



15
UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE INGENIERÍA

FALLA DE ORIGEN

SISTEMA DE CONTROL DE PRESUPUESTOS
BAJO AMBIENTE VISUAL BASIC

T E S I S

Que para obtener el Título de:

Ingeniero en Computación

Presentan:

Pedro Benítez Mejía
Luz María Castañeda de León
Laura Elena Méndez Liceaga
Mónica Orduña Amaro
Omar Zepeda López

Director de tesis: Ing. Adolfo Millán Najera

México, D.F.

1995



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

TESIS CON FALLA DE ORIGEN

DEDICO ESTA TESIS

A MI MADRE

CRISTINA MEJIA DE BENITEZ

Con el cariño y amor que se merece por el apoyo que me ha brindado a través de mi vida y particularmente para que pudiese concluir mis estudios.

A LA MEMORIA DE MI PADRE

PEDRO BENITEZ ALARCON

Por la hermosa herencia de honradez y rectitud que aportó en mi vida.

A MI DIR. DE TESIS

ING. ADOLFO MILLAN NAJERA

Por sus útiles sugerencias y aportaciones.

A MI ESCUELA

Facultad de Ingeniería. Por su esfuerzo constante en la formación de profesionistas.

A MIS AMIGOS

Por significar un motivo que me impulsa a continuar superándome.

PEDRO BENITEZ M.

GRACIAS:

Padre nuestro

*Por darme la vida
en este instante.*

Queridos Padres

*Por estrecharme en su
corazón, aceptandome en él.*

Hermanos

*Por sus consejos, compañía
y ante todo su ejemplo a seguir*

Querida familia

Amigos míos

Universidad y

Humanidad en general

*Que siempre tuvieron una
palabra de aliento y amor,
una enseñanza que dejarme.*

Luz María Castañeda de León

Dedicado ...

A mi mamá:

Por ser la persona a quien más quiero, por su amor, apoyo, confianza, por estar conmigo en cada momento de mi vida, y por tantas cosas bellas que me ha dado. Gracias por todo.

A mi papá:

No sólo le dedicó este trabajo, sino todo mi esfuerzo puesto durante el transcurso de mis estudios como una muestra de mi gran amor por él. Gracias por sus consejos y apoyo, pero sobretodo por su amor.

A todos mis hermanos, en especial a Jessica y Rissela:

Por ser las niñas más lindas que pude tener como hermanas y porque con su compañía me han llenado de amor, ternura y alegría.

A todas las personas que han estado conmigo de alguna u otra forma, a todos mis amigos de la Facultad que compartieron conmigo una de las mejores etapas de mi vida.

A la Facultad de Ingeniería y profesores que hicieron posible cumplir una de mis grandes metas, ser un Ingeniero en Computación.

Gracias a Dios ...

LAURA MENDEZ

GRACIAS

A Dios

Por darme la existencia.

A mi mami

Por todo su amor, apoyo y comprensión que siempre me ha brindado.

A la memoria de mi padre

Por haber sido mi guía y fortaleza, por todo su esfuerzo hecho para que yo pudiera tener una formación universitaria.

A mis hermanos

Toño, Mario, Jorge y Angélica por que siempre están a mi lado.

A Bere

Porque siempre tiene una sonrisa de aliento para mi.

A todos mis amigos y compañeros que compartieron conmigo alegrías y tristezas.

A mi querida Universidad y profesores que me dieron toda su sabiduría.

MONICA

*A mi madre Rosa por su amor,
espíritu de lucha y sacrificio.*

*A mis Hermanas Claudia y Diana
por su amor y compañía, ojalá y
algún día pasen por este camino.*

*A mi esposa Cristina y mi hijo en
camino por darle una razón más a
mi vida*

*A alguien muy especial que desde
mis primeros días guía mi vida y
mi pensamiento.*

A Todos Gracias

*A Julieta Pérez y familia por
esos primeros años que me
inculcaron tenacidad y
disciplina.*

*A cada uno de mis amigos por
todos esos momentos, ojalá y
siempre sigamos juntos.*

*A cada una de las personas
que durante toda mi vida me
aconsejaron y apoyaron, y que
gracias a ellos pude llegar a
este momento.*

Omar Fernando López

INDICE

I INTRODUCCION

I.1	¿Cómo Surge la Necesidad?	3
I.2	El Entorno de la Empresa	3
I.3	Esquema del Problema	4
I.4	Sistemas de Presupuestos en el Mercado	5

II AMBIENTE OPERATIVO

II.1	Antecedentes de los Sistemas Operativos	15
II.2	El Sistema Operativo MS - DOS	15
II.3	El Sistema Operativo OS - 2, como una Alternativa	18
II.3.1	¿Por qué Tenemos que Cambiar de Sistema?	20
II.4	Windows, Historia	22
II.5	Antecedentes de Programación bajo Windows	23
II.5.1	Etapas para la Construcción de una Aplicación en Windows	29
II.5.2	El Compilador de C	30
II.5.3	Editores de Recursos	31
II.5.4	Compilador de Recursos	31
II.5.5	Enlazador	32
II.5.6	MAKE/NMAKE	32
II.5.7	Componentes Básicos de una Aplicación en Windows	33

III VISUAL BASIC

III.1	¿Qué es Visual Basic?	37
III.2	Programación en Visual Basic	38
III.2.1	El Ambiente de Visual Basic	39
III.2.2	Propiedades	40
III.2.3	Controles	41
III.2.4	Variables	47
III.2.5	Menús	49
III.3	Ventajas y Limitantes	49

IV METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

IV.1	Introducción	55
IV.2	Método del Ciclo de Vida	56
IV.2.1	Investigación Preliminar	57
IV.2.2	Determinación de los Requerimientos del Sistema	60
IV.2.2.1	Actividades de la Determinación de Requerimientos	60
IV.2.3	Diseño del Sistema	61
IV.2.4	Desarrollo de Software	62
IV.2.5	Pruebas del Sistema	63
IV.2.6	Implantación y Evaluación	64
IV.3	Método del Desarrollo del Análisis Estructurado	66
IV.3.1	¿Qué es el Análisis Estructurado?	67
IV.3.2	Elementos del Análisis Estructurado	68
IV.3.3	¿Qué es el Diseño Estructurado?	71
IV.3.4	Empleo del Análisis Estructurado con otros Métodos de Desarrollo	72
IV.4	Método del Prototipo de Sistemas	73
IV.4.1	¿Qué es un Prototipo?	73
IV.4.2	Etapas del Método de Prototipos	74
IV.4.3	Identificación de Requerimientos Conocidos	74
IV.4.4	Desarrollo de un Modelo de Trabajo	75
IV.4.5	El Prototipo y el Usuario	77
IV.4.6	Revisión del Prototipo	77
IV.4.7	Repetición del Proceso las veces que sea necesario	78
IV.4.8	Uso de Prototipos	78
IV.4.9	Abandono de la Aplicación	78
IV.4.10	Implantación del Prototipo	79
IV.4.11	Redesarrollo de la Aplicación	80
IV.4.12	Inicio de un Nuevo Prototipo	81
IV.5	Definición de una Metodología Propia de Desarrollo	82

V ANALISIS DEL SISTEMA

V.1	Introducción	93
V.2	Definición del Problema	94
V.2.1	Objetivo	95
V.2.2	Plan de Trabajo	96
V.3	Determinación de los Requerimientos	96
V.3.1	Hardware	101
V.3.2	Software	102
V.3.3	Recursos Humanos	103
V.4	Estudio de Factibilidad	103
V.5	Diseño Conceptual	107

VI DISEÑO DEL SISTEMA

VI.1	Introducción	111
VI.2	Diseño de Pantallas	112
VI.2.1	Fases del Sistema	112
VI.2.2	Diseño de la Pantalla Principal	113
VI.2.3	Pantalla de Catálogo de Proveedores	114
VI.2.4	Pantalla de Catálogo de Usuario	115
VI.2.5	Pantalla de Planeación del Presupuesto	118
VI.2.5	Pantalla de Consulta General de Gasto	122
VI.2.6	Pantalla de Captura de Gasto	122
VI.3	Esquema Pseudo-prototipo (EP)	125
VI.4	Validación del Esquema Pseudo-prototipo	128
VI.5	Ventajas y Desventajas del EP	133
VI.6	Diseño de la Base de Datos	133
VI.6.1	Tablas del Sistema	134
VI.6.2	Modelo Entidad-Relación del Sistema	134
VI.6.3	Diccionario de Datos	135

VII DESARROLLO DEL SISTEMA

VII.1	Introducción	145
VII.2	Entrada al Sistema	147
VII.3	Módulo de Presupuesto	151
VII.4	Módulo de Gasto	159
VII.5	Módulo de Usuarios	164
VII.6	Módulo de Proveedores	169

VIII PRUEBAS, IMPLANTACION Y EVALUACION DEL SISTEMA

VIII.1	Introducción	179
VIII.2	Pruebas del Sistema	180
VIII.2.1	Pruebas de Integración	181
VIII.3	Implantación del Sistema	187

IX CONCLUSIONES

IX.1	Conclusiones	191
------	--------------	-----

APENDICE A

Controles de Visual Basic	201
---------------------------	-----

APENDICE B

Propiedades de Visual Basic	209
-----------------------------	-----

GLOSARIO

Glosario 217

BIBLIOGRAFIA

Bibliografia 229

PROLOGO

El presente trabajo de tesis que lleva como título *Sistema de Control de Presupuestos Bajo Ambiente Visual Basic*, tiene como objetivo llevar a la práctica los conocimientos adquiridos durante la carrera aplicándolos a un problema real, a través del análisis, diseño y desarrollo de un sistema presupuestal que de cumplimiento a los requerimientos de una compañía de seguros, proporcionando un medio para poder identificar y evaluar eficazmente las actividades de la compañía, aprovechando sus recursos disponibles bajo la plataforma de ambiente Windows, utilizando la herramienta gráfica Visual Basic.

Podemos mencionar que para dar cumplimiento, se tuvo que definir como base los recursos con los que la compañía contaba, permitiendo de esta forma utilizar una herramienta gráfica que diera la mejor solución para el desarrollo del sistema, esta herramienta es Visual Basic.

Visual Basic es una de las nuevas generaciones de software que corre bajo ambiente Windows, es también uno de los más poderosos que existen en el mercado. Desde sus inicios ha ganado bien merecida fama y es ahora ampliamente reconocido como uno de los más prácticos, debido a su amplia variedad de herramientas. Visual Basic se ha convertido en un lenguaje con mayor perspectiva en cuestión de negocios y en las diferentes áreas de aplicación de una empresa.

En los siguientes capítulos se pretende dar a conocer toda la información concerniente al desarrollo del sistema, el cual nos podrá llevar a un mejor entendimiento, partiendo de conocimientos previos tales como:

Capítulo I **Introducción.** Se da una explicación del origen del problema, los alcances del sistema y qué esperan los usuarios del mismo. Asimismo, se analizan los diferentes sistemas de presupuesto que existen en el mercado haciendo un análisis de sus alcances y limitantes.

Capítulo II **Ambiente Operativo.** Se tratan las diferentes alternativas de los principales sistemas operativos, estudiando las ventajas y desventajas con las que nos encontraríamos al hacer uso de ellos durante el desarrollo del sistema.

Capítulo III **Visual Basic.** Es el punto donde estudiamos su filosofía, así como su forma de programación, es donde se ve claramente que en vez del laborioso proceso para crear una interfaz de usuario necesaria en otros lenguajes de programación, en Visual Basic esta interfaz forma parte del mismo, por lo que uno sólo tiene que dibujar los gráficos en pantalla.

Capítulo IV **Metodologías para el Desarrollo de Sistemas.** Es una información básica, ya que la elaboración de un sistema de información es un proceso formado por diferentes etapas, y es en esta parte donde se explora los diferentes enfoques de desarrollo, abordando sus características y condiciones en las que se obtenga el mayor beneficio. Así como el de proponer una metodología propia de desarrollo.

Estos primeros cuatro capítulos dan como resultado una base sólida y confiable para la realización del sistema, la cual es presentada dentro de los capítulos siguientes: Análisis, Diseño y Desarrollo los cuales conforman la parte primordial del fin a alcanzar.

Capítulo V **Análisis del Sistema.** Se realiza un estudio que permite evaluar los requerimientos, sus características, posibilidades y limitaciones que tendría el sistema para su realización. Obteniéndose de esta forma un panorama de la factibilidad técnica, económica y operacional.

Capítulo VI **Diseño del Sistema.** Se conjuntan dos puntos importantes, el primero la información obtenida a partir del análisis del sistema, y el segundo, la interacción constante con el usuario. Obteniéndose con esto, el Esquema Pseudo-prototipo, el cual dará inicio a la realización del mismo.

Capítulo VII **Desarrollo del Sistema.** Se da la descripción paso a paso de los procedimientos realizados en cada uno de los módulos con el que cuenta el sistema, los cuales darán solución a las necesidades planteadas en el análisis.

Capítulo VIII **Pruebas e Implantación del Sistema.** Se realiza la descripción de cómo se lleva a cabo la instalación y pruebas que verifiquen su correcto desempeño, así como comentarios recibidos sobre las ventajas del funcionamiento del sistema.

Capítulo IX **Conclusiones.** En este punto se dan a conocer todos los aspectos a los que se llegó y que consideramos importantes durante el desarrollo de la presente tesis.

Por último para una mayor información sobre la operación del sistema se realizó un Manual de Usuario que se encuentra como un anexo a esta tesis.

CAPITULO I

INTRODUCCION

I.1 ¿COMO SURGE LA NECESIDAD?

Al estar trabajando en una empresa surge la necesidad de hacer una planeación flexible del presupuesto y llevar un control fácil y estricto del gasto. Pero uno voltea al mercado y que es lo que ve, sistemas que manejan el presupuesto muy bien pero de una manera ya sea técnica o muy poco flexible y en el peor de los casos no se adaptan a la idea de la necesidad que tienen los directivos del manejo de su presupuesto.

Por lo cual se requiere de un sistema:

- Que sea amigable y fácil de usar.
- Que la tecnología utilizada sea de punta y con portabilidad.
- Que su precio no sea muy alto.
- Que realice un buen trabajo y con tiempo de respuesta satisfactorio.

I.2 EL ENTORNO DE LA EMPRESA

Normalmente dentro de la empresa al final de un año se planea lo que será el presupuesto del año siguiente, es decir, los directivos se plantean un tope del gasto. Pero estos gastos son muy diversos y lo son más si los vemos desde las diferentes direcciones dentro de la misma empresa. Dado que cada área maneja conceptos propios, esto es, mientras que casi todas las áreas manejan un presupuesto de papelería, normalmente la Dirección de Ventas es la que se encarga de promover a la empresa y manejar los gastos de representación, así como Sistemas será la Dirección que normalmente se encargue de los gastos de equipo de cómputo y compras de software.

INTRODUCCION

Además el control de lo que podemos gastar con lo que hemos gastado se vuelve más complejo, si de los gastos que hemos hecho nos interesa saber que departamento de la dirección lo realizó, y cuál es su proporción con respecto a las demás gerencias o departamentos de la empresa.

1.3 ESQUEMA DEL PROBLEMA

Pero qué pasa si una de las direcciones está organizada con un director, subdirectores, gerentes, etc., y a su director no le interesa saber qué departamento específico realizó el gasto, sino un detalle de cada concepto, por ejemplo si en la Dirección de Recursos Humanos le interesa saber cuánto se ha gastado en capacitación del personal, pero además el desea saber ¿cuánto ha sido el gasto de cursos para la Dirección de Ventas?, ¿cuánto para la Dirección Técnica?, ¿cuánto para la Dirección de Sistemas?, etc. Si seguimos revisando cada dirección por separado nos daremos cuenta que cada una se maneja de una manera muy específica, compartiendo el mismo problema, **¿cómo poder planear su presupuesto?**, de una forma fácil, flexible, y algo muy importante, que lo puedan pensar hasta el detalle que ellos deseen, y posteriormente poder capturar su gasto como se les vaya presentando, y que un directivo de una manera automática vaya monitoreando las erogaciones y con esto llevar un buen control del presupuesto y del gasto. Todo esto sin tener que introducirnos a terminos técnicos que maneja normalmente el Area de Contabilidad.

I.4 SISTEMAS DE PRESUPUESTOS EN EL MERCADO

Cuando da inicio un proyecto, se debe tener en cuenta una secuencia de pasos que permitan que éste se realice con la mayor eficiencia, para lo cual se debe de obtener información que permita poder entender, interpretar, analizar y principalmente dar solución a la problemática existente. La información se obtiene a través de diferentes formas como son: encuestas, entrevistas e investigaciones principalmente.

En el mundo de la computación una vez que se ha analizado el problema se empieza a plantear una serie de alternativas que le den solución, una de las alternativas que suele ser primordial es investigar si existe en el mercado algún sistema que pueda darle solución a éste. Para nuestro problema se realizó una investigación referente a los sistemas de presupuestos que existen en el mercado.

Antes de empezar a mencionar el software que existe en el mercado debemos hacer notar que existe un sinnúmero de sistemas que cubren necesidades de contabilidad, administración, inventarios, nóminas, control bancario, producción, etc., y que son contados los sistemas que existen de control de presupuestos, la mayoría de los sistemas principalmente de contabilidad o administración ofrecen un módulo dedicado a presupuestos, el cual en la mayoría de los casos tiene una dependencia directa de la información que contiene el sistema en sí.

A continuación se muestra la información recopilada de los diferentes sistemas que existen en el mercado los cuales hacen referencia al presupuesto.

ALTERNATIVAS EN EL MERCADO

Concepto	Presupuesto: Es un sistema o un módulo.		Ambientes de trabajo		Facilidad de uso		Flexibilidad (adaptación por medio de programación)	Seguridad (por medio de clave)	Trabajo en línea	Ayuda y Consulta	Reportes
	Sistema	Módulo	Mono- usuario	Red	Menús o Pant.	Teclas					
ASPEL-COI		X	X	X	X	X	SI	SI	SI	S/I	S/I
ADMIN		X			X	X	SI	S/I	SI	SI	con Flex
MICROSHIP	X	X					NO	SI	S/I	SI	con Flex
SOFTLAND	X						NO	NO	S/I	Ayuda	S/I

Tabla 1.1

S/I Sin Información

REQUERIMIENTOS

	Tipo de computadora	Disco Duro	Versión del Sistema Operativo	Memoria en RAM	Impresora
ASPEL-COI	XT/AT/PS-2	SI	3.0 ó Sup	512 MB	-cualquiera
ADMIN	XT	SI	3.1 ó Sup	640 KB	-cualquiera
MICROSHIP	PC, PS	SI	3.0 ó Sup	640 KB	-cualquiera
SOFTLAND	PC, PS	SI	3.0 ó Sup	640 KB	capacidad de imprimir 132 columnas

Tabla 1.2

SOFTLAND

- Período tributario variable.
- Maneja pólizas de todo el año en línea.
- Saldo de apertura ingresados en cualquier momento.
- Consulta por pantalla de saldos y movimientos de las cuentas.
- Impresión de Pólizas.
- Manejo de catálogo de cuentas paralelo para las cuentas de la C.N.B.
- Conexión directa desde sistemas de Proveedores, Clientes.

ASPE- COI

- Catálogo de cuentas definible por la empresa.
- Control de Centros de costos (departamentalización).
- Consultas y reportes de auxiliares mensuales y anuales.
- Consolidación de la contabilidad de varias empresas, póliza de cierre anual.
- Presupuestos para cualquier tipo de cuenta.

INTRODUCCION

Como se ha observado los diversos sistemas de control de presupuestos existentes en el mercado tienen una manera de controlar el presupuesto muy técnica, general y definitivamente no están vistos para ser manejados a nivel directivo.

A todo esto debemos sumar que el ejecutivo se encargará de planear el presupuesto, pero definitivamente no será él mismo el que se encargue de alimentar el gasto, en el mejor de los casos le delegará esta responsabilidad a su subdirector, pero finalmente la que capture las facturas será la secretaria.

Y seamos sinceros ¿vamos a capacitar a la secretaria de un director o subdirector en el manejo de un paquete como COI por ejemplo?, que es un sistema diseñado para capturar, procesar y mantener actualizada la información contable principalmente, conteniendo en una de sus funciones el módulo de presupuestos con una orientación técnica, por lo que el sistema tendría que ser adecuado a las necesidades e información que requieran las diferentes direcciones, cabe hacer resaltar que contiene funciones que no son de utilidad y cuyo manejo no les corresponde analizar a los directores por lo que se estaría utilizando un sistema sobrado de funciones y cuyo costo es entonces no justificable. O un paquete como Project que es un administrador de proyectos bajo ambiente Windows.

Los sistemas hoy en día se deben de ir simplificando en su uso, con lo anterior no queremos decir que los usuarios no han ido evolucionando, por el contrario, debido a que cada momento que pasa se vuelven más críticos y exigentes, las gentes de sistemas nos vemos más exigidos. En el pasado han quedado los días en que los usuarios veían al personal de sistemas como un ser venido de otro planeta, el cual se introduce a una máquina y hace maravillas.

INTRODUCCION

Son por estas razones y de acuerdo a las necesidades de la empresa, que se decidió en realizar un Sistema de Control de Presupuestos hecho a la medida, este sistema deberá tener como características principales las siguientes:

1. Debe ser sencillo de usar, con la característica de que una persona con los conocimientos básicos tanto de cómputo, como de contabilidad o administración, lo pueda operar.
2. La interfaz con el usuario debe ser además de sencilla de usar, dar una impresión de modernidad y principalmente de calidad ya que serán ejecutivos los que lo emplearán.
3. Definitivamente el software debe ser barato, además se debe considerar que la empresa está en constante modernización del equipo de cómputo, lo cual nos da una ventaja, ya que las limitaciones de hardware no serán parte de nuestra historia.
4. También debemos tomar en cuenta que la empresa pertenece a un Grupo Financiero y que ésta tiene tendencias de tecnología que se deben adoptar.

CAPITULO II

AMBIENTE OPERATIVO

II.1 ANTECEDENTES DE LOS SISTEMAS OPERATIVOS

Como una definición tradicional de sistema operativo tenemos que es un conjunto de programas que administran los recursos de la máquina, el principal recurso que administra, es el hardware de la computadora, es decir el procesador, la memoria, los dispositivos de entrada-salida, los dispositivos de comunicación y datos.

II.2 EL SISTEMA OPERATIVO MS-DOS

Al hablar de los Sistemas Operativos para PC's no podemos dejar de mencionar al DOS. Este Sistema Operativo se presenta en 1981¹ con características para ejecutarse en una modesta XT, con una memoria principal de 256K y un procesador 8088.

El MS-DOS realmente gana importancia cuando IBM lo adopta como el sistema operativo para la computadora personal IBM. Siendo el primer sistema operativo para microcomputadora de 16 bits, el MS-DOS nace con una necesidad de 12K de memoria¹. Esta organizado en 3 archivos. El IBMBIO.COM que maneja el disco y el sistema de entrada-salida de caracteres. El IBMDOS.COM que contiene el administrador de archivos de disco y el manejador de caracteres de entrada-salida. El archivo COMMAND.COM que contiene el procesador de comandos externos. Esta primera versión muestra compatibilidad con CP/M; lo cual era muy importante en el mercado de la época.

¹ Harvey M. Deitel. An Introduction to OS. Addison-Wesley Publishing Company. 1990. Pag. 633.

AMBIENTE OPERATIVO

La empresa Microsoft en su primera versión del sistema operativo MS-DOS incluye independencia en los dispositivos de entrada-salida al tratarlos como archivos (CON para la consola, teclado para la entrada y la pantalla para la salida; PRN para la impresora, y AUX para los puertos seriales auxiliares). Esta independencia de dispositivos hace más fácil a los usuarios escribir programas.

Mientras el CP/M de la época manejaba registros físicos y lógicos de 128 bytes¹, los discos de la IBM tenían sectores de 512 bytes. Con la versión 2.0 se podía manejar discos duros de || 10 Mb || algo que para la época era mucho.

Pero para 1994 qué nos ofrece MS-DOS. Cuando ya todo mundo ha comprendido la necesidad de poder manejar varias tareas al mismo tiempo, cuando la GUI (interfaz gráfica de usuario), es el pan nuestro de cada día y si nos basamos en la definición de que un sistema operativo es un conjunto de programas que nos administran el hardware, la pregunta es ¿MS-DOS realiza esto eficientemente?

La versión 6.0 del sistema operativo MS-DOS nos ofrece de entrada la barrera de 640 KB que ofrecían sus antecesores², ya que no corre programas en modo protegido y sólo los corre en modo real, la mayoría de las aplicaciones de DOS, usan el sistema operativo básicamente para las entradas y salidas y van directamente al hardware, si alguna aplicación intenta escribir en el área del DOS lo logrará ya que no hay nada que se lo impida, lo que sigue haciendo a MS-DOS un sistema operativo muy inestable; pero DOS 6.0 nos ofrece hacernos la vida más fácil al incluir una utilería llamada MEMMAKER que nos da la configuración "idónea" para cargar nuestros TSR's en memoria superior.³

¹ Harvey M. Deitel. An Introduction to OS. Addison-Wesley Publishing Company. 1990. Pag. 633.

² PCMagazine (en español). Vol. 4. Num. 2. America S. A. 1993. Pag. 59.

³ PCWord. Octubre 1993. PCWord Communications. Pag. 42.

AMBIENTE OPERATIVO

Y las demás mejoras son una serie de utilerías¹ que ya estaban disponibles antes en el mercado con otros desarrolladores, algunas de éstas son:

- El MSBACKUP permite respaldar información en discos flexibles, obteniéndose de esta forma un backup compactado.
- El Microsoft Anti-Virus ofrece protección contra cientos de virus conocidos al momento.
- Undelete ofrece tres niveles de protección para archivos borrados.
- DoubleSpace ofrece duplicar el espacio en disco duro, desde el sistema operativo.

Al ver todas estas características del DOS podemos pensar que es un sistema operativo que brinda muchas libertades ya que no restringe casi nada, pero por el contrario no ofrece: ninguna ayuda para recuperarse de errores, no permite trabajar en multitarea y aún más, no ofrece una interfaz amigable con el usuario, algo que un usuario de nuestro tiempo exige mucho y es él, el que tiene que implementar todo vía programa.

Pero a contraparte de esto tenemos que en el mercado hay millones de PC's que corren bajo ambiente MS-DOS lo cual es algo importante de considerar cuando se piensa en qué plataforma será la idónea para desarrollar un sistema.

¹ PCMagazine (en español). Vol. 4. Num. 7. America S. A. 1993. Pag. 62.

AMBIENTE OPERATIVO

II.3 EL SISTEMA OPERATIVO OS/2, COMO UNA ALTERNATIVA

Este sistema operativo nace de un trabajo conjunto de IBM y MICROSOFT en 1985¹, OS-2 se ve como el sistema operativo de los 90's, el sustituto de MS-DOS, ya que provee multitarea, lleva a las aplicaciones a tener un acceso directo a periféricos de un amplio ancho de banda y provee un ambiente protegido para asegurar la estabilidad del sistema.

La versión original OS/2 1.0¹ utilizaba las características del 80286 y corría en el 80386 también como en el 80286 pero no explotaba sus características. Esta versión incluía una interfaz orientada a texto como el MS-DOS. Pero la versión 1.1 del OS/2 fue liberada e incluyó una interfaz gráfica llamada Administrador de Presentaciones (Presentation Manager PM)¹.

OS/2 protege aplicaciones una de otra y de daños al área del sistema operativo. La protección es especialmente crítica en redes de computadora en la cual amplio número de usuarios interactúan a través de la red.

Mientras en el MS-DOS, un ambiente de programa usa el sistema completo, en OS/2 muchos procesos comparten la máquina. Los procesos compiten por recursos y pueden solicitar acceso simultáneamente a recursos particulares. OS/2 aísla cada programa del otro, así que ellos aparecen en un ambiente consistente cada vez que se ejecutan.

En un ambiente unitarea, un programa incluye todas las capacidades necesarias y que no son proporcionadas directamente por el sistema operativo. Esto tiende a acrecentar el tamaño de los programas y hacerlo por ende más complejos. En un ambiente multitarea OS/2, lo mejor es hacer

¹ Harvey M. Deitel. An Introduction to OS. Addison-Wesley Publishing Company. 1990. Pag. 785.

programas pequeños, en el que cada uno realice una función o un conjunto de funciones bien, y entonces tener estos programas intercomunicados para completar las tareas de usuario.

Los patrones en los sistemas operativos son tan fundamentales e igualmente difíciles de cambiar. Un Sistema Operativo provee una plataforma para aplicaciones en la forma de una interfaz de programa de aplicación (API). El API es el lenguaje mediante el cual las aplicaciones se comunican con el usuario, con el sistema de archivos y la memoria del sistema.

La versión de OS/2 2.0¹ corre bajo 32 bits, es protegido y puede recuperar el procesador en caso de fallas, pero el gran problema es que casi no hay aplicaciones bajo su ambiente disponible, lo que lo hace poco atractivo tanto para los usuarios, como para los desarrolladores ya que no desean arriesgarse hasta que no vean al sistema consolidado en el mercado.

La GUI de esta versión de OS/2 el Workplace Shell (WPS), tiene un diseño basado en objetos. No tiene barra superior de menú y aunque el WPS sustituye a la interfaz de usuario en OS/2 1.x, los programadores todavía pueden seguir usando el PM API.

Pero sigue la gran desventaja de OS/2, que carece de manejadores para muchos periféricos populares y tanto la instalación como la configuración requieren de un antecedente técnico, lo cual lo hace poco atractivo al usuario común.

Las aplicaciones están tan estrechamente ligadas al API del sistema operativo, que un sistema operativo muy exitoso, puede en realidad ser un

¹ PCMagazine (en español), Vol. 4. Num. 2 America S. A. 1993. Pag. 50.

AMBIENTE OPERATIVO

impedimento al progreso de la industria. Los sistemas operativos sobreviven por mucho más tiempo de los que justifican sus méritos técnicos. Por ejemplo, ¿Qué otro sistema operativo además del DOS se ha estado usando continuamente desde 1981?

Un nuevo sistema operativo requiere un fuerte compromiso de los usuarios y programadores. Pero los usuarios no comprarán un nuevo sistema operativo a menos que existan aplicaciones disponibles para el mismo, y los programadores no escribirán las aplicaciones sin tener la garantía de un mercado. Un sistema operativo exitoso, uno que sea excelente técnicamente y que gane una buena porción del mercado, es raro. En el mundo de los sistemas operativos, se proponen muchos y se seleccionan pocos.

II.3.1 ¿Por qué Tenemos que Cambiar de Sistema?

Sencillamente porque DOS no puede apoyar las aplicaciones sofisticadas de hoy en día. DOS, que ya tiene más de diez años, fue desarrollado para ser ejecutado en máquinas 8088 de 16 bits en modo real. Como DOS sólo opera en el modo real, el sistema no está protegido; los programas pueden escribir por encima de cualquier área de memoria que DOS utiliza y de este modo trabar el sistema. DOS sólo puede ejecutar una tarea a la vez. La interfaz del usuario bajo DOS es poco atractiva y difícil de usar ya que está basada en comandos.

El hecho de que DOS force a los programadores a controlar el hardware para tantas operaciones básicas, es el obstáculo más grande para abandonarlo. Debido a las miles de aplicaciones que los usuarios de PCs utilizan, no es aceptable reemplazarlo con algo que no pueda ejecutar aplicaciones de DOS,

y como sus aplicaciones se ocupan de tantas funciones que debieran ser la responsabilidad del sistema operativo, asegurar su compatibilidad no es fácil.

Windows y OS/2 ofrecen dos soluciones diferentes para eliminar las limitaciones de DOS. Windows ofrece un API para las características que no existen en el API de DOS. Puede tener Windows en su sistema y ejecutar a DOS en una caja de DOS, o puede salir de Windows y regresar a DOS; en realidad no hay que escoger entre los dos.

Por otra parte, OS/2 es un sistema operativo completamente nuevo que reemplaza a DOS. Puede usar la incómoda opción de doble arranque y mantener a DOS o a OS/2 en su computadora. Arrancar de nuevo para cargar a DOS es inconveniente, y puede experimentar complicaciones desagradables cuando intente operar varios sistemas operativos en una computadora.

La superioridad técnica de OS/2 sobre DOS y Windows según los expertos¹ es casi indudable. Pero los patrones no nacen de la excelencia técnica. Los patrones son el resultado de la excelencia técnica, las tendencias del mercado y de la fortaleza política.

Aquí el mercado (los usuarios y los programadores) habló claramente: Windows, sí. OS/2, no. La actividad prematura de algunos programadores que escribieron aplicaciones para OS/2 decayó, a la vez que todo el mundo se concentró en Windows.

¹ PCMagazine (en español), Vol. 4. Num. 2. América S. A. 1993. Pag. 50,56.

AMBIENTE OPERATIVO

II.4 WINDOWS, HISTORIA

Microsoft empieza a trabajar con Windows en 1983, solo dos años después que la PC aparece y por ende DOS. Sin embargo, la versión original, Windows 1.01, no aparece hasta 1985. Esta versión fue liberada en la máquina estándar de la época: una IBM PC con dos manejadores de disco de 360K, 256K de memoria de procesador. El display era automáticamente llenado; esto es, la versión original de Windows estuvo diseñada a utilizar la ventana completa. Se veía en dos dimensiones y lejos de ser impresionante.

La versión más sobresaliente vino dos años después. Por primera vez Windows pudo sobreponer en la pantalla. Sin embargo, Windows 2 podía correr sólo en modo real 80x86, lo cual significaba que estaba limitado a un megabyte de memoria de procesador. Por aquel tiempo Windows cambia hacia Windows 286 y Windows 386 para aprovechar las capacidades del (nuevo) chip 80386.

Finalmente, en mayo de 1990, Microsoft introduce Windows 3.0. La vista y el sentir de Windows 3.0 fue un gran progreso de sus antecesores, y su característica proporcional, hace a la vista, las pantallas más refinadas. La versión 3.0 tiene también mejor soporte para programas de DOS, por lo cual muchos usuarios usan ahora Windows como el primer ambiente operativo de la PC.

El MS-DOS ejecutivo de las primeras versiones fue reemplazado por una trinidad de ventanas que administra Windows: el administrador de programas, la lista de tareas, y el administrador de archivos. Desde el punto de vista de programación una de las más importantes características de Windows 3+ es que puede soportar memoria extendida: hasta 16 Mbytes de RAM. Y con su

modo 386 extendido, Windows usa un constructor de memoria virtual de la 80/3/486 esto da a los programadores acceso hasta 4 veces la cantidad de memoria actualmente instalada. En una máquina de hasta 16 MBytes, Windows puede proveer hasta 64 MBytes. El remover las restricciones de memoria había sido siempre una de las ventajas de OS/2, pero ahora la mayoría de los programadores voltean a Windows.

Windows ofrece a los programadores un completo API que incluye llamadas para la Interfaz estándar con el usuario, multitareas, administración de memoria e independencia de dispositivos.

II.5 ANTECEDENTES DE PROGRAMACION BAJO WINDOWS

Una aplicación típica bajo Windows presenta todas las opciones posibles (en forma de objetos visuales) en la pantalla para ser seleccionadas por los usuarios. Por este camino, Windows representa un nuevo tipo de programación orientada a eventos. Es decir, el programador no es completamente responsable del flujo del programa mientras que el usuario si lo es. Los usuarios seleccionan entre todas las opciones presentadas, y ésta es tomada por el programa para responder correctamente. Los programadores no pueden escribir los programas asumiendo que el usuario pulsará en una misma secuencia las opciones.

En cambio, el programador tiene que escribir código separadamente para cada botón. Esto es en todos los casos en general. El código es necesariamente dividido dentro de pequeñas secciones, una sección para una clase de evento. Por ejemplo, se puede adicionar una caja de texto a la ventana, en la cual se

AMBIENTE OPERATIVO

pueda escribir un mensaje para que sea desplegado cuando el usuario haga un clic al botón, el programa puede ser semejante a éste:

```
Sub ClickMe_Click()  
    Message.text='Hola'  
End Sub
```

Este código es específicamente diseñado para manejar un tipo de evento. Además de ser event-driven, la programación Windows es también programación orientada a objetos. Esto es fácil de ver en la pantalla: justamente recoger un objeto tal como un icono o paint-brush y moverlos por todos lados. Estos corresponden de cerca a programas llamados orientados a objetos. Este tipo de programación rompe un programa dentro de objetos discretos, cada uno tiene que tener un código propio y datos asociados con este. En este camino, cada objeto puede ser un poco independiente de los otros.

La programación en Windows ofrece tres ventajas tanto a usuarios como a los programadores: interfaz gráfico de usuarios, multitarea e independencia del hardware.

Los programas se identifican mediante barras de títulos, y se accede a muchas de las funciones básicas de manipulación de archivos mediante menús sin más que apuntar y hacer clic con el mouse. La mayoría de los programas de Windows poseen un interfaz de teclado y otro de mouse. Si bien la mayoría de las funciones de los programas para Windows se pueden controlar desde el teclado.

AMBIENTE OPERATIVO

Dado que todos los programas para Windows tienen la misma apariencia, los programadores sólo tienen que utilizar las subrutinas construidas directamente dentro de Windows para generar menús y cuadros de diálogo. Todos los menús poseen la misma interfaz de teclado y de mouse, ya que Windows, y no el programa de aplicación, gestiona el trabajo.

Una aplicación escrita para Windows debe ser diseñada de modo que incluya controladores (drivers) para todos los posibles dispositivos.

Gran parte de la funcionalidad de Windows la ofrecen las bibliotecas de enlace dinámico, llamadas DLL (Dynamic link Libraries), que mejoran el sistema operativo base ofreciendo un potente y flexible interfaz gráfico de usuario. Las bibliotecas de enlace dinámico contienen funciones predefinidas que se unen a un programa de aplicación cuando éste se carga, en lugar de hacerlo cuando se genera el archivo .EXE.

Las bibliotecas de funciones evitan al programador el tener que volver a crear un nuevo procedimiento para una operación común, como leer un carácter o dar formato a la salida. Los programadores pueden construir fácilmente sus propias bibliotecas para incluir características y posibilidades adicionales, tales como cambiar una fuente o justificar texto.

Junto con las funciones, las bibliotecas también pueden codificar datos e incluso incorporar recursos gráficos, como formas de cursores o matrices de puntos.

Los programas de Windows utilizan un formato de archivos .EXE, denominado nuevo formato de ejecutable. Este nuevo formato incluye una cabecera de "nuevo estilo".

AMBIENTE OPERATIVO

Entre los módulos de la biblioteca de Windows se encuentran: KERNEL, USER y GDI, que contienen rutinas que ayudan a los programas que se ejecutan con Windows a realizar diversas tareas, tales como enviar y recibir mensajes. Estos módulos de biblioteca proporcionan funciones denominadas exportaciones que pueden ser llamadas desde el programa de aplicación o desde otros módulos de biblioteca. El nuevo formato de ejecutables identifica dichas funciones exportadas mediante un nombre y un número de orden. El nuevo formato de ejecutable incluye una sección de tabla de acceso, que indica la dirección de cada una de las funciones exportadas dentro del módulo.

A la inversa, desde el punto de vista del programa de aplicación, las funciones de la biblioteca que utilizan un programa se conocen como importaciones. Casi todos los programas de Windows contienen al menos una función exportada, o una función a la que se llama desde fuera del programa. Esta función de ventana, generalmente de uno de los módulos de la biblioteca, es la que recibe mensajes de Windows.

El nuevo formato de ejecutables ofrece también información adicional sobre cada uno de los segmentos de código y datos de un programa o de una biblioteca. Normalmente, los segmentos de código se marcan como "móviles" y "descartables", mientras que los segmentos de datos se marcan sólo como "movibles". Esto permite a Windows mover segmentos de código y datos en memoria, e incluso descartar segmentos de código si se necesita memoria adicional. Si, más tarde, se necesita un segmento de código descartado, puede volver a cargarse fácilmente a partir del archivo .EXE original. Windows posee una categoría denominada "cargar en la llamada". Con ella se define a un segmento de código de un programa o de una biblioteca que no se cargará

en memoria a meros que se realice una llamada a una función de dicho segmento de código desde otro segmento de código. Esta técnica de gestión de memoria permite a Windows ejecutar simultáneamente varios programas en un espacio de memoria que, normalmente, sólo sería suficiente para un único programa.

El lenguaje de programación más indicado para crear aplicaciones de Windows es C/C++. Las aplicaciones de Windows se pueden desarrollar también en Pascal y en lenguaje ensamblador.

Los conceptos y la terminología de Windows se pueden dividir en dos categorías principales: aquellas características que son visibles, como pueden ser los menús y los iconos, y las operaciones ocultas, como los mensajes y el acceso a funciones.

Una ventana en Windows es el interfaz visual entre el usuario y la aplicación que genera la ventana. Para la aplicación, la ventana es una zona rectangular de la pantalla que está bajo control de la aplicación. La aplicación crea y controla todo lo referente a la ventana, incluyendo su tamaño.

El papel que representa la ventana en la representación de la subestructura fundamental del sistema de Windows es muy importante.

Los diez componentes fundamentales de una ventana son: borde, barra de título, cuadro de control, menú del sistema (menú control), cuadro minimizar, cuadro maximizar, barra de desplazamiento vertical, barra de desplazamiento horizontal, barra de menú y área de trabajo. Todos estos componentes ayudan a definir la apariencia de una aplicación.

AMBIENTE OPERATIVO

Las ventanas que se parecen y se comportan de forma similar se dice que son de la misma "clase de ventanas". Las ventanas creadas por el programador pueden tener diversas características (diferentes tamaños, estar situadas en diferentes lugares de la pantalla, etc.).

Todas las ventanas que el programador crea, deben estar basadas en una clase de ventanas.

La programación en Windows es programación orientada a objetos. En la programación orientada a objetos, un objeto es un tipo de dato abstracto que consta de una estructura de datos y de varias funciones que actúan sobre dicha estructura de datos.

Un objeto gráfico es una colección de datos que se pueden manipular como una única entidad y que se presenta al usuario como parte del interfaz visual. En particular, un objeto gráfico representa tanto a los datos como a la presentación de los mismos. Algunos ejemplos de objetos gráficos pueden ser los menús, las barras de título, los cuadros de control y las barras de desplazamiento.

Los objetos gráficos como iconos, cursores, puntos de inserción, cuadros de mensajes, cuadros de diálogo, fuentes, matrices de puntos, los lápices y los pinceles, son ejemplos de recursos. Un recurso representa los datos que se incluyen en el archivo .EXE de un programa, si bien, técnicamente hablando, no residen en el segmento de datos normal de un programa.

Habitualmente, una aplicación define sus recursos como de sólo lectura y descartables. Esto posibilita que Windows pueda desechar un recurso cuando se necesite más memoria.

Windows ofrece cerca de 1500 funciones al creador de aplicaciones. Algunos ejemplos de estas funciones son: DispatchMessage, PostMessage, RegisterWindowsMessage y SetActiveWindows. La interacción con estas funciones se realiza mediante una llamada lejana entre segmentos.

Existe un archivo de cabecera de gran importancia para Windows llamado WINDOWS.H. Este archivo contiene cerca de 1200 declaraciones de constantes, las declaraciones typedef, y cerca de 500 declaraciones de funciones de Windows.

En Windows los nombres de variables comienzan por una letra o letras en minúscula que describen el tipo de dato de la variable. A este prefijo del nombre de la variable le sigue el nombre de la variable, que se representa mediante una combinación significativa de letras en mayúsculas y minúsculas. Este enfoque hace posible que cada variable adjunte un mnemónico que representa su tipo de dato.

II.5.1 Etapas para la Construcción de una Aplicación en Windows

1. Crear WinMain y las funciones de ventanas asociadas en C o C++.
2. Crear las descripciones de menús, cuadros de diálogo y de otros recursos adicionales; situarlas en un archivo guión de recursos.
3. (Opcional). Usar las herramientas de Windows para crear cursores, matrices de puntos e iconos propios.
4. (Opcional). Usar el Editor de diálogos, que se suministra con las herramientas de Windows, para crear cuadros de diálogo.

AMBIENTE OPERATIVO

5. Crear todas las definiciones de los módulos y situarlas en el archivo de definición de módulos.
6. Compilar y enlazar todo el código fuente en lenguaje C/C++.
7. Compilar el archivo guión de recursos y añadirlo al archivo ejecutable.

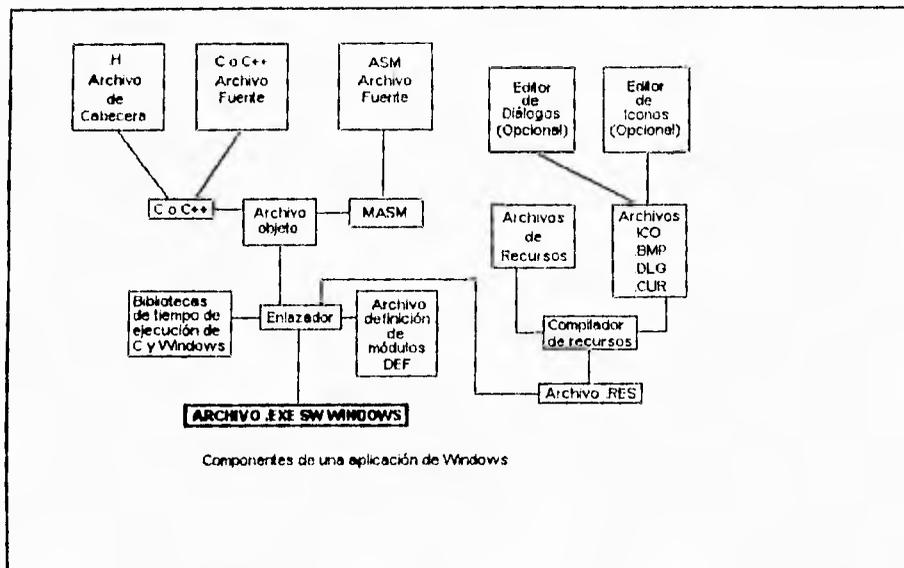


Figura 2.1

Siete pasos necesarios para construir una aplicación de Windows

En la creación de una aplicación real para Windows se requiere el uso de varias herramientas de desarrollo. A continuación se describirán brevemente algunas de ellas.

II.5.2 El Compilador de C

Cuando se compila un programa estándar en C/C++, se tienen dos opciones obligatorias para Windows. Estas opciones son -Gw y -Zp. La primera opción,

-Gw, añade a cada función el código inicial y final típico de Windows; -Zp hace que el compilador de C/C++ empaque todas las estructuras. Esto garantizará que las estructuras utilizadas en el programa sean del mismo tamaño que las correspondientes estructuras usadas por Windows.

II.5.3 Editores de Recursos

En las herramientas de Windows se incluyen tres editores de recursos: Editor de iconos, Editor de diálogos y Editor de fuentes. El Editor de iconos permite la definición rápida de iconos, cursores y matrices de puntos. El Editor de fuentes le ofrece un método cómodo para crear sus propias fuentes. Y el Editor de diálogos simplifica la creación de descripciones de cuadros de diálogo.

II.5.4 Compilador de Recursos

Las aplicaciones de Windows con frecuencia utilizan sus propios recursos, tales como cuadros de diálogos, menús e iconos. Todos estos recursos deben estar predefinidos en un archivo denominado archivo guión de recursos. Este archivo se compila con el compilador de recursos, y la información adicional se añade al archivo .EXE final de la aplicación. Esto permite a Windows cargar y utilizar los recursos a partir del archivo ejecutable.

AMBIENTE OPERATIVO

II.5.5 Enlazador

Todas las aplicaciones de Windows se pueden enlazar con el enlazador de C/C++. Cuando se enlaza una aplicación de Windows, el enlazador necesita un archivo de definición de módulos. El archivo de definición de módulos especifica el nombre de la aplicación y contiene los nombres de los archivos de todas las funciones de retorno de la aplicación.

II.5.6 MAKE/NMAKE

El programa MAKE/NMAKE ofrece un método eficiente para mantener actualizada la versión ejecutable de un programa.

Tipo de datos	Significado
HANDLE	Define un entero sin signo de 16 bits que se utiliza como manipulador.
HWND	Especifica un entero sin signo de 16 bits que se usa como manipulador de una ventana.
LONG	Define un entero de 32 bits con signo.
LPSTR	Define un puntero de 32 bits a un tipo de datos de caracteres.
FARPROC	Define un puntero de 32 bits a una función.
WORD	Define un entero sin signo de 16 bits.

Tabla 2.1

Tipos de datos frecuentes en Windows

Estructura	Uso
MSG	Especifica los campos de un mensaje de entrada.
PAINTSTRU CT	Especifica la estructura de dibujo que se utiliza para dibujar dentro de una ventana.
RECT	Especifica un rectángulo.
WNDCLASS	Especifica una clase de ventanas.

Tabla 2.2

Estructuras frecuentemente usadas en Windows

II.5.7 Componentes Básicos de una Aplicación en Windows

Al nivel más alto, una aplicación en Windows se puede dividir en dos componentes esenciales: la función WinMain y la función de ventana. Microsoft requiere que el cuerpo principal de su programa se llame WinMain. Esta función actúa como punto de entrada de la aplicación y se comporta de modo similar a la función main (principal) de un programa estándar en C/C++. Sin embargo, la función de ventana posee un papel esencial. Su aplicación en Windows nunca accede directamente a ninguna función de ventana. En lugar de esto, su aplicación solicita a Windows que lleve a cabo la operación especificada. A fin de facilitar esta comunicación, Windows necesita una función de retorno. Una función de retorno se declara en Windows y se vuelve a ella cada vez que Windows ejecuta una operación sobre una ventana.

AMBIENTE OPERATIVO

Todas las aplicaciones en Windows deben tener una función WinMain. WinMain es la responsable de:

- Declarar las clases de ventanas de la aplicación.
- Realizar las inicializaciones necesarias.
- Crear e iniciar el bucle de procesamiento de mensajes de la aplicación (que accede a la cola de mensajes de la aplicación).
- Finalizar la ejecución del programa cuando se reciba el mensaje WIN_QUIT.

CAPITULO III

VISUAL BASIC

III.1 ¿QUE ES VISUAL BASIC?

Con la aparición de Windows se incrementó sensiblemente la cantidad de usuarios adeptos a este ambiente operativo. Sin embargo, un segmento de éstos habían permanecido al margen: los programadores, que no contaban con un paquete para crear sus aplicaciones.

Una solución para ello surgió con Visual Basic. Este Software de Microsoft ha puesto a la disposición de prácticamente cualquier usuario, la experiencia de programar aplicaciones para Windows a través del lenguaje BASIC.

Visual Basic es una herramienta gráfica que permite crear de una manera fácil y rápida aplicaciones para el ambiente Windows. La programación en Visual Basic permite explotar completamente la interfaz gráfica de usuario (GUI) y además tener una mayor producción en la generación de sus aplicaciones por el tipo de comandos implementados.

Se basa en la creación de formas y controles por medio de su ambiente de programación amigable. Cada forma tiene asociado un segmento de código que establece su funcionalidad.

Este sistema es un gran avance para desarrolladores, combina la sintaxis simplificada de BASIC y GWBASIC con la estructura de programación de QBASIC y QuickBasic.¹

La programación en Visual Basic permite crear objetos, llamados controles, establece y cambia sus propiedades, y con esto se atribuye el código BASIC para la parte funcional.

¹ Keynon Brown. Programmer's Introduction to V.B.: SYBEX Inc. 1992. Pag.4.

III.2 PROGRAMACION EN VISUAL BASIC

Kenyon Brown se vió tentado a definir a Visual Basic como un *sistema de programación orientados a objetos*¹, aunque, como él afirma se equivocaría. Ya que nosotros creamos objetos, llamados *formas* y *controles* que hacen que su aplicación funcione. Sin embargo, sin profundizar en el tema, estos objetos carecen de las propiedades de herencia y polimorfismo, las cuales tienen que estar presentes en un verdadero ambiente orientado a objetos. Las aplicaciones creadas corren en Windows, y están en un ambiente *visually-oriented* que provee librerías de herramientas y objetos para programación de aplicaciones.

La filosofía de programación es que en vez del laborioso proceso para crear una interfaz de usuario necesaria en otros lenguajes de programación, en Visual Basic sólo se dibuja en la pantalla. Existen tres pasos generales para escribir una aplicación en Visual Basic, y son los siguientes:

- Dibujar la(s) ventana(s) que se requiera(n).
- Establecer las propiedades a objetos.
- Escribir código para los eventos asociados.

El primer paso es el diseño de la(s) ventana(s); (antes, diseñar la apariencia de una ventana era un proceso tedioso), donde serán colocados los botones, los tamaños y otros tipos de especificaciones. Adicionar y mover características también era una tarea difícil. Sin embargo, bajo Visual Basic, todo este proceso es extraordinariamente fácil. Visual Basic nos permite simplificar el dibujar las ventanas que se quieran, como también todos los botones, cajas y etiquetas.

¹ Kenyon Brown. Programmer's Introduction to V.B. SYBEX Inc. 1992. Pág.5

El siguiente paso es establecer las propiedades que se desean para la ventana, por ejemplo, dar un nombre a una ventana o botón, cambiar sus colores (o si va a ser visible o no) y finalmente, escribir el código que responda al evento.

III.2.1 El Ambiente de Visual Basic

Hay una ventana modelo que aparece en el centro de la pantalla, etiquetada como Form1. Esta es la ventana en la que se diseñará. Visual Basic se refiere a la ventana como una forma. La Form1 tiene ya un sistema de caja de menú, una barra de título, ambas minimizan y maximizan cajas, bordes, y el área cliente. El área cliente es llenada con puntos a intervalos regulares. Estos puntos forman un grid que ayuda a alinear los botones y cajas cuando se diseña una ventana. El ambiente de Visual Basic se muestra en la figura 3.1

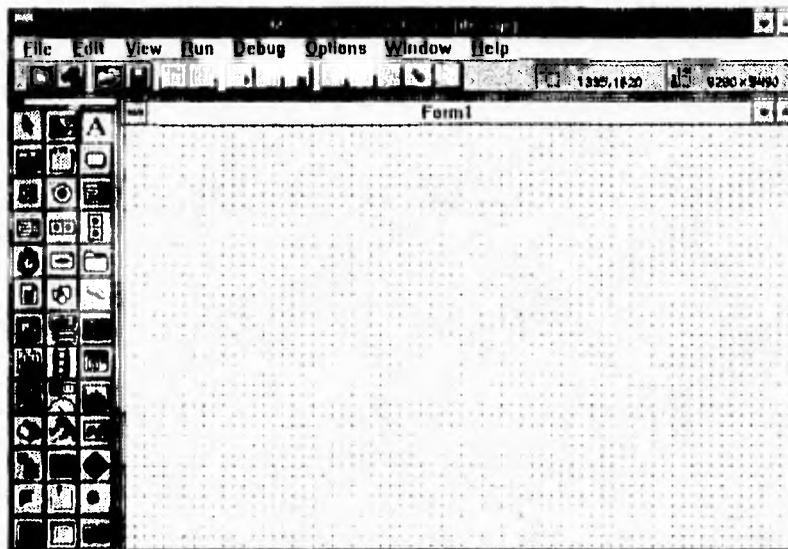


Figura 3.1

El ambiente de Visual Basic

VISUAL BASIC

III.2.2 Propiedades

Las cajas y botones son llamados objetos y cada tipo diferente de objeto puede tener ciertas propiedades. Por ejemplo, nuestro objeto llamado Form1 tiene propiedades asociadas, como: caption, color, visible, name, etc. Una caja de texto puede tener diferentes propiedades establecidas, incluyendo, por ejemplo, que el texto sea desplegado en ella. En otras palabras, las propiedades de un objeto representan todos los datos que son normalmente asociados con este objeto. Como se muestra en la figura 3.2.

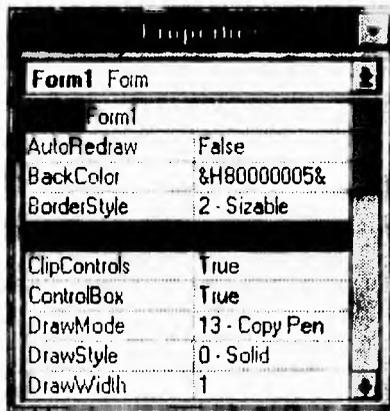


Figura 3.2

Propiedades de los objetos gráficos

Cada objeto (botones, caja de lista, radio botones y etiquetas) tiene establecido ciertas propiedades, esto es, datos que son asociados con él. Cuando se trabaja con diferentes objetos, la barra de propiedades permite cambiar sus propiedades. No todas las propiedades son válidas al momento del diseño. Para mayor información acerca de estas propiedades consultar el apéndice B.

III.2.3 Controles

Hay dos tipos de objetos gráficos en Visual Basic: formas y controles, los cuales el usuario puede manipular. Un control es cualquier objeto gráfico que se puede diseñar y poner en una forma, por ejemplo: caja de lista, botones, etiquetas, even timer, caja de texto, cajas de pintura, etc., por tanto todas las clases de botones son controles. Una forma es la ventana que diseñamos y creamos; un control es usado para I/O con el usuario, semejante a cajas y botones.

Se debe usar una caja de texto cuando se desea desplegar un texto; si deseamos desplegar una pintura, podemos usar una caja de pintura. Para adicionar una caja de texto, se usa la caja de herramientas de Visual Basic (llamada Toolbox). Nosotros podemos dibujar controles directamente en la forma, tomándolos de la caja de herramientas. La caja de herramientas se muestra en la figura 3.3.



Figura 3.3

Caja de herramientas (Toolbox)

VISUAL BASIC

Existen dos caminos para dibujar controles en Visual Basic. El primero es seleccionar el control de la caja de herramientas (dar un clic); cuando se ha hecho, se arrastra el mouse y se coloca en el lugar deseado de la forma. El segundo camino consiste en dar dos clics en el control a seleccionar y entonces éste aparecerá en la parte media de la forma, como se muestra en la figura 3.4.

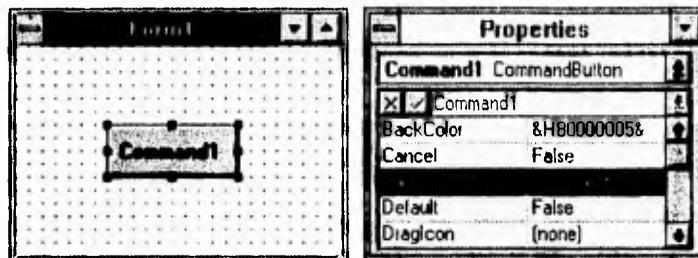


Figura 3.4

Forma para dibujar un control

Estos controles son desplegados con su 'CtlName' que es el nombre que se utiliza en la programación. Como ya mencionamos anteriormente cada objeto tiene propiedades establecidas las cuales pueden ser cambiadas en tiempo de diseño según nuestras necesidades. La lista de propiedades incluye: CtlName, Caption, Left, Top, Height, Width -todas las medidas de propiedades en Visual Basic son en twips por default (1/1440ths en una pulgada)-, Font Name y propiedades adicionales como FontSize, TabStop e Index (el cual permite coordinar las acciones de los botones). Por ejemplo si en la propiedad de Caption se escribe el nombre de SALIR aparecerá como se muestra en la figura 3.5.

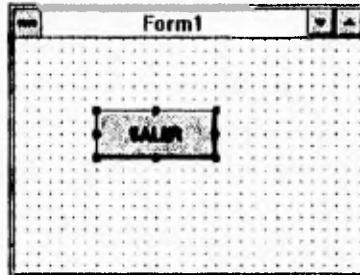


Figura 3.5

Asignando un nombre en las propiedades de los controles

El camino usual para hacer referencia a una propiedad en un programa en Visual Basic es `Object.Property`, donde `Object` es el nombre del objeto -forma o control- que tiene esta propiedad, y `Property` es el nombre de la propiedad.

Los programas son event-driven.* El código se debe conectar a un evento. Para encontrar los eventos asociados con cualquier objeto, se debe dar doble clic al objeto mientras diseñamos nuestro programa, abriéndose una ventana de código. La ventana de código tiene un molde para todos los procedimientos de eventos conectados con un objeto específico. Como se muestra en la figura 3.6.

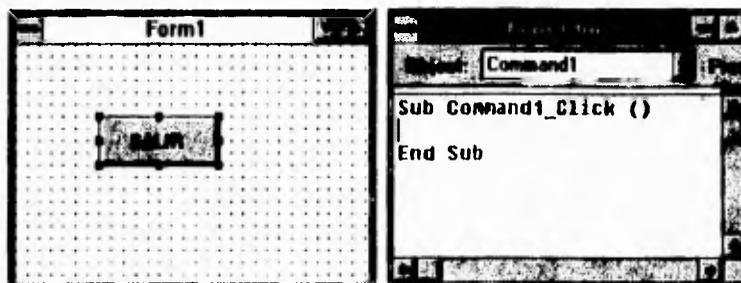


Figura 3.6

Ventana para asignar código a un control

* "event-driven" conocido como programación orientado a eventos

VISUAL BASIC

Hay dos tipos principales de procedimientos en Visual Basic: Sub procedimientos y Funciones de procedimientos (Visual Basic también soporta tipos de subrutinas GoSub en un Basic estándar). La diferencia entre éstas es que las Funciones de procedimiento pueden retornar valores, y los Sub procedimientos no. Los Sub procedimientos y las Funciones de procedimiento pueden también tomar argumentos pasados de otros semejantes a ellos:

```
Sub MySub (A As Integer, B As Integer)
```

```
.  
.
```

```
End Sub
```

```
Function MyFunc(C As Integer, D As Integer)
```

```
.  
.
```

```
End Function
```

Revisando el código que se muestra a continuación, observamos: `Command1_Click()`. El `Command1` es el nombre por default del control; el nombre del Sub procedimiento `Command1_Click()` indica que este procedimiento de evento es conectado con el botón `Command1`, y éste es el Sub procedimiento que se ejecuta cuando el usuario da clic al botón.

```
Sub Command1_Click()
```

```
Text1.Text = 'Bienvenido a Visual Basic'
```

```
End Sub
```

Existen otros eventos asociados con los botones de comandos. En la ventana de código existe una caja a la derecha la cual indica los procedimientos que son válidos para un objeto en particular.

Para correr nuestros programas en Visual Basic, nosotros lo podemos hacer dentro de la aplicación Windows stand-alone. También se nos permite crear un archivo .EXE que es una opción que nos da el menú de Visual Basic. Cuando se elige esta opción, se abre una caja de diálogo. Se debe dar un clic en el botón OK, y Visual Basic crea un archivo llamado proyect.exe, el cual se puede correr bajo Windows directamente.

El proyect.exe no es completamente stand-alone; los archivos .EXE de Visual Basic requieren el archivo vbrunxxx.dll para correr, donde xxx corresponde al número de versión.

Visual Basic da a nuestra aplicación un nombre de proyect.exe por que éste organiza las tareas en proyectos, no en formas. Una aplicación puede tener un número de formas asociadas a ella y colecciona todo junto en un proyecto.

Visual Basic solamente permite abrir un proyecto a un tiempo, y cada proyecto puede tener tres diferentes partes.

El primer componente usual de un proyecto es una forma; las aplicaciones usualmente tienen al menos una forma. Se pueden tener módulos en Visual Basic, los cuales son hechos en código BASIC pero este código no está directamente asociado con una forma en particular. Los procedimientos en un módulo pueden ser alcanzados en cualquier parte de la aplicación.

Un módulo global es parte integral de la aplicación, pero ésta contiene declaraciones, no código. Esto es, se declaran variables en BASIC, así como constantes y tipos. Colocando estas declaraciones dentro del módulo global (hay solamente un módulo global) se hacen más accesibles para el resto de la aplicación.

VISUAL BASIC

Para guardar la ruta del proyecto, Visual Basic mantiene la ventana del proyecto, que es usada cuando se tienen formas múltiples o módulos de código; también permite switchearse al módulo global, para poder adicionar declaraciones globales de constantes, variables o tipos. El nombre por default de una forma en la ventana del proyecto es form.frm; la extensión .frm es como un .bas para módulos. El proyecto es salvado como un archivo .mak.

La ventana de proyecto se muestra en la figura 3.7.



Figura 3.7

Ventana del Proyecto

Para mayor información acerca de estos controles consultar el apéndice A.

III.2.4 Variables

El nombre de variables en Visual Basic puede ser de 40 caracteres de longitud (incluyendo letras, números, y underscores), y solamente existen dos reglas para ellas: El primer caracter debe ser una letra, no se pueden usar palabras reservadas de Visual Basic como nombres de variables. Los tipos de datos que se pueden construir son los siguientes:

Tipo	Bytes	Caracter	Rango
Integer	2	%	-32,768 a 32,767
Long	4	&	-2,147,483 a 2,147,483,647
Single	4	!	-3.37E+38 a 3.37E+38
Double	8	#	-1.67D+308 a 1.67D+308
Currency	8	@	-9.22E+14 a 9.22E+14
String	varios	\$	[No aplicable]

Tabla 3.1

Tipos de datos

Existen ciertos caracteres que se usan para indicar el tipo de variable como: % para enteros, ! para punto flotante, etc.

Hay dos caminos para indicar a Visual Basic que se quiere usar un cierto nombre para una variable. El primero es simplemente usar el nombre donde se quiere, esto es:

```
my_int% = 5
```

que es una declaración implícita. El otro camino es usar una sentencia Dim para especificar al inicio la declaración de variables de un procedimiento o sentencias globales declaradas inicialmente en el módulo global. Ejemplos:

VISUAL BASIC

```
Dim my_int As Integer
Dim my_double As Double
Dim my_variable_string As String
Dim my_fixed_string As String * 20
```

Hay cuatro diferentes niveles de alcance para variables, porque hay cuatro diferentes lugares para declarar variables.

El primer lugar para declarar variables, con cada sentencia Dim o implícitamente, esto en el nivel de procedimiento. Cuando se declara una variable en un procedimiento, esta variable es local para este procedimiento; en otras palabras, este alcance es restringido para el procedimiento en el cual fué declarada. En todo momento en el que se entra al procedimiento, las variables locales son reinicializadas.

Los dos siguientes lugares donde se declaran variables son en la forma y en los niveles del módulo. Si se declara una variable a nivel forma, esta variable es accesada para todos los procedimientos de la forma. Lo mismo pasa con los módulos: si se declara una variable a nivel módulo, esta variable es accesada por todos los procedimientos en el módulo.

El nivel final es a nivel global. Todos los procedimientos en línea de código en una aplicación tienen acceso a estas variables. Para declarar una variable global, se coloca en el módulo global y se declara con la palabra Global ejemplo:

```
Global my_int As Integer
Global my_double As Double
Global my_variable_string As String
Global my_fixed_string As String * 20
```

III.2.5 Menús

Hay un número de elementos en un menú. El primero es la barra de menú, la cual indica todos los posibles menús en una aplicación. Cada línea en un menú lista un menú único; si este elemento es iluminado, este es seleccionado.

Si un elemento de algún menú finaliza con "...", este puede desplegar una caja de diálogo para leer más información. Si un elemento está sombreado, quiere decir que está deshabilitado. Finalmente, los elementos de un menú pueden ser agrupados con una barra separadora.

III.3 VENTAJAS Y LIMITANTES

En la estrategia de Microsoft, Visual Basic se encuentra como el producto ideal para el desarrollo de aplicaciones comerciales y corporativas, dejando a las herramientas Ms Access y Ms FoxPro para el usuario final por su facilidad de uso y a Ms Visual C++ para desarrolladores de aplicaciones más complejas como lo constituye la creación de un driver para un dispositivo, un driver de acceso a una base de datos a los mismos controles que se usan en Visual Basic.

Una de las ventajas de Visual Basic es que se pueden desarrollar poderosas y complejas aplicaciones que los usuarios pueden ejecutar en la forma standard de Windows.

Otra de las ventajas de usar Visual Basic es que provee una interfaz consistente y manejadora para desplegar información en varias aplicaciones

VISUAL BASIC

distintas que tienen similares controles. La filosofía de la interfaz Windows esta basada sobre la utilización de metáforas visuales para la realización de acciones y tareas.

El paradigma es simple: después de que se ha usado una aplicación Windows, es fácil aprender el uso de Visual Basic porque la interfaz es muy familiar.

Para el uso de controles Windows tales como: menús, iconos, barras de scroll y cajas de diálogos, la información es presentada a los usuarios en una forma visualmente interesante y de formatos dinámicos.

El sistema de programación de Visual Basic abre al mundo las posibilidades de diseñar una interfaz al usuario, gráficos efectivos en Windows, lo cual antes no estaba disponible, al menos que se programe en C y se haya utilizado Windows Software Development Kit (SDK). Un Graphical User Interface (GUI) consiste en que los usuarios vean cuando la aplicación Windows esta abierta. Usando un GUI habilita al usuario a interactuar con un programa y manipular funciones.

Visual Basic permite diseñar la interfaz, escribir código en BASIC y atribuir el código a los controles de aplicación rápida y fácilmente. Este proceso es muy diferente a las frecuentemente tareas realizadas en programas en C con el SDK.

Visual Basic reduce drásticamente la dificultad y la cantidad de tiempo envuelta en programación en Windows. Creando una aplicación Windows en C toma meses en desarrollarla, sin embargo, con Visual Basic, se puede construir una aplicación dinámica en Windows en tan sólo un día.

Este lenguaje es interprete, las aplicaciones son leídas para correrlas tan pronto como se termina de escribir el código. En un lenguaje compilado, por contraste, un programa debe ser compilado antes de ser ejecutado.

De esta manera, el compilador convierte a los programas en lenguaje máquina, los cuales consisten de números binarios que la computadora entiende. Esta conversión es hecha sólo una vez. Así el programa se ejecuta, el lenguaje máquina es disponible y puede ser directamente ejecutado por la computadora.

En un lenguaje interprete que guste de Visual Basic, las sentencias de procedimiento pueden ser ejecutadas tan pronto como son tecladas. El interprete lee una línea de un procedimiento y envía las instrucciones apropiadas a la computadora para su ejecución. Cuando la línea esta completa, el interprete se mueve a la siguiente línea, interpreta y ejecuta. Teniendo una aplicación, inmediatamente se realiza la lectura para correr los programas realizándose rápidamente. Por esto, Visual Basic es una gran ventaja como herramienta de programación en Windows.

En comparación con la programación en C y C++ para Windows. Visual Basic está limitado en velocidad y en el conjunto de funciones propias. Sin embargo, cubre estas limitantes otorgando la flexibilidad de acceder las API's de Windows. Controles externos desarrollados por terceros y librerías dinámicas que permiten aumentar el rango de funciones de Visual Basic y la velocidad de respuesta de una aplicación.

CAPITULO IV

**METODOLOGIAS PARA EL
DESARROLLO DE SISTEMAS**

IV.1 INTRODUCCION

Los sistemas de información basados en computadora sirven para diversas finalidades que van desde el procesamiento de las transacciones de una empresa, hasta proveer de la información necesaria para decidir sobre asuntos que se presentan con frecuencia, asistencia a los altos funcionarios con la formulación de estrategias difíciles y la vinculación entre la información de las oficinas y los datos de toda la corporación. En algunos casos los factores que deben considerarse en un proyecto de sistemas de información (tales como el aspecto más apropiado de la computadora o la tecnología de comunicaciones que se va utilizar, el impacto del nuevo sistema sobre los empleados de la empresa y las características específicas que el sistema debe tener), se pueden determinar de una manera secuencial. En otros casos, debe ganarse experiencia por medio de la experimentación conforme el sistema evoluciona por etapas.

A medida que las computadoras son empleadas cada vez más por personas que no son especialistas en computación, el rostro del desarrollo de sistemas de información adquiere una nueva magnitud. Los propios usuarios emprenden ya el desarrollo de algunos de los sistemas que ellos emplean.

Todas estas situaciones están representadas por tres distintos enfoques al desarrollo de sistemas de información basados en computadora¹:

1. Método del ciclo de vida para el desarrollo de sistemas.
2. Método del desarrollo del análisis estructurado.
3. Método del prototipo de sistemas.

¹ James A. Senn. Análisis y Diseño de Sistemas de Información McGraw-Hill. 1992. Pag. 32.

METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

El presente capítulo tiene como finalidad explorar cada enfoque, abordando sus características y las condiciones bajo las que es probable que se obtenga el mayor beneficio para la empresa u organización.

IV.2 METODO DEL CICLO DE VIDA

El desarrollo de sistemas, un proceso formado por diferentes etapas, comienza cuando la administración o algunos de los miembros del personal encargado de desarrollar sistemas, detectan un sistema de la empresa que necesita mejoras o en su defecto, desarrollar uno que satisfaga las necesidades de la misma.

El método de ciclo de vida para el desarrollo de sistemas es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información. En la mayor parte de las situaciones dentro de una empresa todas las actividades están muy relacionadas, en general son inseparables, y quizá sea difícil determinar el orden de los pasos que se siguen para efectuarlas.

Una idea básica del ciclo de vida del desarrollo de sistemas es que es un proceso bien definido en el cual se percibe una aplicación, se desarrolla y se realiza. El ciclo de vida le da estructura a un proceso creativo. Con el fin de manejar y controlar el esfuerzo de desarrollo, se hace necesario conocer qué se debería haber hecho, que se ha hecho y aun qué falta por cumplir. Las fases en el ciclo de vida proveen una base para la administración y el control en razón de que definen los segmentos del flujo de trabajo que se pueden identificar para propósitos administrativos y especificar los documentos u otros resultados intermedios que van a ser producidos en cada fase.

METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

Las fases en el ciclo de vida para el desarrollo de sistemas de información son descritos por varios autores, pero las diferencias están principalmente en la cantidad de detalle y en la forma de categorización. Hay un acuerdo general en el flujo de los pasos de desarrollo y la necesidad de procedimientos de control en cada etapa.

El método de ciclo de vida para el desarrollo de sistemas consta de las siguientes actividades¹:

1. Investigación preliminar.
2. Determinación de requerimientos.
3. Diseño del sistema.
4. Desarrollo de software.
5. Prueba del sistema.
6. Implantación y evaluación.

IV.2.1 Investigación Preliminar

La solicitud para recibir ayuda de un sistema puede originarse por varias razones; sin importar cuáles sean éstas, el proceso se inicia siempre con la petición de una persona (administrador, empleado o especialista en sistemas).

Cuando se formula la solicitud comienza la primera actividad de sistemas: la investigación preliminar. Esta actividad tiene tres partes: aclaración de la solicitud, estudio de factibilidad y aprobación de la solicitud.

¹ James A. Senn. Análisis y Diseño de Sistemas de Información McGraw-Hill. 1992. Pag. 33

METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

Aclaración de la solicitud. Muchas solicitudes que provienen de empleados y usuarios no están formuladas de manera clara. Por consiguiente, antes de considerar cualquier investigación de sistemas, la solicitud de proyecto debe examinarse para determinar con precisión lo que el solicitante requiere.

Estudio de factibilidad. Cuando se propone una nueva aplicación, normalmente se lleva a cabo el estudio de factibilidad antes de que sea aprobada para su desarrollo. Si la aplicación es parte del plan maestro del predefinido sistema de información gerencial, aún debería completarse la fase de determinación de la factibilidad, con el fin de evaluar los enfoques alternativos para su desarrollo aunque la idea en general se haya aprobado con el resto del plan. Si una propuesta es sometida por fuera del plan, el estudio de factibilidad incluye el aspecto de si es consecuente con el plan existente y si debería forzar las prioridades de las otras aplicaciones ya planeadas.

Cinco aspectos dentro de la factibilidad son dirigidos en el estudio. Dan como resultado el reconocimiento tanto de los beneficios como de los riesgos inherentes al desarrollo y a la implementación del sistema de aplicación propuesto:

1. **Factibilidad técnica.** ¿Se puede implementar la aplicación propuesta con la tecnología actual? El análisis de los riesgos del proyecto con relación a la factibilidad técnica incluye si la tecnología esta disponible en el mercado.
2. **Factibilidad económica.** ¿Dará el sistema beneficios mayores que los costos? El estudio de la factibilidad presenta los beneficios tangibles e intangibles de una manera formal. También se presenta un análisis

METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

relativamente detallado de los costos tanto para los desarrollos como para las operaciones de las diferentes alternativas.

3. **Factibilidad Motivacional.** La probabilidad de que la organización esté suficientemente motivada para apoyar el desarrollo e implementación de la aplicación, con la participación necesaria del usuario, los recursos, el tiempo de entrenamiento, etc. Esta motivación usualmente es demostrada por el propietario o por el defensor de la aplicación quien tiene suficiente poder en la organización para proveer los recursos y motivar a otros a ayudar y cooperar.
4. **Factibilidad de planeamiento.** La probabilidad de que la organización pueda completar el proceso oportunamente, en el tiempo permitido para su desarrollo. La adición de recursos para el desarrollo no siempre reduce su duración; en efecto, el aporte de personal que no se puede utilizar efectivamente, puede impedir el desarrollo, en razón del tiempo que se gasta en la comunicación.
5. **Factibilidad operacional.** ¿Trabjará cuando se instale? Este análisis puede implicar una determinación subjetiva del medio ambiente político y administrativo en el cual se realizará el sistema. En general, mientras mayores sean los requerimientos de cambio en el medio ambiente del usuario en que se instalará el sistema, mayores serán los riesgos de fracaso de su implementación.

Aprobación de la solicitud. No todos los proyectos solicitados son deseables o factibles. Algunas organizaciones reciben tantas solicitudes de sus empleados que sólo es posible atender unas cuantas. Sin embargo, aquellos proyectos que son deseables y factibles deben incorporarse en los planes.

IV.2.2 Determinación de los Requerimientos del Sistema

Con el fin de obtener de manera efectiva un conjunto completo y correcto de los requerimientos, es necesario utilizar un método o métodos que tengan en cuenta hasta que punto los requerimientos son completamente conocidos o por el contrario, si se está necesitando ser explorados o descubiertos.

El resultado del análisis de los requerimientos de información del ciclo de vida de desarrollo es un reporte que detalla los requerimientos de la aplicación. Los requerimientos constan de elementos como los siguientes:

- * Informe (incluye los datos elementales en los reportes).
- * Consultas.
- * Esquema conceptual de la base de datos (a partir del modelamiento de los datos u otro análisis).
- * Requerimientos funcionales (incluyendo las características operacionales).
- * Requerimientos de interfaz del usuario.

IV.2.2.1 Actividades de la Determinación de Requerimientos

Es útil ver la determinación de requerimientos a través de 3 grandes actividades: Anticipación, investigación y especificación de requerimientos.

Anticipación de requerimientos. Experiencia de los analistas en una área en particular y el contacto con sistemas en un ambiente similar al que se encuentra bajo su investigación, tiene influencia sobre el estudio que éstos realizan. Su experiencia les permite anticipar ciertos problemas o características y requerimientos para un nuevo sistema.

METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

Por lo tanto, es probable que las características que investigan del sistema actual, las preguntas que formulen o los métodos que utilizan estén basados sobre esta familiaridad.

Investigación de requerimientos. Esta actividad es la más importante del análisis de sistemas, los analistas estudian el sistema actual o en su defecto la creación de uno nuevo, con la ayuda de varias herramientas y habilidades, y documentan sus características para emprender el análisis.

Especificaciones de requerimientos. Los datos obtenidos durante las fases de anticipación e investigación se analizan para determinar las especificaciones de los requerimientos, es decir, la descripción de las características del nuevo sistema. Esta actividad tiene tres partes relacionadas entre sí:

- * Análisis de datos basados en hechos reales.
- * Identificación de requerimientos esenciales.
- * Selección de estrategias para satisfacer los requerimientos.

IV.2.3 Diseño del Sistema

El diseño de un sistema produce los detalles que establecen la forma en la que el sistema cumplirá con los requerimientos identificados durante la fase de análisis.

METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

Se comienza el proceso de diseño identificando los reportes y demás salidas que debe producir el sistema. Hecho lo anterior se determinan con toda precisión los datos específicos para cada reporte y salida, es común que los diseñadores hagan un bosquejo del formato o pantalla que esperan que aparezca cuando el sistema esté terminado.

El diseño de un sistema también indica los datos de entrada, aquellos que serán calculados y los que deben ser almacenados. Asimismo, se escriben con todo detalle los procedimientos de cálculo y los datos individuales. Los diseñadores seleccionan las estructuras de archivo y los dispositivos de almacenamiento, tales como discos y cintas magnéticas o incluso archivos en papel. Los procedimientos que se escriben indican cómo procesar los datos y producir las salidas.

Los documentos que contienen las especificaciones de diseño representan a éste de muchas maneras (diagramas, tablas y símbolos especiales). La información detallada del diseño se proporciona al equipo de programación para comenzar la fase de desarrollo de software.

IV.2.4 Desarrollo de Software

Los encargados de desarrollar software pueden instalar (o modificar y después instalar) software comprado a terceros o escribir programas diseñados a la medida del solicitante. La elección depende del costo de cada alternativa, del tiempo disponible para escribir el software y de la disponibilidad de los programadores. Por regla general, los programadores (o analistas programadores) que trabajan en las grandes organizaciones pertenecen a un grupo permanente de profesionales. En empresas pequeñas,

donde no hay programadores, se pueden contratar servicios externos de programación.

Los programadores también son responsables de la documentación de los programas y de proporcionar una explicación de cómo y por qué de ciertos procedimientos se codifican en determinada forma. La documentación es esencial para probar el programa y llevar a cabo el mantenimiento una vez que la aplicación se encuentra instalada.

IV.2.5 Pruebas del Sistema

Durante la fase de pruebas del sistema, el sistema se emplea de manera experimental para asegurarse de que el software no tenga fallas, es decir que funciona de acuerdo con las especificaciones y en la forma en que los usuarios esperan que lo haga. Se alimentan con un conjunto de datos de prueba para su procesamiento y después se examinan los resultados. En ocasiones se permite que varios usuarios utilicen el sistema para que los analistas observen si tratan de emplearlo en formas no previstas. Es preferible descubrir cualquier sorpresa antes de que la organización implante el sistema y dependa de él.

En muchas organizaciones, las pruebas son conducidas por personas al grupo que escribió los programas originales; con esto se persigue, por una parte, que las pruebas sean completas e imparciales y, por otra, que el software sea más confiable.

METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

IV.2.6 Implantación y Evaluación

La implantación es el proceso de verificar e instalar nuevo equipo, capacitar a los usuarios, instalar la aplicación y construir todos los archivos de datos necesarios para utilizarla.

Dependiendo del tamaño de la organización que empleará la aplicación y el riesgo asociado con su uso, puede elegirse comenzar la operación del sistema sólo en una área de la empresa (prueba piloto), por ejemplo en un departamento con una o dos personas. Algunas veces se deja que los dos sistemas, el viejo y el nuevo, trabajen en forma paralela con la finalidad de comparar los resultados. En otras circunstancias, el viejo sistema deja de utilizarse determinado día para comenzar a emplear el nuevo al día siguiente. Cada estrategia de implantación tiene sus méritos de acuerdo con la situación que se considere dentro de la empresa. Sin importar cuál sea la estrategia utilizada, los encargados de desarrollar el sistema procuran que el uso inicial del sistema se encuentre libre de problemas.

Una vez instaladas, las aplicaciones se emplean durante muchos años. Sin embargo las organizaciones y los usuarios cambian con el paso del tiempo, incluso el ambiente es diferente con el paso de las semanas y los meses. Por consiguiente, es indudable que debe darse mantenimiento a las aplicaciones; realizar cambios y modificaciones en el software, archivos o procedimientos para satisfacer las nuevas necesidades de los usuarios. Dado que los sistemas de las organizaciones junto con el ambiente de las empresas experimentan cambios de manera continua, los sistemas de información deben mantenerse siempre al día. En este sentido, la implantación es un proceso en constante evolución.

La evaluación de un sistema se lleva a cabo para identificar puntos débiles y fuertes. La evaluación ocurre a lo largo de cualquiera de las siguientes dimensiones:

*** Evaluación operacional**

Valoración de la forma en que funciona el sistema, incluyendo su facilidad de uso, tiempo de respuesta, lo adecuado de los formatos de información, confiabilidad global y nivel de utilización.

*** Impacto organizacional**

Identificación y medición de los beneficios para la organización en áreas tales como finanzas (costos, ingresos y ganancias), eficiencia operacional e impacto competitivo. También se incluye el impacto sobre el flujo de información interno y externo.

*** Opinión de los administradores**

Evaluación de las actitudes de directivos y administradores dentro de la organización, así como de los usuarios finales.

*** Desempeño del desarrollo**

La evaluación del proceso de desarrollo de acuerdo con criterios tales como tiempo y esfuerzo de desarrollo, concuerdan con presupuestos y estándares, y otros criterios de administración de proyectos. También se incluye la valoración de los métodos y herramientas utilizadas en el desarrollo.

Desafortunadamente la evaluación de sistemas no siempre recibe la atención que merece. Sin embargo, cuando se conduce en forma adecuada proporciona mucha información que puede ayudar a mejorar la efectividad de los esfuerzos de desarrollo de aplicaciones subsecuentes.

IV.3 METODO DEL DESARROLLO DEL ANALISIS ESTRUCTURADO

Muchos especialistas en sistemas de información reconocen la dificultad de comprender de manera completa sistemas grandes y complejos. El método de desarrollo del análisis estructurado tiene como finalidad superar esta dificultad por medio de 1) la división del sistema en componentes y 2) la construcción de un modelo del sistema.¹ El método incorpora elementos tanto de análisis como de diseño.

Cuando los analistas comienzan a trabajar sobre un proyecto de sistemas de información, a menudo tienen que profundizar en una área de la organización con la que tienen poca familiaridad. A pesar de esto, deben desarrollar un sistema que ayude a los gerentes y personal -los futuros usuarios- de esa área. Cualquier nuevo sistema o conjunto de recomendaciones para cambios en el sistema existente, ya sea este manual o automatizado, debe conducir hacia una mejora. Para alcanzar este resultado, se espera que los analistas de sistemas hagan lo siguiente:

- Conozcan los detalles y procedimientos del sistema en uso.
- Obtengan una idea de las demandas futuras de la organización como resultado del crecimiento, del aumento de la competencia en el mercado, de los cambios en las necesidades de los consumidores, de la evolución de las estructuras financieras, de la introducción de la nueva tecnología y cambios en las políticas del gobierno entre otros.
- Documentar detalles del sistema actual para su revisión y discusión por otros.

¹ James A. Senn. Análisis y Diseño de Sistemas de Información McGraw-Hill. 1992. Pág. 38

- **Evaluar la eficiencia y efectividad del sistema actual y sus procedimientos, tomando en cuenta el impacto sobre las demandas anticipadas para el futuro.**
- **Recomendar todas las revisiones y ampliaciones del sistema actual, señalando su justificación. Si es apropiado, quizá la propuesta de un nuevo sistema completo.**
- **Documentar las características del nuevo sistema con un nivel de detalle que permita comprender a otros sus componentes (y su interrelación), y de una manera que permita manejar el desarrollo del nuevo sistema.**
- **Fomentar la participación de gerentes y empleados en todo el proceso, tanto para aprovechar su experiencia y conocimiento del sistema actual, como para conocer sus ideas, sentimientos y opiniones relacionadas con los requerimientos de un nuevo sistema o de los cambios para el actual.**

IV.3.1 ¿Qué es el Análisis Estructurado?

El análisis estructurado se concentra en especificar lo que se requiere que haga el sistema o la aplicación. No se establece cómo se cumplirán los requerimientos o la forma en que se implantará la aplicación.¹ Más bien permite que las personas observen los elementos lógicos (lo que hará el sistema) separados de los componentes físicos (computadoras, terminales, sistemas de almacenamiento, etc.). Después de esto se puede desarrollar un diseño físico eficiente para la situación donde será utilizado.

¹ James A. Senn. Análisis y Diseño de Sistemas de Información McGraw-Hill. 1992. Pág. 38

METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

El análisis estructurado es un método para el análisis de sistemas manuales o automatizados, que conduce al desarrollo de especificaciones para sistemas nuevos o para efectuar modificaciones a los ya existentes. Cuando los analistas de sistemas abordan una situación poco familiar, siempre existe una pregunta sobre dónde comenzar el análisis. Una situación dinámica siempre puede ser vista como abrumadora debido a que muchas de las actividades se llevan a cabo constantemente. El análisis estructurado permite al analista conocer un sistema o proceso (actividad) en una forma lógica y manejable al mismo tiempo que proporciona la base para asegurar que no se omita ningún detalle pertinente.

IV.3.2 Elementos del Análisis Estructurado

Los elementos esenciales del análisis estructurado son símbolos gráficos, diagramas de flujo de datos y el direccionamiento centralizado de datos.

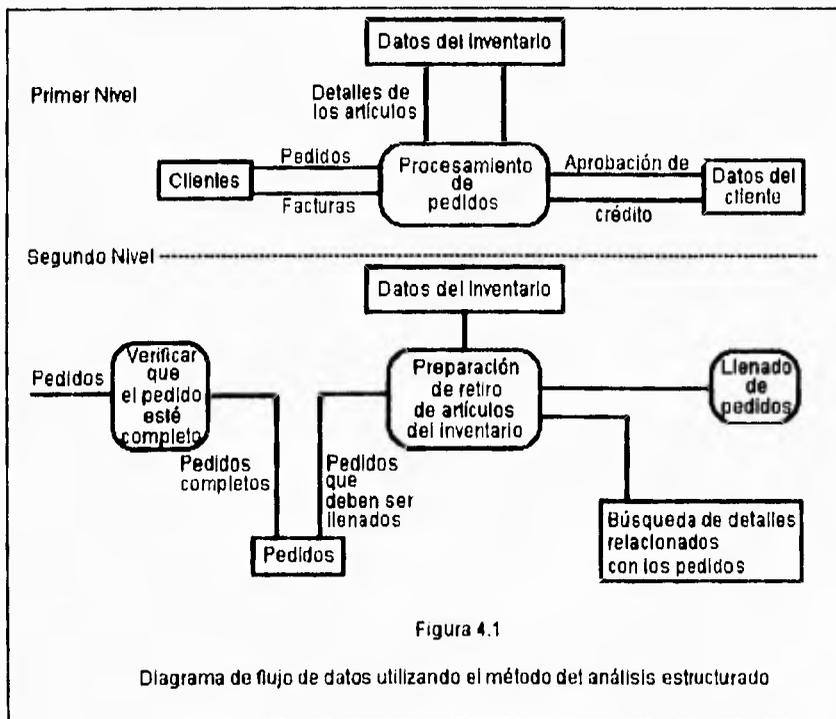
Descripción gráfica. Una de las formas de describir un sistema es preparar un bosquejo que señale sus características, identifique la función para la que sirve e indique como éste interactúa con otros elementos, entre otras cosas. Sin embargo, describir de esta manera un sistema grande es un proceso tedioso y propenso a errores ya que es fácil omitir algún detalle o dar una explicación que quizá los demás no entiendan.

En lugar de las palabras, el análisis estructurado utiliza símbolos o iconos, para crear un modelo gráfico del sistema. Los modelos de este tipo muestran los detalles del sistema pero sin introducir procesos manuales o computarizados, archivos en cinta o disco magnético, o procedimientos operativos y de programas. Si se seleccionan los símbolos y notaciones correctas entonces casi cualquier persona puede seguir la forma en que los componentes se acomodarán entre sí para formar el sistema.

METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

Tal como se indica en la figura 4.1, los iconos identifican los elementos básicos de los procesos, el flujo de datos, el sitio donde se almacenan los datos y las fuentes y destinos de éstos. Se dibuja una línea alrededor del sistema para señalar qué elementos se encuentran dentro del sistema y cuáles fuera de su frontera.

El diagrama lógico de flujo de datos muestra las fuentes y destinos de los datos, identifica y da nombre a los procesos que se llevan a cabo, identifica y da nombre a los grupos de datos que relacionan una función con otra y señala los almacenes de datos a los que tiene acceso.



METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

Diagramas de flujo de datos. El modelo del sistema recibe el nombre de diagrama de flujo de datos (DFD). La descripción completa de un sistema está formado por un conjunto de diagramas de flujo de datos.

Para desarrollar una descripción del sistema por el método del análisis estructurado se sigue un proceso descendente (top-down). El modelo original se detalla en diagrama de bajo nivel que muestran características adicionales del sistema. Cada proceso puede desglosarse en diagramas de flujo de datos cada vez más detallados, esta secuencia se repite hasta que se obtienen suficientes detalles que permiten al analista comprender en su totalidad la parte del sistema que se encuentra bajo investigación.

La figura 4.1 muestra los niveles primero y segundo de una parte de algún sistema. Nótese que el enfoque está sobre los datos y procesos. No se hace mención alguna de computadoras, comunicaciones, personas o departamentos y tampoco se incluyen detalles físicos.

Diccionario de datos. Todas las definiciones de los elementos en el sistema - flujo de datos, procesos y almacenes de datos- están descritos en forma detallada en el diccionario de datos. Si algún miembro del equipo encargado del proyecto desea saber alguna definición del nombre de un dato o el contenido particular de un flujo de datos, esta información debe encontrarse disponible en el diccionario de datos.

IV.3.3 ¿Qué es el Diseño Estructurado?

El diseño estructurado, otro elemento del análisis estructurado que emplea la descripción gráfica, se enfoca en el desarrollo de especificaciones del software. La meta del diseño estructurado es crear programas formados por módulos independientes unos de otros desde el punto de vista funcional. Este enfoque no sólo conduce hacia mejores programas sino que facilita el mantenimiento de los mismos cuando surja la necesidad de hacerlo.

El diseño estructurado es una técnica específica para el diseño de programas¹ y no un método de diseño de comprensión. Es decir, no indica nada relacionado con el diseño de archivos o base de datos, la presentación de entradas o salidas, la secuencia de procesamiento o el hardware que dará soporte a la aplicación. Esta técnica conduce a la especificación de módulos de programa que son funcionalmente independientes.

La herramienta fundamental del diseño estructurado es el diagrama estructurado. Al igual que los diagramas de flujo de datos, los diagramas estructurados son de naturaleza gráfica, evitan cualquier referencia relacionada con el hardware o detalles físicos. Su finalidad no es mostrar la lógica de los programas (que es la tarea de los diagramas de flujo). Los diagramas estructurados describen la interacción entre módulos independientes junto con los datos que en un módulo pasa a otro cuando interacciona con él. Estas especificaciones funcionales para los módulos se proporcionan a los programadores antes que dé comienzo la fase de escritura de código.

¹ James A. Senn. Análisis y Diseño de Sistemas de Información McGraw-Hill. 1992. Pag. 42

IV.3.4 Empleo del Análisis Estructurado con otros Métodos de Desarrollo

El análisis estructurado se combina, con bastante frecuencia, con el método ya presentado de ciclo de vida clásico de desarrollo de sistemas. Por ejemplo los analistas pueden optar por desarrollar diagramas de flujo de datos como una forma para documentar las relaciones entre componentes durante la investigación detallada de algún sistema existente. Asimismo, se pueden definir los archivos y datos en un diccionario centralizado de datos de acuerdo con las reglas del análisis estructurado.

Sin embargo muchas organizaciones optan por no utilizar este método de desarrollo. Por ejemplo, los analistas deciden con frecuencia que el desarrollo de diagramas y esquemas es una tarea que consume mucho tiempo, sobre todo si el sistema es grande y complejo. (Es común que los diagramas tengan que dibujarse una y otra vez conforme se adquiera nueva información).

Otros analistas señalan que los elementos que faltan, tales como las personas y los procedimientos de control, son parte del sistema mismo y no pueden omitirse en la descripción de éste.

IV.4 METODO DEL PROTOTIPO DE SISTEMAS

Este método hace que el usuario participe de manera más directa en la experiencia de análisis y diseño que cualquiera de los métodos ya presentados.¹ La construcción de prototipos es muy eficaz bajo las circunstancias correctas. Sin embargo, al igual que los otros métodos, es útil si se emplea en el momento adecuado y en la forma apropiada.

IV.4.1 ¿Qué es un Prototipo?

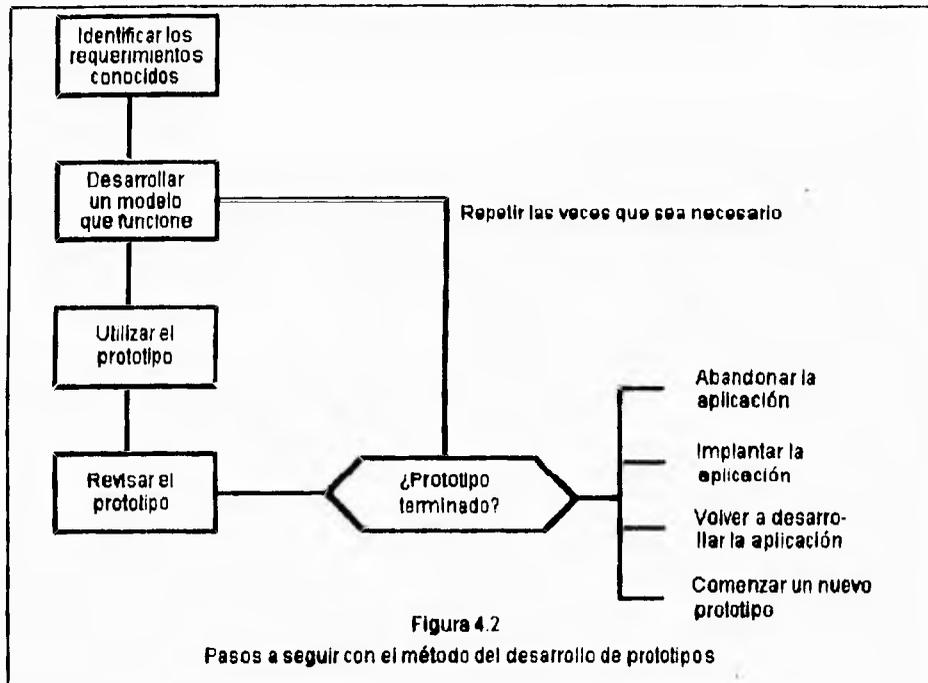
El prototipo es un sistema que funciona -no sólo una idea en el papel-, desarrollado con la finalidad de probar ideas y suposiciones relacionadas con el nuevo sistema. Al igual que el sistema basado en la computadora, está constituido por software que acepta entradas, realiza cálculos, produce información ya sea impresa o presentada en una pantalla, o que lleva a cabo otras actividades significativas. Es la primera versión o iteración de un sistema de información; es el modelo original.

Los usuarios evalúan el diseño y la información generada por el sistema. Lo anterior sólo puede hacerse con efectividad si los datos utilizados, al igual que las situaciones, son reales. Por otra parte, deben esperarse cambios a medida que el sistema es utilizado.

¹ James A. Senn. Análisis y Diseño de Sistemas de Información McGraw-Hill. 1992. Pág. 43

IV.4.2 Etapas del Método de Prototipos

El desarrollo de un prototipo para una aplicación se lleva a cabo en una forma ordenada, sin importar las herramientas utilizadas (figura 4.2).



IV.4.3 Identificación de Requerimientos Conocidos

La determinación de los requerimientos de una aplicación es tan importante para el método de desarrollo de prototipos como lo es para los métodos del ciclo clásico de vida de desarrollo de sistemas o análisis estructurado (aunque las tácticas son diferentes). Por consiguiente, antes de crear el prototipo, los

analistas y usuarios deben trabajar juntos para identificar los requerimientos conocidos que tienen que satisfacerse. Para hacerlo determinan los fines para los que servirá el sistema y el alcance de sus capacidades. En general, sólo un analista de sistemas es el que coordina este paso.

IV.4.4 Desarrollo de un Modelo de Trabajo

La construcción de un prototipo es un proceso iterativo de desarrollo. Antes de la primera iteración, los analistas de sistemas explican el método a los usuarios, las actividades a realizar, la secuencia en que se llevarán a cabo y también discuten las responsabilidades de cada participante. Es útil comenzar el proceso de construcción del prototipo con el desarrollo de un plan general que permita a las personas conocer lo que se espera de ellas y del proceso de desarrollo. Un cronograma para el inicio y fin de la primera iteración es de gran ayuda y por lo tanto, debe elaborarse justo antes de comenzar las actividades. Sin embargo, dada la naturaleza de este método de desarrollo, es difícil y en ocasiones imposible, fijar una fecha tentativa de terminación. La experiencia con el sistema es la que determina eventualmente cuando el sistema está terminado. Para comenzar la primera iteración, usuarios y analistas identifican de manera conjunta los datos que son necesarios para el sistema y especifican la salida que debe producir la aplicación. Esto significa describir 1) los reportes y documentos que el sistema debe proporcionar y 2) el formato de cada uno de ellos. Las decisiones de diseño necesarias para desarrollar la salida del sistema cambian muy poco en relación con las tomadas en otros métodos de desarrollo. Sin embargo, con un prototipo, se espera que las especificaciones iniciales estén incompletas. En general, se necesitan entre dos y tres reuniones para establecer las especificaciones Iniciales.

METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

Asimismo, el analista estima los costos asociados con el desarrollo del prototipo. Este paso es muy importante, aunque sólo se indique una estimación del costo. Lo anterior da a la administración y a los participantes una idea de los gastos necesarios (personal, equipo y artículos de consumo) que les permite revisar el plan de desarrollo. La rapidez con la que se genera el sistema es esencial para que no se pierda el estado de ánimo sobre el proyecto y los usuarios puedan comenzar a evaluar la aplicación a mayor brevedad posible.

En el desarrollo de un prototipo se preparan los siguientes componentes:

- * El lenguaje para el diálogo o conversación entre el usuario y el sistema.
- * Pantallas y formatos para entrada de datos.
- * Módulos esenciales de procesamiento.
- * Salida del sistema.

Al construir el prototipo se deben seguir los estándares para datos que emplean la organización. La incorporación en la interfaz entrada/salida de características representativas de las que serán incluidas en el sistema final permite una mayor exactitud en el proceso de evaluación.

En esta fase no se prepara la documentación ni tampoco las especificaciones de salida o de diseño de software.

En esta etapa es más importante la rapidez con la que se construye el prototipo que la eficiencia de operación. Es por esto que el analista no intenta optimizar la velocidad de operación del sistema (este aspecto será importante más adelante).

IV.4.5 El Prototipo y el Usuario

Es responsabilidad del usuario trabajar con el prototipo y evaluar sus características y operación. La experiencia con el sistema bajo condiciones reales permite obtener la familiaridad indispensable para determinar los cambios o mejoras que sean necesarios, así como la eliminación de características inadecuadas o innecesarias.

IV.4.6 Revisión del Prototipo

Durante la evaluación, los analistas de sistemas desean capturar información sobre lo que les gusta y lo que les desagrada a los usuarios; al mismo tiempo ponen el por qué reaccionan los usuarios en la forma en que lo hacen. La información obtenida tendrá influencia sobre las características de la siguiente versión de la aplicación. Asimismo, la evaluación permite profundizar en los rasgos de los usuarios y también en los de la empresa -detalles que tienen influencia no sólo en la aplicación sino también en la forma en que será implantada-.

Los cambios al prototipo son planificados con los usuarios antes de llevarlos a cabo. Sin embargo, el analista es el responsable de realizar las modificaciones.

IV.4.7 Repetición del Proceso las veces que sea necesario

El proceso antes descrito se repite varias veces; en general, son necesarios entre cuatro y seis iteraciones. El proceso finaliza cuando los usuarios y analistas están de acuerdo en que el sistema ha evolucionado lo suficiente como para incluir todas las características necesarias o cuando ya es evidente que no se obtendrá mayor beneficio con una iteración adicional.

IV.4.8 Uso de Prototipos

Cuando el prototipo esta terminado, el siguiente paso es tomar la decisión sobre cómo proceder. Existen cuatro caminos a seguir después de evaluar la información obtenida con el desarrollo y uso del prototipo: abandonar el proyecto de aplicación, implantar el prototipo, volver a desarrollar la aplicación o comenzar con otro prototipo.

IV.4.9 Abandono de la Aplicación

En algunos de los casos, la decisión es descartar el prototipo y abandonar el desarrollo de la aplicación. Esta conclusión no significa que fuese un error emprender el proceso de desarrollo del prototipo o un desperdicio de recursos -nada más alejado de la realidad-. Más bien, la información y experiencia ganada con el desarrollo y empleo del prototipo condujo hacia una decisión de desarrollo. Es probable que usuarios y analistas hayan aprendido que el sistema era innecesario o hayan descubierto otra solución durante el proceso. También es posible que la experiencia obtenida sugiera que el enfoque utilizado no fue el más adecuado. Y, en algunas ocasiones, el evento que dio

ESTA TESIS DE GRADO
SALIR DE LA BIBLIOTECA

METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

inicio al esfuerzo de desarrollo ocurre solamente una vez. El prototipo cumplió con las necesidades inmediatas y no se espera que el evento se presente de nuevo.

IV.4.10 Implantación del Prototipo

La segunda opción es implantar el prototipo. Algunas veces el prototipo se convierte en el sistema que se necesita. En este caso, se implanta sin ninguna modificación y no se emprenden más esfuerzos de desarrollo. Esta decisión es más probable de tomarse bajo una o más de las siguientes circunstancias.

- La evolución del prototipo condujo a una aplicación que tiene las características, capacidades y desempeño requeridos.
- La aplicación será utilizada con poca frecuencia y no es importante su rapidez o eficiencia operacional.
- La aplicación no tiene efectos sobre otras aplicaciones o datos de la organización y tampoco interacciona con ellos; además satisface las necesidades de los usuarios inmediatos.
- El medio ambiente de la aplicación se encuentra en un estado de flujo; es difícil determinar necesidades a largo plazo o condiciones de operación más estables. En consecuencia no es posible justificar otras actividades de desarrollo. El prototipo es de utilidad para las condiciones actuales.

IV.4.11 Redesarrollo de la Aplicación

Cuando un prototipo tiene éxito puede proporcionar información muy amplia con respecto a los requerimientos de la aplicación y conducir a su completo desarrollo. Terminar el prototipo no significa finalizar el proceso de desarrollo. Más bien señala el comienzo de la siguiente actividad: el desarrollo completo de la aplicación.

La información recopilada durante el desarrollo del prototipo sugiere características que deben añadirse a la aplicación. También permite que los analistas y usuarios tengan mayor información para tomar decisiones relacionadas con la forma en que debe realizarse el procesamiento y la información que se debe producir. La presentación de información, incluida la forma de interactuar con el sistema, es un requerimiento importante; los datos obtenidos a partir de la experiencia con el prototipo contribuyen a considerar con mayor eficacia otras características adicionales que son necesarias.

Con frecuencia, cuando la aplicación se vuelve a desarrollar, se pone mucha atención en el mejor uso posible de los recursos del sistema. La velocidad de procesamiento y el tiempo de respuesta tienen mayor importancia, al igual que el uso eficiente de los medios de almacenamiento.

El redesarrollo de una aplicación puede presentarse como parte del método de ciclo de vida de los sistemas de información. Las dos formas más comunes de incorporar la construcción de un prototipo para la aplicación son las siguientes:

- El prototipo se emplea como una opción para la determinación de requerimientos; las características del prototipo son consideradas como los requerimientos a satisfacer en las subsecuentes actividades de desarrollo.
- El prototipo se utiliza como sustituto para el diseño e implantación de la aplicación, es decir como un esqueleto a partir del que se construye el resto del sistemas.

Cualquiera que sea el camino, la construcción de prototipos de aplicación favorece el proceso de desarrollo.

IV.4.12 Inicio de un Nuevo Prototipo

La cuarta opción es comenzar un nuevo proyecto de prototipo. Algunas veces la información ganada con el desarrollo y uso del prototipo, sugiere el empleo de un enfoque muy diferente para satisfacer las necesidades de la organización. En este caso es posible encontrar que las características de la aplicación son muy diferentes si el prototipo es inadecuado para demostrarlas y evaluarlas.

Como consecuencia de lo anterior, más que emprender directamente el esfuerzo de desarrollo total, la gerencia puede apoyar la creación de otro prototipo que permita añadir más información a la que ya se tiene disponible.

De nuevo, es importante notar que no se han desperdiciado los esfuerzos hasta este momento, aunque ellos revelen requerimientos diferentes a los anticipados. Conocerlos en esta fase del proceso, justo cuando se cambia la

METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

dirección del esfuerzo de desarrollo del prototipo, es mucho mejor que advertirlos después de invertir en él todo el esfuerzo de desarrollo e implantación.

IV.5 DEFINICION DE UNA METODOLOGIA PROPIA DE DESARROLLO

Hasta este punto no se ha explicado la metodología empleada para el desarrollo de la tesis, con la intención de presentar en primera instancia el tema a tratar y su problemática. Una vez familiarizado con el tema es más significativa la explicación de la metodología que tal vez si se hubiera presentado anteriormente no se justificaría.

Por otra parte cabe mencionar que los tipos de decisiones que constantemente toman los gerentes son variables, la información que se necesita es variable. Así pues, no es probable que se pueda encontrar una metodología que se ajuste uniformemente y convenga al sistema en desarrollo.

No existe ningún método correcto para desarrollar un sistema, pero si existen diferentes formas para producir el sistema correcto para una aplicación. En la comunidad empresarial existen muchas variaciones de los métodos expuestos anteriormente en este capítulo.

Algunos métodos tienen más éxitos que otros y esto depende de cuándo se emplean, cómo se aplican y de los participantes en el proceso de desarrollo. En ciertas ocasiones el único método adecuado será un enfoque paso por paso, comparable con el ciclo de vida tradicional. En otros casos, el desarrollo de prototipos es el único método que tiene sentido. En otras situaciones se combinan los métodos y, además, los usuarios desarrollan parte de la aplicación.

METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

El indicador definitivo del éxito de un método de desarrollo de un sistema, en particular, es aquel que se refiere a los resultados obtenidos y no a la "precisión" teórica del método.

Es por estas razones que la metodología que fue utilizada en el "Sistema de Control de Presupuestos bajo ambiente Visual Basic" se basa en los conceptos y principios de Ingeniería de Software.

Las metodologías de Ingeniería de Software no deben verse como alternativas excluyentes, ya que realmente son complementarias y dependen de la naturaleza del sistema a desarrollar.

En Ingeniería de Software, los pasos para aplicar los métodos, herramientas y procedimientos adecuados en la elaboración de un producto dependen de las características y naturaleza del proyecto, las herramientas con que se cuente para trabajar y de los controles de entrega requeridos al analista.

Para la presentación de esta tesis se eligió la utilización de la hibridación entre el ciclo de vida clásico, construcción de prototipos y análisis estructurado dada las características propias del proyecto.

Así tenemos que las fases de este sistema son:

- Análisis del Sistema.
- Diseño del Sistema.
- Desarrollo del Sistema.
- Pruebas e Implantación.

En la figura 4.3 se ejemplifica la metodología utilizada:

METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

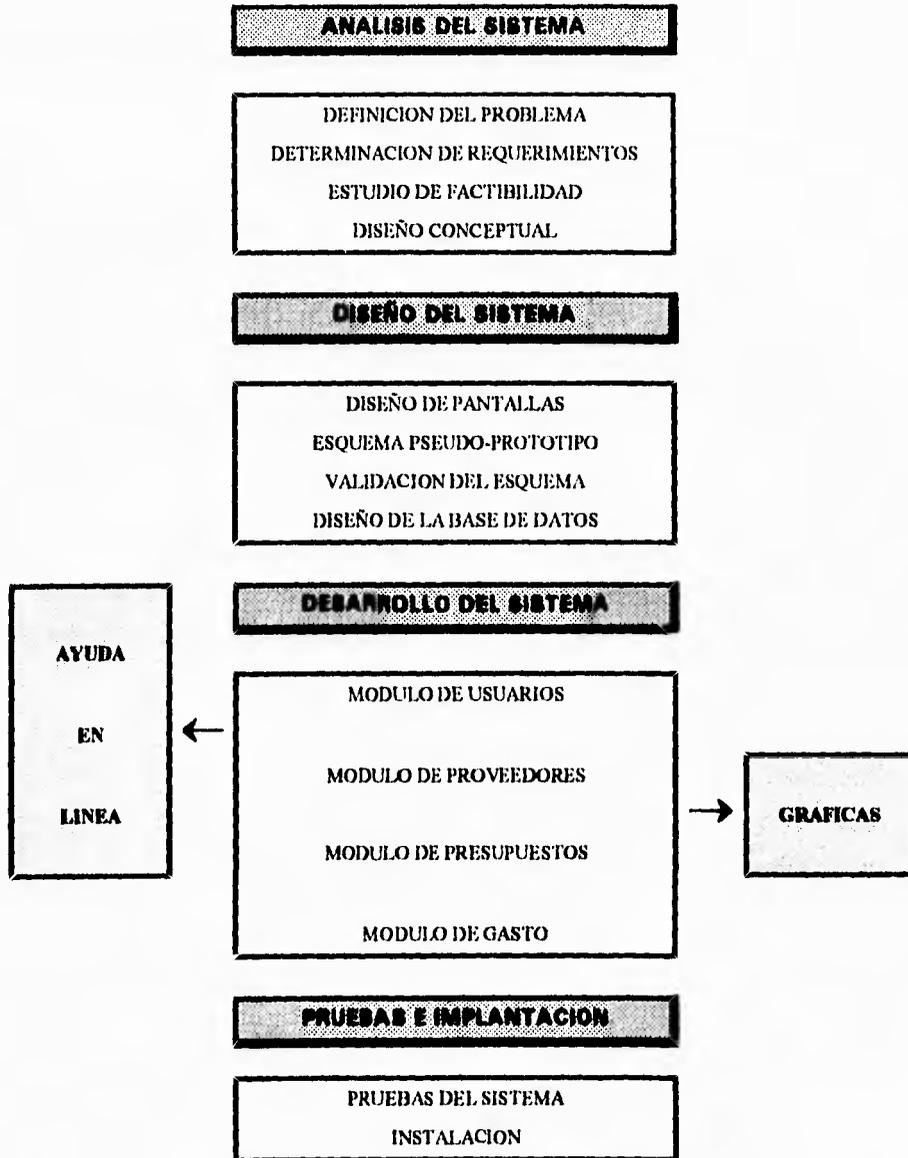


Figura 4.3
Fases de la metodología propuesta

METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

Se puede observar que las fases para el desarrollo de nuestro sistema corresponden a las etapas del método del ciclo de vida tradicional, pero cada una de estas fases cuentan con subfases que nos ayudarán a tener una mejor comprensión y desarrollo de nuestro sistema.

A continuación de una manera breve, haremos una descripción de cada una de estas fases y subfases, explicando el porqué de utilizarlas, así como algunas de las ventajas que consideramos fueron importantes para el desarrollo del sistema.

Dará inicio la elaboración de nuestro sistema con un análisis del mismo. El objetivo de esta fase es obtener una definición clara del sistema que será desarrollado e implementado, así como el de analizar con detalle cada uno de los aspectos que han de contemplarse con el nuevo sistema, buscando su justificación de acuerdo con los objetivos, restricciones y limitaciones impuestas por el usuario y por la alta dirección.

En el análisis de todo sistema tiene que seguirse una secuencia lógica de actividades, la cual daremos inicio con una definición clara y precisa del problema. Esta subfase empieza con la definición del problema y el objetivo. El establecimiento y la cuantificación del objetivo es de gran importancia. La toma de decisiones con la que termina cualquier análisis, depende del objetivo propuesto. Si no se tiene claramente definido el problema y establecido el objetivo, la actividad de uno como analista se convierte en un fin y no en un medio.

METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

En la subfase de determinación de los requerimientos; durante esta subfase identificaremos los requerimientos de los usuarios de la compañía, hardware y software, ya que de estos dependen la optimización de recursos y que el sistema cumpla con las necesidades de la gente que los solicitó.

Seguimos con la subfase de estudio de factibilidad, haremos un estudio sobre la viabilidad del desarrollo e implantación del sistema, ésta nos proporcionará una indicación sobre la conveniencia del sistema en relación con los costos y beneficios del mismo, también nos ayudará a identificar y programar las actividades requeridas para llevar a cabo al sistema a buen fin. Con lo antes mencionado daremos un conjunto de características que se obtendrán al tener el sistema implantado en la compañía.

En base a esto obtendremos un diseño conceptual del mismo, que mostrará su estructura como el flujo de datos, procesos y salidas correspondientes, este diseño será un diagrama el cual permitirá ver a grandes rasgos los diferentes módulos con los que cuenta el sistema, de cómo será vista por quienes operarán y manipularán el sistema. Aquí tomaremos un elemento del análisis estructurado, que es el diseño estructurado, el cual también utiliza un método gráfico para la descripción del sistema, formula las especificaciones funcionales para los módulos de software. Asimismo también se incluye una descripción de la interacción entre los diferentes módulos pero sin mostrar la lógica interna en cada uno de estos.

Continuando con las fases de nuestro método de desarrollo, tenemos lo que es el diseño del sistema; en esta fase desarrollaremos la estructura modular del sistema, para definir las interfases entre los diferentes módulos y programas, así como su estructura y flujo de datos. Esta fase requiere que se trabaje y se participe muy de cerca con el usuario, debido a esto nos auxiliaremos del

METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

método del prototipo de sistemas, por las características y ventajas que este ofrece. Las subfases que contendrán esta fase son: el diseño de pantallas (como se mencionó el en capítulo III, gracias a las facilidades que ofrece Visual Basic), se puede hacer un diseño en un corto tiempo, y dedicarle mayor tiempo a la validación de requerimientos. Esto nos llevará a definir algo que llamaremos Esquema Pseudo-prototipo (EP), que será el de diseñar un prototipo de un sistema previo, esto con la finalidad de simular la captura de datos.

En la tabla 4.1 se muestran las características que encontraremos al utilizar estas subfases con la participación constante del usuario.

TABLA 4.1 CARACTERISTICAS DEL DISEÑO DE PANTALLA Y ESQUEMA PSEUDO-PROTIPO

PARTICIPACION DEL USUARIO

- * Los datos proporcionados por el usuario durante el desarrollo, es una información valiosa sobre el diseño.
- * La experiencia ganada a través del empleo del Esquema Pseudo-prototipo por los usuarios, misma que genera reacciones inmediatas por parte de éstos, evita sorpresas desagradables en la fase de desarrollo; asimismo facilita los cambios y mejoras que son deseables.
- * Se esperan sugerencias que serán bien recibidas, para efectuar cambios en las especificaciones y modificaciones en las características del sistema.

METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

En la tabla 4.2 se muestran las características del desarrollo y uso de los recursos.

TABLA 4.2 USO DE LOS RECURSOS Y CARACTERISTICAS DEL DESARROLLO

- * Se recalca la determinación de los requerimientos correctos; esto evita el desarrollo e instalación de un sistema equivocado.
- * Se concede mayor importancia a los requerimientos del sistema, no a las necesidades secundarias de procesamiento; el desarrollo gira en torno a lo que es esencial.
- * Comparado con otras opciones de desarrollo, facilita la administración del tiempo de los usuarios y analistas.
- * Los requerimientos de información son determinados antes de iniciar el desarrollo de la aplicación; de esta manera se evitan errores y retrasos costosos.
- * La eficiencia del proceso de desarrollo asegura la validación temprana de interfaces y características operacionales.
- * La participación de personal clave fomenta la comunicación durante el desarrollo.
- * Los usuarios ganan experiencia durante el desarrollo, lo que facilita el entrenamiento previo en el uso del sistema.

Otra subfase es la validación del EP, ésta nos permitirá evaluar el flujo de las pantallas del sistema y si realmente se están considerando todas las variables y requerimientos que el usuario previamente discutió.

Una vez, habiendo hecho la validación de las pantallas y considerando los requerimientos, procederemos a la definición de las bases de datos, ya que estas no permitirán, hacer la manipulación de datos que serán utilizados en los diferentes módulos.

METODOLOGIAS PARA EL DESARROLLO DE SISTEMAS

También definiremos el modelo entidad-relación de que conforman a nuestro sistema. Una vez que se ha determinado las entidades y relaciones, podemos concentrarnos en los requerimientos de datos para cada entidad. Construiremos un diagrama de estructura de datos a partir de la información obtenida, al preparar el diagrama de relación entre las entidades (las entidades, y la dirección y alcance de las relaciones entre ellas).

Después de haber efectuado el análisis y el diseño, la siguiente fase es el desarrollo del sistema, aquí se obtendrá la codificación y programación de los diferentes módulos, esto se desarrollará con el uso de algún lenguaje de programación previamente definido. Se hará el desarrollo de los módulos según su importancia y la forma en que éstos interactúan entre si.

Por último las fases de pruebas e implantación, en esta se llevará a cabo las pruebas individuales por programa, así como las pruebas de integración de los módulos que conforman el sistema, para lograr su implantación y liberación dentro de las instalaciones de la compañía.

Con la fusión de estos métodos y la definición de algunas ideas propias, se espera que la metodología propuesta satisfaga las expectativas del nuevo sistema.

CAPITULO V

ANALISIS DEL SISTEMA

V.1 INTRODUCCION

Cualquier sistema implementado por el hombre, pasa por diferentes fases. El ciclo de vida de un sistema empieza en un grupo de planeación, al principio como una idea conceptual muy general, que posteriormente toma una forma en un grupo de diseño. Si se considera factible el proyecto desde diversos puntos de vista, se desarrolla e implementa para ponerse en marcha y culminar con su operación, es decir el sistema satisface una necesidad.

El costo económico y la complejidad tecnológica de los sistemas que actualmente implementa el hombre, es tan grande, que es necesario planear con todo cuidado estos proyectos, analizar diversas alternativas, tomando en cuenta todos los aspectos de un proyecto.

En resumen, se requiere una secuencia lógica de pasos detallados de análisis, para resolver estos problemas.

En el presente capítulo se analizan y desarrollan los pasos que se siguieron en esta fase de análisis del sistema de control de presupuestos para una mejor concepción de su desarrollo, los cuales se mencionan a continuación:

- Definición del problema.
- Determinación de los requerimientos.
- Estudio de factibilidad.
- Diseño conceptual.

ANALISIS DEL SISTEMA

V.2 DEFINICION DEL PROBLEMA

Las instituciones se deben adaptar a un medio en el que la asignación de recursos constituye un problema a resolver. Por lo tanto es sumamente importante considerar a los presupuestos como un sistema, porque esto implica un proceso continuo durante el año que es la clave para buenos presupuestos en cualquier negocio, los presupuestos no son una función financiera realizada por los departamentos de presupuestos, tampoco pueden considerarse como pronósticos, si con ello queremos decir la predicción de eventos futuros en vez de la planeación para la obtención de un resultado y del control para aumentar al máximo las probabilidades de obtener ese resultado.

En muchas de estas instituciones el proceso de planeación y elaboración de presupuestos se realiza a nivel financiero, y los gerentes de mayor jerarquía sólo participan en la toma de decisiones y en la formulación del presupuesto mismo.

En el caso de la compañía la asignación y planeación del presupuesto se realiza mediante una junta de dirección llevada a cabo cada año en donde entre otras cosas se ven los logros y cumplimientos que se tuvieron para la compañía. Por otra parte los directores llevan su plan de trabajo para que les sea asignado el presupuesto del año entrante. Para la empresa esta asignación es considerada de acuerdo a las expectativas de la misma. Una vez hecha la asignación cada director es responsable de su presupuesto y lo único que tiene que hacer es presentar a contabilidad una justificación de los gastos invertidos.

ANÁLISIS DEL SISTEMA

Si en algún momento el director general desea saber el estado que guarda la relación presupuesto-gasto de las diferentes áreas, lo que tendría que hacer es pedirlo a cada una de éstas y a su vez cada director de área recurrir a su control personal y presentar el reporte al director general.

Con lo antes mencionado se observa que no existe una forma homogénea o estandar de llevar el control de presupuesto-gasto, además, se requiere de la participación de todos los gerentes de las diferentes áreas de la que consta la compañía.

Por lo cual surge la necesidad de tener un sistema que estandarice la obtención del presupuesto, que de seguridad y principalmente que cumpla con los requerimientos de cada una de las áreas, un sistema diseñado a los gerentes que están a niveles inferiores, ya que éstos son quienes gastan el dinero proporcionando un servicio; ellos son los expertos dentro de su correspondiente área de actividad, y se deben familiarizar con los procesos de planeación y presupuestación, mas aún la alta gerencia debe ser capaz de recibir sus recomendaciones, análisis y fundamentalmente tener en cualquier momento la información del control de presupuesto que se lleva hasta el momento.

V.2.1 Objetivo

El sistema de control de presupuestos es un medio, no puede esperarse que éste en sí, resuelva todos los problemas administrativos, con lo cual el objetivo es, proporcionar un medio para poder planear el presupuesto y llevar un control estricto del gasto teniendo siempre presente la relación presupuesto-gasto, para que la gerencia pueda tomar decisiones, decidir las medidas necesarias para resolver los problemas, asignar y aprovechar los recursos de la organización en forma eficiente.

ANALISIS DEL SISTEMA

V.2.2 Plan de Trabajo

Para poder llevar a cabo las actividades para el desarrollo del sistema en una forma eficiente nos apoyaremos en el Diagrama de Gantt, técnica simple que consiste en representar mediante barras horizontales las actividades que participan en el proyecto.

Inicialmente se dispusieron las barras en el primer renglón, estimando las actividades a desarrollar. Como medida de control se llenó el segundo renglón sobre la realidad del tiempo incurrido, esto ayudó a evaluar lo planeado con lo real y sirvió como indicador de la eficiencia del proyecto.

Este diagrama se mostrará al termino de la presente tesis para observar el tiempo estimado, así como el tiempo real de desarrollo.

V.3 DETERMINACION DE LOS REQUERIMIENTOS

La determinación de los requerimientos, es el estudio de un sistema para conocer cómo trabaja y dónde es necesario efectuar mejoras. Los estudios de sistemas dan como resultado una evaluación del funcionamiento de los métodos empleados y si es necesario o posible realizar ajustes.

Dado que los analistas del sistema no trabajan como gerentes o empleados en los departamentos de los usuarios (mercadotecnia, compras, producción o contabilidad) no tienen los mismos conocimientos, hechos y detalles que los usuarios y gerentes de esas áreas. Por consiguiente, el primer paso del analista es comprender la situación.

ANALISIS DEL SISTEMA

Antes de empezar a determinar los requerimientos creemos oportuno, definir algunos conceptos manejados por el usuario, que son importantes para tener una mejor comprensión de sus requerimientos.

Concepto:

Nombre que le da el usuario a cada rubro de la planeación, al cual se le podrá asignar un cierto presupuesto, y por ende recibirá ciertos gastos.

Nivel:

Concepto cabecera que es capaz de agrupar a otros conceptos y al cual se le puede asignar un presupuesto y capturar gasto. Los niveles son definidos por el usuario. El sistema cuenta con dos niveles predefinidos:

Nivel 1 Dirección General.

Nivel 2 Areas de la empresa.

Módulo:

Son las diferentes opciones con las que cuenta el sistema siendo éstas: Presupuesto, Gasto, Proveedores y Usuarios.

Presupuesto Anual:

Capital máximo asignado al nivel el cual será ejercido durante el transcurso del año.

Presupuesto Asignado:

Capital máximo disponible para cada uno de los conceptos que integran el nivel.

ANALISIS DEL SISTEMA

Por presupuestar:

Es la diferencia entre el presupuesto anual y la suma de cada uno de los conceptos asignados a ese nivel.

Gasto anual:

Es la suma de todas las erogaciones del nivel y sus subniveles definidos.

Gasto por Ejercer:

Diferencia entre el presupuesto anual y gasto anual.

Gasto Ejercido:

La suma de todas las erogaciones del mes en el nivel y sus subniveles definidos.

Para poder decidir el hecho de realizar el sistema de control de presupuestos, fué necesario identificar los requerimientos del usuario; Hardware, Software, y Recursos Humanos, ya que de éstos depende la optimización de recursos y la satisfacción de las necesidades de las personas que lo solicitaron.

Para llevar a cabo esto se realizó una serie de entrevistas con el usuario que definieron sus necesidades, considerando entre las más importantes las siguientes:

- Contar con dos módulos principalmente: Presupuesto y Gasto.

En el caso del presupuesto.

- Que la planeación del presupuesto sea definida hasta el nivel que uno desee (un máximo de 20 niveles).
- Que tenga una estructura de árbol.
- Que se pueda dividir en 4 zonas principales:
 - a) Zona de Información de nivel. Que proporcione el nombre del nivel, la cantidad asignada presupuestada y por presupuestar que en ese momento se este consultando.
 - b) Zona de conceptos. Poder definir los conceptos que se deseen con su presupuesto asignado.
 - c) Zona de comandos. Poder tener la capacidad de crear y eliminar tanto niveles como conceptos.
 - d) Zona de información específica de cada concepto. Hacer la descripción del concepto, así como la asignación del presupuesto, teniendo la capacidad de poder asignar este presupuesto a los diferentes meses del año. En el caso de que la suma de éstos sobrepase el presupuesto, tener la posibilidad de detectarlo.

ANALISIS DEL SISTEMA

En el caso del Gasto.

- Que también sea manejado como estructura de árbol.
- Manejado y controlado bajo la estructura definida en el presupuesto.
- A cada concepto poderle asociar su gasto.
- Al igual que el presupuesto, poder hacer la asignación del gasto ejercido mensualmente. En caso de que sobrepase lo presupuestado hacerlo notar.
- Poder ver las facturas que tiene asociado cada concepto.
- Tener la opción de capturar, en el caso de una factura, el proveedor y el mes en que se realizó. Tomando en cuenta esto se consideró en definir 2 catálogos, que son: proveedores y usuarios.

En el caso del catálogo de proveedores, poder realizar altas, bajas y cambios contando con información como: Nombre, Dirección, Teléfonos, RFC y el nombre de la persona responsable (con la que se realizó el trato).

En el caso del catálogo de usuarios tener la información relacionada con su Nombre, Puesto que desempeña en la compañía, a que área pertenece, así como el otorgamiento de derechos y restricciones, los cuales permitirán tener acceso a ciertos módulos del sistema.

Además que el sistema cuente con:

- Un alto nivel de seguridad. Esto quiere decir, que los usuarios con los derechos previamente definidos tengan acceso a los diferentes módulos. El sistema hará la asignación de las claves y password de los usuarios y proveedores teniendo en cuenta que estos serán únicos.

ANALISIS DEL SISTEMA

- Que estos niveles de seguridad, permitan al usuario definir derechos de acceso y con esto controlar ellos mismos a las personas que tiene acceso a su información.
- Que el sistema realice un buen trabajo y con un tiempo de respuesta satisfactorio.
- Que realice una estricta validación de las entradas de datos.
- Que presente flexibilidad en la emisión de informes (reportes), gráficas y consultas con interactividad al usuario final.
- Que sea amigable y fácil de usar.
- Que la tecnología utilizada sea de punta y con portabilidad.
- Que el precio del sistema no sea muy alto.

V.3.1 Hardware

El hardware es el equipo que se necesita para que el sistema funcione óptimamente, para evaluar esto, es necesario tener la clara concepción de qué es lo que el usuario espera del sistema.

En el caso específico de este sistema, no se contempla la evaluación y compra de equipo, debido a que, en la compañía donde será instalado cuenta con el equipo que a continuación se describe:

ANALISIS DEL SISTEMA

EQUIPO	CARACTERISTICAS	CANTIDAD
TIPO DE RED	Red Token Ring	1
SISTEMA OPERATIVO	Windows NT	1
SERVIDOR	Pentium 66 MHz	1
NODOS DE RED	Procesador 486 a 33MHz con 4 Mbytes de Memoria Real y 120 Mbytes de disco duro.	20

Las características del equipo antes mencionado garantizan la efectividad y buen desempeño del sistema.

V.3.2 Software

El software se refiere a los programas de aplicación que se pueden utilizar para desarrollar un sistema, éstos se clasifican de acuerdo a: efectividad en el desempeño, eficiencia, facilidad de uso, portabilidad, calidad de la documentación, ayuda en línea, costo, disponibilidad, recursos de hardware, ámbito de trabajo y soporte.

En cuanto al programa de aplicación para el desarrollo del sistema de control de presupuestos existe la restricción por parte de la compañía de utilizar como herramienta gráfica a Visual Basic bajo ambiente Windows.

V.3.3 Recursos Humanos

Los recursos humanos son las personas a quienes va dirigido el sistema, es decir los usuarios del mismo, en este caso constituido por diversos grupos, tales como: Director general, los gerentes de los diferentes departamentos y las secretarias.

V.4 ESTUDIO DE FACTIBILIDAD

En este punto se trata un aspecto de suma importancia. "La factibilidad del sistema de control de presupuestos" llevada a cabo mediante una investigación preliminar donde se examinó la factibilidad del proyecto; es decir, la posibilidad de que el sistema sea benéfico a la compañía.

Factibilidad Operativa.

Para poder determinar si el sistema de control de presupuestos será benéfico para la compañía había que determinar si el sistema cumplía con los requerimientos operativos de ésta, por lo tanto había que cuestionarse si una vez instalado y desarrollado el sistema se iba a trabajar con él. Algunas preguntas para observar si había obstáculos fueron:

ANALISIS DEL SISTEMA

¿Existe suficiente apoyo para el proyecto por parte de la gerencia?,
¿También de los usuarios?.

En este aspecto se puede mencionar que los directores son los principalmente beneficiados, ya que ellos serán los usuarios de dicho sistema. Por otra parte es importante mencionar que no existe hasta el momento un sistema que de un panorama del presupuesto en una forma estandar; cada director se vale de sus medios para dar solución.

¿Se han involucrado los usuarios en la planeación y desarrollo del proyecto?

Hasta este momento han sido los directores los que de alguna forma han planeado el sistema que requieren, anteriormente cada director tenía que realizar y determinar su presupuesto según sus necesidades, con esta experiencia lo único que han hecho es discutirla entre los diferentes directores para poderla estandarizar.

¿Que características podemos encontrar al tener el sistema de control de presupuestos instalado en la empresa?

- a) Tener una visión del futuro en cuanto objetivos a realizar, establecimientos de metas y estrategias, desarrollo de programas, conocimiento del mercado de productos, procurando una directriz en el desarrollo de las operaciones.**

- b) Ayudar a la toma de decisiones y obligar a la consideración temprana de políticas básicas.**

ANALISIS DEL SISTEMA

- c) Forzar a la dirección a planear un uso más económico de la mano de obra, el material, las instalaciones y el capital.**
- d) Exigir la participación de todo el personal directivo para fijar metas y planes de una manera armoniosa de tal manera que se logren los objetivos departamentales y generales de la empresa.**
- e) Es un estándar que ayuda a detectar eficiencias e ineficiencias en los departamentos operativos y no operativos de la empresa.**
- f) Permitir un autoanálisis periódico de la compañía, y la corrección de la diferencia entre lo presupuestado y lo realizado.**
- g) Obligar a establecer una asignación definida de responsabilidad en cada función de la empresa.**
- h) El empeño y esfuerzo que aportan las personas que intervienen en el control presupuestal, ayudará a mostrar y desarrollar aptitudes e iniciativas, obteniendo una mejor calidad del personal ejecutivo.**

En resumen, las características, que bien podemos mencionar como ventajas son la planeación y control de las operaciones, de tal manera que permitan el logro de los objetivos deseados: el equilibrio en la estructura financiera, el aprovechamiento de los recursos y su óptima obtención, la medición de la eficiencia y la delimitación de responsabilidades dando lugar a una toma de decisiones adecuada y oportuna.

ANALISIS DEL SISTEMA

Factibilidad Técnica.

Los aspectos técnicos a los que teníamos que enfrentarnos fueron:

¿Existe la tecnología necesaria para hacer lo que sugieren?

Como antes se mencionó la compañía cuenta con el hardware y software para la realización y uso del sistema, que garantizan el buen desarrollo y funcionamiento del mismo, considerando que el software utilizado es una restricción por parte de la compañía.

¿El sistema propuesto y sus componentes proporcionarán respuestas adecuadas a las preguntas sin importar el número o ubicación de los usuarios?.

¿Se puede agrandar el sistema si se desarrolla?.

Es importante mencionar que las expectativas de los directores para el uso del sistema en primera instancia han sido las que se mencionaron en los requerimientos, pero cabe contemplar que es una compañía que se encuentra a la vanguardia por lo cual, se requiere de un sistema precisamente que de opción a seguir creciendo en cuanto al cumplimiento de las necesidades de la empresa.

V.5 DISEÑO CONCEPTUAL

El diseño conceptual establece un diseño más completo orientado al usuario para la aplicación. Hace énfasis en la aplicación tal como es vista por quienes operarán y manipularán el sistema. Establece las funciones de entrada y salida que ejecutará la aplicación. En términos generales, el diseño conceptual trata las funciones reales de procesamiento como "cajas negras".

En la figura 5.1 se muestra los diferentes módulos con los que cuenta el sistema, así como la interacción del usuario con el mismo.

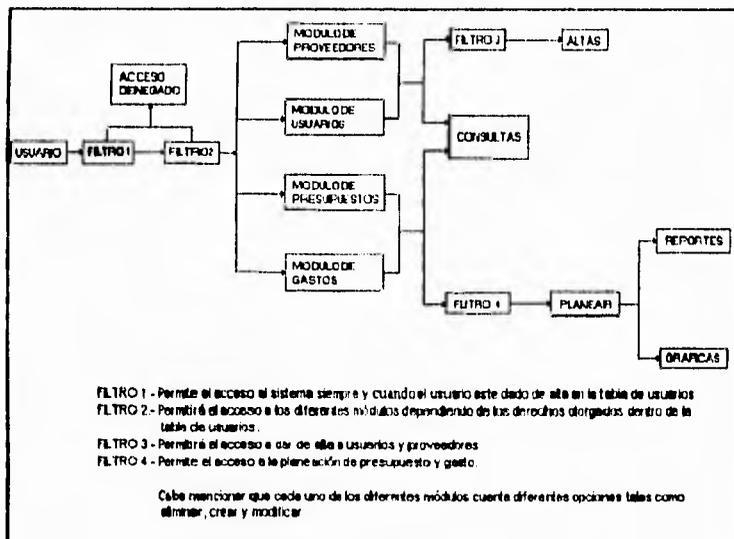


Figura 5.1
Diseño conceptual

108

CAPITULO VI

DISEÑO DEL SISTEMA

110

VI.1 INTRODUCCION

A este momento se han definido de manera general con el usuario los objetivos y características generales del sistema.

En todo proceso de desarrollo de un sistema se tiene que llegar al diseño del mismo, las maneras de afrontar este diseño son muy diversas, lo aprendido al momento nos lleva al punto de decidirlo dependiendo del tipo de lenguaje de programación que se utilizará.

La afirmación anterior pareciera contraponerse con la idea de que un buen diseño no depende de la herramienta. El diseño conceptual mostrado anteriormente nos dice que es lo que el sistema debe hacer, pero no nos dice como lo va a hacer, ni toma en cuenta las ventajas o inconvenientes que pudiera tener en este caso Visual Basic.

Nuestro equipo de trabajo tomó como base de diseño la característica visual de Visual Basic, de allí viene su nombre.

La poderosa interfaz gráfica que nos ofrece Visual Basic nos permite proponer un rápido diseño de pantallas y con ello algo que nosotros llamamos Esquema Pseudo-prototipo (EP).

Los pasos que cubriremos dentro de esta etapa de diseño serán:

- Diseño de pantallas.
- Esquema Pseudo-prototipo.
- Validación del esquema.
- Diseño de la base de datos.

DISEÑO DEL SISTEMA

VI.2 DISEÑO DE PANTALLAS

El Sistema de Control de Presupuestos tiene definidas 6 pantallas básicas.

- Pantalla Principal.
- Pantalla de Catálogo de Proveedores.
- Pantalla de Catálogo de Usuarios.
- Pantalla de Presupuestos.
- Pantalla de Consulta General de Gasto.
- Pantalla de Captura de Gasto.

VI.2.1 Fases del Sistema

Durante el proceso de diseño del sistema se definieron tres fases en las cuales el usuario, interactúa con las pantallas dinámicamente, dichas fases se muestran en la figura 6.1.

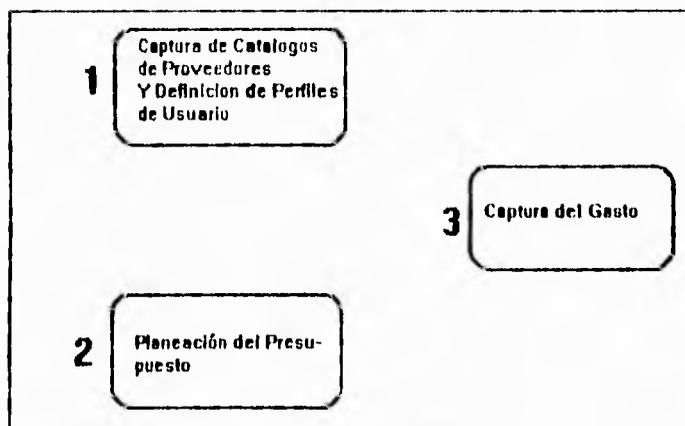


Figura 6.1

Fases de interacción del sistema

Estas fases hacen realmente sencillo al usuario el uso del sistema, dado que las pantallas son pocas y totalmente gráficas.

VI.2.2 Diseño de la Pantalla Principal

Esta pantalla cumple la función de menú principal del sistema, y permite al usuario tener una visión general del sistema y sus opciones.

Su definición básica consta de 4 botones de comando que nos darán acceso a otras opciones del sistema y un botón para salir del sistema.

Su presentación es la siguiente:

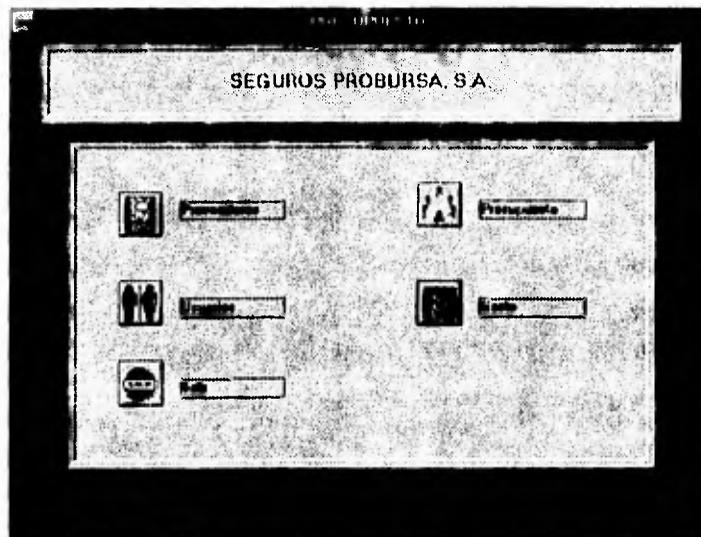


Figura 6.2
Pantalla Principal

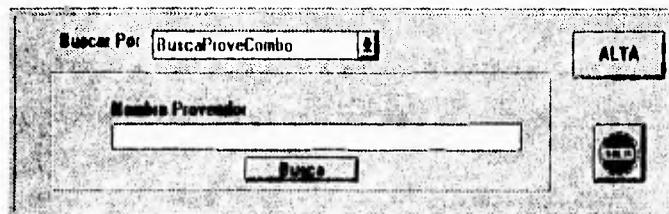
DISEÑO DEL SISTEMA

VI.2.3 La Pantalla de Catálogo de Proveedores

Dentro del catálogo de proveedores las funciones principales serán altas, bajas y cambios de un proveedor, la pantalla de proveedores tiene un diseño sencillo y claro.

Definiendo dos partes principales de la pantalla.

- a) El área de tipo de búsqueda

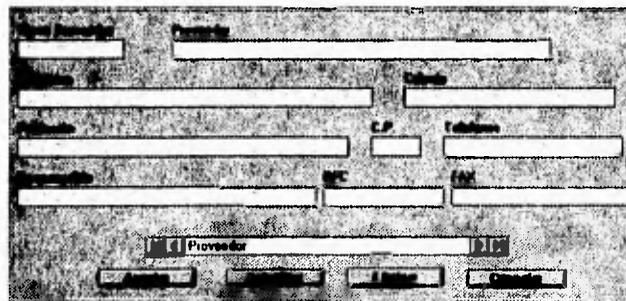


The screenshot shows a search interface with a dropdown menu labeled 'Buscar Por' containing the text 'BuscaProveCombo'. To the right is an 'ALTA' button. Below this is a text input field labeled 'Nombre Proveedor' with a 'Busca' button underneath it. A circular icon is visible on the right side of the form.

Figura 6.3
Area de búsqueda
Pantalla de Proveedor

Esta área esta definida para que el usuario seleccione el tipo de búsqueda para elegir proveedor, o si desea dar de alta a uno nuevo.

- b) Area de consulta y modificación de datos.



The screenshot displays a form for data entry and modification. It includes several input fields for 'Nombre', 'C.P.', 'Teléfono', and 'RFC'. At the bottom, there is a dropdown menu labeled 'Proveedor' and four buttons: 'Alta', 'Baja', 'Modificar', and 'Consulta'.

Figura 6.4
Area de consulta y modificación
Pantalla de Proveedor

Como se puede observar, esta área contiene todos los datos generales del proveedor.

La pantalla de proveedor esta diseñada para tener una interacción directa con la pantalla de captura del gasto (ver pantalla de gasto). Su presentación completa se muestra en la figura 6.5.

The screenshot shows a web-based form for managing suppliers. At the top, there is a search bar labeled 'Busca/Inve/Combo' and a button 'ATA'. Below this is a section for 'Nuevo Proveedor' with a 'Busca' button. The main form consists of several input fields: 'Clave Proveedor', 'Proveedor', 'Dirección', 'Colonia', 'Población', 'C.P.', 'Teléfono', 'Responsable', 'RFC', and 'FAX'. At the bottom, there is a table header 'Proveedores' with a 'Busca' button and an 'Eliminar' button.

Figura 6.5

Pantalla de Proveedor

VI.2.4 Pantalla de Catálogo de Usuario

La pantalla de catálogo de usuario tiene como fin que cada director pueda definir sus propios usuarios, los cuales podrán trabajar dentro de las áreas que él les haya asignado. Cabe aclarar que un usuario podrá dar derechos sólo a las áreas a las que él tiene derecho.

En esta pantalla se definieron tres zonas:

DISEÑO DEL SISTEMA

a) Area de búsqueda y alta.



Figura 6.6
Area de búsqueda y alta
Pantalla de Usuario

Esta área guarda una diferencia en su diseño, con el catálogo de proveedores dado que en el caso de proveedores la lista tenderá a ser muy grande por lo que mostrarlos en un combo resultaría inmanejable, en cambio para el caso de usuarios, éste sólo mostrará los usuarios que hayan sido definidos por el dueño de la clave que esté en ese momento dentro del sistema.

b) Area de consulta de datos generales de usuario.

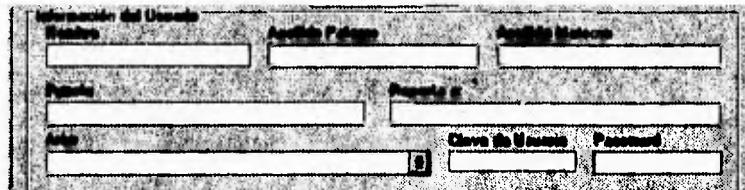


Figura 6.7
Area de consulta de datos
Pantalla de usuario

Esta zona guarda un diseño muy parecido al de proveedores con la particularidad que el sistema asigna la clave de usuario y el usuario tiene oportunidad de elegir su password.

c) Area de definición de perfil de usuario.

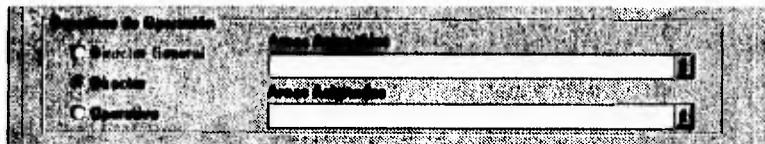


Figura 6.8

Area de perfil de usuario

Pantalla de usuario

Esta zona esta diseñada para que el administrador defina los derechos que el usuario tendrá dentro del sistema, para esto están predefinidos tres perfiles de usuario:

- El perfil de director general, en este caso el usuario tendrá derechos a todas las opciones del sistema, y podrá ver todas las direcciones que están definidas en él.
- El perfil de director de área, en este perfil el usuario podrá ejercer todas las opciones del sistema pero sólo dentro de el área o las áreas a las que tenga acceso.
- El perfil de operativo podrá sólo capturar facturas dentro de captura de gasto, esto sólo dentro de las áreas a las que tenga acceso.

Los usuarios con perfil de Director General y Director podrán definir sus propios usuarios sólo con derechos a las direcciones a las que tengan acceso.

DISEÑO DEL SISTEMA

La pantalla de usuario se muestra en la figura 6.9

USUARIO MuestraUsuarioCombo

ALTA USUARIO

Información del Usuario

Nombre Apellido Paterno Apellido Materno

Puesto Región

Área Clave de Usuario Password

ÁreaUsuarioCombo

Derechos de Operación

Director General

Director

Operativo

Áreas Asignadas

DerechosUsuarioCombo

Áreas Asignadas

ÁreasAsignadaCombo

Salir Eliminar Cancelar

Figura 6.9
Pantalla de Usuario

VI.2.5 Pantalla de Planeación del Presupuesto

En esta pantalla el usuario realizará toda la planeación del presupuesto, con todas las opciones comentadas en el análisis, como son:

- Crear conceptos.
- Definir niveles.
- Visualizar el presupuesto por nivel y por concepto, etc.

La pantalla de presupuesto se divide en 4 zonas principales:

a) Area de información de nivel

The screenshot shows a form with the following elements:

- A label "Nivel:" followed by a text input field.
- Below this, two columns of data:

Presupuesto Anual	Por Presupuestar
<input type="text"/>	<input type="text"/>

Figura 6.10
Area información de nivel
Pantalla de presupuesto

Como se puede observar en la figura 6.10 los datos que nos muestra, nos permite darnos cuenta el nombre del concepto que es cabecera de nivel, el presupuesto que tiene asignado y el monto de que le resta por presupuestar.

b) Area de concepto

El área de concepto nos muestra todos los conceptos que están definidos dentro del nivel, así como el monto de presupuesto que tiene asignado cada uno de ellos, también nos muestra el monto de dinero por presupuestar que maneja cada concepto.

The screenshot shows a table with the following data:

Concepto	Presupuesto Anual	Por Presupuestar
SECCION DE ATENCION	10000000	10000000
SECCION COMERCIO	650000	412000

Figura 6.11
Area de concepto
Pantalla de presupuesto

DISEÑO DEL SISTEMA

c) Área de comandos

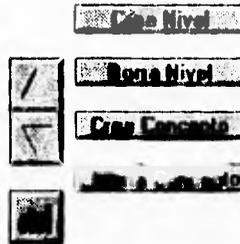


Figura 6.12

Área de comandos Pantalla de presupuesto

Dentro del área de comandos se presentan todas las herramientas con las que cuenta la pantalla de presupuestos, como son:

- Crear un nuevo nivel.
- Borrar un nivel.
- Eliminar un concepto.
- Desplazarse entre niveles.
- Generar gráficas.

Una descripción detallada de cada una de estas funciones se realizará durante la etapa de desarrollo del sistema

d) Área de Información específica del concepto

El área de información específica del concepto nos muestra el presupuesto detallado del concepto señalado en el área de conceptos.

La información que nos proporciona el área de información específica de concepto se visualiza en la figura 6.13.

The screenshot shows a web-based interface for budget management. At the top, there are two input fields labeled 'Presupuesto Anual' and 'Presupuesto Asignado'. Below these is a section titled 'Presupuesto Mensual' containing a 3x4 grid of input boxes for the months: Ene, Feb, Mar, Apr, May, Jun, Jul, Ago, Sep, Oct, Nov, and Dic. At the bottom of this grid are two buttons labeled 'Aceptar' and 'Cancelar'.

Figura 6.13

**Area de información específica del concepto
Pantalla de presupuesto**

La pantalla de presupuesto nos proporciona los iconos de salir y de acceso al gasto. El icono de acceso al gasto se representa como un billete con alas. La pantalla de presupuesto se muestra en la figura 6.14.

This screenshot shows a more complex budget management interface. At the top, there is a 'Nivel:' dropdown menu. Below it are two input fields for 'Presupuesto Anual' and 'Por Presupuestar'. A large, mostly blank table occupies the center of the screen. To the right of the table is a vertical toolbar with several icons, including one that looks like a winged bill. At the bottom, there are two input fields for 'Presupuesto' and 'Presupuesto Asignado', followed by a 'Presupuesto Mensual' section with a 3x4 grid of monthly input boxes (Ene to Dic) and 'Aceptar' and 'Cancelar' buttons.

Figura 6.14

Pantalla de presupuesto

DISEÑO DEL SISTEMA

VI.2.6 Pantalla de Consulta General de Gasto

Esta pantalla es totalmente informativa y en ella podrá el usuario visualizar las áreas de información de nivel, de conceptos y de Información específica del concepto que la pantalla de presupuesto maneja, adicionándole a ésta la información del gasto, como es gasto por ejercer, gasto mensual, etc.; esto con el fin de no confundir al usuario.

Presupuesto	Gasto Ejercido	Gasto Por Ejercer	Total Anual

Presupuesto Anual	Gasto Anual	Gasto Ejercido Anual

Actualizar Gasto

Gasto Ejercido Mensual

Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic

Figura 6.15
Pantalla de consulta de gasto

VI.2.7 Pantalla de Captura de Gasto

La pantalla de captura guarda un diseño diferente al que se maneja en las pantallas de presupuesto y consulta de gasto, dado que en esta pantalla además de mostrar información general tanto de nivel, como del concepto, el usuario podrá relacionar gastos (facturas) al concepto que él desee, teniendo una interacción con la pantalla de proveedor.

En la pantalla de captura de gasto se definieron tres zonas principales:

- Area de consulta de nivel.
- Area de consulta de concepto.
- Area de captura de gasto.

a) Area de consulta de nivel.

En dicha área el usuario podrá visualizar información general del nivel tanto el acumulado anual como la información del mes seleccionado en el área de consulta del gasto. El área de consulta de nivel se muestra en la figura 6.16.

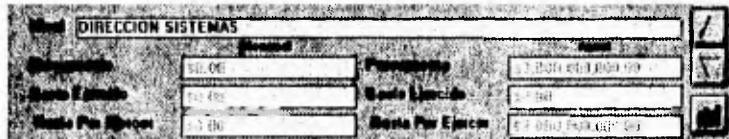


Figura 6.16

Area consulta de nivel
Pantalla captura de gasto

b) Area de consulta de concepto.

Esta área es una imagen de la de consulta de nivel, pero refiriéndose al concepto seleccionado en el combo de conceptos. En la figura 6.17 se muestra el área de consulta de concepto.

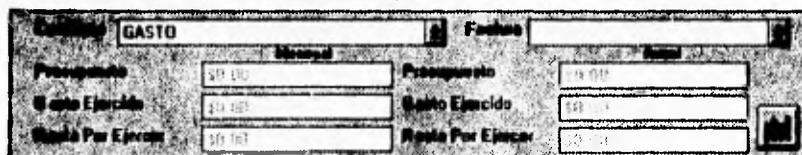


Figura 6.17

Area consulta de concepto
Pantalla captura de gasto

DISEÑO DEL SISTEMA

c) Area de captura de gasto.

El área de captura de gasto es uno de los puntos fuertes del sistema junto con la planeación del presupuesto, dado que en dicha área el usuario podrá capturar todos sus gastos, relacionándolos al concepto que él haya definido dentro de la planeación, cada vez que el usuario captura una nueva factura el sistema recalcula y actualiza la información de la pantalla.

Cualquier factura generada debe de relacionarse con un proveedor existente en el catálogo de proveedores, si el sistema no reconoce el proveedor digitado el sistema automáticamente accesa la pantalla de proveedor, y el usuario podrá realizar las búsquedas definidas en la misma.

Dado que en la realidad algunos gastos no tendrán necesariamente una relación con algún proveedor como son los viáticos, en el catálogo de proveedores existe un proveedor con la clave 0000 que funge las funciones de proveedor. El área de captura de gasto se muestra en la figura 6.18.

The image shows a screenshot of a software interface for capturing expenses. It features several input fields and buttons. At the top, there are fields for 'Clave Proveedor' and 'Proveedor', followed by a 'Buscar Proveedor' button. Below these are fields for 'No. de Factura', 'Fecha de Factura' (with a date selection box), 'Mes de Que se Emite' (with a 'MesCombo' dropdown menu), and 'Monto'. A larger field labeled 'Presupuesto del Estado' is positioned below the date and month fields. At the bottom of the form, there are four buttons, likely for navigation or action.

Figura 6.18

Area de captura de gasto
Pantalla de captura de gasto

La pantalla de captura de gasto también guarda un acceso directo a la pantalla de presupuesto y de consulta de gasto. Su presentación completa se muestra en la figura 6.19.

Figura 6.19
Pantalla captura de gasto

VI.3 ESQUEMA PSEUDO-PROTOTIPO (EP)

Posterior al diseño de pantallas, el usuario siente la necesidad de ver si el sistema funcionará de la forma en que él lo desea.

Esto nos llevó a proponer una implementación que le llamamos Esquema Pseudo-prototipo (EP).

DISEÑO DEL SISTEMA

Este EP no es otra cosa que la simulación de un prototipo, esto aprovechando las ventajas gráficas que nos ofrece Visual Basic. Normalmente en el diseño de un prototipo, se gasta demasiado tiempo y dinero, y algunas veces este prototipo no cumple con las características y objetivos totalmente, lo que lo hace poco atractivo debido a la carga de programación que conlleva realizarlo (a menos de que se este trabajando con una herramienta CASE, pero este tipo de herramientas normalmente son muy caras).

El concepto general del EP es, al igual que en el diseño de un prototipo generar un sistema previo, pero en un EP el sistema simula la captura de datos, y la interacción con el usuario se hace mediante el flujo de pantallas, como se muestra en la pantalla 6.20 y 6.21.

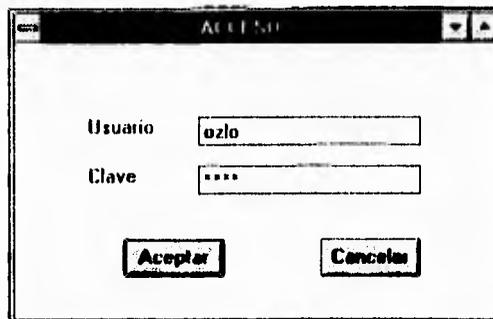


Figura 6.20

Ventana de acceso al sistema

En la pantalla de la figura 6.20 el usuario captura su usuario y su clave de acceso y presiona el comando aceptar, mostrando el sistema la siguiente pantalla.

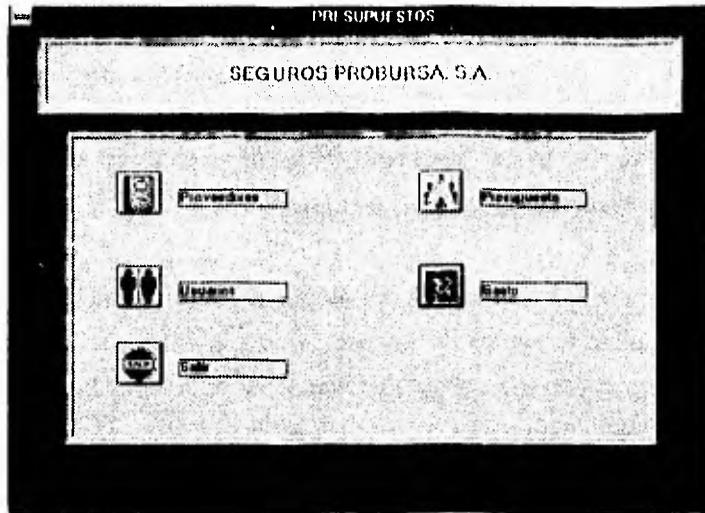


Figura 6.21

Primera pantalla presentada a usuarios
Pantalla principal

La pantalla de la figura 6.21 la muestra el sistema si el usuario y su clave de acceso fueron validas, en caso de que el usuario hubiera digitado una clave o usuario invalidos el sistema desplegaría la pantalla de la figura 6.22.

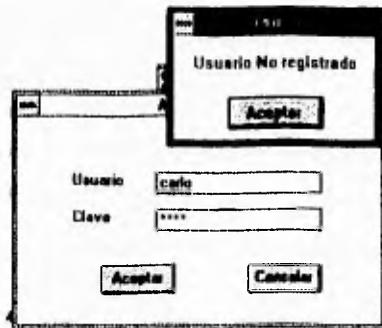


Figura 6.22

Respuesta a usuarios no registrados
Pantalla de acceso

DISEÑO DEL SISTEMA

Realmente estas pantallas bajo Visual Basic requieren de una programación mínima para poder darle la presentación que guardan. Esto nos da realmente la posibilidad de desarrollar LA INTERFAZ del sistema en un tiempo muchísimo menor, y poder dedicarle más tiempo a la validación de requerimientos y fronteras del sistema con el usuario.

Es importante aclarar, que en el Esquema Pseudo-prototipo el usuario realmente no captura todos los datos, uno como analista debe de considerar cuales son los puntos claves de las pantallas y así poder simularlas en el EP, dando oportunidad al usuario de aclarar las pantallas y en general todos los puntos que él considere pertinentes, con la ventaja de que el usuario validará en un tiempo realmente corto como se verá su sistema al final del proceso de desarrollo, lo que nos permite hacer aclaraciones muy pero muy valiosas antes de comenzar realmente, a diseñar nuestra base de datos y posteriormente a desarrollar el sistema de cómputo en sí. Todas las pantallas del sistema no se discutirán dentro de la tesis, éstas se explicarán dentro del manual de usuario.

VI.4 VALIDACION DEL ESQUEMA PSEUDO-PROTOTIPO

Este paso y el anterior son totalmente necesarios y tal vez sean los más importantes para que el sistema llegue a buen fin. Puesto que en estos dos pasos la interacción con el usuario es constante, y es él, el que guía el diseño de sus pantallas y los datos que contienen éstas.

En general la validación del Esquema Pseudo-prototipo consiste en ir evaluando el flujo del sistema, si realiza todas las validaciones, y si en el sistema se están considerando todas las variables que el usuario previamente discutió con nosotros como analistas.

Un ejemplo muy sencillo se basa en la pantalla de la figura 6.20, en la cual el sistema recibe el usuario y clave de acceso, el sistema tiene que validar esta clave y permitir el acceso al sistema si es que está registrado en el sistema.

Pero adicionalmente a esto el sistema tiene que validar los derechos que este usuario tiene dentro del sistema, por lo que no cualquier usuario podrá entrar a planear el presupuesto dado que esta tarea esta destinada sólo a los directores de área, esto se valida dentro del EP con dos pantallas más.

La figura 6.23 nos muestra qué desplegaría el sistema si el usuario no tiene derecho a capturar el presupuesto.

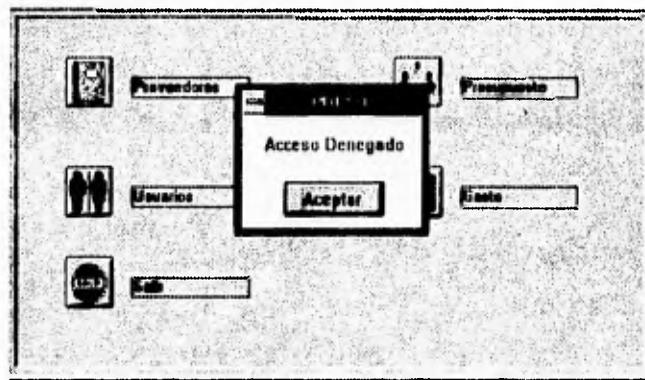


Figura 6.23

**Opción no válida para el usuario
Pantalla principal**

DISEÑO DEL SISTEMA

Al contrario si el usuario si tiene el nivel de usuario que le permita planear el presupuesto el sistema nos mostrará la figura 6.24.

The screenshot shows a software window titled 'PRESUPUESTO'. At the top, there is a 'Nivel:' field. Below it are two input fields: 'Presupuesto Anual' and 'Por Presupuestar'. A large, mostly empty grid occupies the center of the screen. To the right of the grid is a vertical toolbar with four buttons: 'Crea Nivel', 'Borra Nivel', 'Crea Concepto', and 'Elimina Concepto'. Below the grid, there are two input fields: 'Desagregación' and 'Presupuesto Asignado'. Underneath these is a section for 'Presupuesto Mensual' with a grid of 12 input fields for the months: Ene, Feb, Mar, Abr, May, Jun, Jul, Ago, Sep, Oct, Nov, and Dic. At the bottom of this section are two buttons: 'Activar' and 'Cancelar'.

Figura 6.24

Pantalla de presupuesto

Y ya dentro de esta pantalla sigue el proceso de validación, con los comentarios del usuario como una parte muy importante, en el Esquema Pseudo-prototipo.

Validaciones dentro de la opción de presupuesto

ACCION	VALIDACION
AREA DE CONCEPTO.	La suma de presupuesto asignado de cada una de los conceptos del grid no puede ser mayor al presupuesto del nivel.
AREA DE INFORMACION ESPECIFICA.	La suma del presupuesto de los meses no puede ser mayor al presupuesto asignado.
CREAR NIVEL.	El número de niveles no puede ser mayor a 20.

Validaciones dentro de captura de gasto

ACCION	VALIDACION
ACCESO A LA OPCION	el sistema valida si el usuario tiene los derechos de entrada a la opción.
CLAVE DE PROVEEDOR	Verificar que la clave de proveedor exista. Si no existe nos muestra la pantalla de catálogo de proveedores.
NO. DE FACTURA	Verificar que el número de factura con relación al proveedor no exista.
FECHA DE FACTURA	La fecha no debe ser mayor a la fecha del sistema.
MES EN QUE SE EJERCE LA FACTURA	El mes en que se ejerce no puede ser menor al mes de la fecha de factura.

DISEÑO DEL SISTEMA

Validaciones dentro del módulo de Proveedores

ACCION	VALIDACION
ALTA PROVEEDOR	Verificar que no existan campos nulos.
PROVEEDOR	No existan proveedores duplicados.
CLAVE PROVEEDOR	Campo asignado por el sistema.

Validaciones dentro del módulo de Usuarios

ACCION	VALIDACION
ALTA USUARIO	Verificar que no existan campos nulos.
PERFIL DE USUARIO	El usuario tenga un perfil de usuario definido.
AREA ASIGNADA	El usuario tenga un área de operación asignada.
CLAVE USUARIO	Asignada por el sistema.

El diseño de las pantallas, la implementación de el Esquema Pseudo-prototipo y la validación del mismo, son totalmente recursivos, y estas fases se repetirán las veces que el analista y el usuario consideren necesarios, pudiendo llegar en un momento dado a la decisión de no desarrollar el sistema.

VI.5 VENTAJAS Y DESVENTAJAS DEL EP

Dentro del desarrollo de este sistema se detectaron las siguientes Ventajas y Desventajas del EP, comparado con la implementación de un prototipo como tal:

VENTAJAS	DESVENTAJAS
El tiempo de implementación es realmente corto.	No permite al usuario capturar directamente los datos.
Permite al usuario evaluar los datos que manejara el sistema, antes de que este realmente desarrollado.	
Los costos por tiempo de desarrollo se reducen.	
Da las bases para un sistema exitoso.	
No requiere gran cantidad de programación.	

Cabe recalcar que este EP lo definimos basándonos en las facilidades que ofrece Visual Basic, para implementar una interfaz muy atractiva al usuario, sin ningún tipo de programación y lo único que realmente se programa es el flujo de las pantallas. Es por esto que los tiempos son tan cortos y este EP es factible de implementar.

VI.6 DISEÑO DE LA BASE DE DATOS

Habiendo concluido la validación de nuestro pseudo-prototipo, seguimos con una definición de nuestras tablas y la relación que existirá entre cada una de ellas.

DISEÑO DEL SISTEMA

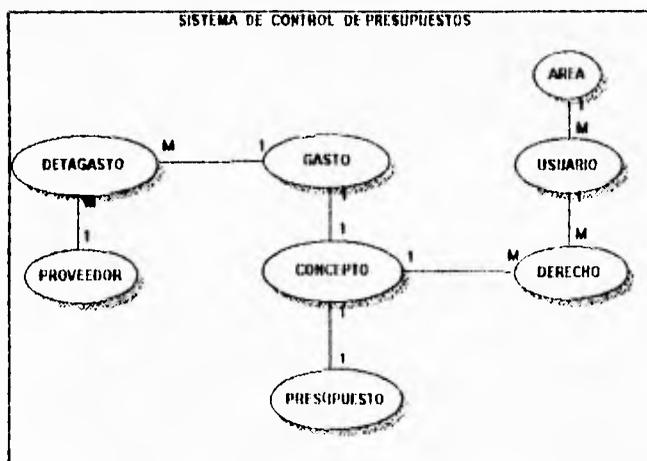
VI.6.1 Tablas del Sistema

Las tablas que se definieron durante el proceso de diseño son las siguientes:

TABLA	DESCRIPCION
PRESUPUESTO	Almacena toda la información referente al presupuesto.
GASTO	Guarda la información general del gasto (consolidada).
DETAGASTO	Guarda el detalle del gasto por meses (todas las facturas asociadas a cada concepto).
PROVEEDOR	Catálogo de proveedores.
USUARIO	Catálogo de usuarios.
DERECHO	Almacena los derechos de cada usuario.
AREA	Catálogo de áreas.

VI.6.2 Modelo Entidad-Relación del Sistema

Un modelado sencillo del sistema es el siguiente:



VI.6.3 Diccionario de Datos

TABLA AREA

CAMPO	DESCRIPCION	TIPO	LONGITUD
CLAVE	Identificador del área.	TEXT	3
DESCRIPCION	Descripción del área de la compañía	TEXT	50

INDICES

CAMPOS	TIPO
CLAVE	LLAVE PRIMARIA

TABLA CONCEPTO

CAMPO	DESCRIPCION	TIPO	LONGITUD
YEAR	Año del presupuesto.	TEXT	4
NIVEL ACTUAL	Nivel al que pertenece el concepto.	TEXT	2
CUENTAS	Clave con el que se reconoce el concepto y su liga entre niveles.	TEXT	60
DESCRIPCION	Descripción del concepto.	TEXT	50

DISEÑO DEL SISTEMA

INDICES

CAMPOS	TIPO
YEAR+NIVELACTUAL+CIENTAS	LLAVE PRIMARIA

TABLA PRESUPUESTO

CAMPO	DESCRIPCION	TIPO	LONGITUD
YEAR	Año del presupuesto.	TEXT	4
NIVEL ACTUAL	Nivel al que pertenece el concepto.	TEXT	2
CIENTAS	Clave con el que se reconoce el concepto y su liga entre niveles.	TEXT	60
DESCRIPCION	Descripción del concepto.	TEXT	50
ENE.	Presupuesto del mes de enero.	DOUBLE	
FEB.	Presupuesto del mes de febrero.	DOUBLE	
MAR	Presupuesto del mes de marzo.	DOUBLE	
ABR.	Presupuesto del mes de abril.	DOUBLE	
MAY.	Presupuesto del mes de mayo.	DOUBLE	
JUN.	Presupuesto del mes de junio.	DOUBLE	
JUL.	Presupuesto del mes de julio.	DOUBLE	
AGO.	Presupuesto del mes de agosto.	DOUBLE	
SEP.	Presupuesto del mes de septiembre.	DOUBLE	
OCT.	Presupuesto del mes de octubre.	DOUBLE	
NOV.	Presupuesto del mes de noviembre.	DOUBLE	
DIC.	Presupuesto del mes de diciembre.	DOUBLE	
PRESUPUESTO	Presupuesto anual.	DOUBLE	

INDICES

CAMPOS	TIPO
YEAR+NIVELACTUAL+CUENTAS	LLAVE PRIMARIA

TABLA GASTO

CAMPO	DESCRIPCION	TIPO	LONGITUD
YEAR	Año del gasto.	TEXT	4
NIVEL ACTUAL	Nivel al que pertenece el concepto.	TEXT	2
CUENTAS	Clave con el que se reconoce el concepto y su liga entre niveles.	TEXT	60
DESCRIPCION	Descripción del concepto.	TEXT	50
ENE	Gasto del mes de enero.	DOUBLE	
FEB	Gasto del mes de febrero.	DOUBLE	
MAR	Gasto del mes de marzo.	DOUBLE	
ABR	Gasto del mes de abril.	DOUBLE	
MAY	Gasto del mes de mayo.	DOUBLE	
JUN	Gasto del mes de junio.	DOUBLE	
JUL	Gasto del mes de julio.	DOUBLE	
AGO	Gasto del mes de agosto.	DOUBLE	
SEP	Gasto del mes de septiembre.	DOUBLE	
OCT	Gasto del mes de octubre.	DOUBLE	
NOV	Gasto del mes de noviembre.	DOUBLE	
DIC	Gasto del mes de diciembre.	DOUBLE	

DISEÑO DEL SISTEMA

INDICES

CAMPOS	TIPO
YEAR+NIVELACTUAL+CUENTAS	LLAVE PRIMARIA

TABLA DETAGASTO

CAMPO	DESCRIPCION	TIPO	LONGITUD
YEAR	Año del gasto.	TEXT	4
NIVEL ACTUAL	Nivel al que pertenece el concepto.	TEXT	2
CUENTAS	Clave con el que se reconoce el concepto y su liga entre niveles.	TEXT	60
IDPROVEEDOR	Clave del proveedor.	TEXT	4
FACTURA	Número de factura.	TEXT	6
MONTOFAC	Monto de la factura.	DOUBLE	
FECHAFACTURA	Fecha de factura.	DATE	
MES	Mes en que se ejerce.	TEXT	2
DESCRIPCION	Descripción del gasto.	TEXT	50
IDUSUARIO	Usuario que capturo el gasto	TEXT	4

INDICES

CAMPOS	TIPO
YEAR+NIVELACTUAL+CUENTAS+IDPROVEEDOR+FACTURA	LLAVE PRIMARIA

TABLA PROVEEDOR

CAMPO	DESCRIPCION	TIPO	LONGITUD
IDPROVEEDOR	Clave del proveedor.	TEXT	4
NOMBRE	Razón social del proveedor.	TEXT	50
CALLE	Calle y número del domicilio fiscal.	TEXT	40
COLONIA	Colonia del domicilio fiscal.	TEXT	30
CP	Código postal del domicilio fiscal.	TEXT	5
TELEFONO	TELEFONO.	TEXT	10
FAX	FAX.	TEXT	10
CELULAR	CELULAR.	TEXT	15
RESPONSABLE	Nombre del representante.	TEXT	50
RFC	Registro federal de causantes del proveedor.	TEXT	14

INDICES

CAMPOS	TIPO
IDPROVEEDOR	LLAVE PRIMARIA
NOMBRE	NORMAL

DISEÑO DEL SISTEMA

TABLA USUARIO

CAMPO	DESCRIPCION	TIPO	LONGITUD
IDUSUARIO	Identificador del usuario.	TEXT	4
CVEUSUARIO	Clave de usuario.	TEXT	5
PASSWORD	Clave de acceso.	TEXT	5
NOMBRE	Nombre de pila del usuario.	TEXT	30
PATERNO	Apellido paterno.	TEXT	30
MATERNO	Apellido materno.	TEXT	30
AREA	Clave del área a la que pertenece.	TEXT	4
PUESTO	Puesto que ejerce.	TEXT	50
CAPTURA	Fecha de alta del usuario.	TEXT	DATE
TIPO	Perfil de usuario.	TEXT	1

INDICES

CAMPOS	TIPO
IDUSUARIO	LLAVE PRIMARIA
CVEUSUARIO	NORMAL
PASSWORD	NORMAL
AREA	NORMAL

TABLA DERECHOS

CAMPO	DESCRIPCION	TIPO	LONGITUD
IDUSUARIO	Identificador del usuario.	TEXT	4
AREA	Clave del área a la cual tiene acceso.	TEXT	3

INDICES

CAMPOS	TIPO
IDUSUARIO+ AREA	LLAVE PRIMARIA

CAPITULO VII

DESARROLLO DEL SISTEMA

144

VII.1 INTRODUCCION

En el presente capítulo, se dará una explicación de cómo se llevó a cabo el desarrollo del sistema, así como algunos ejemplos de la programación de los controles más importantes del mismo.

Cabe mencionar que la manera de programación en Visual Basic se basa en un esquema de programación al objeto sin ser éste una herramienta de desarrollo propia de programación orientada a objetos. Permite crear, establecer y cambiar sus propiedades y, posteriormente, asignar un código BASIC funcional.

Visual Basic no maneja conceptos como los son poder definir clases y herencia de objetos, conceptos básicos en la programación orientada al objeto, por lo que nosotros manejaremos que Visual Basic es una programación orientada a eventos,¹ ya que cada acción que realice el usuario estará relacionada a un evento dado del control o forma. En otras palabras, los controles de Visual Basic esperan que sucedan eventos particulares, antes de que respondan.

La filosofía de programación, por tanto consiste en tener los siguientes objetos gráficos: Formas y Controles. Una forma puede contener uno o varios controles, pero un control no puede contener una forma.

Por lo cual la filosofía de programación Visual Basic consiste, primero en crear objetos como ventanas, iconos y menús, y después elaborar procedimientos que sean llamados por cada uno de estos objetos. Esto es diferente del método tradicional de elaboración de un programa, en el cual

¹ Keynon Brown. Programm's Introduction to V.B. SYBEX Inc. 1992. Pag. 5

DESARROLLO DEL SISTEMA

existen estructuras para controlar el flujo del programa de un procedimiento a otro de manera lógica hasta que el programa termina.

Para la elaboración de un programa se dedicará la mayor parte del tiempo en crear una interfaz de aplicación durante el diseño. En ese momento, se están creando formas, dibujando controles y escribiendo módulos de código. Cuando se inicia una aplicación durante la corrida sólo las formas que se crearon y programaron serán visibles.

Una *forma* es una ventana que creamos y adaptamos a nuestras necesidades para cada aplicación. Esto es la base de cualquier aplicación Visual Basic que eventualmente correrá como un programa independiente en Windows. Un *control* es el nombre preciso para cualquier objeto que dibujemos en una forma, pero también se refiere a la forma misma. Los botones de comando, menús, cajas de diálogo y barras de desplazamiento, son ejemplos de controles. Cuando un usuario activa un control al seleccionarlo, el código BASIC que está asignado al control es invocado y realiza una tarea específica. Un *módulo* es una estructura para escribir el código que se le asigna a una forma y sus controles. Después de dibujar los controles que deseamos que aparezcan en nuestra forma, se escribe el código asociado funcional para cada control que los usuarios pueden invocar.

Visual Basic utiliza la metáfora del "evento" para describir su paradigma de programación. Utilizaremos este enfoque orientado a los eventos cuando creamos una aplicación. *Orientado a eventos*, significa que todos los 4589 controles¹ que se dibujen en una forma, especifican cómo se comportará la interfaz. Un *evento* es una acción que es reconocida por un control. Se

¹ Keynon Brown. *Programmer's Introduction to V.B.* SYBEX Inc. 1992. Pag. 7

DESARROLLO DEL SISTEMA

escribe un código para cada control a fin de que algo suceda cuando el usuario final interactue con la forma.

Sin embargo, se necesita comprender que el código no está almacenado en un lugar único donde se pueda abrir, observar, cambiar o imprimir. De hecho, no existe un solo programa que se pueda editar y depurar. La utilización de Visual Basic es diferente del enfoque lineal de escribir, correr y depurar un programa, en el cual el proceso comienza en la parte inicial, pasa el control de un procedimiento a otro y, continúa hasta que termina.

VII.2 ENTRADA AL SISTEMA

El diseño de las pantallas es una parte muy importante de la aplicación, ya que es donde se establece el nivel de comunicación y facilidad en el manejo entre el usuario y el sistema; por esto se trató de realizar de la manera más sencilla, buscando siempre que el desarrollo de cada módulo fuera amigable para el usuario.

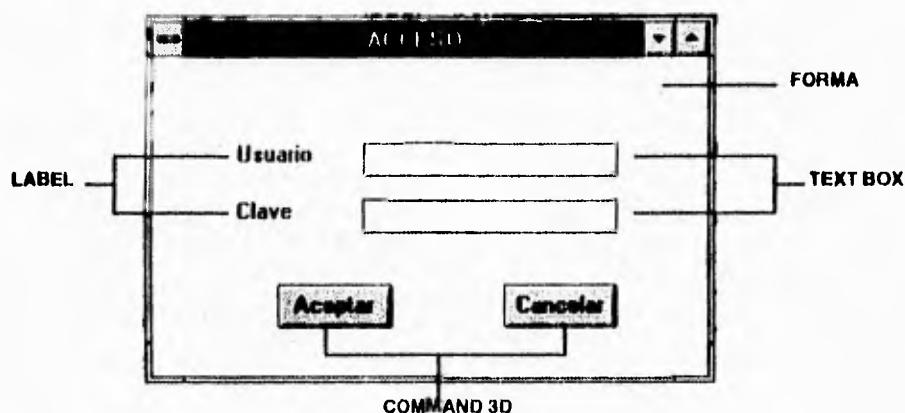
Se realizaron varios prototipos de diseño, de los cuales se eligió el que reuniera la mayor parte de las características para su buen funcionamiento, pero en el transcurso del desarrollo se fueron contemplando nuevas ideas tanto en el diseño de la pantalla como el de la programación, por ejemplo, se tomaron ciertas restricciones para lograr seguridad de accesos al sistema. Todo esto con una constante comunicación con el usuario.

Cabe mencionar que la definición de las siguientes pantallas son las definitivas de un conjunto de prototipos creados.

DESARROLLO DEL SISTEMA

Como se definió en el análisis y en el diseño del sistema, nuestra aplicación de control de presupuestos debe de guardar un esquema por niveles y cada nivel puede tener definidos los conceptos que el usuario desee y a su vez cada concepto puede pasar a ser un nivel inferior y tener definidos dentro de él sus propios conceptos.

Para poder entrar al sistema, se creó una ventana de acceso al mismo, como se muestra a continuación:



Esta ventana contiene los objetos gráficos como son: una forma y controles (Label, Text box, Command 3D) la mayoría de las ventanas utilizadas en el sistema contienen estos objetos gráficos y algunos otros controles, por lo cual únicamente se mencionaran en las diferentes ventanas, ya que en los apéndices A y B se da una mayor explicación de éstos.

Cada uno de estos controles tiene un código asociado dentro del cual son manipuladas sus propiedades, tanto para llevar a cabo una función específica como para lograr un aspecto agradable a la vista del usuario final.

DESARROLLO DEL SISTEMA

Aquí el programa debe de validar el tipo de clave del usuario que esté utilizando el sistema, por lo que habrá usuarios que puedan o no entrar. De entrada, el usuario proporciona su clave y password.

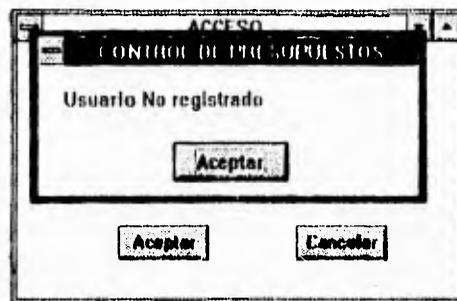
Esta validación se hace por medio de una búsqueda en la tabla de usuarios. Si el usuario presiona el comando de aceptar la aplicación ejecuta el siguiente código:

```
Sub AcepCom_Click ()
  Call AbreBases
  gYear = Year(Date)
  UsuariosTable.Index = "CveUsuario"
  UsuariosTable.Seek "=", UsuText
  If UsuariosTable.NoMatch Then
    MsgBox "Usuario No registrado"
    UsuText.SetFocus
    UsuText = ""
    PassText = ""
    Exit Sub
  Else
    If UsuariosTable("PassWord") <> UCase$(PassText) Then
      MsgBox "Clave No Valida"
      UsuText.SetFocus
      UsuText = ""
      PassText = ""
      Exit Sub
    Else
      'Se acepta el usuario y su Clave
      VUsuario = UCase$(UsuText)
      VPass = UCase$(PassText)
      Call derechos
      PrinciForm.Show
      Unload Me
    End If
  End If
End Sub
```

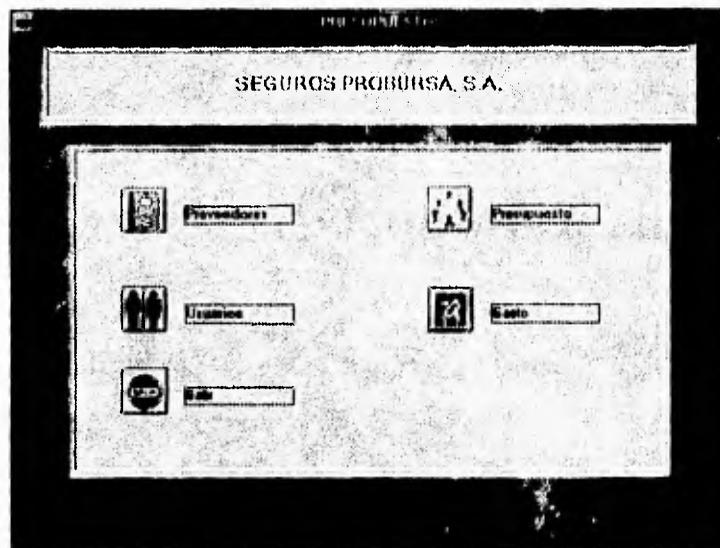
Las dos variables marcadas con negrita son básicas en el sistema, ya que en ellas se guardan el nombre del usuario y su clave, para posteriormente validar sus

DESARROLLO DEL SISTEMA

derechos en la tabla correspondiente para poder tener acceso a los diferentes módulos con los que cuenta el sistema. En caso de que el usuario y su clave no sean válidas, se desplegará el siguiente mensaje:



Una vez de que el sistema haya validado las claves, permitirá pasar a la pantalla general del sistema, la cual se muestra a continuación:



A continuación se explicará cómo se llevó a cabo el desarrollo de cada uno de los módulos que conforman al sistema.

VII.3 MODULO DE PRESUPUESTO

Si el usuario presiona la opción de presupuesto el sistema ejecuta el siguiente código:

```
Sub PresuPrinCom_Click ()
  UsuariosTable.Index = "CveUsuario"
  UsuariosTable.Seek "=", vUsuario
  If UsuariosTable.NoMatch Then
    MsgBox "Usuario No registrado"
    Exit Sub
  End If
  Select Case UsuariosTable("Tipo")
    Case Is = "O"
      MsgBox "Acceso Denegado"
      Exit Sub
  End Select
  Nivel = "01"
  Linea = 1
  PresuForm.Show
  Unload Me
  Unload PrinciForm
End Sub
```

En el código anterior, el sistema valida si el usuario tiene derecho a entrar al módulo de presupuestos, de ser así llama a la forma de presupuestos y se descarga de memoria la forma general del sistema.

La pantalla general de captura y consulta de presupuesto es la siguiente:

DESARROLLO DEL SISTEMA

The screenshot shows a software window titled "DEP SUPLENTO". At the top, there is a "Nivel:" dropdown menu. Below it are two input fields: "Presupuesto Anual" and "Por Presupuestar". A large, mostly empty grid occupies the center. To the right of the grid are four buttons: "Crea Nivel", "Borra Nivel", "Crea Cuenta", and "Elimina Cuenta". Below the grid, there are two input fields: "Descripción" and "Presupuesto Asignado". Underneath these is a section titled "Presupuesto Mensual" containing 12 input boxes, one for each month from "Ene" to "Dic". At the bottom of the window are two buttons: "Aceptar" and "Cancelar".

Todas las formas cuando son llamadas con el método **.Show** ejecutan su evento llamado **load**, el evento load de la forma PresuForm tiene el siguiente código asociado.

```
Sub Form_Load ()  
  aMes(1) = "Ene"  
  aMes(2) = "Feb"  
  aMes(3) = "Mar"  
  aMes(4) = "Abr"  
  aMes(5) = "May"  
  aMes(6) = "Jun"  
  aMes(7) = "Jul"  
  aMes(8) = "Ago"  
  aMes(9) = "Sep"  
  aMes(10) = "Oct"  
  aMes(11) = "Nov"  
  aMes(12) = "Dic"  
  Call TituGrid  
  Call LLenaGrid("Sube", "Presupuesto", "01", gCuentas)  
  Linea = 1  
  Call LlenaMeses(1)  
End Sub
```

En el código anterior identificamos dos funciones que resultan básicas en el módulo de presupuestos.

LLenaGrid (Movi As String, Tipo As String, PNivel As String, pCuentas As String):

Se encarga de llenar el grid con los conceptos que corresponden al nivel en el que se encuentra el usuario.

Recibiendo los siguientes parámetros:

- Movi:** Indica a la función si el usuario desea subir o bajar de nivel.
- Tipo:** Indica si nos encontramos en el área de presupuestos o en la de gasto.
- pNivel:** Indica el nivel en el que se encuentra el usuario.
- pCuentas:** Esta cadena indica al programa el identificador general del nivel dentro del sistema.

Este identificador es básico para la aplicación, ya que si el nivel es el '01' o nivel uno, la variable pCuentas deberá de ser de tres caracteres '001','002', etc. si el nivel fuera el '02' o nivel dos, la cadena sería de 6 caracteres '0001001', por ejemplo esto nos diría que este concepto en el nivel uno pertenece al concepto '001', por lo que pCuentas es una liga entre niveles de los diferentes conceptos.

Esta cadena se guarda en la tabla de presupuesto en el campo cuentas.

DESARROLLO DEL SISTEMA

El procedimiento llena grid guarda el siguiente código:

```
Sub LLenaGrid (Movi As String, Tipo As String, PNivel As String, pCuentas As
String)
    Dim l As Integer
    Dim j As Integer
    Dim Tempo As String
    Dim Suma As Double
    Dim SumaPresu As Double
    Dim SumaPresuAsig As Double
    Forma = "F"
    PresuTable.Index = "PrimaryKey"
    Call Carga(Movi, Tpresup, PNivel, pCuentas)
    PresuDin.MoveFirst
    PresuDin.MoveNext
    If PresuDin.EOF = True And PNivel <> "01" Then
        MsgBox "No Hay Nivel Inferior Definido"
    Else
        PresuDin.MoveFirst
        Call TituGrid
        SumaPresuAsig = 0
        PresenGrid.Rows = 1
        PresenGrid.Col = 0
        j = 0
        l = 1
        ReDim Arre(1 To Lim, 0 To 0)
        Tempo = "Seguros Probursa S.A."
        PresenGrid.AddItem " "
        PresenGrid.Row = 1
        If Val(PNivel) > nDirec Then
            NivelLab = PresuDin("Descripcion")
            Arre(iNivel, j) = PresuDin("NivelActual")
            Arre(iCuentas, j) = PresuDin("Cuentas")
            Arre(iDescri, j) = PresuDin("Descripcion")
            Arre(iPresuAsig, j) = PresuDin("Presupuesto")
            SumaPresu = 0
            PresuDin.Edit
            For M = 1 To 12
                If IsNull(PresuDin(aMes(M))) Then
                    PresuDin(aMes(M)) = 0
                End If
                Arre(M, j) = PresuDin(aMes(M))
                'Arre(iDescri + M, J) = PresuTable(aMes(M))
                SumaPresu = SumaPresu + PresuDin(aMes(M))
            Next
        End If
    End Sub
```

DESARROLLO DEL SISTEMA

```

PresuDin.Update
j = j + 1
PresuMask = SumaPresu
ReDim Preserve Arre(1 To Lim, 0 To j)
PresuDin.MoveNext
Else
    NivelLab = Tempo
    Arre(iNivel, j) = "00"
    Arre(iCuentas, j) = ""
    j = j + 1
    PresuMask = SumaPresu
    ReDim Preserve Arre(1 To Lim, 0 To j)
End If
If PresuDin.EOF = True Then
    Exit Sub
End If
Do While PresuDin.EOF = False
    PresenGrid.Col = 0
    Arre(iNivel, j) = PresuDin("NivelActual")
    Arre(iCuentas, j) = PresuDin("Cuentas")
    Arre(iDescri, j) = PresuDin("Descripcion")
    Arre(iPresuAsig, j) = PresuDin("Presupuesto")
    SumaPresu = 0
    PresuDin.Edit
    For M = 1 To 12
        If IsNull(PresuDin(aMes(M))) Then
            PresuDin(aMes(M)) = 0
        End If
        Arre(M, j) = PresuDin(aMes(M))
        SumaPresu = SumaPresu + PresuDin(aMes(M))
    Next
    PresuDin.Update
    'SumaPresuAsig = SumaPresuAsig + SumaPresu
    SumaPresuAsig = SumaPresuAsig + PresuDin("Presupuesto")
    PresenGrid.Text = PresuDin("Descripcion")
    PresenGrid.Col = 1
    PresenGrid.Text = PresuDin("Presupuesto")
    PresenGrid.Col = 2
    PresenGrid.Text = PresuDin("Presupuesto") - SumaPresu
    PresenGrid.AddItem " "
    l = l + 1
    j = j + 1
    ReDim Preserve Arre(1 To Lim, 0 To j)
    PresenGrid.Row = l
    PresuDin.MoveNext
Loop

```

DESARROLLO DEL SISTEMA

```
If PresenGrid.Rows > 2 Then
    PresenGrid.FixedRows = 1
    PresenGrid.FixedCols = 1
End If
PresenGrid.RemoveItem PresenGrid.Rows - 1
' Checa Si el Registro Tiene Miembros
If PresenGrid.Rows = 1 Then
    BajaCom.Enabled = False
Else
    'ModiCom.Enabled = True
End If
Linea = 1
If Val(PNivel) > nDirec Then
    PresuAnualMask = Val(Arre(iPresuAsig, 0))
    PresuDispoMask = Val(PresuAnualMask) - SumaPresuAsig
End If
End If
Forma = "T"
End Sub
```

Esta función se encarga de llenar el grid de conceptos y generar dos variables:

PresuDin: Variable global tipo dynaset que guarda toda la información del nivel y sus conceptos que la conforman.

Arre: Es un arreglo global que nos sirve como ayuda para acceder rápidamente a la información de los conceptos del nivel.

Las variables anteriores son accesadas por todos los controles en la forma y se regeneran cada vez que el usuario cambia de nivel.

Al ser Visual Basic un lenguaje orientado a eventos, el módulo de presupuesto responde a diferentes eventos que el usuario indique:

Eventos relacionados al grid de conceptos:

Si el usuario accesa alguno de los renglones del grid de conceptos el sistema ejecutara el siguiente código asociado al evento SelChange() del Grid

```
Sub PresenGrid_SelChange ()
  If Linea <> PresenGrid.Row And Forma = "T" Then
    Linea = PresenGrid.Row
    Call LlenaMeses(Linea)
  End If
  If PresenGrid.Rows = 1 Then
    Linea = 1
  Else
    Linea = PresenGrid.Row
  End If
End Sub
```

En este evento se actualiza la región de información específica del concepto, cada vez que el usuario cambia de línea en el grid, la información de esta región se actualiza con respecto al concepto señalado en el grid. Esto lo realiza la función `llenaMes(pLinea As Integer)`.

El sistema responde a los eventos haciendo clic en los siguientes controles:



Con estos scroll bar, el usuario podrá ascender o descender del nivel en el



que se encuentra.



Define un nivel siguiente al actual.



Borra el nivel actual y todos sus conceptos y niveles anteriores.



Permite definir un nuevo concepto en el nivel actual.

DESARROLLO DEL SISTEMA

Elimina Concepto

Elimina el concepto seleccionado en el grid de conceptos de nivel y sus niveles inferiores.

Actualizar

Guarda los cambios realizados en el área de información específica del concepto.

Cancelar

Cancela los cambios realizados en el área de información específica del concepto y restablece los valores que se encuentran antes de él.

VII.4 MODULO DE GASTO

Pensando en la importancia que para el usuario representa el poder contar con información general y completa acerca del presupuesto y gasto de los diferentes niveles que este maneja, se desarrolló un módulo el cual le dé respuesta a estas necesidades, para ello se tuvo que definir la prioridad que la información tendría en la primera forma causando un impacto hacia lo que el usuario en primera instancia requeriría, además de poder seleccionar entre un conjunto de opciones que dan paso a otras tareas. La pantalla general de captura y consulta de gasto es la siguiente:

The screenshot displays a software interface for budget and expenditure management. At the top, it shows the level 'DIRECCION DE SISTEMAS'. Below this, there are input fields for 'Presupuesto', 'Gasto', and 'Gasto Por Ejecutar', along with a 'Total Anual' field. A table below these fields shows 'Presupuesto Anual' and 'Gasto Anual' with values of 0. A button labeled 'Actualizar Gasto' is positioned below the table. At the bottom, a section titled 'Gasto Ejecido Mensual' contains a grid of input fields for each month of the year: Ene, Feb, Mar, Abr, May, Jun, Jul, Ago, Sep, Oct, Nov, and Dic.

A continuación se mencionará el desarrollo efectuado para este módulo, el cual constó principalmente de:

- Información general del gasto.
- Actualización del gasto.

DESARROLLO DEL SISTEMA

En las cuales se utilizaron controles y formas con sus diferentes propiedades y haciendo uso del código se pudo combinar dichos controles con eventos que permitieron una mejor realización de las tareas aunado con una buena presentación.

En esta opción se programó para que, una vez que el usuario seleccione la opción de GASTO, se presente una nueva forma, la cual contiene:

Dos command buttom a los cuales se les asoció eventos. En uno de ellos dada la relación que existe entre el gasto y el presupuesto se tuvo que realizar una opción en la que por medio de código y eventos el usuario pudiera cambiarse a la pantalla de presupuesto. Por otro lado se contempló el hecho de que el usuario requeriría en algún momento salir de esta pantalla por lo cual se realizó el procedimiento para tal situación.

En esta forma también se colocó información de interés acerca del nivel en el que el usuario se encontraba como es el caso del presupuesto, gasto ejercido y por ejercer tanto anual como mensual.

Para el caso de los títulos e información, el usuario sólo puede visualizar y no modificar. Se utilizó el control label con diferentes propiedades, panel para diferenciar entre cada grupo de información contenida en la forma, como es el caso de la información del presupuesto tanto anual como mensual, un command buttom para poder llevar a cabo una acción como es la actualización del gasto y del presupuesto.

Y por último un spin buttom para poder incrementar o decrementar el concepto brindándole al usuario la opción de poder seleccionar el deseado, en este caso además de hacer uso de los controles se tuvieron que realizar dos

procedimientos, en uno de ellos se realizó una rutina para ascender de nivel y en el otro para descender, a continuación se muestra el código de dichos procedimientos.

Procedimiento para SUBIR DE NIVEL

```
Sub SubeCom_Click ()
  Dim Rompe As String
  If gNivel = "01" Then
    SubeCom.Enabled = False
  Else
    BajaCom.Enabled = True
    gNivel = Arre(iNivel, 0)
    gCuentas = Arre(iCuentas, 0)
    Call LLenaCombo("SUBE", gNivel, gCuentas)
    ConceptoCombo.ListIndex = 0
  End If
End Sub
```

La parte más compleja de la sección de Gasto fue realizar la actualización del gasto ya que se tenía que considerar varios aspectos entre los cuales se encuentran:

- Colocar llaves de seguridad, ya que no todos los usuarios deben realizar cambios y a éstos a su vez se les agrego otras llaves de seguridad para no acceder a los diferentes niveles.
- Considerar que los cambios del gasto eran efectuados por cada concepto y éste formaba parte de un nivel dentro de la empresa.
- Realizar una buena búsqueda para el área de proveedores.

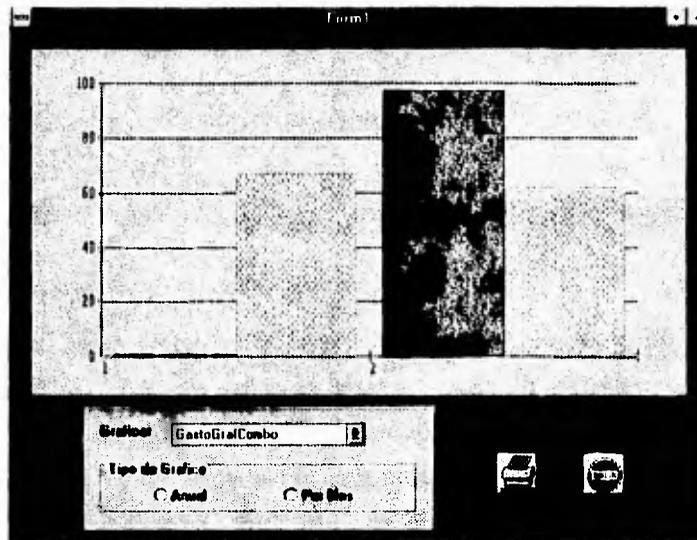
DESARROLLO DEL SISTEMA

Una vez hechas estas consideraciones se realizó un estudio que permitió manejar toda la información necesaria en una sola pantalla y además brindarle de una manera fácil un mejor manejo como es: realizar búsqueda del proveedor, poder seleccionar entre los diferentes conceptos que contiene el nivel y tener el acceso a entrar a graficar por lo que se explica a continuación el desarrollo de esta sección.

El desarrollo del módulo de Gasto incluye dentro de su opción Actualizar Gasto como se muestra a continuación:

En donde existen dos opciones que tienen como finalidad graficar por nivel y por concepto respectivamente.

Su desarrollo se basa en una forma que contiene los siguientes comandos: un graph, un frame, un list box, dos option button y dos command button.



El código asociado a la forma presenta dos casos:

Caso 1. Grafica Nivel, que llama a una función llamada GrafNivel.

Caso 2. Grafica Concepto, que llama a otra función llamada GrafConcepto, que a su vez tiene dos opciones, la de graficar un sólo concepto (concepto actual) o graficar todos los conceptos existentes.

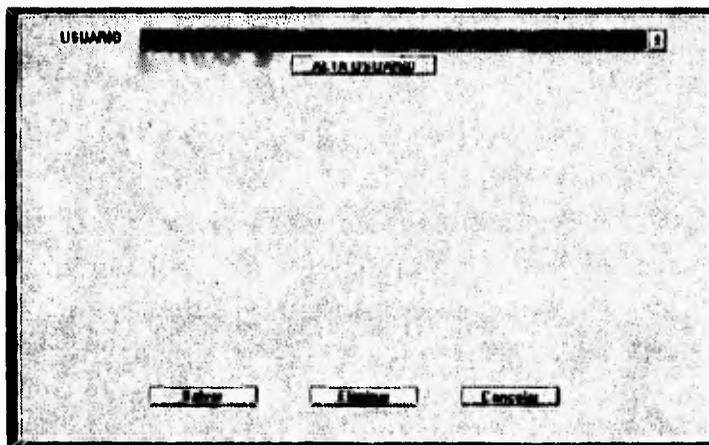
En el caso de que se elija graficar por Concepto Actual, se habilitan el frame, y los option button para tener la posibilidad de graficar por año o por sus meses. En cambio si se elige graficar por todos los conceptos se deshabilitan las opciones de graficar por año o por meses.

VII.5 MODULO DE USUARIOS

Este módulo debe presentar las siguientes opciones:

- Consulta de un usuario.
- Alta de un usuario.

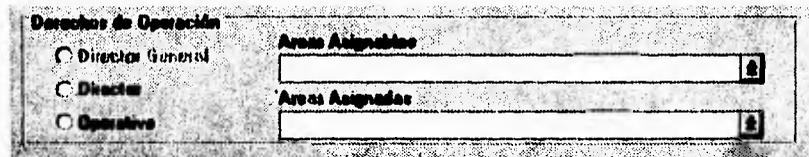
Cada una de estas opciones cuenta con su respectivo command3D para poder acceder a ellas, no hay que estar entrando a menús ramificados, sino que es de una manera directa, por lo que se logra una óptima interfaz entre el usuario y el sistema como se muestran en la siguiente pantalla:



En seguida entraremos a ver un poco más detalladamente el desarrollo de cada una de las funciones que componen el módulo de usuarios.

En el caso de la opción de CONSULTA; que realiza la función de desplegar la información de un usuario se utilizó un combo box y se programó de tal manera que aparezca como una ventana, la cual con un clic mostrará los usuarios que estén dados de alta por el usuario actual, es decir, por la clave y password del usuario que entró al sistema, ésta es una de las restricciones y medidas de seguridad que se tomó muy en cuenta durante la programación, ya que así ninguna persona que tenga acceso a este módulo podrá ver todos los usuarios que tengan permiso para entrar al sistema y sus claves de acceso.

Dentro de la consulta que se hace por medio de este combo box se tiene la opción de elegir a algún usuario, de esta manera aparecerá un frame3D con la información sobre el usuario elegido. Dentro de este frame3D hay varios option button que sirven para la selección de la categoría del usuario, como se muestra en la siguiente figura:



El código que se programó en este combo box para realizar lo anterior es como sigue:

```
Sub Form_Load ()
    AltGastoForm.Enabled = False
    ClaveBusMask.Text = AltGastoForm.ClaveProveeMask.Text
    ProveeBusText.Text = AltGastoForm.ProveeText.Text
    BuscaNomProveFrame.Visible = False
    BuscaCveProveFrame.Visible = False
    AltaProvePanel.Visible = False
    BuscaProvePanel.Visible = False
    If vgBuscaProvee Then
        AceptarProveCom.Visible = True
    End If
End Sub
```

DESARROLLO DEL SISTEMA

```
Else
  AceptarProveCom.Visible = False
  ProveeForm.SalirCom.Visible = True
End If
ProveeForm.ActualizaProveCom.Visible = True
ProveeForm.EliminaProveCom.Visible = True
BuscaProveCombo.AddItem "Clave"
IClave = 0
BuscaProveCombo.AddItem "Nombre"
IProveedor = 1
End Sub
```

La pantalla general de consulta y alta de usuario queda como sigue:

The screenshot shows a graphical user interface for user management. At the top, there is a title bar with the text 'USUARIO' and a button labeled 'ALTA USUARIO'. Below the title bar, the form is organized into several sections. The first section, 'Información del Usuario', contains three text input fields for 'Nombre', 'Apellido Paterno', and 'Apellido Materno'. The second section has two fields for 'Puesto' and 'Reporta a.'. The third section includes fields for 'Area', 'Clave de Usuario', and 'Password'. The fourth section, 'Derechos de Operación', features three radio button options: 'Directivo General', 'Directivo', and 'Operativo'. To the right of these options are two text input fields labeled 'Áreas Asignadas'. At the bottom of the form, there are three buttons: 'Salir', 'Eliminar', and 'Cancelar'.

En el caso del comando ALTA USUARIO, que permite adicionar a una persona a la lista de usuarios del sistema. El desarrollo de la programación se realizó de tal forma que funcione de la siguiente manera:

Al dar un clic en el comando de ALTA USUARIO, inmediatamente aparecerá otra ventana en donde se podrán capturar todos los datos del nuevo usuario, como son: nombre, dirección, puesto que ocupa en la empresa, y sobre todo

datos importantes para el sistema como clave, password y permisos de acceso a determinadas áreas.

Está diseñado de tal manera que en la entrada del sistema se capture la clave y password del usuario, el sistema entrará directamente a esta parte del módulo para validar su entrada y los permisos otorgados con los que cuenta.

Como ya se había mencionado se analizó y se llevó a cabo el proceso de la programación tomando grandes medidas de seguridad.

Se puede observar que en la pantalla general de usuarios tenemos los comandos de Salvar, Eliminar y Cancelar.

El comando Salvar se refiere a guardar los cambios que se hayan realizado en los datos de algún usuario consultado, o bien, guardar los datos del nuevo usuario que se haya dado de alta en el sistema, y cuenta también con un comando para elegir esta opción de operación y su código asociado es el siguiente:

Procedimiento para salvar a un usuario:

```
Sub SalvaUsuarioCom_Clic ()
  If DirecGenOption.Value Then
    IOpcion = ""
  Else
    If DirecOption.Value Then
      IOpcion = "D"
    Else
      If OperaOption.Value Then
        IOpcion = "O"
      End If
    End If
  End If
  DerechosTable.Index = "PrimaryKey"
```

DESARROLLO DEL SISTEMA

```
UsuariosTable.Index = "Index2"  
If AreaAsignadaCombo.ListCount = 0 Then  
    MsgBox "Tiene que Asignar Area", 48, "Area de Operación"  
    DerechosUsuarioCombo.SetFocus  
    Exit Sub  
End If  
If vfUsuarioAlta = True Then  
    ClaveUsuarioLabel = Left$(PaternoUsuarioText, 1)  
    left$(MaternoUsuarioText, 1)  
    UsuariosTable.AddNew  
    UsuariosTable("IdUsuario") = IldNuevo  
Else  
    UsuariosTable("Nombre") = NombreUsuarioText  
    UsuariosTable("Puesto") = MaternoUsuarioText  
    UsuariosTable("Reporta") = ReportaUsuarioText  
    UsuariosTable("Area") =  
LlenaCade(Str(AreaUsuarioCombo.ItemData(AreaUsuarioCombo.ListIndex)),  
"0", 3)  
    UsuariosTable("Clave") = ClaveUsuarioLabel  
    UsuariosTable(Password) = PassUsuarioMask  
    UsuariosTable("Tipo") = IOpcion  
    UsuariosTable.Update  
    EliminaUsuarioCom.Enabled = False  
    SalvaUsuarioCom.Enabled = False  
End Sub
```

El comando Eliminar realiza la operación de eliminación, que es por supuesto borrar el registro indicado del sistema, por lo cual se diseño para que se elimine todo lo referente al usuario, tal como información personal, permisos y accesos del usuario en el sistema.

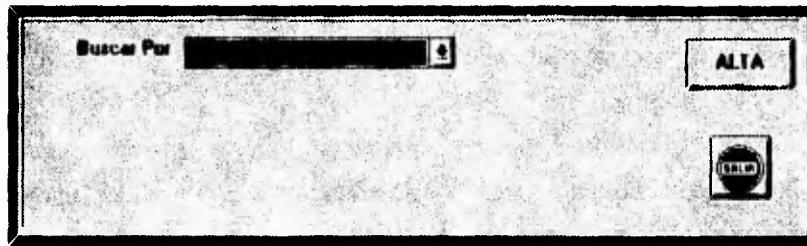
El comando Cancelar es la función que permite cancelar alguna operación que se este llevando a cabo, por ejemplo, si de pronto el usuario se da cuenta que no esta haciendo lo correcto, con tan sólo dar un clic en el command button de Cancelar, se borrará inmediatamente lo hecho hasta ese momento en la ventana, no realizando los cambios hechos.

VII.6 MODULO DE PROVEEDORES

Este módulo debe presentar las siguientes opciones:

- Alta de proveedor.
- Búsqueda de proveedor.

Para crear una pantalla la cual tuviera estas opciones se utilizaron los siguientes controles: Frame, dentro del cual se encuentra un label, un combo box y un text box, como se muestra a continuación:



Habilita un panel el cual contiene varios text box, los cuales van a permitir la entrada de los diferentes datos del proveedor, éstas a su vez estarán asociadas a etiquetas las cuales dan un título a sus correspondientes text box. La definición de la pantalla se muestra a continuación.

DESARROLLO DEL SISTEMA

The image shows a screenshot of a data entry form for a provider. The form is enclosed in a rectangular border and contains the following fields and buttons:

- Clave Proveedor**: A small text input field.
- Proveedor**: A larger text input field.
- Dirección**: A text input field.
- Colonia**: A text input field.
- Población**: A text input field.
- C.P.**: A small text input field.
- Telefonos**: A text input field.
- Responsable**: A text input field.
- RFC**: A text input field.
- FAX**: A text input field.
- Salvar**: A button located at the bottom left.
- Cancelar**: A button located at the bottom right.

En la parte inferior del panel se muestran comandos SALVAR y CANCELAR. En el comando SALVAR es donde se encuentra todo el código principal asociado a este panel, el cual realiza lo siguiente:

Para poder actualizar la Tabla de Proveedores es indispensable tener la condición de que se introduzca la información referente al nombre del proveedor, su RFC, así como sus teléfonos, ya que consideramos que es información importante para la empresa. Una vez introducida toda la información, se realiza una búsqueda en la tabla, por un índice cuya llave primaria es el nombre del proveedor, esto con el fin de asegurarnos de que no halla sido ya dado de alta, si el proveedor ya fue dado de alta se desplegará un mensaje indicando al usuario de que dicho proveedor ya fue encontrado. En caso de que no halla sido encontrado se lleva a cabo una función para que el sistema automáticamente asigne una clave al nuevo proveedor dado de alta y al mismo tiempo actualizar todos los campos de la tabla.

El comando CANCELAR permite deshacer los datos introducidos en las text box, de esta manera volver a iniciar la captura.

Como un ejemplo se muestra a continuación la manera de programación en Visual Basic que se llevó a cabo para el comando SALVAR.

PROCEDIMIENTO PARA SALVAR A UN PROVEEDOR DADO DE ALTA

```

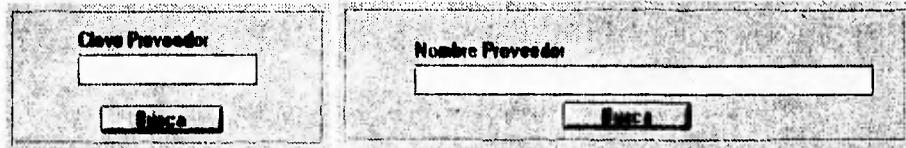
Sub SalvaAltaProveCom_Click ()
  Dim lcCveProvee As String
  If RTrim$(AltaNomProveText) = "" Then
    MsgBox "Nombre del Proveedor Indispensable"
    AltaNomProveText.SetFocus
  Else
    If RTrim$(AltaRfcProveText) = "" Then
      MsgBox "RFC Indispensable"
      AltaRfcProveText.SetFocus
    Else
      if RTrim$(AltaTelProveeMask) = "" Then
        MsgBox "Telefono Indispensable"
        AltaTelProveeMask.SetFocus
      Else
        ProveeTable.Index = "Index2"
        ProveeTable.Seek "=", AltaNomProveText.Text
        If ProveeTable.NoMatch Then
          lcCveProvee = GeneraCveProvee()
          If lcCveProvee <> "ERROR" Then
            ProveeTable.AddNew
            ProveeTable("CveProvee") = lcCveProvee
            ProveeTable("Nombre") = UCase$(AltaNomProveText)
            ProveeTable("Calle") = UCase$(AltaDireProveText)
            ProveeTable("Colonia") = UCase$(AltaColProveText)
            ProveeTable("CP") = AltaCpProveeMask
            ProveeTable("Telefonos") = AltaTelProveeMask
            ProveeTable("Fax") = AltaFaxProveeMask
            ProveeTable("Responsable") = UCase$(AltaRespoProveText)
            ProveeTable("Poblacion") = UCase$(AltaPoblaProveText)
            ProveeTable("RFC") = UCase$(AltaRfcProveText)
            ProveeTable.Update
            MsgBox "Proveedor " + lcCveProvee
            ProveeData.Refresh
            ProveeData.Recordset.FindFirst "CveProvee = " + lcCveProvee + ""
            If ProveeData.Recordset.NoMatch Then
              MsgBox "La clave del proveedor no existe"
            Else
              BuscaProvePanel.Visible = True
              AltaProvePanel.Visible = False
            End If
          End If
        End If
      End If
    End If
  End If
End Sub

```

DESARROLLO DEL SISTEMA

```
                End If
            End If
        Else
            MsgBox "Proveedor Ya Dado de Alta"
        End If
    End If
End If
End Sub
```

Para el caso de realizar una búsqueda del proveedor se utilizó un combo que contiene la opción ya sea por clave o por nombre. Habilita dos Frame's dependiendo de la opción que se elija, una para realizar la búsqueda por clave o para realizar la búsqueda por nombre como se muestra a continuación:



El código asociado al botón BUSCA es semejante en ambos casos, ya que hace la búsqueda en la tabla (en el caso por clave, mediante el índice de clave y en el caso por nombre, mediante un índice cuya llave primaria es el nombre del proveedor) en caso de no encontrar al proveedor (ya sea por clave o por nombre) este desplegará un mensaje indicando que no existe información sobre dicho proveedor. En caso de que exista información del proveedor se habilitará un panel el cual contiene controles como: un horizontal scroll bar, text box, label y command3D. Como se muestra a continuación:

The image shows a web form for managing providers. It contains the following fields:

- Clave Proveedor
- Proveedor
- Direccion
- Colonia
- Población
- C.P.
- Teléfono
- Responsable
- RFC
- FAX

Below the fields is a horizontal scroll bar labeled "Proveedor" with left and right arrow buttons. At the bottom of the form are two buttons: "Actualizar" and "Eliminar".

El código asociado al horizontal scroll bar, cumple con la función de ir recorriendo la tabla ya sea hacia adelante o ir retrocediendo, dependiendo de la opción que se le indique, cada vez que vaya cambiando de registro, se irá desplegando la información en las cajas de diálogo, pero éstas permanecerán deshabilitadas para el usuario final, en caso de que el usuario haya escogido a un proveedor para realizarle modificaciones, se encuentra el comando de ACTUALIZAR, cuya función es dejar habilitadas las text box y así poder realizar todas las modificaciones que el usuario requiere hacer, automáticamente el sistema hará la actualización en la tabla.

El comando ELIMINAR, su función principal es de dar de baja a todos los campos en la tabla que en ese momento este activo. Para llevar a cabo esto se creó una tabla que únicamente guarda todas las claves que fueron dadas de baja esto con la finalidad de que vuelvan a ser ocupadas por proveedores que vayan a ser dados de alta, todo esto con la seguridad de que cada clave asignada a cada proveedor sea única.

Como ejemplo de la programación llevada a cabo por estos comandos se muestra a continuación dicho código:

DESARROLLO DEL SISTEMA

PROCEDIMIENTO PARA REALIZAR LA BUSQUEDA DE PROVEEDORES POR MEDIO DE LA BARRA DE DESPLAZAMIENTO

```
Sub ProveeData_Validate (Action As Integer, Save As Integer)
    NomProveeText = UCase$(NomProveeText)
    DirecText = UCase$(DirecText)
    ColProveText = UCase$(ColProveText)
    PobiaProveText = UCase$(PobiaProveText)
    CpProveeMask.Mask = ""
    TelProveeMask.Mask = ""
    RespProveeText = UCase$(RespProveeText)
    RfcProveeText = UCase$(RfcProveeText)
    FaxProveeMask.Mask = ""
    CveProveeMask.Enabled = False
    NomProveeText.Enabled = False
    DirecText.Enabled = False
    ColProveText.Enabled = False
    PobiaProveText.Enabled = False
    CpProveeMask.Enabled = False
    CpProveeMask.Mask = ""
    TelProveeMask.Enabled = False
    TelProveeMask.Mask = ""
    RespProveeText.Enabled = False
    RfcProveeText.Enabled = False
    FaxProveeMask.Enabled = False
    FaxProveeMask.Mask = ""
End Sub
```

PROCEDIMIENTO PARA ELIMINAR UN PROVEEDOR

```
Sub EliminaProveCom_Click ()
    Dim lcCveProvee As String
    lcCveProvee = CveProveeMask
    ProveeData.Recordset.Delete
    ProveeData.Recordset.MoveNext
    If ProveeData.Recordset.EOF Then
        ProveeData.Recordset.MoveLast
    End If
    CveProveeBajaTable.AddNew
    CveProveeBajaTable("CveProvee") = lcCveProvee
    CveProveeBajaTable.Update
    CveProveeBajaTable.MoveFirst
End Sub
```

DESARROLLO DEL SISTEMA

Por último una de las operaciones que también realiza este módulo y no se mencionó al principio, por ser una operación general y que no puede faltar en ningún segmento del sistema es la de **Salir** y su código es el siguiente:

```
Sub SalirCom_Clic()  
  PrinciForm.Show  
  Unload Me  
End Sub
```

Todo esto se programó al igual que los demás módulos en código Visual Basic como se mostró anteriormente, utilizando así mismo algunas de las variables globales y parámetros, los cuales nos permitieron realizar las diferentes operaciones, y siempre se trató de utilizar la misma técnica de escritura de código y lógica en el desarrollo de la programación de todos los módulos.

Es de esta forma, de como se llevó a cabo la programación de los diferentes módulos con los que cuenta el sistema, cabe mencionar que estas son algunas de las pantallas definitivas del sistema y código asociado de algunos de los controles más importantes en su respectivo módulo.

176

CAPITULO VIII

**PRUEBAS E IMPLANTACION
DEL SISTEMA**

178

VIII.1 INTRODUCCION

La calidad de un sistema depende de su diseño, desarrollo, pruebas e implantación. Un aspecto de la calidad del sistema es su confiabilidad. Un sistema es confiable si al usarse de manera razonable, no produce fallas peligrosas o costosas. Esta definición distingue entre los errores de software, en los que el sistema no produce los resultados esperados, y las fallas que se presentan. Aunque es prácticamente imposible desarrollar software que se pueda demostrar que está libre de errores, los analistas hacen lo posible por prevenir la aparición de errores, usando métodos y técnicas que incluyen la detección de errores, su corrección y la tolerancia a éstos.

El aseguramiento de la calidad incluye pruebas para garantizar que el sistema se desempeña de forma adecuada y que cumpla con sus requerimientos. El propósito de la prueba es hallar errores, no el demostrar lo correcto de un sistema.

La implantación de un sistema se conforma por las actividades primarias de capacitación y revisión después de la implantación.

Después de implantar el sistema, debe hacerse una evaluación para determinar si el sistema cumple con las expectativas y donde serán necesarias las mejoras.

La calidad de los sistemas, la confianza del usuario y las estadísticas de operación quedan determinadas por técnicas como el registro de eventos, evaluación del impacto y las encuestas de actitud. Los métodos de recopilación de datos utilizados durante el análisis son igualmente efectivos durante la revisión después de la implantación.

PRUEBAS E IMPLANTACION DEL SISTEMA

La revisión no sólo dice qué tan bien está diseñado e implantado el sistema, sino que también es una valiosa fuente de información que puede aplicarse a un nuevo proyecto del sistema.

En el presente capítulo se explica como se llevaron a cabo estos últimos pero muy importantes aspectos del sistema:

- Pruebas del Sistema.
- Implantación del Sistema.

VIII.2 PRUEBAS DEL SISTEMA

La primera y normalmente la principal característica de un buen sistema es que sea correcto. Por esto fue de gran importancia llevar a cabo una serie de pruebas al sistema de control de presupuestos.

Las pruebas del sistema consistieron en un conjunto de pasos y procedimientos que se realizaron hasta poder estar seguros de que el sistema fuera correcto, es decir, que realice exactamente la función para lo que fue diseñado.

Las pruebas de integración dictan el orden en que los módulos deben estar disponibles para su depuración y las pruebas de unidad de los módulos.

VIII.2.1 Pruebas de Integración

Se requirió de una cuidadosa planeación y programación a tiempo para asegurarnos que los módulos estuvieran disponibles para su integración dentro del sistema cuando se necesitara.

Las pruebas realizadas al sistema se llevaron acabo mediante una serie de ensayos sencillos, cada uno de los cuales no consumió tiempo en la programación inicial, pero cuya suma total hubiera requerido una gran cantidad de tiempo. Es por esto, que las pruebas se realizaron a la par del desarrollo del sistema.

Cabe mencionar la importancia que tuvo la participación del usuario tanto en el diseño del Esquema Pseudo-prototipo como en muchas de las pruebas, ya que con su experiencia se evaluaron muchos casos no contemplados, los cuales fueron incluidos y también probados en todo momento de su desarrollo, para realizar las modificaciones que se requirieran.

Para llevar a cabo ésto, nos inclinamos en dos enfoques de integración: pruebas de integración ascendente y pruebas de integración descendente.

La integración ascendente es la estrategia tradicional empleada para integrar los componentes de un sistema de software en un todo funcionando.

El enfoque de integración ascendente aplicado al sistema consistió en desarrollar y probar los módulos individualmente antes de su integración.

PRUEBAS E IMPLANTACION DEL SISTEMA

Esto se llevó a cabo de la siguiente manera:

Primero se hicieron pruebas para el acceso al sistema, éstas consistieron en teclear un nombre de usuario y password erróneos, esto con la finalidad de validar el acceso al sistema, después un nombre de usuario correcto y el password incorrecto (y viceversa), esto con la intención de que las personas que tuvieran conocimiento de algún nombre de usuario y desconocieran el password no tuvieran acceso al mismo. Posteriormente se teclearon el nombre de usuario y el password correctos para verificar que dependiendo de la información tecleada se tuviera acceso a los módulos del sistema, dependiendo de los derechos previamente otorgados en el módulo de usuarios. Una vez hechas estas validaciones para el acceso al sistema, se procedió a probar los módulos individualmente.

Para el caso del módulo de presupuesto se creó una matriz de prueba, esto fue para realizar una simulación manual con datos de prueba que se introducirán al sistema, llevando a cabo los cálculos necesarios para comprobar que los datos arrojados por el sistema sean los correctos.

Esta matriz consistió en definir niveles con sus respectivos conceptos, información tal como: presupuesto anual, por presupuestar y presupuesto mensual, como se muestra en la figura 8.1.

MATRIZ DE PRUEBA PARA EL MODULO DE PRESUPUESTOS

NIVEL	CONCEPTO	PRESUPU.	GASTO	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICEMBRE	GASTO POR
		ANUAL	ANUAL													
SEGUROS		3,000,000.0	2,079,600.0	211,600.0	118,500.0	215,800.0	239,200.0	223,300.0	171,000.0	227,000.0	181,100.0	138,500.0	122,600.0	122,500.0	114,500.0	920,400.0
	DIRECCION DE SISTEMAS	2,200,000.0	1,858,600.0	181,600.0	83,500.0	175,800.0	173,200.0	153,300.0	121,000.0	192,000.0	126,100.0	128,500.0	117,600.0	116,500.0	114,500.0	543,400.0
	DIRECCION COMERCIAL	800,000.0	423,000.0	50,000.0	35,000.0	40,000.0	66,000.0	70,000.0	50,000.0	35,000.0	55,000.0	10,000.0	5,000.0	6,000.0	1,000.0	377,000.0
SISTEMAS		2,200,000.0														
	INVERSIONES	700,000.0	381,500.0	80,000.0	0.0	90,000.0	60,000.0	60,000.0	0.0	85,000.0	0.0	10,000.0	1,500.0	0.0	0.0	318,500.0
	GASTO	1,500,000.0	1,275,100.0	81,600.0	83,500.0	85,800.0	113,200.0	93,300.0	121,000.0	107,000.0	126,100.0	118,500.0	116,100.0	116,500.0	114,500.0	224,900.0
INVERSIONES		700,000.0														
	AS-400	400,000.0	121,500.0	40,000.0	0.0	55,000.0	10,000.0	0.0	0.0	20,000.0	0.0	0.0	1,500.0	0.0	0.0	278,500.0
	PC's	300,000.0	260,000.0	40,000.0	0.0	35,000.0	50,000.0	60,000.0	0.0	65,000.0	0.0	10,000.0	0.0	0.0	0.0	40,000.0
GASTO		1,500,000.0														
	TRANSPORTE	400,000.0	328,200.0	22,200.0	19,500.0	20,000.0	28,000.0	27,000.0	30,000.0	30,000.0	31,000.0	31,500.0	30,000.0	29,500.0	29,500.0	73,800.0
	MANTENIMIENTO	400,000.0	325,400.0	21,500.0	22,500.0	22,800.0	23,500.0	20,500.0	22,000.0	25,000.0	35,000.0	33,000.0	33,100.0	33,500.0	33,000.0	74,800.0
	NOMINA	600,000.0	554,500.0	22,900.0	40,000.0	41,000.0	43,700.0	45,800.0	49,000.0	52,000.0	52,100.0	52,000.0	52,000.0	52,000.0	52,000.0	45,500.0
	PAPELERIA	100,000.0	69,000.0	15,000.0	1,500.0	2,000.0	20,000.0	0.0	20,000.0	0.0	8,000.0	0.0	1,000.0	1,500.0	0.0	31,000.0

Figura 8.1

MATRIZ DE PRUEBA PARA EL MODULO DE PRESUPUESTOS

NIVEL	CONCEPTO	PRESUPU.	GASTO	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DIEMBRE	GASTO POR EJERCER
		ANUAL	ANUAL													
SEGUROS		3.000.000 D	2.079.600 D	211.600 D	118.500 D	215.800 D	239.200 D	223.300 D	171.000 D	227.000 D	181.100 D	136.500 D	122.600 D	122.500 D	114.500 D	920.400 D
	DIRECCION DE SISTEMAS	2.200.000 D	1.656.600 D	161.600 D	63.500 D	175.800 D	173.200 D	153.300 D	121.000 D	192.000 D	126.100 D	126.500 D	117.600 D	116.500 D	114.500 D	543.400 D
	DIRECCION COMERCIAL	800.000 D	423.000 D	50.000 D	35.000 D	40.000 D	66.000 D	70.000 D	50.000 D	35.000 D	55.000 D	10.000 D	5.000 D	6.000 D	1.000 D	377.000 D
SISTEMAS		2.200.000 D														
	INVERSIONES	700.000 D	381.500 D	80.000 D	0 D	90.000 D	60.000 D	60.000 D	0 D	85.000 D	0 D	10.000 D	1.500 D	0 D	0 D	318.500 D
	GASTO	1.500.000 D	1.275.100 D	81.600 D	63.500 D	85.800 D	113.200 D	93.300 D	121.000 D	107.000 D	125.100 D	116.500 D	116.100 D	116.500 D	114.500 D	224.900 D
INVERSIONES		700.000 D														
	AS-400	400.000 D	121.500 D	40.000 D	0 D	55.000 D	10.000 D	0 D	0 D	20.000 D	0 D	0 D	1.500 D	0 D	0 D	278.500 D
	PC s	300.000 D	250.000 D	40.000 D	0 D	35.000 D	50.000 D	60.000 D	0 D	65.000 D	0 D	10.000 D	0 D	0 D	0 D	40.000 D
GASTO		1.500.000 D														
	TRANSPORTE	400.000 D	328.200 D	22.200 D	19.500 D	20.000 D	26.000 D	27.000 D	30.000 D	30.000 D	31.000 D	31.500 D	30.000 D	29.600 D	29.500 D	73.800 D
	MANTENIMIENTO	400.000 D	325.400 D	21.500 D	22.500 D	22.800 D	23.500 D	20.500 D	22.000 D	25.000 D	35.000 D	33.000 D	33.100 D	33.500 D	33.000 D	74.600 D
	NOMINA	600.000 D	554.500 D	22.900 D	40.000 D	41.000 D	43.700 D	45.800 D	49.000 D	52.000 D	52.100 D	52.000 D	52.000 D	52.000 D	52.000 D	45.500 D
	PAPELERIA	100.000 D	69.000 D	15.000 D	1.500 D	2.000 D	20.000 D	0 D	20.000 D	0 D	8.000 D	0 D	1.000 D	1.500 D	0 D	31.000 D

Figura 8.1

PRUEBAS E IMPLANTACION DEL SISTEMA

Como se puede observar, se asignan datos de entrada a la matriz en sus diferentes niveles y conceptos en cada uno de los meses del año realizándose los cálculos necesarios para que la suma de los conceptos de cada nivel no sobrepase al presupuesto asignado a ese nivel, así mismo también se calculó la cantidad por presupuestar, y así sucesivamente con todos los niveles y conceptos correspondientes.

Con respecto al módulo de gasto, se creó una matriz similar como se muestra en la figura 8.2, teniendo los mismos niveles y conceptos que la matriz de prueba de presupuesto, con la información referente al presupuesto anual, gasto anual, realizando los mismos cálculos que en presupuesto.

MATRIZ DE PRUEBA PARA EL MODULO DE GASTO

NIVEL	CONCEPTO	PRESUPUESTO	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE	POR
		ANUAL													PRESUPUESTAR
SEGUROS		3.000.000,0													
	DIRECCION DE SISTEMAS	2.200.000,0	180.000,0	131.500,0	139.800,0	153.900,0	137.900,0	173.300,0	139.000,0	144.490,0	87.000,0	68.500,0	40.000,0	42.000,0	762.810,0
	DIRECCION COMERCIAL	800.000,0	50.000,0	35.000,0	40.000,0	66.000,0	70.000,0	50.000,0	35.000,0	0,0	0,0	0,0	20.000,0	22.000,0	413.000,0
DIRECCION DE SISTEMAS		2.200.000,0													
	INVERSIONES	700.000,0	100.000,0	83.000,0	64.000,0	59.000,0	20.000,0	47.400,0	7.800,0	6.500,0	0,0	9.000,0	0,0	0,0	303.300,0
	GASTO	1.500.000,0	80.000,0	48.500,0	75.800,0	94.900,0	117.900,0	125.900,0	131.200,0	137.990,0	87.000,0	59.500,0	40.000,0	42.000,0	459.310,0
INVERSIONES		700.000,0													
	AS-400	400.000,0	60.000,0	48.000,0	15.000,0	20.000,0	15.000,0	29.500,0	0,0	6.500,0	0,0	0,0	0,0	0,0	206.000,0
	PC s	300.000,0	40.000,0	35.000,0	49.000,0	39.000,0	5.000,0	17.900,0	7.800,0	0,0	0,0	9.000,0	0,0	0,0	47.300,0
GASTO		1.500.000,0													
	TRANSPORTE	400.000,0	20.000,0	16.000,0	20.900,0	25.000,0	26.000,0	50.000,0	39.900,0	43.500,0	19.000,0	20.000,0	5.000,0	4.000,0	118.700,0
	MANTENIMIENTO	400.000,0	20.000,0	12.500,0	19.900,0	29.900,0	27.900,0	35.000,0	46.000,0	42.900,0	14.000,0	14.500,0	10.000,0	8.000,0	125.400,0
	NOMINA	600.000,0	20.000,0	20.000,0	35.000,0	40.000,0	40.000,0	40.900,0	51.300,0	51.590,0	39.000,0	25.000,0	25.000,0	30.000,0	182.210,0
	PAPELERIA	100.000,0	20.000,0	0,0	0,0	0,0	30.000,0	0,0	0,0	0,0	15.000,0	0,0	0,0	0,0	35.000,0

Figura 8.2

PRUEBAS E IMPLANTACION DEL SISTEMA

Posteriormente después de haber realizado estas dos matrices con la información previamente definida y haber calculado los datos de salida; se prosiguió a capturar dicha información en el sistema real, esto con la finalidad de llevar a cabo una comparación entre los resultados obtenidos manualmente y los datos arrojados por el sistema. Cabe señalar que no hubo errores en cuanto a la información obtenida del sistema.

El fin del módulo de usuarios es dar de alta a usuarios, asignar derechos de operación, otorgar privilegios para el acceso a los diferentes módulos, aquí se probaron las opciones de alta, validar la no duplicidad, tanto de usuarios como de los password, así mismo la opción de eliminar.

En el caso del módulo de proveedores, se tiene el catálogo de todos los proveedores que tienen contacto con la compañía, aquí se manejan las opciones de alta, consulta y eliminación, checando que estas tres opciones funcionen de manera adecuada, que la asignación que hace el sistema de una clave al proveedor, sea única, y que cuando un proveedor sea dado de baja, esa clave volvería a ocupar para otro proveedor, teniendo siempre la seguridad de que estas claves no están duplicadas. En estos dos últimos módulos se dieron de alta a proveedores y usuarios, actualizando diferente información, así como eliminando alguna de ésta.

Después de esto, se continuo con las pruebas bajo el enfoque de integración descendente, el cual consistió en poner mayor énfasis en el sistema, donde el sistema es el programa total, que consta de la colección de los módulos. Este enfoque descendente de pruebas necesitó implementarse en etapas.

PRUEBAS E IMPLANTACION DEL SISTEMA

Primero se probaron los módulos de presupuesto y gasto, para ver si la interacción presupuesto-gasto fuera la correcta. Como gasto tiene la opción de pasar al módulo de proveedores, se prosiguió a probar esta unión, esto lo hicimos para consultar datos sobre los diferente proveedores.

La ventaja que obtuvimos de seguir esta prueba de integración descendente nos aseguró que el sistema integrado totalmente trabajará correctamente antes de su liberación.

VIII.3 IMPLANTACION DEL SISTEMA

Una vez llevadas a cabo las pruebas del sistema y verificando el buen funcionamiento de éste, se procedió a su instalación.

El proceso de instalación dentro de Visual Basic, consiste en preparar discos llamados de setup o de instalación. Estos discos se prepararon con una herramienta que nos proporciona el mismo Visual Basic llamada Setup Wizard.

El proceso de instalación es de lo más sencillo, y se reduce a ejecutar una instrucción dentro de la línea de comando de Windows y el sistema va solicitando los discos, como Excel o Word.

PRUEBAS E IMPLANTACION DEL SISTEMA

Se instaló el sistema en la red token-ring de la compañía. En este sentido se realizaron pruebas para verificar que las bases de datos del sistema fueran compartidas, que el tiempo de respuesta (acceso a los registros de las bases de datos) fuera satisfactorio.

La revisión del sistema se sigue dando dentro de la empresa esto debido a la importancia de los datos que maneja el sistema, como es el presupuesto mismo de la empresa.

Se plantea un funcionamiento provisional dentro de la dirección de sistemas para probar su operabilidad, para posteriormente presentarlo a las diversas áreas de la empresa.

CAPITULO IX

CONCLUSIONES

IX.1 CONCLUSIONES

Los capítulos contenidos dentro del presente trabajo de tesis se elaboraron de tal manera que el lector pueda comprenderlos de una manera clara y sencilla, ya que la información contenida en ellos se presenta de tal forma que cualquier lector inexperto en sistemas computacionales pueda introducirse a este ambiente, tomando una visión amplia de lo que es un sistema de información.

Cabe mencionar que aunado a esto, también se logró evaluar la herramienta gráfica Visual Basic, que aunque fue una restricción por parte de la empresa como herramienta de aplicación para el desarrollo del sistema, se llegó a la conclusión de que fue una buena decisión por parte de los directivos.

El diseño y desarrollo del sistema bajo Visual Basic mostró características que varían con respecto a una programación tradicional, como son el manejo de eventos y la comunicación entre aplicaciones, esto aprovechando la característica multitarea que nos ofrece el ambiente Windows.

La interfaz gráfica resulta muy atractiva para el usuario, ya que nos permite manejar bitmaps y gráficas de una manera muy sencilla, esto nos ahorra tiempo en el diseño de las pantallas, ya que en la programación tradicional bajo MS-DOS se dedica la mayor parte del tiempo y muchas líneas de código en el diseño de una interfaz pobre, y poco agradable.

CONCLUSIONES

El manejo de ayuda dentro del sistema resulta fundamental en las aplicaciones modernas, para que el usuario pueda manejar el sistema sin necesidad de recurrir a manuales o bien sin salirse del sistema para poder resolver sus dudas acerca del manejo del mismo.

Para el caso del nuestro sistema esto facilitó la implantación y puesta en marcha de la aplicación, ya que el usuario puede resolver sus dudas generales del sistema con solo oprimir una tecla o abrir un archivo en la ayuda general de Windows. Esto siguiendo la interfaz vistosa que nos ofrece Windows.

Si bien durante el desarrollo del sistema de control de presupuestos bajo Visual Basic no nos ofreció mayores ventajas en cuanto al manejo de diccionario de datos o diagramas de flujo, su capacidad de diseñar pantallas en poco tiempo y con el mínimo de programación nos permitió definir un Esquema Pseudo-prototipo el cual resultó muy importante para validar el buen funcionamiento del sistema.

Visual Basic al ser una programación que nosotros llamamos orientada a eventos, no nos permitió pasar parámetros a estos eventos (solamente maneja los parámetros ya definidos por Visual Basic) esto en ocasiones resultó inconveniente, ya que nos obligó a tener un sistema con demasiadas variables globales lo que puede hacer a nuestro código poco portable y difícil de depurar.

La forma que tiene de manejar el código es muy conveniente ya que en un primer paso realiza un análisis de las variables manejadas dentro del proyecto para seguir con una interpretación del código, lo que nos evita el fastidioso proceso de compilado y ligado de los compiladores convencionales. Y

CONCLUSIONES

posteriormente cuando el sistema ya esta listo podemos generar el programa ejecutable.

En cuanto a los requerimientos de hardware necesarios para que el sistema tenga un tiempo de respuesta aceptable, éstos siguen la política que fija Microsoft, de sistemas amigables y vistosos sin importar el hardware, ya que para que el sistema corra con un desempeño aceptable se necesitan al menos 4 Mbytes de memoria real y un procesador 386 o mayor como mínimo.

Las necesidades de memoria secundaria (Disco Duro), dependen de la cantidad de controles y bibliotecas utilizadas en el sistema; nuestro sistema de control de presupuestos requiere al menos 1.6Mb de disco duro, además de tener previamente instalado Windows 3.1.

Podemos mencionar que de los puntos más importantes que observamos con el desarrollo sistemático, se pudo llegar a obtener una serie de ventajas que dependieron principalmente en haber alcanzado el objetivo deseado como lo fue el de analizar y diseñar un sistema presupuestal que proporcionará un medio para poder identificar y evaluar eficientemente las actividades de la compañía, aprovechando sus recursos disponibles, mediante la utilización de una metodología propia, ya que sirvió para llevarnos a un mejor desempeño del mismo, por lo que podemos mencionar las siguientes ventajas:

- * **Comprensión.** Se mantuvo relaciones más estrechas entre uno como analista y los usuarios. Mejor comprensión por parte de estos últimos al tener una vinculación con lo que iba a ser el sistema.

CONCLUSIONES

Realización y pruebas de prototipos. De esta forma, ya no fue necesario esperar la finalización del desarrollo para poder realizar la validación del sistema. Las ventajas fueron múltiples, pues se detectaron errores sin haber escrito gran cantidad de código; además, los usuarios, o las personas encargadas de validar el futuro del sistema pudieron comprobar como iría funcionando el mismo.

En relación al Sistema de Control de Presupuesto concretamente; algunas de las características que se encontraran al tenerlo instalado en la empresa son las siguientes:

- * Tener una visión en cuanto a los objetivos a realizar, estableciendo metas y estrategias, desarrollo de programas, procurando una directriz en el desarrollo de operaciones.
- * Ayuda a la toma de decisiones y obliga a la consideración temprana de políticas básicas.
- * Exige la participación de todo el personal directivo para fijar metas y planes de una manera eficiente, de tal manera que se logren los objetivos departamentales y generales de la empresa.
- * Permite un autoanálisis de la empresa, y la corrección de la diferencia entre lo presupuestado y lo realizado.
- * Obliga a establecer una asignación definida de responsabilidad en cada función de la empresa.

CONCLUSIONES

- El empeño y esfuerzo que aportan las personas que intervienen en el control presupuestal, ayuda a mostrar y desarrollar aptitudes e iniciativas, obteniéndose una mejor calidad del personal ejecutivo.

En resumen, las características, que bien podemos mencionar como ventajas del sistema, son la planeación, de tal manera que permita el logro de los objetivos deseados, el equilibrio en la estructura financiera, el aprovechamiento de los recursos y su óptima obtención, la medición de la eficiencia y la delimitación de responsabilidades dando lugar a una toma de decisiones adecuada y oportuna.

Las perspectivas del sistema son de crecimiento dado que la recepción de parte de los directivos fue de su agrado, se considera que es una herramienta que ve los presupuestos de una manera diferente a la tradicional.

En los tiempos modernos hablar de presupuesto-gasto es un renglón coyuntural en la vida de cualquier persona y por supuesto de una empresa. La mayoría de la gente lleva un control de lo que gasta pero muy pocos planean lo que van a gastar.

Con el sistema de control de presupuestos un directivo esta obligado a planear su presupuesto, lo que lo lleva a una mejor evaluación de resultados. Pero el siguiente año surge un nuevo presupuesto y los directivos desean tomar como base el presupuesto del año que acaba de terminar; de ahí surge la necesidad de que el sistema de control de presupuesto maneje un módulo de proyecciones de presupuesto, para que cada uno de los directivos de área puedan tomar el presupuesto del año anterior y llevarlo al año que comienza con un factor de proyección que en junta de consejo se haya decidido.

CONCLUSIONES

En resumen, podemos mencionar que las ventajas del sistema, son la planeación, el equilibrio en la estructura financiera, el aprovechamiento de los recursos, la medición de la eficiencia y la delimitación de responsabilidades.

Por lo cual se concluye que en la elaboración de la presente tesis se logró dar cumplimiento al objetivo previamente establecido el cual fue, llevar a la práctica los conocimientos adquiridos durante la carrera aplicándolos a un problema real.

Cliente: **COMPANÍA ASEGURADORA**
 Proyecto: **Sistema de Control de Presupuestos**

Descripción de la Actividad	Periodo: Semanal																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1 Definición del Problema	■																							
2 Determinación de Requerimientos	■	■																						
3 Estudio de Factibilidad		■	■																					
4 Diseño Conceptual			■	■																				
5 Diseño de Pantallas				■	■	■																		
6 Esquema Pseudo-Prototipo					■	■																		
7 Validación del EP					■	■	■																	
8 Diseño de la Base de Datos								■	■															
9 Módulo de Presupuestos									■	■	■													
10 Módulo de Gasto										■	■	■												
11 Módulo de Usuarios											■	■												
12 Módulo de Proveedores												■	■											
13 Integración de Módulos														■	■									
14 Diseño de la Matriz de Prueba																■	■							
15 Aplicación de la Matriz de Prueba																	■	■						
17 Pruebas de Integración Asc																		■	■					
18 Pruebas de Integración Desc																			■	■				
20 Instalación																				■	■			
21 Evaluación de Resultados																					■	■		

 Tiempo Estimado
 Tiempo Real
 Evaluación en Proceso

197

821

APENDICE A

CONTROLES DE VISUAL BASIC

FALTA PAGINA

199.a la 2.00

APENDICE A

Anibutton

Es un control flexible que permite usar iconos, bitmaps o metafiles para definir controles propios.

Combo Box

Combina una caja de texto con una lista. Esta opción se usa para dar al usuario la opción de capturar en la porción de texto, o seleccionar un elemento de la lista.

Comm

Este control hace posible la comunicación serial de la aplicación permitiendo la transmisión y recepción de datos a través de un puerto serial.

Command Button

Lleva a cabo un comando o acción cuando el usuario lo selecciona (por lo general se le oprime).

Command3D

En adición al botón de comando sencillo, despliega textos en 3era. dimensión, así como un icono o un bitmap, además de la apariencia de relieve en la ventana.

Common Dialog

Proporciona un conjunto estándar de cajas de diálogo, necesarias para operaciones como abrir, salvar e impresión de archivos.

APENDICE A CONTROLES DE VISUAL BASIC

Check Box, Check3D

Despliega una opción verdadera/falsa o si/no, y pueden ser elegidos muchos de estos controles a la vez.

Data

Proporciona una manera de acceso a información en bases de datos, permite navegar registro por registro de una tabla y desplegar los datos de un registro en los controles de una pantalla.

Directory List box

Despliega los directorios y las rutas de acceso de una unidad de disco, en tiempo de ejecución solamente.

Drive List Box

Permite al usuario elegir una de las unidades de disco disponibles, en tiempo de ejecución. Es de gran utilidad si se desea crear cajas de diálogo que permitan al usuario abrir un archivo desde cualquier unidad de disco válida en su sistema.

File List Box

Encuentra y lista los archivos que existen en el directorio especificado a través de la propiedad Path en tiempo de ejecución.

Frame, Frame3D

Agrupador de controles.

Graph

Permite diseñar gráficas interactivas en las formas. En tiempo de ejecución es posible enviar nuevos datos a las gráficas y rediseñarlas, imprimirlas y copiarlas en el portapapeles.

Grid

Despliega un conjunto de renglones y columnas, donde la intersección de ellos es una celda. Es de gran utilidad cuando el sistema produce como salida gran cantidad de información. Es posible leer y asignar el contenido de las celdas por programa.

Horizontal Scroll Bar

Proporciona una navegación fácil a través de una larga lista de elementos o de una gran cantidad de información. El recorrido que hace en el control es de izquierda a derecha y viceversa.

Image

Control gráfico que despliega imágenes. Básicamente tiene el mismo funcionamiento que el control Picture Box, sólo que soporta un subconjunto de las propiedades, eventos y métodos del mismo.

KeyStat

Este control se usa para desplegar o modificar el estado de las teclas CAPS LOCK, NUM LOCK, INS y SCROLL LOCK.

APENDICE A CONTROLES DE VISUAL BASIC

Label

Despliega textos, con los cuales el usuario no puede interactuar ni modificar.

Line

Es un control gráfico que despliega líneas verticales, horizontales y diagonales.

List Box

Despliega una lista de elementos de los cuales el usuario puede seleccionar uno o más. Si el número de elementos excede al que puede ser desplegado, una barra de desplazamiento vertical aparece automáticamente.

MaskedEdit

Permite restringir la entrada de datos a la forma como debe producirse la salida. Suple el formato establecido mediante la propiedad **Mask**, con los datos que están siendo capturados.

Ole

OLE (Object Linking and Embedding) es una tecnología que permite al programador de aplicaciones bajo Windows crear una aplicación que pueda desplegar datos de muchas otras aplicaciones.

Option Button, Option3D

Como parte de un conjunto de opciones con otros controles de su tipo, despliega múltiples elecciones, de las cuales el usuario sólo puede escoger una.

APENDICE A CONTROLES DE VISUAL BASIC

Panel3D

Control que permite desplegar textos planos o tridimensionales y agrupar otros controles en un fondo tridimensional, o dar una apariencia tridimensional a controles sencillos como cajas de listas, combos, barras de desplazamiento, etc.

Picture Box

Despliega un gráfico bitmap (.BMP), icono (.ICO) o metafile. Despliega sólo la parte de la gráfica que se ajuste bien al tamaño del rectángulo.

Picture Clip

Este control permite seleccionar un área de un bitmap y desplegar la imagen en una forma o en una caja de imagen. Es invisible en tiempo de ejecución.

Pointer

Proporciona una manera de mover los objetos y cambiar su tamaño (no es en sí un control).

Report

Control visible en modo de diseño e invisible en tiempo de ejecución, que actúa en conjunto con la utilería propia de Visual Basic. El uso de estos dos elementos permite generar reportes a la medida, transparentemente, vía programa. Al colocar este control en la forma y asignarle sus propiedades, se pueden agregar sofisticadas capacidades de reporte a la aplicación, sin necesidad de escribir código extenso.

APENDICE A CONTROLES DE VISUAL BASIC

Shape

Es un control gráfico utilizado para desplegar figuras geométricas como rectángulos, cuadrados, óvalos, círculos, rectángulos redondeados y cuadrados redondeados.

Spin Button

Usado con otro control (una caja de texto, por ejemplo), permite incrementar y decrementar números. se puede usar también para navegar hacia adelante o hacia atrás a través de un rango de valores o una lista de términos.

Text Box

Proporciona un área para capturar o desplegar textos.

Timer

Ejecuta el código especificado en intervalos regulares de tiempo, establecidos en su propiedad interval.

Vertical Scroll Bar

Tiene la misma función que el control anterior, sólo que la navegación se efectúa de arriba a abajo y viceversa.

APENDICE B

PROPIEDADES DE VISUAL BASIC

208

APENDICE B

Action

Determina el tipo de acción por ejecutarse. Los valores que puede tomar son numéricos.

ActiveControl

Especifica el control que posee el focus; cuando se refiere a una forma, la propiedad indica el control que debiera tener el focus siempre que aquella esté activa.

Align

Determina si la imagen es desplegada en cualquier parte de la forma, necesariamente arriba o abajo de la misma.

Alignment

Indica la alineación del texto en un control.

AutoSize

Determina si el control se debe de ajustar al contenido (imagen, texto, etc.).

BackColor

Determina el color de fondo del control.

BackStyle

Indica si el fondo de un control es opaco o transparente.

APENDICE B PROPIEDADES DE VISUAL BASIC

BorderWidth

Especifica al ancho del borde de un control.

BorderStyle

Determina el tipo de borde del control. Los valores que puede tomar son del 0 al 3.

Caption

Determina el título de la forma o el texto desplegado en un control.

Cols

Especifica el número de columnas de un grid (mínimo 1).

CopiestoPrinter

Determina el número de copias mandadas a la impresora.

Count

Indica el número de copias de formas cargadas, el número de controles en una forma o el número de objetos en una colección.

CurrentX, Y

Determina las coordenadas horizontales y verticales de un gráfico, o para imprimir un objeto.

DatabaseName

Determina el nombre y la ruta de localización de la base de datos a la que está conectado un control data.

DataChanged

Indica si los datos en el control han sido cambiados por algún proceso, y por lo tanto la información no concuerda con la del registro actual.

DataField

Vincula a un control con un campo.

DataSource

Determina el control data a través del cual el control actual está ligado a la base de datos.

DragIcon

Define el icono desplegado para el apuntador del mouse, cuando se realiza un movimiento de arrastre.

Drive

Determina la unidad de disco seleccionada en tiempo de ejecución.

Enabled

Determina la unidad de disco seleccionada en tiempo de ejecución.

Exclusive

Define si el acceso a una base de datos para un data control es exclusivo (un solo usuario) o es un acceso múltiple.

APENDICE B PROPIEDADES DE VISUAL BASIC

FillColor

Define el color del relleno de un control.

FixedCols, Rows

Determina el total de columnas y renglones fijos para un grid.

FontBold, FontItalic, FontStrikethru, FontTransparent, FontUnderline, FontSize

Determina el formato del texto desplegado en los controles, como negritas, subrayados, cursivas, tamaño, etc.

ForeColor

Determina el color del frente usado para desplegar texto o gráficas en un objeto.

Height, Width

Define las dimensiones de un objeto.

Hide Selection

Especifica si un texto debe aparecer iluminado cuando un control pierda el focus.

Icon

Determina el icono que será desplegado en tiempo de ejecución, cuando la forma sea minimizada.

Index

Determina el índice que identifica al control, si este forma parte de un arreglo de controles.

Interval

Define el número de milisegundos entre cada una de las llamadas al evento Timer del control.

List

Determina los elementos contenidos en una porción de una lista de un control.

ListCount

Determina el número de elementos en una porción de una lista de un control

ListIndex

Determina el índice del elemento actualmente seleccionado en un control.

MaxLength

Define si hay un límite para la captura de datos en una caja de texto, y cual es ese límite.

Mousepointer

Determina el tipo de apuntador del mouse.

MultLine

Especifica si una caja de texto puede aceptar y desplegar múltiples selecciones en una lista y como debe hacerse esta selección.

APENDICE B PROPIEDADES DE VISUAL BASIC

MultiSelect

Determina si el usuario puede hacer múltiples selecciones en una lista y como debe hacerse esta selección.

Name

Es el nombre que identifica a cualquier control o forma, y que debe especificarse en el código cuando se quiera hacer referencia al objeto en cuestión.

Picture

Determina el gráfico que será desplegado en el control.

Rows

Determina el número de renglones de un grid.

Visible

Determina si un control está visible u oculto en tiempo de ejecución.

GLOSARIO

GLOSARIO

API (Application Program Interface)

Interfaz de programa de aplicación. Lenguaje y formato utilizados por un programa para comunicarse con otro programa. También puede incluir los comandos utilizados para interrumpir a la computadora con el fin de llamar la atención a otro programa.

BASE DE DATOS

- (1) Un conjunto de archivos interrelacionados que es creado y manejado por un sistema de gestión o de administración de bases de datos (DBMS).
- (2) Cualquier conjunto de datos almacenado electrónicamente.

CICLO DE DESARROLLO DE UN SISTEMA

Ciclo de desarrollo de sistema. Secuencia de eventos en el desarrollo de un sistema de información (aplicación), lo cual requiere del esfuerzo mutuo, tanto de parte del usuario como del equipo técnico.

- I. ANALISIS Y DISEÑO DE SISTEMAS
 - estudio de factibilidad
 - diseño general
 - creación de prototipos
 - diseño detallado
 - especificaciones funcionales
- II. APROBACION DEL USUARIO
- III. PROGRAMACION
 - diseño
 - codificación
 - prueba
- IV. IMPLEMENTACION
 - entrenamiento
 - conversión
 - instalación

CLIC

Seleccionar un objeto presionando el botón del ratón cuando el cursor está apuntando la opción de menú o icono deseados.

GLOSARIO

CODIGO

Un conjunto de simbolos de máquina que representa datos o instrucciones.

COMPILADOR

(1) Software que traduce lenguajes de programación de alto nivel en lenguaje máquina. Un compilador habitualmente genera en primer lugar lenguaje ensamblador y a continuación traduce este lenguaje a lenguaje máquina.

(2) Software que convierte un lenguaje de alto nivel a un juego de órdenes y un texto en una representación de muy bajo nivel que pueda ejecutarse.

CONTROL

Termino general utilizado para describir cualquier forma o elemento gráfico que se dibuje sobre una forma, incluyendo cajas de texto, caja de listados, botones de comando, cajas de imágenes, barras de desplazamiento e iconos.

CP/M (Control Program for Microprocessors)

Programa de control para microprocesadores. Un sistema operativo monousuario para los microprocesadores 8080 y Z80, de Digital Research Inc.

DBMS (Database Management System)

Sistema de administración o gestión de bases de datos. Software que controla la organización, almacenamiento, recuperación, seguridad e integridad de los datos en una base de datos. Acepta pedidos de datos desde un programa de aplicación y le ordena al sistema operativo transferir los datos apropiados.

DLL (Dinamic Link Library)

Biblioteca de enlaces dinámicos. Un conjunto de rutinas de programa que están disponibles para las aplicaciones en tiempo de ejecución.

DICCIONARIO DE DATOS

Catálogo que contiene los nombres y las estructuras de todos los tipos de datos.

ENSAMBLADOR

Un programa de computadora que funciona con datos simbólicos de entrada, para producir instrucciones de máquina, mediante el desempeño de funciones tales como la traducción de código, simbólicos de operaciones a instrucciones operativas de computadora, la asignación de posiciones en el almacenamiento para instrucciones sucesivas y la computación de direcciones absolutas a partir de direcciones simbólicas.

EVENTO

Una acción reconocida por un control de Visual Basic.

EVENT-DRIVEN (manejador de eventos)

Aplicación que responde a la entrega del usuario o de otra aplicación a intervalos irregulares. Se controla mediante las elecciones del usuario (seleccionar menú, pulsar botón, etc.).

GDI (Graphics Device Interface)

Interfaz de dispositivos para gráficos. El lenguaje de gráficos utilizado en Microsoft Windows.

GUI (Graphical User Interface)

Interfaz gráfica de usuario basada en gráficos que incorpora iconos, menús enrollables y un ratón, tal como se encuentra en los entornos Macintosh, Windows, OS/2 Presentation Manager y GEM.

HARDWARE

El equipo físico o los dispositivos que forman una computadora y el equipo periférico.

GLOSARIO

HMA (High Memory Area)

Area alta de memoria. En PC, una región de 64K de memoria extendida por encima de un megabyte (1.024 - 1.088K) a la que se puede acceder por el DOS. El controlador HIMEM.SYS de Microsoft utiliza esta memoria.

ICONO

Una diminuta representación pictórica de un objeto, tal como una aplicación, archivo o unidad de disco, que se utiliza en interfaces gráficas de usuario (GUI). El usuario selecciona un objeto apuntando el icono correspondiente y presionando la tecla del ratón. Los iconos pueden ser desplazados en la pantalla.

INGENIERIA DE SOFTWARE

Es la disciplina tecnológica preocupada de la producción sistemática y mantenimiento de los productos de software que son desarrollados y modificados en tiempo y dentro de un presupuesto definido, la ingeniería de software difiere de la programación tradicional, en que se utilizan técnicas de ingeniería para especificar, diseñar, instrumentar, validar y mantener los productos dentro del tiempo y el presupuesto establecidos para el proyecto, además esta ingeniería se preocupa por aspectos administrativos que quedan fuera del dominio normal de la programación.

INTERFAZ

Una conexión e interacción entre hardware, software y usuario. Las interfaces de hardware son los conectores, zócalos y cables que transportan las señales eléctricas en un orden prescrito. Las interfaces de software son los lenguajes, códigos y mensajes que utilizan los programas para comunicarse unos con otros, tal como entre un programa de aplicación y el sistema operativo. Las interfaces de usuario son los teclados, ratones, diálogos, lenguajes de comando y menús empleados para la comunicación entre el usuario y la computadora.

KERNEL (núcleo)

La parte fundamental de un programa, tal como un sistema operativo, que reside en memoria todo el tiempo.

MEMORIA CONVENCIONAL

Primeros 640Kb. de memoria que MS-DOS utiliza para ejecutar aplicaciones.

MEMORIA EXPANDIDA

Memoria adicional a la memoria convencional, disponible para aplicaciones. Se asigna en bloques de 16Kb. Por ejemplo Windows hará uso de memoria expandida sólo si se inicia con la opción /R En el modo extendido de 386, Windows simula memoria expandida para las aplicaciones que la necesiten. En modo estándar o extendido del 386, permite a las aplicaciones utilizar memoria expandida pero no hace uso de la misma para el manejo de aplicaciones.

MEMORIA EXTENDIDA

Memoria adicional a la memoria, a la que no tiene acceso MS-DOS y sus aplicaciones. Esta memoria no puede ser utilizada por las computadoras 8086/8088. En el modo extendido del 386, de Windows hace uso de la memoria extendida para manejar y ejecutar aplicaciones.

MEMORIA VIRTUAL

Sistema de manejo de la memoria utilizada por ejemplo por Windows en el modo extendido del 386 que le permite comportarse como si tuviera más memoria de la actualmente disponible en el sistema. Es equivalente a la memoria RAM libre más la cantidad de espacio de disco asignada al archivo de intercambio que Windows utiliza para simular memoria RAM adicional.

METODO

Una palabra de código Visual Basic que es similar a una función o instrucción, pero que siempre actúa sobre un control en particular. Para cada control, Visual Basic predefine una serie de métodos que se pueden utilizar.

MODO PROTEGIDO

Modo operativo de una computadora capaz de direccionar memoria extendida directamente.

GLOSARIO

MODO REAL

Modo operativo en el que se ejecuta Windows para obtener la máxima compatibilidad con versiones de aplicaciones previas a la 3.0. Es el único modo disponible para computadoras con menos de 1 MB de memoria extendida.

MS-DOS (Microsoft-Disk Operating System)

Sistema Operativo en disco de Microsoft. Sistema operativo de un solo usuario para PC, de Microsoft. Es casi idéntico a la versión de IBM, que se llama DOS, y ambas versiones se llaman DOS genéricamente.

OS/2

Sistema operativo multitarea de un solo usuario para PC 286 y superiores. Las versiones de 16 bits se han desarrollado conjuntamente por IBM y Microsoft. Las versiones de 32 bits se han desarrollado independientemente.

Es un sistema avanzado que no está confinado al límite ínfimo de 1 MB de DOS. Aunque se han añadido órdenes nuevas, muchas de OS/2 son las mismas que las de DOS. Las versiones de 16 bits pueden direccionar 16MB de RAM y 1GB de memoria virtual. OS/2 requiere 4 MB de RAM(excepto la versión 1.3), aunque se encuentra ejecutándose en computadoras con 8MB y más de 8.

PLATAFORMA

La arquitectura del hardware de un modelo particular o familia de computadoras. La plataforma es el estándar con que los diseñadores de software escriben sus programas.

PRESUPUESTAR

Significa básicamente: asignar un valor monetario a cada plan, balancear ingresos y gastos para un uso eficaz del capital, y comparar el movimiento real del efectivo con el que se planeó.

PROGRAMA DE APLICACION

Programa utilizado para un trabajo en particular, tal como procesamiento de texto, manejo de base de datos, etc.

PROTOTIPO

Es el estudio donde se aplican los principios de análisis y diseño, para la construcción de un modelo de software. Es un modelo mediante el cual se pueden obtener los requisitos del sistema de forma efectiva.

RAM (Random Access Memory)

Memoria utilizada para ejecutar aplicaciones y realizar otras tareas necesarias mientras la computadora esté encendida. Al apagar la computadora, toda la información que RAM contiene se pierde.

REGISTRO FISICO

Es la unidad de información leída o escrita en un dispositivo.

REGISTRO LOGICO

Es una colección de datos tratados como una unidad desde el punto de vista usuario.

REQUERIMIENTO

Es una característica que debe incluirse en un sistema, esta puede ser la inclusión de una determinada forma de capturar o procesar datos, producir información, mejorar el tiempo de respuesta, controlar una actividad de la empresa o brindar soporte a la gerencia.

SISTEMA

- (1) Grupo de componentes relacionados que interactúan para realizar una tarea.
- (2) Un sistema de informática (computer system) está compuesto por la CPU, el sistema operativo y los dispositivos periféricos.

GLOSARIO

(3) Un sistema de información (information system) está constituido por la base de datos, todos los programas de ingreso, actualización, consulta e informes de datos y los procedimientos manuales y por máquina.

(4) "El sistema" se refiere frecuentemente al sistema operativo.

SISTEMA DE INFORMACION

Conjunto de elementos y procedimientos que utilizados metódica y adecuadamente, son capaces de manipular datos, procesarlos y emitir un conocimiento sobre determinadas situaciones o aspectos de una organización, lo cual permitirá tomar decisiones para la consecución de los objetivos de la empresa.

SISTEMA MONOUSUARIO

Es aquel sistema que solamente puede ejecutar un proceso a la vez, cuenta con el control absoluto de los recursos de la computadora. Generalmente es un sistema operativo propietario y dependiente de la arquitectura.

SISTEMA MULTIPROCESO

Son más de dos aplicaciones ejecutándose en dos o más procesadores.

SISTEMA MULTIUSUARIO

Sistema de computación con varias terminales, habilitadas para que varios usuarios, cada uno con su propia terminal, utilicen la computadora.

SISTEMA MULTITAREA

Son más de dos aplicaciones ejecutándose aparentemente al mismo tiempo por un solo procesador.

SISTEMA OPERATIVO

Un programa maestro de control que maneja la computadora y actúa como planificador y agente de tránsito. Es el primer programa que se carga en la memoria de la computadora después de que ésta sea encendida, y el núcleo central (kernel) del mismo debe estar siempre residente en memoria. El

sistema operativo puede ser desarrollado por el fabricante del hardware en el que corre o por una casa independiente de software.

SOFTWARE

Instrucciones para una computadora. Una serie de instrucciones que realizan una tarea en particular se llama programa o programa de software. Las dos categorías principales son software de sistemas de aplicaciones.

El software de sistemas se compone de programas de control, incluyendo el sistema operativo, software de comunicaciones y administrador de bases de datos.

El software de aplicaciones es cualquier programa que procesa datos para el usuario (inventario, nómina, hoja de cálculo, procesador de texto, etc.).

TSR's (Terminate and Stay Resident)

Programa residente en memoria. Se refiere a programas que permanecen en memoria de forma que se puedan extraer en cualquier aplicación pulsando simplemente una combinación de teclas.

UNITAREA

Es cuando a un usuario se le permite establecer solo una trayectoria de control para operar en su beneficio.

VENTANA

Area rectangular de la pantalla en la que se encuentra una aplicación o documento.

WINDOWS

Entorno operativo para gráficos de Microsoft que se integra con DOS. Proporciona un entorno de sobremesa similar al Macintosh, en el cual cada aplicación activa se visualiza en una pantalla movable y redimensional sobre la pantalla.

Con objeto de usar todas las funciones del Windows, las aplicaciones deben escribirse específicamente para él. Sin embargo, Windows también ejecuta aplicaciones de DOS y se puede usar como el entorno operativo desde el que se lanzan todos los programas.

BIBLIOGRAFIA

FALTA PAGINA

227..a la 228

BIBLIOGRAFIA

Análisis y Diseño de Sistema de Información.
James A. Senn
Editorial Mc Graw-Hill; 1992

Sistema de Información por Computadora.
Marquez Vite, Juan Manuel
Editorial Trillas; 1987

Programer's Introduction to Visual Basic.
Keynon Brown
Editorial Sysbex Inc.
Alameda, California USA; 1992

Visual Basic 3.0 Programming
Torgerson, Thomas W.
New York; 1994

Visual Basic Programming with windows applications
Hergert, Douglas A.
New York; 1991

Programming in Windows
Murray, William H. & Papas Chris H.
Editorial Osborne McGraw-Hill
Madrid, España; 1992

Diccionario de Computación
Alan Freedman
Editorial McGraw-Hill
Marzo; 1994

El Lenguaje de Proigramación en C
Brianw. Kernighan
Pentice Hall
1992

BIBLIOGRAFIA

Clipper 5 Referencia Rapida
Francisco Marín Quiroz
Macrobit Editores S.A. C.V.
1991

An Introduction to OS.
Harvey M. Deitel.
Addison-Wesley Publishing Company.
1990.

PCMagazine (en español).
Vol. 4. Num. 2.
America S. A. 1993.

PCMagazine (en español).
Vol. 4. Num. 7
America S. A. 1993.

PCWord.
PCWord Communications.
Octubre 1993.