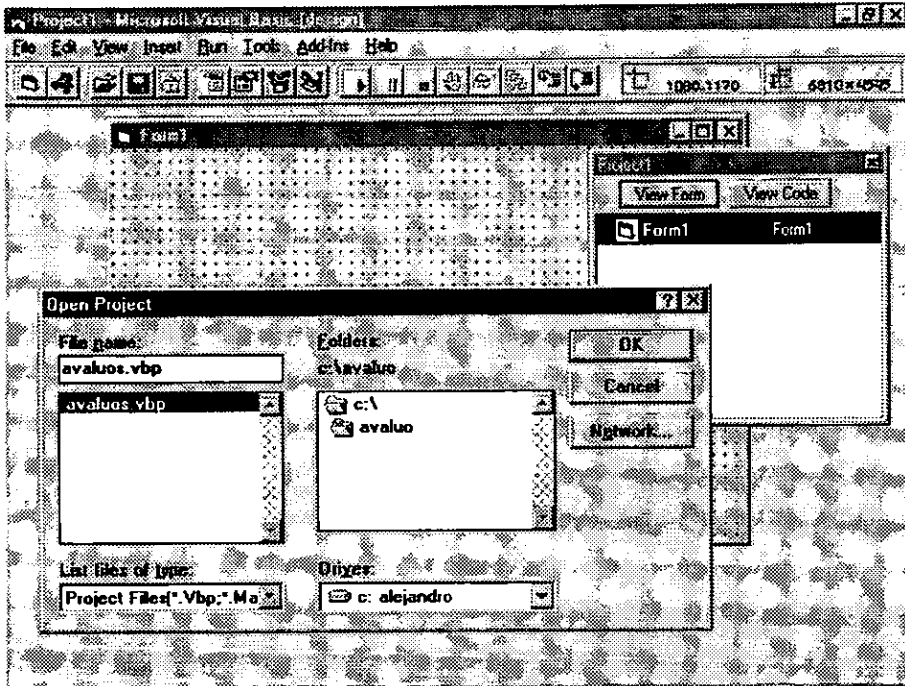


Figura II. Selección del proyecto que se va a cargar

Cuando ya se selecciono el proyecto y ya se cargo, entonces se procede a seleccionar de la lista de formas que utiliza el sistema, la forma a la cual se le desea realizar la modificación y/o actualización. Al realizar esta selección se puede escoger que el sistema muestre el código de la forma que deseamos seleccionar o que muestre el diseño de la misma, para ver como es que está constituida y cuales son los objetos que la conforman.

Si se selecciona la opción de ver la forma y se desean observar las propiedades de alguno de los objetos que la conforman, lo que se tiene que realizar es seleccionar el objeto y presionar la tecla F4 para que muestre sus propiedades.

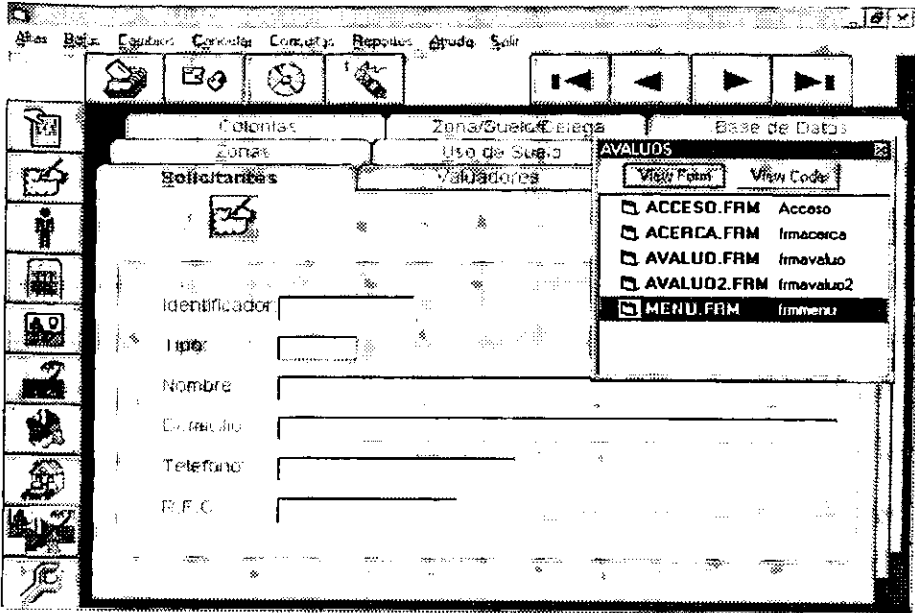


Figura III. Selección de la forma que se desea analizar y/o modificar.

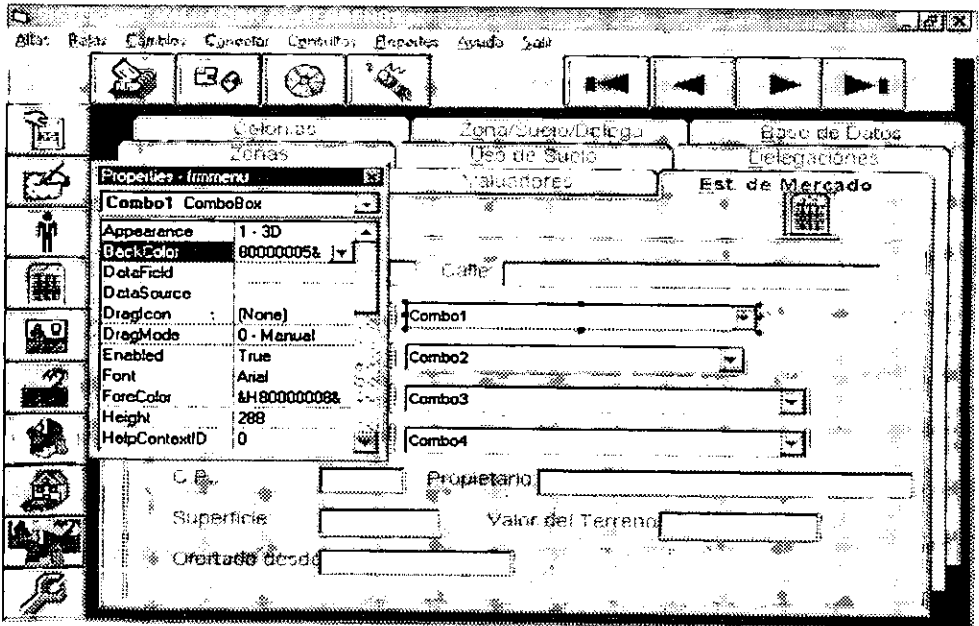


Figura IV. Propiedades de un objeto seleccionado.

Si se desea observar el código que tiene el objeto que se selecciono en alguno de sus eventos, lo que se realiza es presionar dos veces el botón derecho del Mouse sobre el objeto y aparecerá una pantalla que muestra el código del objeto en alguno de sus eventos. Además de que en esta pantalla se puede seleccionar otro objeto ya que también se encuentra la lista de todos los objetos que están contenidos en la forma que se esta analizando y todos sus eventos. Aquí es posible modificar el código del objeto para solucionar el problema y posteriormente volver a ejecutar el sistema y si observamos que ya esta funcionando correctamente procedemos a salvar el proyecto, para poder realizar nuevamente el archivo ejecutable.

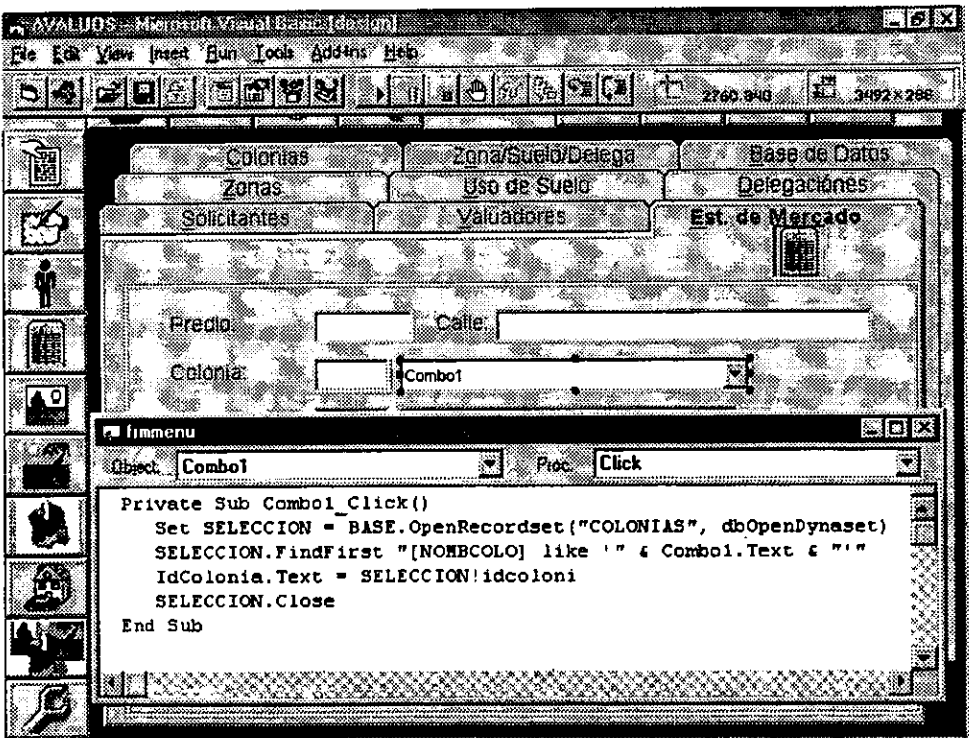


Figura V Código de un objeto

Con lo que respecta a la base de datos se puede utilizar Microsoft Access 7.0 ó se puede utilizar el manejador de datos que tiene Visual Basic. Para poder acceder la base de datos se tiene que seleccionar la opción Data Manager de la opción Add-Ins del menú de opciones de Visual Basic, como se muestra en la siguientes figura.

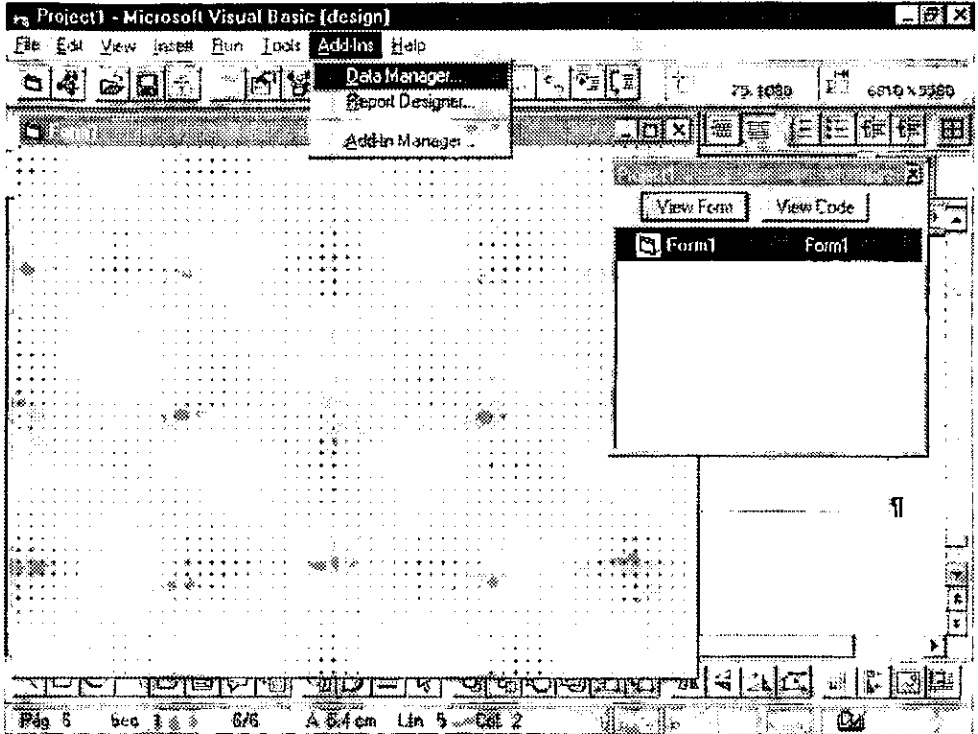


Figura VI. Manejador de datos de Visual Basic

Al realizar la selección de esta opción se ejecuta el manejador de datos de Visual Basic y traerá consigo una nueva pantalla en la cual se tiene que realizar la selección de la base de datos que se desea abrir o bien crear una nueva base de datos, podemos compactar una base de datos, encriptar o decriptar una base de datos existente o reparar una base de datos que se haya dañado.

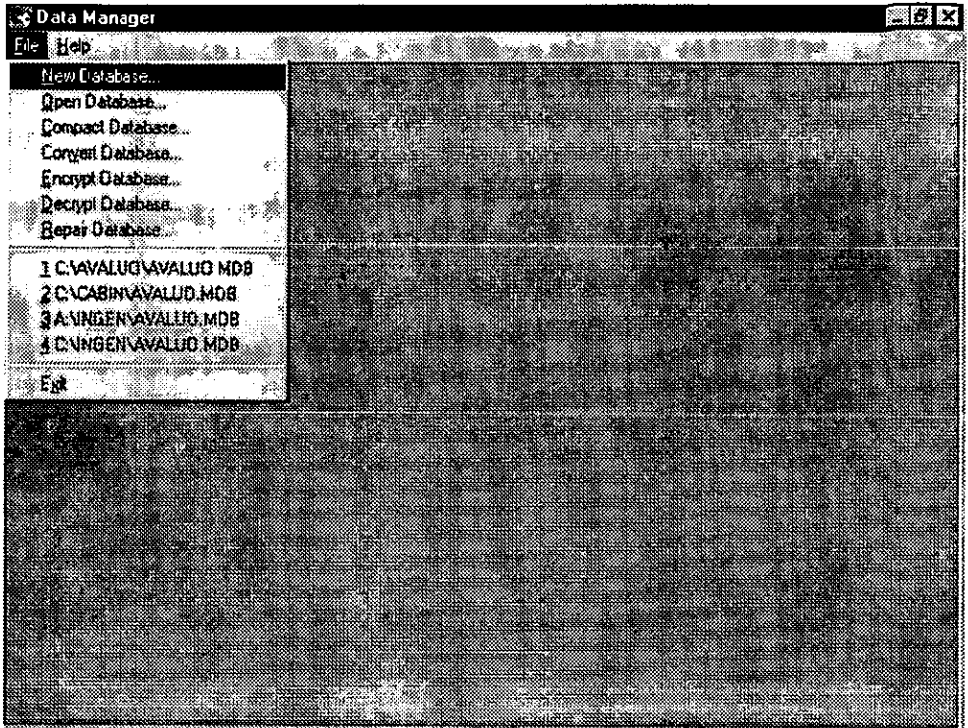


Figura VII. Pantalla de acceso al Data Manager de Visual Basic.

Para nuestro caso como ya existe la base de datos lo que procedemos a realizar es abrir la base de datos y en seguida podemos observar las tablas que conforman la base de datos de avalúos, en esta ventana se puede abrir una tabla y ver el contenido de cada uno de los registros que tiene esa tabla, también se puede crear una nueva tabla si es que el sistema lo requiere, la siguiente opción es ver el diseño de alguna de las tablas y cuales son sus propiedades que tiene, también se puede llevar a cabo el borrado de alguna tabla que el sistema ya no requiera y se pueden crear relaciones entre tablas.

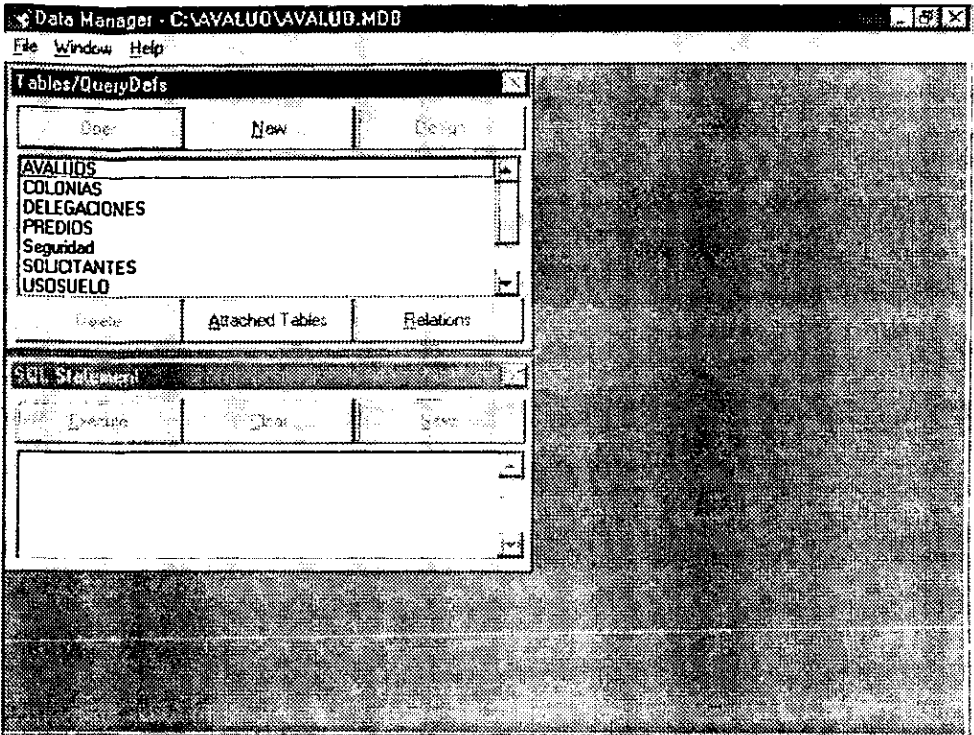


Figura VIII. Tablas de la Base de Datos.

Cuando ya tenemos las tablas que conforman la base de datos, se puede seleccionar alguna de ellas y después presionar el botón que tiene la etiqueta de Abrir, la actividad que realizara el Manejador de Datos es abrir esa tabla y se observara el primer registro que contiene la tabla, si se desean observar los demás registros de la tabla, se puede ir presionando las flechas que están en la parte inferior o si se desea encontrar un registro en especial se puede presionar el botón de buscar y escribir el campo que deseamos buscar y cual es el valor a buscar dentro de ese campo. Además en esta misma pantalla se pueden dar de alta nuevos registros, se pueden actualizar registros, se pueden borrar registros y se pueden refrescar los datos de la tabla.

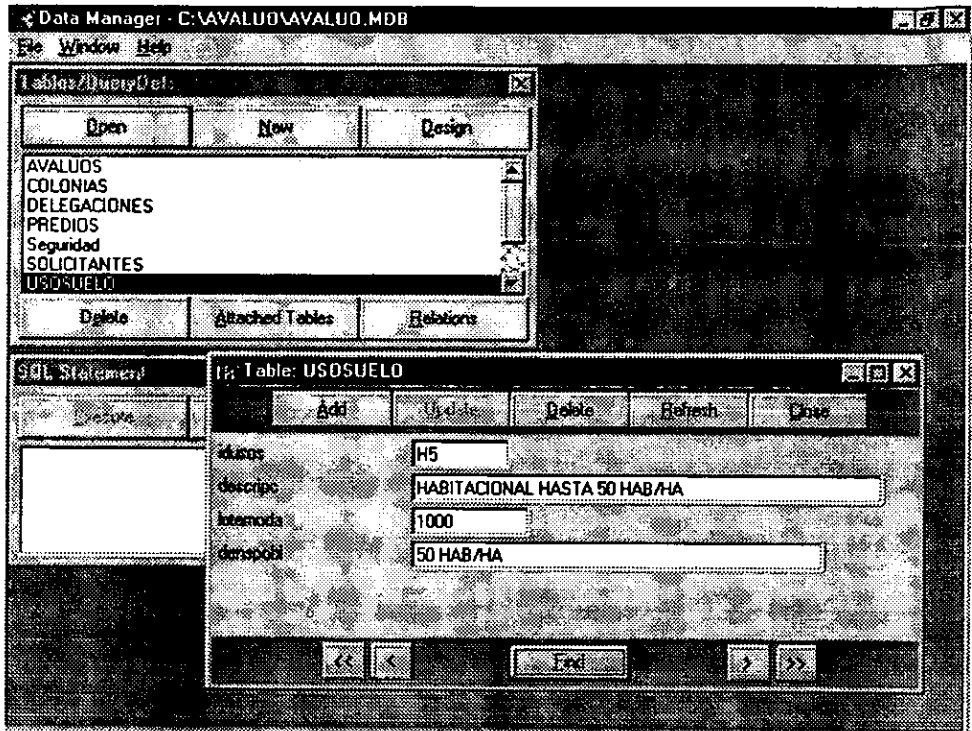


Figura IX. Abriendo una tabla de la base de datos.

Otra opción que se presenta al momento de abrir el manejador de datos es crear una nueva tabla para el sistema, en esta pantalla hay que escribir el nombre de la nueva tabla y definir cada uno de los campos que contendrá la tabla, así como de que tipo va a ser ese campo de la tabla.

La siguiente opción que encontramos es ver el diseño de los campos de la tabla, esto con el fin de cambiar las propiedades que tiene algún campo en especial, agregar un campo mas a la tabla o eliminar algún campo de la tabla.



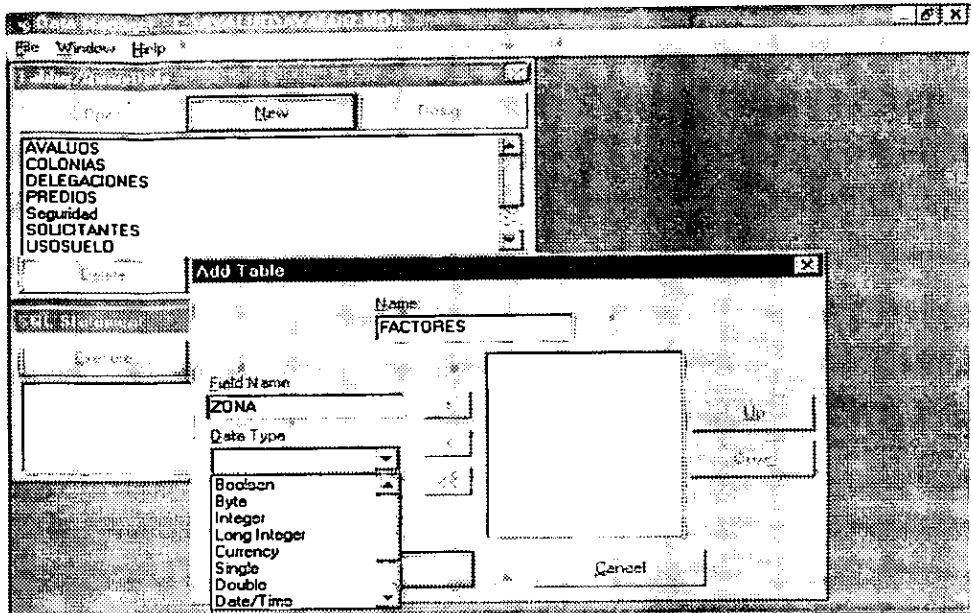


Figura X. Nueva tabla de la base de datos.

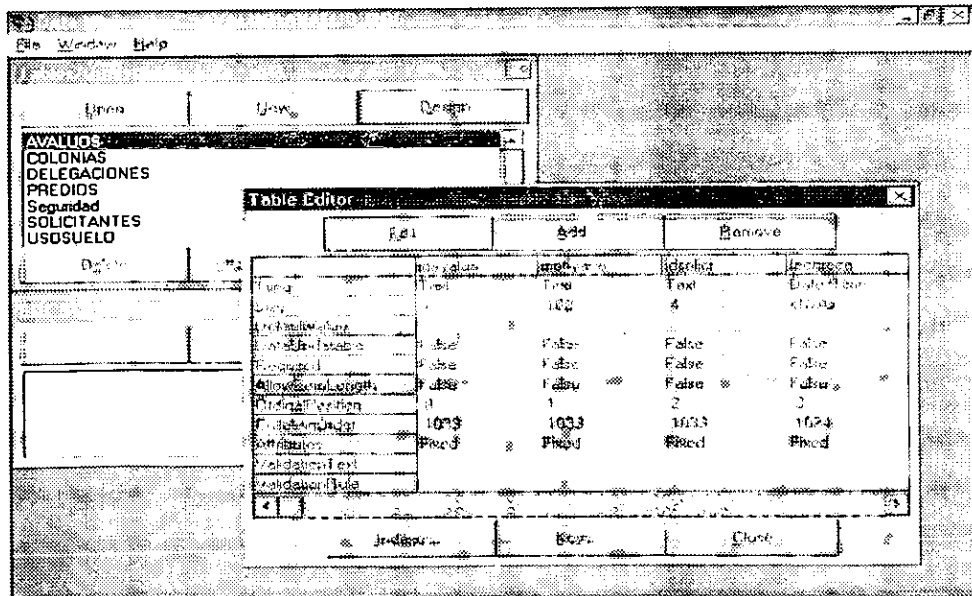


Figura XI. Diseño de una tabla.

## APÉNDICE C

### Código del sistema

```

Private Sub chkpwd()
' Verifica que el usuario este dado de alta en la base de datos
Dim MyBase As Database, MyTabla As Recordset
Dim res As Recordset, nivel As String, consulta As String
Dim wsuserid As String, wspassword As String

Set MyBase = Workspaces(0).OpenDatabase("c:\avaluo\avaluo.mdb")
wsuserid$ = LTrim(txtUserId)
wspassword$ = LTrim(txtPassword)
btnOk.MousePointer = 11

consulta = "Select * from seguridad where UserID = " & wsuserid & " and Password = " & wspassword &
""

Set res = MyBase.OpenRecordset(consulta, dbOpenDynaset)
If res.EOF Then
    MsgBox "USUARIO NO DADO DE ALTA", vbInformation
    Exit Sub
Else
    MsgBox "BIENVENIDO AL SISTEMA DE VALUACION", vbInformation
    txtnivel = res!seguridad
    If txtnivel = 1 Then 'Acceso a todo el sistema
        frmmenu.Show
        Acceso.Hide
    End If
    If txtnivel = 2 Then 'Todo el sistema menos mantenimiento
        frmmenu.Show
        frmmenu.SSTab1.TabEnabled(8) = False
        frmmenu.SSCommand18.Enabled = False
    End If
    If txtnivel = 3 Then 'Consultas y reportes
        frmmenu.Show
        frmmenu.SSTab1.TabEnabled(8) = False
        frmmenu.SSCommand1.Enabled = False
        frmmenu.SSCommand2.Enabled = False
        frmmenu.SSCommand3.Enabled = False
        frmmenu.SSCommand18.Enabled = False
        frmmenu.Altas.Enabled = False
        frmmenu.Bajas.Enabled = False
        frmmenu.Cambios.Enabled = False
    End If
End If
*****aquí se actualiza la fecha de acceso.*****
res.Edit
UltFecha_Acceso = Format(Now, "dd-mm-yy")
res!ultfecha = UltFecha_Acceso
res.Update
btnOk.Enabled = True
btnOk.MousePointer = 0
res.Close
MyBase.Close
End Sub

```

```

Private Sub SSSCommand1_Click()
' Registra la información que se desea dar de alta
' dependiendo del folder que se encuentre activo
Select Case SSTab1.Tab
Case 0      'FOLDER DE SOLICITANTES
  If Text1.Text = "" Or Text2.Text = "" Or _
    Text3.Text = "" Or Text4.Text = "" Or _
    Text5.Text = "" Or Text6.Text = "" Then
    MsgBox "INFORMACION INCOMPLETA ", 16
  Else
    Set SELECCION = BASE.OpenRecordset("SOLICITANTES", dbOpenDynaset)
    BeginTrans
    SELECCION.AddNew
    SELECCION!idsolici = Text1.Text
    SELECCION!tiposoli = Text2.Text
    SELECCION!nombre = Text3.Text
    SELECCION!domicili = Text4.Text
    SELECCION!telefono = Text5.Text
    SELECCION!rfc = Text6.Text
    SELECCION.Update
    CommitTrans
    SELECCION.Close
    Text1.Text = Empty
    Text2.Text = Empty
    Text3.Text = Empty
    Text4.Text = Empty
    Text5.Text = Empty
    Text6.Text = Empty
  End If
Case 1      'FOLDER DE VALUADORES
  If Text7.Text = "" Or Text8.Text = "" Or _
    Text9.Text = "" Or Text10.Text = "" Or _
    Text11.Text = "" Or Text12.Text = "" Or _
    Text13.Text = "" Then
    MsgBox "INFORMACION INCOMPLETA ", 16
  Else
    Set SELECCION = BASE.OpenRecordset("VALUADORES", dbOpenDynaset)
    BeginTrans
    SELECCION.AddNew
    SELECCION!idvaluad = Text7.Text
    SELECCION!nombre = Text8.Text
    SELECCION!tpovalua = Text9.Text
    SELECCION!registro = Text10.Text
    SELECCION!especiali = Text11.Text
    SELECCION!eficiencia = Text12.Text
    SELECCION!observac = Text13.Text
    SELECCION.Update
    CommitTrans
    SELECCION.Close
    Text7.Text = Empty
    Text8.Text = Empty
    Text9.Text = Empty
    Text10.Text = Empty
    Text11.Text = Empty
  End If

```

```

Text12.Text = Empty
Text13.Text = Empty
End If
Case 2      'FOLDER DE ESTUDIO DE MERCADO
If Text14.Text = "" Or Text15.Text = "" Or _
Text16.Text = "" Or Text17.Text = "" Or _
Text18.Text = "" Or Text19.Text = "" Or _
Text20.Text = "" Or Text21.Text = "" Or _
Text22.Text = "" Or Text23.Text = "" Or _
Text24.Text = "" Then
MsgBox "INFORMACION INCOMPLETA ", 16
Else
Set SELECCION = BASE.OpenRecordset("PREDIOS", dbOpenDynaset)
BeginTrans
SELECCION.AddNew
SELECCION!idpredio = Text14.Text
SELECCION!calle = Text15.Text
SELECCION!idcoloni = Text16.Text
SELECCION!iddelega = Text17.Text
SELECCION!idzona = Text18.Text
SELECCION!IDUSOSUE = Text19.Text
SELECCION!codpost = Text20.Text
SELECCION!nompropi = Text21.Text
SELECCION!superficie = Text22.Text
SELECCION!objeto = ""
SELECCION!boletpre = ""
SELECCION!escritura = ""
SELECCION!factorm = 0
SELECCION!factsupe = 0
SELECCION!factren = 0
SELECCION!factubic = 0
SELECCION!factopo = 0
SELECCION!factzona = 0
SELECCION!valorava = Text23.Text
SELECCION!fechreal = Text24.Text
SELECCION!calle1 = ""
SELECCION!calle2 = ""
SELECCION!calle3 = ""
SELECCION!calle4 = ""
SELECCION!linderos = " "
SELECCION.Update
CommitTrans
SELECCION.Close
Text14.Text = Empty
Text15.Text = Empty
Text16.Text = Empty
Text17.Text = Empty
Text18.Text = Empty
Text19.Text = Empty
Text20.Text = Empty
Text21.Text = Empty
Text22.Text = Empty
Text23.Text = Empty
Text24.Text = Empty
End If
Case 3      'FOLDER DE ZONAS

```

```

If Text25.Text = "" Or Text26.Text = "" Then
    MsgBox "INFORMACION INCOMPLETA ", 16
Else
    Set SELECCION = BASE.OpenRecordset("ZONAS", dbOpenDynaset)
    BeginTrans
    SELECCION.AddNew
    SELECCION!idzona = Text25.Text
    SELECCION!descripc = Text26.Text
    SELECCION.Update
    CommitTrans
    SELECCION.Close
    Text25.Text = Empty
    Text26.Text = Empty
End If
Combo3.Clear
cmb3_zonas
Combo5.Clear
cmb5_zonas
Case 4 'FOLDER DE USO DE SUELO
If Text27.Text = "" Or Text28.Text = "" Or _
    Text29.Text = "" Or Text30.Text = "" Then
    MsgBox "INFORMACION INCOMPLETA ", 16
Else
    Set SELECCION = BASE.OpenRecordset("USOSUELO", dbOpenDynaset)
    BeginTrans
    SELECCION.AddNew
    SELECCION!idusos = Text27.Text
    SELECCION!descripc = Text28.Text
    SELECCION!lotermoda = Text29.Text
    SELECCION!denspobl = Text30.Text
    SELECCION.Update
    CommitTrans
    SELECCION.Close
    Text27.Text = Empty
    Text28.Text = Empty
    Text29.Text = Empty
    Text30.Text = Empty
End If
Combo4.Clear
cmb4_uso
Combo6.Clear
cmb6_uso
Case 5 'FOLDER DE DELEGACIONES
If Text31.Text = "" Or Text32.Text = "" Then
    MsgBox "INFORMACION INCOMPLETA ", 16
Else
    Set SELECCION = BASE.OpenRecordset("DELEGACIONES", dbOpenDynaset)
    BeginTrans
    SELECCION.AddNew
    SELECCION!iddelega = Text31.Text
    SELECCION!nombdele = Text32.Text
    SELECCION.Update
    CommitTrans
    SELECCION.Close
    Text31.Text = Empty
    Text32.Text = Empty

```

```

End If
Combo2.Clear
cmb2_delegaciones
Combo7.Clear
cmb7_delegaciones
Case 6      'FOLDER DE COLONIAS
If Text13.Text = "" Or Text14.Text = "" Then
    MsgBox "INFORMACION INCOMPLETA ", 16
Else
    Set SELECCION = BASE.OpenRecordset("COLONIAS", dbOpenDynaset)
    BeginTrans
    SELECCION.AddNew
    SELECCION!idcoloni = Text13.Text
    SELECCION!nombcolo = Text14.Text
    SELECCION.Update
    CommitTrans
    SELECCION.Close
    Text13.Text = Empty
    Text14.Text = Empty
End If
Combo1.Clear
cmb1_colonias
Case 7      'FOLDER DE ZONA/SUELO/DELEGACION
If Text35.Text = "" Or Text36.Text = "" Or _
    Text37.Text = "" Or Text38.Text = "" Then
    MsgBox "INFORMACION INCOMPLETA ", 16
Else
    Set SELECCION = BASE.OpenRecordset("ZOSUDE", dbOpenDynaset)
    BeginTrans
    SELECCION.AddNew
    SELECCION!idzona = Text35.Text
    SELECCION!IDUSOSUE = Text36.Text
    SELECCION!iddelega = Text37.Text
    SELECCION!intecons = Text38.Text
    If SSCheck1 = True Then
        SELECCION!aguapota = "S"
    Else
        SELECCION!aguapota = "N"
    End If
    If SSCheck2 = True Then
        SELECCION!drenaje = "S"
    Else
        SELECCION!drenaje = "N"
    End If
    If SSCheck3 = True Then
        SELECCION!paviment = "S"
    Else
        SELECCION!paviment = "N"
    End If
    If SSCheck4 = True Then
        SELECCION!eneretec = "S"
    Else
        SELECCION!eneretec = "N"
    End If
    If SSCheck5 = True Then
        SELECCION!cablesub = "S"
    End If

```

```
Else
  SELECCION!cablesub = "N"
End If
If SSCheck6 = True Then
  SELECCION!redtelef = "S"
Else
  SELECCION!redtelef = "N"
End If
If SSCheck7 = True Then
  SELECCION!recobasu = "S"
Else
  SELECCION!recobasu = "N"
End If
If SSCheck8 = True Then
  SELECCION!vialidad = "S"
Else
  SELECCION!vialidad = "N"
End If
If SSCheck9 = True Then
  SELECCION!tvxcable = "S"
Else
  SELECCION!tvxcable = "N"
End If
If SSCheck10 = True Then
  SELECCION!comeserv = "S"
Else
  SELECCION!comeserv = "N"
End If
If SSCheck11 = True Then
  SELECCION!educacio = "S"
Else
  SELECCION!educacio = "N"
End If
If SSCheck12 = True Then
  SELECCION!asistenc = "S"
Else
  SELECCION!asistenc = "N"
End If
If SSCheck13 = True Then
  SELECCION!diversio = "S"
Else
  SELECCION!diversio = "N"
End If
If SSCheck14 = True Then
  SELECCION!recreaci = "S"
Else
  SELECCION!recreaci = "N"
End If
If SSCheck15 = True Then
  SELECCION!trabajo = "S"
Else
  SELECCION!trabajo = "N"
End If
If SSCheck16 = True Then
  SELECCION!parqpubl = "S"
Else
```

```

    SELECCION!parqpubl = "N"
End If
If SSCheck17 = True Then
    SELECCION!iglesias = "S"
Else
    SELECCION!iglesias = "N"
End If
If SSCheck18 = True Then
    SELECCION!servtele = "S"
Else
    SELECCION!servtele = "N"
End If
SELECCION!contamin = Text40.Text
SELECCION!observac = Text41.Text
SELECCION.Update
CommitTrans
SELECCION.Close
End If
Text35.Text = Empty
Text36.Text = Empty
Text37.Text = Empty
Text38.Text = Empty
SSCheck1 = False
SSCheck2 = False
SSCheck3 = False
SSCheck4 = False
SSCheck5 = False
SSCheck6 = False
SSCheck7 = False
SSCheck8 = False
SSCheck9 = False
SSCheck10 = False
SSCheck11 = False
SSCheck12 = False
SSCheck13 = False
SSCheck14 = False
SSCheck15 = False
SSCheck16 = False
SSCheck17 = False
SSCheck18 = False
Text40.Text = Empty
Text41.Text = Empty
Case 8      ' FOLDER DE MANTENIMIENTO PARA DAR DE ALTA USUARIOS
If SSFrame9.Visible = False Then
    MsgBox "NO ESTA HABILITADO EL MARCO DE USUARIOS ", 16
Else
    If Text45.Text = "" Or Text44.Text = "" Or _
        Text39.Text = "" Or Text43.Text = "" Or _
        Text42.Text = "" Then
        MsgBox "INFORMACION INCOMPLETA ", 16
    Else
        Set SELECCION = BASE.OpenRecordset("SEGURIDAD", dbOpenDynaset)
        BeginTrans
        SELECCION.AddNew
        SELECCION!userid = Text45.Text
        SELECCION!Password = Text44.Text
    End If
End If

```



```

SELECCION!seguridad = Text39.Text
SELECCION!nombre = Text43.Text
SELECCION!ultfecha = Text42.Text
SELECCION.Update
CommitTrans
SELECCION.Close
Text45.Text = Empty
Text44.Text = Empty
Text39.Text = Empty
Text43.Text = Empty
Text42.Text = Empty
End If
End If
End Select
End Sub

```

```

Private Sub SSCommand2_Click()
'Realiza la baja de un registro de la base de datos
'dependiendo del folder que se encuentre activo y del
'registro que se encuentre en la pantalla
Dim nreg As Integer
Select Case SStab1.Tab
Case 0 'SOLICITANTES
If Text1.Text = "" Or Text2.Text = "" Or _
Text3.Text = "" Or Text4.Text = "" Or _
Text5.Text = "" Or Text6.Text = "" Then
MsgBox "NO EXISTE REGISTRO A ELIMINAR ", 16
Else
Set SELECCION = BASE.OpenRecordset("SOLICITANTES", dbOpenDynaset)
SELECCION.FindFirst "[IDSOLIC] like "" & Text1.Text & ""
nreg = SELECCION.AbsolutePosition
BeginTrans
SELECCION.Delete
Text1.Text = Empty
Text2.Text = Empty
Text3.Text = Empty
Text4.Text = Empty
Text5.Text = Empty
Text6.Text = Empty
CommitTrans
If nreg = SELECCION.RecordCount Then
SELECCION.MovePrevious
Else
SELECCION.MoveNext
End If
SELECCION.Close
llena_solicitantes
End If
SSCommand1.Enabled = True
frmmenu.Altas.Enabled = True
Case 1 'VALUADORES
If Text7.Text = "" Or Text8.Text = "" Or _
Text9.Text = "" Or Text10.Text = "" Or _
Text11.Text = "" Or Text12.Text = "" Or _
Text13.Text = "" Then

```

```

    MsgBox "NO EXISTE REGISTRO A ELIMINAR ", 16
Else
Set SELECCION = BASE.OpenRecordset("VALUADORES", dbOpenDynaset)
SELECCION.FindFirst "[IDVALUAD] like "" & Text7.Text & ""
nreg = SELECCION.AbsolutePosition
BeginTrans
SELECCION.Delete
Text7.Text = Empty
Text8.Text = Empty
Text9.Text = Empty
Text10.Text = Empty
Text11.Text = Empty
Text12.Text = Empty
Text13.Text = Empty
CommitTrans
If nreg = SELECCION.RecordCount Then
    SELECCION.MovePrevious
Else
    SELECCION.MoveNext
End If
SELECCION.Close
llena_valuadores
End If
SSCommand1.Enabled = True
frmmenu.Altas.Enabled = True
Case 2 'ESTUDIO DE MERCADO
If Text14.Text = "" Or Text15.Text = "" Or _
Text16.Text = "" Or Text17.Text = "" Or _
Text18.Text = "" Or Text19.Text = "" Or _
Text20.Text = "" Or Text21.Text = "" Or _
Text22.Text = "" Or Text23.Text = "" Or _
Text24.Text = "" Then
    MsgBox "NO EXISTE REGISTRO A ELIMINAR ", 16
Else
Set SELECCION = BASE.OpenRecordset("PREDIOS", dbOpenDynaset)
SELECCION.FindFirst "[IDPREDIO] like "" & Text14.Text & ""
nreg = SELECCION.AbsolutePosition
BeginTrans
SELECCION.Delete
Text14.Text = Empty
Text15.Text = Empty
Text16.Text = Empty
Text17.Text = Empty
Text18.Text = Empty
Text19.Text = Empty
Text20.Text = Empty
Text21.Text = Empty
Text22.Text = Empty
Text23.Text = Empty
Text24.Text = Empty
CommitTrans
If nreg = SELECCION.RecordCount Then
    SELECCION.MovePrevious
Else
    SELECCION.MoveNext
End If

```

```

SELECCION.Close
llena_mercado
End If
SSCommand1.Enabled = True
frmmenu.Altas.Enabled = True
Case 3 'ZONAS
If Text25.Text = "" Or Text26.Text = "" Then
    MsgBox "NO EXISTE REGISTRO A ELIMINAR ", 16
Else
    Set SELECCION = BASE.OpenRecordset("ZONAS", dbOpenDynaset)
    SELECCION.FindFirst "[IDZONA] like " & Text25.Text & ""
    nreg = SELECCION.AbsolutePosition
    BeginTrans
    SELECCION.Delete
    Text25.Text = Empty
    Text26.Text = Empty
    CommitTrans
    If nreg = SELECCION.RecordCount Then
        SELECCION.MovePrevious
    Else
        SELECCION.MoveNext
    End If
    SELECCION.Close
    llena_zonas
    cmb3_zonas
    cmb5_zonas
End If
SSCommand1.Enabled = True
frmmenu.Altas.Enabled = True
Case 4 'USO DE SUELO
If Text27.Text = "" Or Text28.Text = "" Or _
    Text29.Text = "" Or Text30.Text = "" Then
    MsgBox "NO EXISTE REGISTRO A ELIMINAR ", 16
Else
    Set SELECCION = BASE.OpenRecordset("USOSUELO", dbOpenDynaset)
    SELECCION.FindFirst "[IDUSOS] like " & Text27.Text & ""
    nreg = SELECCION.AbsolutePosition
    BeginTrans
    SELECCION.Delete
    Text27.Text = Empty
    Text28.Text = Empty
    Text29.Text = Empty
    Text30.Text = Empty
    CommitTrans
    If nreg = SELECCION.RecordCount Then
        SELECCION.MovePrevious
    Else
        SELECCION.MoveNext
    End If
    SELECCION.Close
    llena_uso
    cmb4_uso
    cmb6_uso
End If
SSCommand1.Enabled = True
frmmenu.Altas.Enabled = True

```

```

Case 5 'DELEGACIONES
If Text31.Text = "" Or Text32.Text = "" Then
    MsgBox "NO EXISTE REGISTRO A ELIMINAR ", 16
Else
    Set SELECCION = BASE.OpenRecordset("DELEGACIONES", dbOpenDynaset)
    SELECCION.FindFirst "[IDDELEGA] like "" & Text31.Text & ""
    nreg = SELECCION.AbsolutePosition
    BeginTrans
    SELECCION.Delete
    Text31.Text = Empty
    Text32.Text = Empty
    CommitTrans
    If nreg = SELECCION.RecordCount Then
        SELECCION.MovePrevious
    Else
        SELECCION.MoveNext
    End If
    SELECCION.Close
    llena_delegaciones
    cmb2_delegaciones
    cmb7_delegaciones
End If
SSCommand1.Enabled = True
frmmenu.Altas.Enabled = True
Case 6 'COLONIAS
If Text33.Text = "" Or Text34.Text = "" Then
    MsgBox "NO EXISTE REGISTRO A ELIMINAR ", 16
Else
    Set SELECCION = BASE.OpenRecordset("COLONIAS", dbOpenDynaset)
    SELECCION.FindFirst "[IDCOLONI] like "" & Text33.Text & ""
    nreg = SELECCION.AbsolutePosition
    BeginTrans
    SELECCION.Delete
    Text33.Text = Empty
    Text34.Text = Empty
    CommitTrans
    If nreg = SELECCION.RecordCount Then
        SELECCION.MovePrevious
    Else
        SELECCION.MoveNext
    End If
    SELECCION.Close
    llena_colonias
    cmb1_colonias
End If
SSCommand1.Enabled = True
frmmenu.Altas.Enabled = True
Case 7 'ZONA/SUELO/DELEGACION
If Text35.Text = "" Or Text36.Text = "" Or _
    Text37.Text = "" Or Text38.Text = "" Then
    MsgBox "NO EXISTE REGISTRO A ELIMINAR ", 16
Else
    Set SELECCION = BASE.OpenRecordset("ZOSUDE", dbOpenDynaset)
    SELECCION.FindFirst "[IDZONA] like "" & Text35.Text & ""
    nreg = SELECCION.AbsolutePosition
    BeginTrans

```

```

SELECCION.Delete
Text35.Text = Empty
Text36.Text = Empty
Text37.Text = Empty
Text38.Text = Empty
SSCheck1 = False
SSCheck2 = False
SSCheck3 = False
SSCheck4 = False
SSCheck5 = False
SSCheck6 = False
SSCheck7 = False
SSCheck8 = False
SSCheck9 = False
SSCheck10 = False
SSCheck11 = False
SSCheck12 = False
SSCheck13 = False
SSCheck14 = False
SSCheck15 = False
SSCheck16 = False
SSCheck17 = False
SSCheck18 = False
Text40.Text = Empty
Text41.Text = Empty
CommitTrans
If nreg = SELECCION.RecordCount Then
    SELECCION.MovePrevious
Else
    SELECCION.MoveNext
End If
SELECCION.Close
llena_zosude
End If
SSCommand1.Enabled = True
frmmenu.Altas.Enabled = True
Case 8 'BAJA DE USUARIO
If SSFrame9.Visible = False Then
    MsgBox "NO ESTA HABILITADO EL MARCO DE USUARIOS ", 16
Else
    If Text45.Text = "" Or Text44.Text = "" Or _
        Text39.Text = "" Or Text43.Text = "" Or _
        Text42.Text = "" Then
        MsgBox "NO EXISTE REGISTRO A ELIMINAR ", 16
    Else
        Set SELECCION = BASE.OpenRecordset("SEGURIDAD", dbOpenDynaset)
        SELECCION.FindFirst "[USERID] like " & Text45.Text & ""
        nreg = SELECCION.AbsolutePosition
        BeginTrans
        SELECCION.Delete
        Text45.Text = Empty
        Text44.Text = Empty
        Text39.Text = Empty
        Text43.Text = Empty
        Text42.Text = Empty
        CommitTrans
    End If
End If

```

```

If nreg = SELECCION.RecordCount Then
    SELECCION.MovePrevious
Else
    SELECCION.MoveNext
End If
SELECCION.Close
End If
End If
SSCommand1.Enabled = True
frmmenu.Altas.Enabled = True
End Select
End Sub

```

```

Private Sub SSCommand3_Click()
'Realiza una actualización de un registro que se haya modificado
'dependiendo del folder que se encuentre activo
Select Case SSTab1.Tab
Case 0 'SOLICITANTES
    If Text1.Text = "" Or Text2.Text = "" Or _
        Text3.Text = "" Or Text4.Text = "" Or _
        Text5.Text = "" Or Text6.Text = "" Then
        MsgBox "INFORMACION INCOMPLETA ", 16
    Else
        Set SELECCION = BASE.OpenRecordset("SOLICITANTES", dbOpenDynaset)
        BeginTrans
        SELECCION.Edit
        SELECCION!idsolici = Text1.Text
        SELECCION!tiposoli = Text2.Text
        SELECCION!nombre = Text3.Text
        SELECCION!domicili = Text4.Text
        SELECCION!telefono = Text5.Text
        SELECCION!rfc = Text6.Text
        SELECCION.Update
        CommitTrans
        SELECCION.Close
        Text1.Text = Empty
        Text2.Text = Empty
        Text3.Text = Empty
        Text4.Text = Empty
        Text5.Text = Empty
        Text6.Text = Empty
    End If
Case 1 'VALUADORES
    If Text7.Text = "" Or Text8.Text = "" Or _
        Text9.Text = "" Or Text10.Text = "" Or _
        Text11.Text = "" Or Text12.Text = "" Or _
        Text13.Text = "" Then
        MsgBox "INFORMACION INCOMPLETA ", 16
    Else
        Set SELECCION = BASE.OpenRecordset("VALUADORES", dbOpenDynaset)
        BeginTrans
        SELECCION.Edit
        SELECCION!idvaluad = Text7.Text
        SELECCION!nombre = Text8.Text
        SELECCION!tpoalua = Text9.Text
    End If

```

```

SELECCION!registro = Text10.Text
SELECCION!especiali = Text11.Text
SELECCION!eficiencia = Text12.Text
SELECCION!observac = Text13.Text
SELECCION.Update
CommitTrans
SELECCION.Close
Text7.Text = Empty
Text8.Text = Empty
Text9.Text = Empty
Text10.Text = Empty
Text11.Text = Empty
Text12.Text = Empty
Text13.Text = Empty
End If
Case 2 'ESTUDIO DE MERCADO
If Text14.Text = "" Or Text15.Text = "" Or _
Text16.Text = "" Or Text17.Text = "" Or _
Text18.Text = "" Or Text19.Text = "" Or _
Text20.Text = "" Or Text21.Text = "" Or _
Text22.Text = "" Or Text23.Text = "" Or _
Text24.Text = "" Then
MsgBox "INFORMACION INCOMPLETA ", 16
Else
Set SELECCION = BASE.OpenRecordset("PREDIOS", dbOpenDynaset)
BeginTrans
SELECCION.Edit
SELECCION!idpredio = Text14.Text
SELECCION!calle = Text15.Text
SELECCION!idcoloni = Text16.Text
SELECCION!iddelega = Text17.Text
SELECCION!idzona = Text18.Text
SELECCION!IDUSOSUE = Text19.Text
SELECCION!codpost = Text20.Text
SELECCION!nompropi = Text21.Text
SELECCION!superficie = Text22.Text
SELECCION!objeto = ""
SELECCION!boletpre = ""
SELECCION!escritura = ""
SELECCION!factform = 0
SELECCION!factsupe = 0
SELECCION!factfren = 0
SELECCION!factubic = 0
SELECCION!facttopo = 0
SELECCION!factzona = 0
SELECCION!valorava = Text23.Text
SELECCION!fechreal = Text24.Text
SELECCION!calle1 = ""
SELECCION!calle2 = ""
SELECCION!calle3 = ""
SELECCION!calle4 = ""
SELECCION!linderos = " "
SELECCION.Update
CommitTrans
SELECCION.Close
Text14.Text = Empty

```

```

Text15.Text = Empty
Text16.Text = Empty
text17.Text = Empty
Text18.Text = Empty
Text19.Text = Empty
Text20.Text = Empty
Text21.Text = Empty
Text22.Text = Empty
Text23.Text = Empty
Text24.Text = Empty
End If
Case 3 'ZONAS
If Text25.Text = "" Or Text26.Text = "" Then
  MsgBox "INFORMACION INCOMPLETA ", 16
Else
  Set SELECCION = BASE.OpenRecordset("ZONAS", dbOpenDynaset)
  BeginTrans
  SELECCION.Edit
  SELECCION!idzona = Text25.Text
  SELECCION!descripc = Text26.Text
  SELECCION.Update
  CommitTrans
  SELECCION.Close
  Text25.Text = Empty
  Text26.Text = Empty
End If
Combo3.Clear
cmb3_zonas
Combo5.Clear
cmb5_zonas
Case 4 'USO DE SUELO
If Text27.Text = "" Or Text28.Text = "" Or _
  Text29.Text = "" Or Text30.Text = "" Then
  MsgBox "INFORMACION INCOMPLETA ", 16
Else
  Set SELECCION = BASE.OpenRecordset("USOSUELO", dbOpenDynaset)
  BeginTrans
  SELECCION.Edit
  SELECCION!idusos = Text27.Text
  SELECCION!descripc = Text28.Text
  SELECCION!lotemoda = Text29.Text
  SELECCION!denspobl = Text30.Text
  SELECCION.Update
  CommitTrans
  SELECCION.Close
  Text27.Text = Empty
  Text28.Text = Empty
  Text29.Text = Empty
  Text30.Text = Empty
End If
Combo4.Clear
cmb4_uso
Combo6.Clear
cmb6_uso
Case 5 'DELEGACIONES
If Text31.Text = "" Or Text32.Text = "" Then

```



```

    MsgBox "INFORMACION INCOMPLETA ", 16
Else
    Set SELECCION = BASE.OpenRecordset("DELEGACIONES", dbOpenDynaset)
    BeginTrans
    SELECCION.Edit
    SELECCION!iddelega = Text31.Text
    SELECCION!nombdele = Text32.Text
    SELECCION.Update
    CommitTrans
    SELECCION.Close
    Text31.Text = Empty
    Text32.Text = Empty
End If
Combo2.Clear
cmb2_delegaciones
Combo7.Clear
cmb7_delegaciones
Case 6 'COLONIAS
    If Text33.Text = "" Or Text34.Text = "" Then
        MsgBox "INFORMACION INCOMPLETA ", 16
    Else
        Set SELECCION = BASE.OpenRecordset("COLONIAS", dbOpenDynaset)
        BeginTrans
        SELECCION.Edit
        SELECCION!idcoloni = Text33.Text
        SELECCION!nombcolo = Text34.Text
        SELECCION.Update
        CommitTrans
        SELECCION.Close
        Text33.Text = Empty
        Text34.Text = Empty
    End If
    Combo1.Clear
    cmb1_colonias
Case 7 'ZONA/SUELO/DELEGACION
    If Text35.Text = "" Or Text36.Text = "" Or _
        Text37.Text = "" Or Text38.Text = "" Then
        MsgBox "INFORMACION INCOMPLETA ", 16
    Else
        Set SELECCION = BASE.OpenRecordset("ZOSUDE", dbOpenDynaset)
        BeginTrans
        SELECCION.Edit
        SELECCION!idzona = Text35.Text
        SELECCION!IDUSOSUE = Text36.Text
        SELECCION!iddelega = Text37.Text
        SELECCION!intecons = Text38.Text
        If SSCheck1 = True Then
            SELECCION!aguapota = "S"
        Else
            SELECCION!aguapota = "N"
        End If
        If SSCheck2 = True Then
            SELECCION!drenaje = "S"
        Else
            SELECCION!drenaje = "N"
        End If
    End If

```

```
If SSCheck3 = True Then
  SELECCION!paviment = "S"
Else
  SELECCION!paviment = "N"
End If
If SSCheck4 = True Then
  SELECCION!enerelec = "S"
Else
  SELECCION!enerelec = "N"
End If
If SSCheck5 = True Then
  SELECCION!cablesub = "S"
Else
  SELECCION!cablesub = "N"
End If
If SSCheck6 = True Then
  SELECCION!redtelef = "S"
Else
  SELECCION!redtelef = "N"
End If
If SSCheck7 = True Then
  SELECCION!recobasu = "S"
Else
  SELECCION!recobasu = "N"
End If
If SSCheck8 = True Then
  SELECCION!vialidad = "S"
Else
  SELECCION!vialidad = "N"
End If
If SSCheck9 = True Then
  SELECCION!tvxcable = "S"
Else
  SELECCION!tvxcable = "N"
End If
If SSCheck10 = True Then
  SELECCION!comeserv = "S"
Else
  SELECCION!comeserv = "N"
End If
If SSCheck11 = True Then
  SELECCION!educacio = "S"
Else
  SELECCION!educacio = "N"
End If
If SSCheck12 = True Then
  SELECCION!asistenc = "S"
Else
  SELECCION!asistenc = "N"
End If
If SSCheck13 = True Then
  SELECCION!diversio = "S"
Else
  SELECCION!diversio = "N"
End If
If SSCheck14 = True Then
```

```

    SELECCION!recreaci = "S"
Else
    SELECCION!recreaci = "N"
End If
If SSCheck15 = True Then
    SELECCION!trabajo = "S"
Else
    SELECCION!trabajo = "N"
End If
If SSCheck16 = True Then
    SELECCION!parqpubl = "S"
Else
    SELECCION!parqpubl = "N"
End If
If SSCheck17 = True Then
    SELECCION!iglesias = "S"
Else
    SELECCION!iglesias = "N"
End If
If SSCheck18 = True Then
    SELECCION!servtele = "S"
Else
    SELECCION!servtele = "N"
End If
SELECCION!contamin = Text40.Text
SELECCION!observac = Text41.Text
SELECCION.Update
CommitTrans
SELECCION.Close
End If
Text35.Text = Empty
Text36.Text = Empty
Text37.Text = Empty
Text38.Text = Empty
SSCheck1 = False
SSCheck2 = False
SSCheck3 = False
SSCheck4 = False
SSCheck5 = False
SSCheck6 = False
SSCheck7 = False
SSCheck8 = False
SSCheck9 = False
SSCheck10 = False
SSCheck11 = False
SSCheck12 = False
SSCheck13 = False
SSCheck14 = False
SSCheck15 = False
SSCheck16 = False
SSCheck17 = False
SSCheck18 = False
Text40.Text = Empty
Text41.Text = Empty
Case 8 'USUARIOS
If SSFrame9.Visible = False Then

```

```

MsgBox "NO ESTA HABILITADO EL MARCO DE USUARIOS ", 16
Else
If Text45.Text = "" Or Text44.Text = "" Or _
Text39.Text = "" Or Text43.Text = "" Or _
Text42.Text = "" Then
MsgBox "INFORMACION INCOMPLETA ", 16
Else
Set SELECCION = BASE.OpenRecordset("SEGURIDAD", dbOpenDynaset)
BeginTrans
SELECCION.Edit
SELECCION!userid = Text45.Text
SELECCION!Password = Text44.Text
SELECCION!seguridad = Text39.Text
SELECCION!nombre = Text43.Text
SELECCION!ultfecha = Text42.Text
SELECCION.Update
CommitTrans
SELECCION.Close
Text45.Text = Empty
Text44.Text = Empty
Text39.Text = Empty
Text43.Text = Empty
Text42.Text = Empty
End If
End If
End Select
End Sub

Private Sub SSCommand5_Click()
'Busca el primer registro en la base de datos y lo trae a pantalla
'dependiendo del folder que se encuentre activo
Dim res As Recordset, consulta As String, Paso As Integer
SSCommand1.Enabled = False
frmmenu.Altas.Enabled = False
Select Case SSTab1.Tab
Case 0 'SOLICITANTES
Set SELECCIONSOLICITANTES = BASE.OpenRecordset("SOLICITANTES", dbOpenDynaset)
If (SELECCIONSOLICITANTES.EOF And SELECCIONSOLICITANTES.BOF) Then
MsgBox " NO EXISTEN REGISTROS", 16
Exit Sub
End If
SELECCIONSOLICITANTES.MoveFirst
Text1.Text = SELECCIONSOLICITANTES!idsolici
Text2.Text = SELECCIONSOLICITANTES!tiposoli
Text3.Text = SELECCIONSOLICITANTES!nombre
Text4.Text = SELECCIONSOLICITANTES!domicili
Text5.Text = SELECCIONSOLICITANTES!telefono
Text6.Text = SELECCIONSOLICITANTES!irfc
If Not SELECCIONSOLICITANTES.EOF Then
SELECCIONSOLICITANTES.MoveNext
SELECCIONSOLICITANTES.MovePrevious
End If
Case 1 'VALUADORES
Set SELECCIONVALUADORES = BASE.OpenRecordset("valuadores", dbOpenDynaset)
If (SELECCIONVALUADORES.EOF And SELECCIONVALUADORES.BOF) Then

```

```

    MsgBox " NO EXISTEN REGISTROS", 16
    Exit Sub
End If
SELECCIONVALUADORES.MoveFirst
Text7.Text = SELECCIONVALUADORES!idvaluad
Text8.Text = SELECCIONVALUADORES!nombre
Text9.Text = SELECCIONVALUADORES!tpovalua
Text10.Text = SELECCIONVALUADORES!registro
Text11.Text = SELECCIONVALUADORES!especiali
Text12.Text = SELECCIONVALUADORES!eficiencia
Text13.Text = SELECCIONVALUADORES!observac
If Not SELECCIONVALUADORES.EOF Then
    SELECCIONVALUADORES.MoveNext
    SELECCIONVALUADORES.MovePrevious
End If
Case 2 'EST. DE MERCADO
Set SELECCIONMERCADO = BASE.OpenRecordset("predios", dbOpenDynaset)
If (SELECCIONMERCADO.EOF And SELECCIONMERCADO.BOF) Then
    MsgBox " NO EXISTEN REGISTROS", 16
    Exit Sub
End If
SELECCIONMERCADO.MoveFirst
Text14.Text = SELECCIONMERCADO!idpredio
Text15.Text = SELECCIONMERCADO!calles
Text16.Text = SELECCIONMERCADO!idcoloni
Text17.Text = SELECCIONMERCADO!iddelega
Text18.Text = SELECCIONMERCADO!idzona
Text19.Text = SELECCIONMERCADO!IDUSOSUE
Text20.Text = SELECCIONMERCADO!codpost
Text21.Text = SELECCIONMERCADO!nompropi
Text22.Text = SELECCIONMERCADO!superficie
Text23.Text = SELECCIONMERCADO!valorava
Text24.Text = SELECCIONMERCADO!fechreal
If Not SELECCIONMERCADO.EOF Then
    SELECCIONMERCADO.MoveNext
    SELECCIONMERCADO.MovePrevious
End If
consulta = " select * from colonias where idcoloni = " & Text16.Text & ""
Set res = BASE.OpenRecordset(consulta, dbOpenDynaset)
Combo1.Text = res!nombcolo
res.Close
consulta = " select * from delegaciones where iddelega = " & text17.Text & ""
Set res = BASE.OpenRecordset(consulta, dbOpenDynaset)
Combo2.Text = res!nombdele
res.Close
consulta = " select * from zonas where idzona = " & Text18.Text & ""
Set res = BASE.OpenRecordset(consulta, dbOpenDynaset)
Combo3.Text = res!descripc
res.Close
consulta = " select * from usosuelo where idusos = " & Text19.Text & ""
Set res = BASE.OpenRecordset(consulta, dbOpenDynaset)
Combo4.Text = res!descripc
res.Close
Case 3 'ZONAS
Set SELECCIONZONAS = BASE.OpenRecordset("zonas", dbOpenDynaset)
If (SELECCIONZONAS.EOF And SELECCIONZONAS.BOF) Then

```

```

    MsgBox " NO EXISTEN REGISTROS", 16
    Exit Sub
End If
SELECCIONZONAS.MoveFirst
Text25.Text = SELECCIONZONAS!idzona
Text26.Text = SELECCIONZONAS!descripc
If Not SELECCIONZONAS.EOF Then
    SELECCIONZONAS.MoveNext
    SELECCIONZONAS.MovePrevious
End If
Case 4 'USO DE SUELO
Set SELECCIONUSO = BASE.OpenRecordset("usosuelo", dbOpenDynaset)
If (SELECCIONUSO.EOF And SELECCIONUSO.BOF) Then
    MsgBox " NO EXISTEN REGISTROS", 16
    Exit Sub
End If
SELECCIONUSO.MoveFirst
Text27.Text = SELECCIONUSO!idusos
Text28.Text = SELECCIONUSO!descripc
Text29.Text = SELECCIONUSO!lotemoda
Text30.Text = SELECCIONUSO!denspobl
If Not SELECCIONUSO.EOF Then
    SELECCIONUSO.MoveNext
    SELECCIONUSO.MovePrevious
End If
Case 5 'DELEGACIONES
Set SELECCIONDELEGACIONES = BASE.OpenRecordset("delegaciones", dbOpenDynaset)
If (SELECCIONDELEGACIONES.EOF And SELECCIONDELEGACIONES.BOF) Then
    MsgBox " NO EXISTEN REGISTROS", 16
    Exit Sub
End If
SELECCIONDELEGACIONES.MoveFirst
Text31.Text = SELECCIONDELEGACIONES!iddetega
Text32.Text = SELECCIONDELEGACIONES!nombdele
If Not SELECCIONDELEGACIONES.EOF Then
    SELECCIONDELEGACIONES.MoveNext
    SELECCIONDELEGACIONES.MovePrevious
End If
Case 6 'COLONIAS
Set SELECCIONCOLONIAS = BASE.OpenRecordset("colonias", dbOpenDynaset)
If (SELECCIONCOLONIAS.EOF And SELECCIONCOLONIAS.BOF) Then
    MsgBox " NO EXISTEN REGISTROS", 16
    Exit Sub
End If
SELECCIONCOLONIAS.MoveFirst
Text33.Text = SELECCIONCOLONIAS!idcoloni
Text34.Text = SELECCIONCOLONIAS!nombcolo
If Not SELECCIONCOLONIAS.EOF Then
    SELECCIONCOLONIAS.MoveNext
    SELECCIONCOLONIAS.MovePrevious
End If
Case 7 'ZONA/SUELO/DELEGACION
Set SELECCIONZOSUDE = BASE.OpenRecordset("zosude", dbOpenDynaset)
If (SELECCIONZOSUDE.EOF And SELECCIONZOSUDE.BOF) Then
    MsgBox " NO EXISTEN REGISTROS", 16
    Exit Sub

```

```

End If
SELECCIONZOSUDE.MoveFirst
Text35.Text = SELECCIONZOSUDE!idzona
Text36.Text = SELECCIONZOSUDE!IDUSOSUE
Text37.Text = SELECCIONZOSUDE!iddelega
Text38.Text = SELECCIONZOSUDE!intecons
If SELECCIONZOSUDE!aguapota = "S" Then
  SSCheck1 = True
Else
  SSCheck1 = False
End If
If SELECCIONZOSUDE!drenaje = "S" Then
  SSCheck2 = True
Else
  SSCheck2 = False
End If
If SELECCIONZOSUDE!paviment = "S" Then
  SSCheck3 = True
Else
  SSCheck3 = False
End If
If SELECCIONZOSUDE!enerelec = "S" Then
  SSCheck4 = True
Else
  SSCheck4 = False
End If
If SELECCIONZOSUDE!cablesub = "S" Then
  SSCheck5 = True
Else
  SSCheck5 = False
End If
If SELECCIONZOSUDE!redtelefon = "S" Then
  SSCheck6 = True
Else
  SSCheck6 = False
End If
If SELECCIONZOSUDE!recobasu = "S" Then
  SSCheck7 = True
Else
  SSCheck7 = False
End If
If SELECCIONZOSUDE!vialidad = "S" Then
  SSCheck8 = True
Else
  SSCheck8 = False
End If
If SELECCIONZOSUDE!tvxcable = "S" Then
  SSCheck9 = True
Else
  SSCheck9 = False
End If
If SELECCIONZOSUDE!comeserv = "S" Then
  SSCheck10 = True
Else
  SSCheck10 = False
End If

```

```

If SELECCIONZOSUDE!educacio = "S" Then
  SSCheck11 = True
Else
  SSCheck11 = False
End If
If SELECCIONZOSUDE!asistenc = "S" Then
  SSCheck12 = True
Else
  SSCheck12 = False
End If
If SELECCIONZOSUDE!diversio = "S" Then
  SSCheck13 = True
Else
  SSCheck13 = False
End If
If SELECCIONZOSUDE!recreaci = "S" Then
  SSCheck14 = True
Else
  SSCheck14 = False
End If
If SELECCIONZOSUDE!trabajo = "S" Then
  SSCheck15 = True
Else
  SSCheck15 = False
End If
If SELECCIONZOSUDE!parqpubl = "S" Then
  SSCheck16 = True
Else
  SSCheck16 = False
End If
If SELECCIONZOSUDE!iglesias = "S" Then
  SSCheck17 = True
Else
  SSCheck17 = False
End If
If SELECCIONZOSUDE!servtele = "S" Then
  SSCheck18 = True
Else
  SSCheck18 = False
End If
Text40.Text = SELECCIONZOSUDE!contamin
Text41.Text = SELECCIONZOSUDE!observac
If Not SELECCIONZOSUDE.EOF Then
  SELECCIONZOSUDE.MoveNext
  SELECCIONZOSUDE.MovePrevious
End If
consulta = " select * from zonas where idzona = " & Text35.Text & ""
Set res = BASE.OpenRecordset(consulta, dbOpenDynaset)
Combo5.Text = res!descripc
res.Close
consulta = " select * from usosuelo where idusos = " & Text36.Text & ""
Set res = BASE.OpenRecordset(consulta, dbOpenDynaset)
Combo6.Text = res!descripc
res.Close
consulta = " select * from delegaciones where iddelega = " & Text37.Text & ""
Set res = BASE.OpenRecordset(consulta, dbOpenDynaset)

```



```

Combo7.Text = res!nombdele
res.Close
Case 8
If SSFrame9.Visible = False Then
  MsgBox "NO ESTA HABILITADO EL MARCO DE USUARIOS ", 16
  SSCommand1.Enabled = True
  frmmenu.Altas.Enabled = True
Else
  Set SELECCIONUSUARIOS = BASE.OpenRecordset("SEGURIDAD", dbOpenDynaset)
  If (SELECCIONUSUARIOS.EOF And SELECCIONUSUARIOS.BOF) Then
    MsgBox " NO EXISTEN REGISTROS", 16
    Exit Sub
  End If
  SELECCIONUSUARIOS.MoveFirst
  Text45.Text = SELECCIONUSUARIOS!userid
  Text44.Text = SELECCIONUSUARIOS!Password
  Text39.Text = SELECCIONUSUARIOS!seguridad
  Text43.Text = SELECCIONUSUARIOS!nombre
  Text42.Text = SELECCIONUSUARIOS!ultfecha
  consulta = " select * from seguridad where userid = " & Text45.Text & ""
  Set res = BASE.OpenRecordset(consulta, dbOpenDynaset)
  If res!seguridad = "1" Then
    Combo9.Text = "Todo el Sistema"
  End If
  If res!seguridad = "2" Then
    Combo9.Text = "Sin Mantenimiento"
  End If
  If res!seguridad = "3" Then
    Combo9.Text = "Consultas y Reportes"
  End If
  res.Close
  If Not SELECCIONUSUARIOS.EOF Then
    SELECCIONUSUARIOS.MoveNext
    SELECCIONUSUARIOS.MovePrevious
  End If
End If
End Select
End Sub

```

## APÉNDICE D

### Información utilizada para pruebas del sistema

En este apéndice se muestra información proporcionada por la compañía para la prueba de él sistema. La información que a continuación se muestra son datos reales obtenidos por los valuadores para el avalúo de un predio.

**Asunto:** Actualización de avalúo del predio ubicado en el Boulevard Puerto Aéreo No. 63, Colonia Ignacio Zaragoza, Delegación Venustiano Carranza, México D.F.

**Solicitante:** Servicios Metropolitanos S.A. de C.V.

#### Datos del Predio.

- Ubicación: Boulevard Puerto Aéreo No. 63 Col. Ignacio Zaragoza, Del. Venustiano Carranza, Ciudad México, Entidad Federativa D.F.
- Propietario: Departamento del Distrito Federal
- Objeto del avalúo: Determinar el precio mínimo de venta del terreno en estudio
- Calles que forman la manzana: al Norte avenida seis, al Este Boulevard Puerto Aéreo, al Sur Galindo Villa y Puerto Aéreo y al Oeste calle A.
- Superficie: 86.03 m<sup>2</sup>

#### Datos de la zona.

- Uso de suelo: H4
- Clasificación de la zona: Habitacional
- Servicios públicos: Cuenta con todos los servicios como son tomas de agua domiciliaria, drenaje, alcantarillado, banquetas, guarniciones, pavimentos, alumbrado publico, vías de comunicación
- Tipo de construcción dominante: Casas habitación de tipo medio resueltas en uno y dos niveles en general
- Saturación de construcción en la zona: 100%
- Densidad de población en la zona: 100%