



U.N.A.M.

**SISTEMA INTEGRAL PARA EL INSTITUTO FEDERAL ELECTORAL
DENTRO DE LA DIRECCIÓN DE RECURSOS FINANCIEROS**

Por: Rubén Reyes Olivera

1998

**TESIS CON
FALLA DE ORIGEN**

257708



Universidad Nacional
Autónoma de México

UNAM



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 está 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.



INDICE

	Pág.
Introducción.	i
Objetivos.	iii
1.- Antecedentes del I.F.E.	1
1.1.- Antecedentes y fundamentos legales del Instituto Federal Electoral.	1
1.1.1.- Código Federal de Instituciones y Procedimientos Electorales.	3
1.1.2.- Creación del Instituto Federal Electoral.	3
1.2.- Estructura orgánica y funciones del Instituto Federal Electoral.	4
1.2.1.- Organos Centrales.	5
1.2.2.- Organos desconcentrados permanentes.	9
1.2.3.- Organos desconcentrados temporales.	10
2.- Ciclos de Vida del Software	11
2.1.- Importancia de los Ciclos de Vida del Software.	11
2.2.- Ciclo de Vida Clásico.	12
2.3.- Ciclo de Vida de Construcción de Prototipos.	13
2.4.- Ciclo de Vida del Modelo en Espiral.	15
2.5.- Técnicas de la Cuarta Generación (4GL).	16
2.6.- Selección del Modelo de Ciclo de Vida.	17
2.6.1.- Ventajas y Desventajas del Ciclo de Vida Clásico	17
2.6.2.- Ventajas y Desventajas del Ciclo de Vida de Prototipos	18
2.6.3.- Ventajas y Desventajas del Ciclo de Vida en Espiral.	19
2.6.4.- Ventajas y Desventajas de las Técnicas de Cuarta Generación (4GL)	19



3.- Fundamentos de Bases de Datos.	21
3.1.- Conceptos Básicos.	21
3.2.- Conceptos Asociados a bases de datos.	22
3.3.- Necesidad de las bases de datos.	24
3.3.1.- Ventajas del modelo de bases de datos.	24
3.4.- Sistema manejador de bases de datos (DBMS).	25
3.4.1.- Objetivo de un DBMS.	25
3.4.2.- Características de los DBMS.	26
3.5.- Modelos de los DBMS.	26
3.5.1.- Modelo de Red.	26
3.5.2.- Modelo Jerárquico.	27
3.6.- Enfoque Relacional.	28
3.6.1.- Definición de un DBMS Relacional.	28
3.6.2.- Reglas de Integridad.	28
3.7.- Asociaciones.	30
3.8.- Normalización.	30
3.9.- Desnormalización.	34
4.- Análisis del Sistema.	36
4.1.- Determinación de los Requerimientos de la Información.	36
4.1.1.- Técnicas para la recopilación de hechos de estudio.	37
4.1.1.1.- La Entrevista.	37
4.1.1.2.- Método de Análisis en Grupo.	38
4.1.1.3.- El Cuestionario.	38
4.1.1.4.- La Observación.	39
4.2.- Análisis de las Necesidades del Sistema.	39
4.2.1.- Preparación del reporte de Terminación del Análisis de Sistemas.	40
4.2.2.- Relación costo beneficio.	41
4.2.3.- Conceptos de red de área local.	42
4.3.- Justificación del sistema.	45
4.3.1.- Objetivo del sistema.	45
4.3.2.- Prioridades para las características del sistema.	45
4.3.3.- Análisis Costo-Beneficio.	46



5.- Diseño y Documentación del Sistema.	49
5.1.- Técnicas Estructuradas Aplicadas.	49
5.1.1.- Diseño Ascendente.	49
5.1.2.- Diseño Descendente.	50
5.1.3.- Desarrollo Modular.	51
5.1.4.- Diagrama estructural.	52
5.2.- Técnicas de diseño y documentación.	54
5.2.1.- El método HIPO.	54
5.2.2.- Diagramas de Flujo.	57
5.2.3.- Diagramas Nassi-Schneiderman.	58
5.2.4.- Diagramas Warnier-Orr.	60
5.2.5.- Pseudocódigo.	61
5.2.6.- Manuales de procedimientos.	61
5.2.7.- El método Folklore.	61
5.3.- Interfaz con el usuario.	63
5.3.1.- Factores Humanos.	63
5.3.2.- Estilos de interacción entre hombre y máquina.	64
5.3.3.- Diseño de la interfaz hombre-máquina.	65
5.3.4.- Aspectos de Diseño.	65
5.3.5.- Evaluación del Diseño.	67
5.3.6.- Directrices para el diseño de interfaces.	69
5.3.6.1.- Interacción general.	69
5.3.6.2.- Visualización de la información.	70
5.3.6.3.- Entrada de datos.	71
5.3.7.- Estándares de Interfaz.	72
5.4.- Definición de Módulos.	72
5.5.- Diseño de la ayuda en línea.	75
5.5.1.- Definición del Tipo de Usuario.	76
5.5.2.- Planeación del Contenido del Sistema de Ayuda.	77
5.5.3.- Planeación de la Estructura de los tópicos de ayuda.	78
5.5.4.- Planeación de la Sensibilidad al Contexto.	80
5.5.5.- Determinación de la estructura del archivo de tópicos.	80
5.5.6.- Diseño de los tópicos de Ayuda.	82
5.5.7.- Diseño del texto de los tópicos de Ayuda.	82



6.- Desarrollo del Sistema Integral de Oficinas Centrales.	84
6.1.- Calendario de Actividades.	84
6.2.- Definición.	84
6.3.- Usuarios involucrados.	86
6.4.- Descripción de Hardware.	86
6.5.- Estrategia de Solución.	87
6.6.- Diagrama de Flujo de Datos.	87
6.7.- Selección del producto de Software.	88
6.8.- Diseño.	95
6.8.1.- Diseño de datos.	95
6.8.2.- Diagramas de transición de estado.	97
6.8.3.- Definición de Módulos.	97
7.- Implementación y Evaluación del Sistema Integral de Oficinas Centrales.	103
7.1.- Pruebas de Arranque.	103
7.2.- Implementación.	104
7.3.- Capacitación.	106
7.4.- Mantenimiento.	111
8.- Seguridad y Acceso al Sistema.	115
8.1.- Seguridad: Consideraciones generales.	115
8.2.- Seguridad en SQL.	116
8.3.- Otros aspectos de seguridad.	117
8.4.- Seguridad dentro del Sistema Integral de Oficinas Centrales.	118
8.5.- Seguridad dentro del entorno de Red.	120
8.6.- Seguridad en Access.	122
Conclusiones.	124
Bibliografía.	127
Glosario de Términos	131



Apendice A. Diagramas de Flujo de Datos.	148
Apendice B. Diccionario de Datos.	181
Apendice C. Diagrama de Transición de Estados.	195



A MIS PADRES

Dedico este trabajo a ustedes, a quienes por sus consejos y enseñanzas debo mucho de lo que actualmente soy; en especial a ti mamá, que me has brindado tu apoyo incondicional en todo momento y que cualquier palabra de agradecimiento es pequeña; comparado con las grandes acciones que realizas día con día con ese amor incondicional que siempre has tenido. Espero algún día poder agradecer como tú realmente mereces la infinidad de acciones hechas por ti. Gracias.



INTRODUCCIÓN

El presente trabajo muestra cómo se elaboró el Sistema Integral de Oficinas Centrales del Instituto Federal Electoral, así como las bases teóricas que determinaron el desarrollo del Sistema Integral de Oficinas Centrales que controla los Recursos Financieros de las Subdirecciones de Presupuesto, Tesorería y Contabilidad.

Anteriormente la mayor parte de la información se procesaba con paquetería como Word, Excel, Lotus, Framework, etc. Cada subdirección tenía pequeños sistemas hechos en Dbase para el control de oficios, expedición de cheques, control de ingresos y egresos, independientemente uno de otro. En virtud de la gran información que día a día se genera, fue necesario evaluar la posibilidad de desarrollar un sistema de acuerdo a las necesidades del Instituto Federal Electoral, utilizando herramientas que estuvieran a la vanguardia en el desarrollo de aplicaciones de cómputo, y que permitiera que la recaudación de datos fuera una labor ágil, además de mejorar el flujo de información y comunicación entre las Subdirecciones.

Teniendo como base la problemática expuesta por las Subdirecciones de contabilidad, presupuesto y tesorería, se busco diseñar un sistema que permitiera integrar toda la información que se tenía en diversas aplicaciones en una sola base de datos, eliminando la redundancia en la información y mejorando el flujo y control de datos para mantener al día el registro presupuestal, contable y financiero, automatizando una parte importante de las Subdirecciones de tesorería, contabilidad y presupuesto.

El desarrollo del Sistema Integral de Oficinas Centrales para el Instituto Federal Electoral desempeña parte importante para la toma de decisiones en un menor tiempo sin perder credibilidad y confiabilidad.

La Tesis se divide en ocho capítulos en los que se explican los trabajos que se realizaron, así como la selección, diseño, desarrollo e instalación del Sistema Integral de Oficinas Centrales.

El capítulo uno describe brevemente la historia y estructura del Instituto Federal Electoral, así como las funciones de los diversos organismos del instituto, como una introducción al conocimiento de las actividades de la institución para la cual se realizo este trabajo.

El capítulo dos proporciona las bases teóricas de los Ciclos de vida del Software, como el ciclo de vida clásico, la construcción de prototipos, el modelo en espiral y las técnicas de cuarta generación (4GL), así como la selección de uno de ellos para realizar un óptimo desarrollo del Sistema Integral de Oficinas Centrales.



El capítulo tres aborda los fundamentos de las bases de datos, abarcando conceptos básicos de bases de datos, siguiendo por la explicación de la necesidad del uso de las bases de datos, usando un modelo relacional y técnicas de normalización para conservar la integridad de la información.

El capítulo cuatro explica como realizar un análisis de un sistema, determinando los requerimientos de información, que abarcan técnicas para la recopilación de datos como la entrevista, el análisis en grupos, el cuestionario y la observación. Posteriormente se explica como hacer el análisis de las necesidades del sistema incluyendo la explicación para la formulación de los reportes de terminación del análisis y estudios de relación costo-beneficio. Se añade en este capítulo un breve apartado de conceptos de redes de área local dado su importancia en las organizaciones actuales; su inclusión en el análisis, es dada la importancia que tendría un análisis sin incluir una red local cuando en realidad existe. En el caso de Instituto Federal Electoral, las Subdirecciones para las cuales se desarrollo el sistema cuentan con una red de área local.

En el capítulo cinco se describe la teoría para poder hacer el diseño y la documentación inicial de un sistema. Se explican técnicas estructuradas tales como el diseño ascendente, el diseño descendente, el desarrollo modular, el diagrama estructural, el método HIPO, los diagramas de flujo, los diagramas Nassi-Schneiderman, diagramas Warnier-Orr, pseudocódigo, manuales de procedimientos y el método Folklore. Se explican aspectos del diseño de la interfaz con el usuario, en la cual debemos de considerar los factores humanos, algunos estilos de interacción hombre y máquina y el diseño de la interfaz hombre máquina. Se mencionan aspectos de evaluación del diseño de interfaces y los estándares para el diseño de la interfaz. Finalmente se trata el tópico del diseño de la ayuda en línea.

En el capítulo seis se inicia con el desarrollo del Sistema Integral de Oficinas Centrales, en la primera parte del capítulo se muestra el calendario de actividades y la justificación para el desarrollo del sistema, se da una alternativa de solución al problema, se determinan los usuarios involucrados con el desarrollo del sistema, se menciona la técnica utilizada para la recopilación de datos esquematizando la información a través de diagramas de flujo, se hace la selección del producto, se determinan los requerimientos mínimos para el funcionamiento del producto seleccionado y finalmente se hace el diseño de la base de datos y las interfaces de entrada y salida de datos.

El capítulo siete describe como se realizaron las pruebas de arranque, la puesta en marcha del sistema, la capacitación de los usuarios y el plan de mantenimiento.

El capítulo ocho define la responsabilidad de cada usuario, la seguridad que nos ofrece un entorno de red, la definición de grupos y categorización de los usuarios, las claves de acceso al sistema y los permisos para cada módulo.

Finalmente se exponen las conclusiones del presente trabajo.



OBJETIVOS

Objetivo General

Desarrollar un Sistema que permita controlar el Flujo del Gasto Presupuestal para la Dirección de Recursos Financieros, dentro del Instituto Federal Electoral utilizando un modelo, bajo un entorno de base de datos.

Objetivos Específicos

- 1.- Conocer a detalle el Control del Registro Presupuestal Financiero y Contable en el Instituto Federal Electoral.
- 2.- Evaluación de la infraestructura del equipo de computo en el Instituto Federal Electoral.
- 3.- Recopilación de información utilizando técnicas específicas para la obtención de datos.



CAPÍTULO 1

ANTECEDENTES DEL I.F.E.

1.1.- Antecedentes y fundamentos legales del Instituto Federal Electoral

Para conocer la finalidad y objetivos de la institución con la cual se trabajó, se describirán sus antecedentes y fundamentos legales. A continuación se hace una breve reseña de cómo se manejaron los procesos electorales desde el inicio del presente siglo en México:

El 19 de diciembre de 1911, se creó una ley electoral, la cual se reformó en 1912 y aportó características tales como: el otorgamiento de personalidad jurídica a los Partidos Políticos, la organización de colegios municipales encargados de organizar la elección, la división del territorio en distritos electorales renovables bianualmente, la entrega de la boleta electoral al votante en la casilla, entre otros. En la reforma de esta ley, se establece la elección directa para los miembros del Congreso de la Unión, considerando este procedimiento electoral por primera vez desde la consumación de la Independencia.

El 2 de julio de 1918, surge la Ley para las Elecciones de los Poderes Federales que aporta nuevos conceptos como la garantía del secreto del voto, el carácter permanente del padrón electoral, así como la creación de Consejos Distritales y Municipales. Posteriormente se reformó ésta ley, con nuevos procedimientos que insistían en que los partidos tenían que elaborar sus propias boletas y entregárselas a los presidentes municipales, además de que las credenciales estaban dispuestas para ser parte de los expedientes electorales y darle constancia al elector de haber votado; se hicieron nuevas reformas a esta ley el 24 de diciembre de 1921, el 24 de noviembre de 1931, el 19 de enero de 1942 y el 4 de enero de 1943 las cuales se dirigieron fundamentalmente a fortalecer el padrón de electores, debido al incremento tan notable de la población en ese periodo.

Posteriormente, con la Ley Electoral Federal del 7 de enero de 1946 la preparación, vigilancia y desarrollo del proceso electoral quedan bajo la responsabilidad del



Gobierno Federal, para lo que se crea la Comisión Federal de Vigilancia Electoral, las Comisiones Locales Electorales, los Comités Distritales Electorales y el Consejo del Padrón Electoral. El 21 de febrero de 1949 ésta ley se reformó y entre los agregados principales estuvieron: la obligación de los partidos para tener un comité directivo en cada Entidad Federativa donde cuenten con más de mil afiliados y la prohibición de que un miembro de las Comisiones Federal Electoral y Locales, así como de los Comités Distritales sean candidatos a cargos de elección popular, a menos que se separen de sus puestos seis meses antes de la elección.

Otra ley electoral se expide el 4 de diciembre de 1951, y contempla el aumento de representantes de partido en la Comisión Federal Electoral, pero elimina los representantes partidarios en las comisiones locales y distritales, además de la creación del Registro Nacional de Electores. En 1954 se reforma para otorgar el voto a la mujer y, nuevamente, el 20 de junio de 1963 para incluir a los diputados de partido como complemento del sistema de mayoría y para establecer la credencial permanente de elector.

El derecho al voto a los 18 años entra en vigor con la reforma a la Constitución hecha el 26 de diciembre de 1969, posteriormente se fijan en 21 y 30 años las edades mínimas para ser diputado y senador respectivamente, y aumenta a 250,000 el número de habitantes por Distrito Electoral Federal. Esta reforma elevó igualmente de 20 a 25 el número de diputados de partido, según la votación obtenida por los partidos políticos. También se redujo el requisito de acreditar el 2.5% de la votación al 1.5% de la misma para la asignación de diputados de partido.

El 30 de diciembre de 1977 se publica en el Diario Oficial la llamada reforma política, mediante la promulgación de la Ley Federal de Organizaciones Políticas y Procesos Electorales (LOPPE), que introduce algunas novedades como: el registro de los partidos políticos condicionado al resultado en las elecciones, el establecimiento de prerrogativas para éstos, el reconocimiento a las asociaciones políticas, la ampliación de la representación nacional plural mediante el sistema mixto de representación en la Cámara de Diputados, es decir, diputados electos por el principio de mayoría relativa y diputados electos por el principio de representación proporcional.

En 1987 se derogó la LOPPE con la promulgación del Código Federal Electoral que atribuyó al Gobierno Federal la responsabilidad de la organización de los comicios federales, al tiempo que asignó a ciudadanos y partidos la facultad de participar en los organismos electorales. También se creó el tribunal de lo Contencioso Electoral como órgano jurisdiccional en materia electoral.

Ya en nuestro decenio, el 6 de abril de 1990 se publica en el Diario Oficial de la Federación las modificaciones constitucionales aprobadas por el Congreso de la Unión y por la mayoría de los congresos, esta reforma trajo consigo la obligación en el desempeño de los cargos de elección popular, la claridad de la asociación libre y pacífica para participar en política, la creación del Registro Nacional de Ciudadanos, la definición del proceso electoral como función estatal que se ejerce por los poderes Legislativo y Ejecutivo de la Unión, con la participación de los partidos políticos



nacionales y de los ciudadanos. Esta función se realizará a través de un organismo público dotado de personalidad jurídica y patrimonio propios, que es precisamente el Instituto Federal Electoral.

1.1.1.- Creación del Instituto Federal Electoral

Aún cuando desde la Ley Federal Electoral de 1946 se establece la responsabilidad igualitaria entre Estado y Ciudadanos para la vigilancia y desarrollo del proceso electoral, haciéndose extensiva a los partidos políticos en la Ley de 1951; es hasta la reforma electoral a partir de la consulta pública cuando se hace la promulgación del Código Federal de Instituciones y Procedimientos Electorales, el 14 de agosto de 1990, cuando se crea el INSTITUTO FEDERAL ELECTORAL (I.F.E.), como el organismo público responsable del "Proceso Electoral Federal".

La constitución política señala en su artículo 41 lo siguiente:

"La organización de las elecciones federales es una función estatal que se ejerce por los poderes Legislativo y Ejecutivo de la Unión, con la participación de los partidos políticos nacionales y de los ciudadanos según lo disponga la ley. Esta función se realizará a través de un organismo público dotado de personalidad jurídica y patrimonio propios. La certeza, legalidad, imparcialidad, objetividad y profesionalismo serán principios rectores en el ejercicio de esta función estatal."

"El organismo público será autoridad en la materia, profesional en su desempeño y autónomo en sus decisiones; contará en su estructura con órganos de dirección, así como con órganos ejecutivos y técnicos. De igual manera, contará con órganos de vigilancia que se integrarán mayoritariamente por representantes de los partidos políticos nacionales. El órgano superior de dirección se integrará por consejeros y consejeros magistrados designados por los poderes Legislativo y Ejecutivo y por representantes nombrados por los partidos políticos. Los órganos ejecutivos y técnicos dispondrán del personal calificado necesario para prestar el servicio electoral profesional, los ciudadanos formarán las mesas directivas de casillas."

"El organismo público agrupará para su desempeño, en forma integral y directa, además de las que le determine la ley, las actividades relativas al padrón electoral, preparación de la jornada electoral, cómputo y otorgamiento de constancias, capacitación electoral y educación cívica e impresión de materiales electorales. Asimismo, atenderá lo relativo a los derechos y prerrogativas de los partidos políticos. Las sesiones de todos los órganos colegiados electorales serán públicas en los términos que disponga la ley."



El Instituto Federal Electoral tiene entonces los siguientes fines:

- 1).- Contribuir al desarrollo de la vida democrática.
- 2).- Preservar el fortalecimiento del régimen de partidos políticos.
- 3).- Integrar el Registro Federal de Electores.
- 4).- Asegurar a los ciudadanos el ejercicio de los derechos político - electorales y vigilar el cumplimiento de sus obligaciones.
- 5).- Garantizar la celebración periódica y pacífica de las elecciones para renovar a los integrantes de los poderes Legislativo y Ejecutivo de la Unión.
- 6).- Velar por la autenticidad y efectividad del sufragio.
- 7).- Coadyuvar en la promoción y difusión de la cultura política.

1.2.- Estructura orgánica y funciones del Instituto Federal Electoral

El Instituto Federal Electoral realiza las funciones que la ley le establece a través de órganos centrales, dentro de una estructura orgánica por medio de la cual cumple con los fines que le han sido asignados. Para cumplir estos fines el Instituto Federal Electoral trata de hacer valer una política de legalidad haciendo valer los principios establecidos en la constitución, imparcialidad teniendo equidad en la acción, objetividad para que apegando a los hechos pueda tener el conocimiento real de cualquier acontecimiento, certeza para tener veracidad en las acciones y profesionalismo.

Como en toda organización, para que las políticas y funciones se cumplan de manera efectiva, se crea una estructura organizacional que coordina a cada función y a cada acción dentro de los diferentes niveles de la organización. El Instituto Federal Electoral tiene una estructura de tipo jerárquica en el nivel vertical, esto quiere decir que a medida en que se asciende por la escala vertical aumenta la jerarquía y la autoridad, y departamental en el nivel horizontal, esto implica igual jerarquía y autoridad con diferentes funciones. En la figura 1.1 se encuentra el esquema organizacional del Instituto Federal Electoral.



1.2.1.- Organos Centrales

1.- Consejo General

Es el órgano superior de dirección de todo el Instituto, que da las resoluciones y lineamientos que este habrá de seguir. Está integrado por:

- Un consejero del Ejecutivo (el Secretario de Gobernación) que lo preside.

DIAGRAMA DE ORGANIZACIÓN.

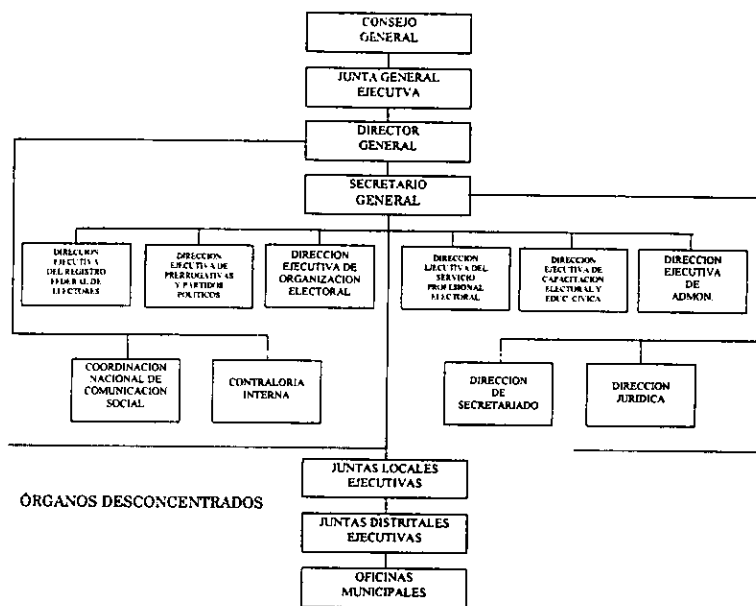


Figura. 1.1 Estructura del Instituto Federal Electoral²

- Cuatro consejeros del poder Legislativo (un diputado y un senador por la mayoría y un diputado y un senador por la primera minoría de cada cámara).
- Seis consejeros magistrados (designados por mayoría calificada en la Cámara de Diputados de un listado propuesto por el Ejecutivo Federal).

² Ricardo O. Zurita - Norma González Ehrlich
1993 *Instituto Federal Electoral. Curso de Inducción*
Impreso en México por I.F.E., México. Pág. 15



- Representantes de los partidos políticos nacionales (de acuerdo a la votación obtenida en la última elección uno por cada diez por ciento de la votación hasta un máximo de cuatro por partido).
- Así como un representante por cada partido político con registro condicionado, con voz pero sin voto.
- Y un secretario con voz pero sin voto (el Secretario General del propio Instituto).

El Director General del Instituto asiste a las sesiones del consejo General con voz, pero sin voto. Este consejo tiene las siguientes funciones:

- Resolver el otorgamiento del registro, así como sobre la pérdida del mismo por los partidos políticos en los casos previstos por el Código.
- Resolver sobre los convenios de fusión frente y coalición que celebren los partidos políticos nacionales.
- Vigilar que las actividades de los partidos políticos nacionales se desarrollen con apego al Código y cumplan con las obligaciones a que están sujetos.
- Vigilar que en lo relativo a las prerrogativas de los partidos políticos nacionales se actúe con apego al Código.
- Registrar la plataforma electoral que para cada proceso electoral deben presentar los partidos políticos.
- Registrar las candidaturas a presidente de los Estados Unidos Mexicanos.
- Registrar las listas regionales de candidatos a diputados de representación proporcional.
- Registrar as fórmulas de candidatos a diputados por el principio de mayoría relativa y de senadores.
- Efectuar el cómputo total de la elección de todas las listas de diputados electos según el principio de representación proporcional y determinar la asignación de diputados para cada partido político nacional y otorgar las constancias correspondientes, dando cuenta al colegio electoral de la Cámara de Diputados.
- Informar a los Colegios Electorales de las cámaras sobre el desarrollo de los trabajos realizados por el Instituto Federal Electoral en las elecciones que habrán de calificar.
- Conocer los informes trimestrales y anual que la Junta General Ejecutiva rinda por conducto del Director General del Instituto.
- Requerir a la Junta General Ejecutiva investigue, por los medios a su alcance, hechos que afecten de modo relevante los derechos de los partidos políticos o el Proceso Electoral Federal.
- Resolver los recursos de revisión que le competan.
- Aprobar anualmente el anteproyecto de presupuesto del Instituto que le proponga el Presidente del propio Consejo.
- Dictar los acuerdos necesarios para hacer efectivas sus atribuciones.

2.- Junta General Ejecutiva

Es el Organismo Ejecutivo de mayor jerarquía del Instituto, encargado de llevar a cabo las resoluciones dictadas por el Consejo General, a la vez que fija las políticas generales,



los programas y los procedimientos que rigen a las instancias del mismo, se integra con:

- El Director General, quien la preside
- El Secretario General
- Los Directores Ejecutivos del Registro Federal de Electores, de Prerrogativas y Partidos Políticos, de Organización Electoral, del Servicio Profesional Electoral, de Capacitación Electoral y Educación Cívica de Administración.

3.- Director General

Es quien preside y coordina la Junta General Ejecutiva, conduce la administración y supervisa el desarrollo adecuado de las actividades de los Organos Ejecutivos y Técnicos del Instituto. El Director General es designado por el Consejo General a propuesta de su Presidente y con el voto calificado de las dos terceras partes de sus miembros. Dura en el cargo ocho años.

4.- Secretario General

Cuando el Director General se ausenta de su puesto, este lo suple, además tiene los cargos de Secretario del Consejo General y de la Junta General Ejecutiva, recibe los informes de los órganos desconcentrados del Instituto, expide las certificaciones que requieran y sustancia los recursos que deben ser resueltos por la Junta durante los dos años anteriores al proceso electoral. El Secretario General es designado por el Consejo General a propuesta del Presidente del Consejo. Dura en su cargo ocho años.

5.- Direcciones Ejecutivas

Como su denominación lo expresa, las Direcciones Ejecutivas son los órganos centrales de ejecución parcial de las atribuciones del Instituto de acuerdo al área de su competencia; estos cargos los ocupan funcionarios nombrados por el Director General, preferentemente de entre los miembros del Servicio Profesional Electoral cuando exista igualdad de circunstancias de los candidatos. Estas son:

- 1.- *Dirección Ejecutiva del Registro Federal de Electores:* Que se encarga de integrar el Catálogo General de Electores, formar, revisar y actualizar el Padrón Electoral, expedir la Credencial para Votar con Fotografía, proporcionar a los órganos del Instituto y a los partidos políticos nacionales las listas nominales de electores, proponer la división territorial nacional en los trescientos distritos y cinco circunscripciones, mantener actualizada y clasificada la cartografía electoral del país, asegurar la integración y el funcionamiento de las comisiones de vigilancia, registrando la asistencia de los partidos políticos, y solicitándoles los estudios y las consultas de su competencia, por último se encarga de acordar con el Director General, los asuntos que le competen.
- 2.- *Dirección Ejecutiva de Prerrogativas y Partidos Políticos:* La cual se encarga de conocer las notificaciones que formulen las organizaciones que pretenden constituirse en partidos políticos; recibir las solicitudes de las organizaciones que



cumplen con los requisitos de ley e integrar el expediente respectivo; inscribir en el libro correspondiente el registro de los partidos y los convenios de fusión, frentes y coaliciones; ministrar a los partidos políticos nacionales el financiamiento público legal. Tramitar las franquicias postales y telegráficas a las que tienen derecho los partidos políticos nacionales; llevar a cabo el libro de registro de los representantes de los partidos políticos nacionales ante los diversos órganos del Instituto a nivel nacional, local y distrital, así como de sus candidatos a puestos de elección popular; acordar con el Director General los asuntos de su competencia. Cabe mencionar que el titular de esta dirección preside la Comisión de Radiodifusión.

- 3.- *Dirección Ejecutiva de Organización Electoral:* Le corresponde apoyar la integración, instalación y funcionamiento de las juntas ejecutivas del Instituto; elaborar los formatos de la documentación electoral y proveer lo necesario para su impresión y distribución; recabar las copias de la documentación del proceso electoral, recabar la documentación necesaria e integrar los expedientes para que se efectúe el cómputo electoral; llevar la estadística electoral; y acordar con el Director General los asuntos de su competencia.
- 4.- *Dirección Ejecutiva del Servicio Profesional Electoral:* Es responsable de formular el anteproyecto de estatuto del servicio profesional electoral; de cumplir y hacer cumplir las normas y procedimientos del servicio profesional electoral; llevar a cabo los programas de reclutamiento, selección, formación y desarrollo de personal profesional; y acordar con el Director General los asuntos de su competencia.
- 5.- *Dirección Ejecutiva de Capacitación Electoral y Educación Cívica:* Se hace responsable de elaborar y proponer los programas de educación cívica y capacitación electoral que desarrollarán las juntas, a la vez que vigila y coordina su cumplimiento; prepara el material didáctico y los instructivos electorales; exhorta a los ciudadanos a cumplir con las obligaciones del Código, en particular su inscripción al Registro Federal de Electores; acordar con el Director General los asuntos de su competencia.
- 6.- *Dirección Ejecutiva de Administración:* Está integrada por 3 Subdirecciones del área: *Subdirección de Recursos Humanos, Subdirección de Recursos Materiales y de Servicio, y Subdirección de Recursos Financieros;* dentro de ésta última área, finalmente se concentra el registro de todas las operaciones monetarias que realiza el I.F.E.; como es el manejo Contable para la elaboración y control del Presupuesto, así como el control de sus Recursos Financieros. A quien corresponde, entre otras funciones (Artículo 97 del COFIPE):
 - A.- Aplicar las políticas, normas y procedimientos para la administración de los Recursos Financieros y Materiales del Instituto.
 - B.- Organizar, dirigir y controlar la administración de los Recursos Materiales y Financieros, así como la prestación de los servicios generales en el Instituto.
 - C.- Formular el Anteproyecto Anual del Presupuesto del Instituto.
 - D.- Establecer y operar los sistemas administrativos para el ejercicio y control presupuestales.



- E.- Elaborar el Manual de Organización y el Catálogo de Cargos y Puestos del Instituto.
- F.- Atender las necesidades administrativas de los órganos del Instituto y acordar con el Director General los asuntos de su competencia.

Adicionalmente, por acuerdo del Consejo General, se crearon cuatro unidades centrales con nivel de dirección ejecutiva, dos dependen del Director General y dos del Secretario General, y son las siguientes:

- 7.- *Coordinación Nacional de Comunicación Social*: Coordina las acciones en materia de comunicación social del Instituto (interna y externa), edita las revistas Foro Electoral y la Gaceta Electoral. Acuerda con el Director General los asuntos de su competencia.
- 8.- *Contraloría Interna*: Vigila el cumplimiento de la función pública del Instituto con apego a los lineamientos establecidos. Acuerda con el Director General los asuntos de su competencia.
- 9.- *Dirección Jurídica*: Maneja los asuntos jurídicos en torno al que hacer y funciones del Instituto. Acuerda con el Secretario General los asuntos de su competencia.
- 10.- *Dirección del Secretariado*: apoya los procesos documentales y de archivo. Acuerda con el Secretario General los asuntos de su competencia.

1.2.2.- Organos desconcentrados permanentes

- 1.- *Juntas Locales Ejecutivas*: Son órganos permanentes de ejecución de las acciones del Instituto en cada Entidad Federativa. Se integran por un Vocal Ejecutivo, quien la preside; un Vocal Secretario, un Vocal del Registro Federal de Electores; un Vocal de Organización Electoral y un Vocal de Capacitación Electoral y Educación Cívica.
- 2.- *Juntas Distritales Ejecutivas*: Son órganos permanentes de ejecución de las acciones del Instituto en cada Distrito Electoral Federal; se integran por un Vocal Ejecutivo, quien la preside; un Vocal Secretario, un Vocal del Registro Federal de Electores, un Vocal de Organización Electoral y un Vocal de Capacitación Electoral y Educación Cívica.
- 3.- *Oficinas Municipales*: El Instituto contará con oficinas municipales en aquellos lugares donde la Junta General Ejecutiva determine su instalación. De conformidad con el acuerdo de la propia Junta del 10 de enero de 1992, se establece que las oficinas municipales que funcionaban bajo la dependencia de la Dirección Ejecutiva del Registro Federal de Electores se conviertan en oficinas municipales del Instituto Federal Electoral, reascribiéndose en las Juntas Distritales Ejecutivas que correspondan. Se trató de 1,121 oficinas municipales.



1.2.3.- Organos desconcentrados temporales

- 1.- *Consejos Locales:* Son órganos de vigilancia de las disposiciones electorales y los acuerdos y resoluciones de los órganos electorales superiores en las Entidades Federativas. Se integran por los miembros de las Juntas Locales Ejecutivas respectivas, por seis Consejeros Ciudadanos nombrados por el Consejo General a propuesta del Director General, y por representantes de los Partidos Políticos en la Entidad Federativa, en igual proporción que en el Consejo General. Los Consejos Locales funcionan exclusivamente durante el proceso electoral federal.
- 2.- *Consejos Distritales:* Son órganos de vigilancia de las disposiciones electorales y los acuerdos y resoluciones de los órganos electorales superiores en los Distritos Electorales Federales. Se integran por los miembros de las Juntas Distritales Ejecutivas respectivas, por seis Consejeros Ciudadanos nombrados por el Consejo Local a propuesta de la Junta Local Ejecutiva, y por representantes de los Partidos Políticos en el distrito, en igual proporción que en el Consejo Local. Los Consejos Distritales sólo funcionan durante el proceso electoral federal.
- 3.- *Mesas Directivas de Casilla:* Las mesas directivas de casilla son, los Organos Electorales formados por ciudadanos facultados para recibir la votación y hacer el escrutinio y el cómputo en cada una de las secciones electorales en las que se dividen los 300 distritos uninominales. Las mesas directivas de casilla, autoridad electoral, tienen a su cargo, durante la jornada electoral, respetar y hacer respetar la libre emisión y efectividad del sufragio, garantizar el secreto del voto y asegurar la autenticidad del escrutinio y cómputo. Cada mesa se integra por un presidente, un secretario y dos escrutadores. Para cada cargo hay propietarios y suplentes, es decir, ocho ciudadanos. Su designación corresponde a las Juntas Distritales, haciéndolo de entre los ciudadanos que han sido insaculados de las listas nominales de electores.

A su vez, los Consejos Distritales notifican y toman la protesta de ley a los integrantes de las mesas directivas de casilla.

Después de conocer brevemente la historia del Instituto Federal Electoral, evaluamos dependiendo de sus características funcionales, cual es el modelo a seguir para el desarrollo del Sistema Integral de Oficinas Centrales.



CAPÍTULO 2

CICLOS DE VIDA DEL SOFTWARE

2.1.- Importancia de los Ciclos de Vida del Software

La ingeniería del Software abarca tres elementos claves que son: -métodos, herramientas y procedimientos- los cuales facilitan al analista controlar el proceso del desarrollo del software y suministrar a los que practiquen dicha ingeniería las bases para construir software de alta calidad de una forma productiva.

Los métodos de la ingeniería del software suministran el “cómo” construir técnicamente el software, estos abarcan un amplio espectro de tareas que incluyen: planificación y estimación de proyectos; análisis de los requerimientos del sistema y del software; diseño de estructuras de datos, arquitectura de programas y procedimientos algorítmicos; codificación; prueba y mantenimiento. Los métodos de la ingeniería del software introducen frecuentemente una notación especial orientada a lenguaje o gráfica y un conjunto de criterios para la calidad del software.

Las herramientas de la ingeniería del software suministran soporte automático para los métodos. Hoy, existen herramientas para soportar cada uno de los métodos. Cuando se integran las herramientas de forma que la información creada por una herramienta pueda ser utilizada por otra, se establece un sistema para el soporte del desarrollo del software, llamado ingeniería del software asistido por computadora (CASE - computer-aided software engineering-). CASE combina el software, hardware y bases de datos de la ingeniería del software para crear un entorno de ingeniería del software análogo al diseño/ingeniería asistida por computadora, CAD/CAE (del inglés: computer-aided design/engineering), para el hardware.

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



La ingeniería del software está compuesta de etapas que abarcan métodos, herramientas y procedimientos. Estas etapas se denominan frecuentemente modelos de la ingeniería del software. Un modelo para la ingeniería del software se elige basándose en la naturaleza del proyecto y de la aplicación, los métodos y herramientas a usar y los controles y entregas requeridos. Los ingenieros de software han tratado ampliamente y debatido tres modelos que se describen a continuación.

2.2.- Ciclo de Vida Clásico

La figura 2.1 ilustra el modelo del ciclo de vida clásico para la ingeniería del software. Algunas veces llamado el "modelo en cascada", el modelo del ciclo de vida exige un enfoque sistemático, secuencial, del desarrollo del software que comienza en el nivel del sistema y progresa a través del ciclo convencional de una ingeniería, el modelo del ciclo de vida clásico abarca las siguientes actividades:

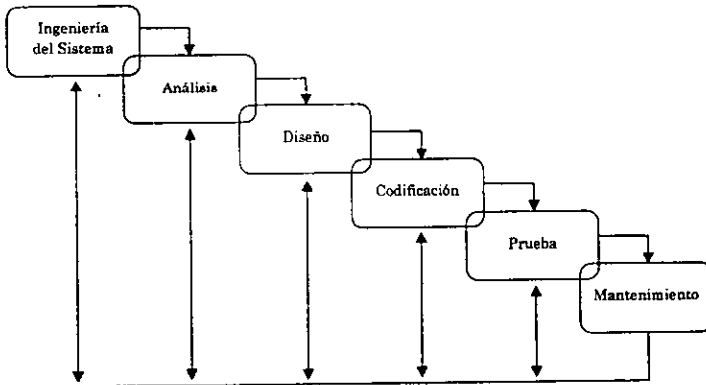


Figura. 2.1 Ciclo de Vida Clásico.¹

Ingeniería y análisis del sistema. Debido a que el software es siempre parte de un sistema mayor, el trabajo inicia estableciendo los requerimientos de todos los elementos del sistema para después asignar algún subconjunto de estos requerimientos al software. Esta visión del sistema es esencial cuando el software debe interrelacionarse con otros elementos tales como hardware, personas y bases de datos. La ingeniería y análisis del sistema abarca los requerimientos globales a nivel del sistema con una pequeña cantidad de análisis y diseño a nivel superior.

Análisis de los requerimientos del software. El proceso de recopilación de los requerimientos se concentra e intensifica especialmente en el software. Para comprender la naturaleza de los programas que se deben construir, el ingeniero de software

¹ Roger S. Pressman. 1995. *Ingeniería del Software*. McGraw-Hill. 3era. Edición. México. Pág. 26



("analista") debe conocer el software, así como la función, rendimiento e interfaces requeridas. Los requerimientos tanto del sistema como del software se documentan y revisan con el cliente.

Diseño. El proceso de diseño traduce los requerimientos en una representación del software que puede ser establecida de forma que obtenga la calidad requerida antes de que comience la codificación. Como los requerimientos, el diseño se documenta y forma parte de la configuración del software.

Codificación. El diseño debe traducirse en una forma legible para la máquina. El paso de la codificación ejecuta esta tarea. Si el diseño se ejecutara de una manera detallada, la codificación puede realizarse mecánicamente.

Prueba. Una vez que se ha generado el código, comienza la prueba del programa, que se enfoca sobre la lógica interna del software, asegurando que todas las instrucciones se han probado, y sobre las funciones externas, esto es, realizando pruebas para asegurar que la entrada definida producirá los resultados que realmente se requieren.

Mantenimiento. El software sufrirá indudablemente cambios después de que se entregue al cliente. Los cambios ocurrirán debido a que se han encontrado errores, debido a que el software debe adaptarse por cambios del entorno externo (por ejemplo, un cambio solicitado debido a que se tiene un nuevo sistema operativo o dispositivo periférico), o debido a que el cliente requiere aumentos funcionales o del rendimiento. El mantenimiento del software se aplica a cada uno de los pasos precedentes del ciclo de vida a un programa existente en vez de a uno nuevo.

El ciclo de vida clásico es el más viejo y más ampliamente utilizado modelo en la ingeniería del software. Sin embargo, con el paso de unos años, se han producido críticas al modelo, incluso por seguidores activos que cuestionan su aplicabilidad a todas las situaciones.

2.3.- Ciclos de Vida de Construcción de Prototipos

Normalmente un cliente definirá un conjunto de objetivos generales para el software, pero no identificará los requerimientos detallados de entrada, procesamiento o salida. En otros casos el programador puede no estar seguro de la eficiencia de un algoritmo, la adaptabilidad de un sistema operativo o la forma en que debe realizarse la interacción hombre-máquina. En estas y muchas otras situaciones, puede ser mejor método de ingeniería del software realizar un prototipo.

La construcción de un Prototipo es un proceso que facilita al programador la creación de un modelo del software a construir. El modelo tomará una de las tres formas siguientes: un prototipo en papel que describa la interacción hombre-máquina de forma que facilite al usuario la comprensión de cómo se producirá tal interacción; un prototipo que funcione que implementa algunos subconjuntos de la función requerida al software deseado; o un

programa existente que ejecute parte o toda la función deseada y que tenga otras características que deban ser mejoradas por el grupo de desarrollo.

La figura 2.2 muestra la secuencia de sucesos del modelo de construcción de prototipos. Como en todos los métodos de desarrollo de software, la construcción de prototipos comienza con la recolección de los requerimientos. El técnico y el cliente se reúnen y definen los objetivos globales para el software, identifican todos los requerimientos conocidos y perfilan las áreas en donde será necesaria una mayor definición, en seguida se produce un "diseño rápido". El diseño rápido se enfoca sobre la representación de los aspectos del software, visibles al usuario (por ejemplo, métodos de entrada y formatos de salida). El diseño rápido conduce a la construcción de un prototipo. El prototipo es evaluado por el cliente/usuario y se utiliza para refinar los requerimientos del software a desarrollar. Se produce un proceso interactivo en el que el prototipo es "afinado" para satisfacer las necesidades del cliente. Idealmente, el prototipo sirve como un mecanismo para identificar los requerimientos del software. Si se construye un prototipo que funciona, el realizador intenta hacer uso de los fragmentos de programas existentes o aplica herramientas (por ejemplo, generadores de informes, gestores de ventana, etc.) que faciliten la rápida generación de programas que funcionen.

En la mayoría de los proyectos, el primer sistema construido apenas es utilizable. Puede ser demasiado lento, demasiado grande, difícil de usar o las tres cosas. No hay más alternativa que comenzar de nuevo y construir una versión rediseñada que resuelva los problemas que se presenten. Cuando se utiliza un nuevo concepto de sistemas o tecnología, hay que construir un sistema para desecharlo, por que incluso la mejor planificación no puede asegurar que vaya a ser bueno la primera vez.

Por lo tanto la pregunta no es si hay que construir un sistema piloto y tirarlo. La única pregunta es si planificar de antemano la construcción de algo que se va a desechar, o prometer entregar el desecho a los clientes. El prototipo puede servir como "el primer sistema"; pero esto puede ser una visión idealizada.

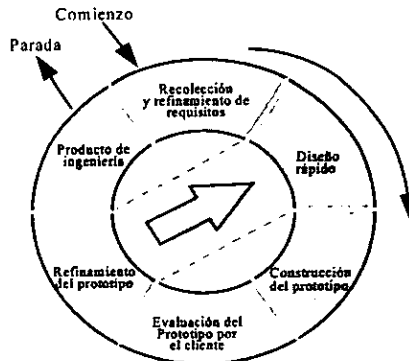


Figura. 2.2 Ciclo de Vida de Construcción de Prototipos.²

² Roger S. Pressman. 1995. *Ingeniería del Software*. McGraw-Hill. 3era. Edición. México. Pág. 29

2.4.- Ciclo de Vida del Modelo en Espiral

El modelo en espiral ha sido desarrollado para cubrir las mejores características tanto del ciclo de vida clásico, como de la creación de prototipos, añadiendo al mismo tiempo un nuevo elemento; el análisis de riesgo, que falta en esos paradigmas. El modelo, representado mediante la espiral de la figura 2.3, define cuatro actividades principales, representadas por los cuatro cuadrantes de la figura:

1. *Planificación*: Determinación de objetivos, alternativas y restricciones.
2. *Análisis de riesgo*: Análisis de alternativas e identificación/resolución de riesgo.
3. *Ingeniería*: Desarrollo del producto de "siguiente nivel".
4. *Evaluación de cliente*: Valoración de los resultados de la ingeniería.

Un aspecto integrante del modelo en espiral es evidente cuando consideramos la dimensión radial representada en la figura 2.3. Con cada ciclo alrededor de la espiral (comenzando en el centro y siguiendo hacia el exterior), se construyen sucesivas versiones del software, cada vez más completas. Durante la primera vuelta alrededor de la espiral se definen objetivos, las alternativas y las restricciones, se analizan e identifican riesgos. Si el análisis de riesgo indica que hay incertidumbre en los requisitos, se puede usar la creación de prototipos en el cuadrante de ingeniería para la asistencia tanto al encargado del desarrollo como al cliente.

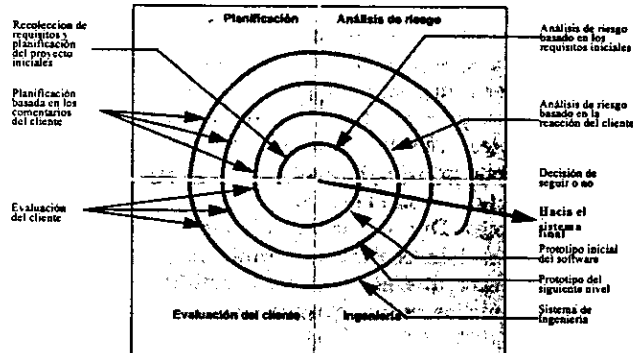


Figura. 2.3 Ciclo de Vida del Modelo en Espiral.³

El cliente evalúa el trabajo de ingeniería (cuadrante de evaluación del cliente) y sugiere modificaciones. En base a los comentarios del cliente se produce la siguiente fase de planificación y de análisis de riesgo. En cada bucle alrededor de la espiral, la culminación del análisis de riesgo resulta en una decisión de "seguir o no seguir". Si los riesgos son demasiado grandes, se puede dar por terminado el proyecto.

³ Roger S. Pressman. 1995. *Ingeniería del Software*. McGraw-Hill. 3era. Edición. México. Pág. 31



En la mayoría de los casos, se sigue avanzando alrededor del camino de la espiral y ese camino lleva a los desarrolladores hacia fuera, hacia un modelo más completo del sistema, y al final, al propio sistema operacional. Cada vuelta alrededor de la espiral requiere ingeniería (cuadrante inferior derecho), que se puede llevar a cabo mediante el enfoque del ciclo de vida clásico o de la creación de prototipos. Debe tenerse en cuenta que el número de actividades de desarrollo que ocurren en el cuadrante inferior derecho aumenta al alejarse del centro de la espiral.

Análisis de riesgo

Cuando se considera el riesgo en el contexto de desarrollo de un sistema, son evidentes tres consideraciones conceptuales:

El futuro, ¿cuáles son los riesgos que pueden hacer que fracase el proyecto de software? Los cambios, ¿cómo afectarán al éxito global y a los plazos, los cambios en los requisitos del cliente, en las tecnologías de desarrollo, en las computadoras destino y en todas las demás entidades relacionadas con el proyecto?. Por último, nos enfrentamos con las elecciones, ¿que métodos y herramientas debemos utilizar, cuánta gente se debe involucrar, así como si es necesario darle importancia a la calidad y de que manera?

El análisis del riesgo se compone de tres actividades diferentes:

- A. *Identificación del riesgo*. Consiste en enumerar los riesgos concretos de un proyecto, los técnicos y los de negocios. Uno de los mejores métodos para comprender cada uno de los riesgos es responder a una serie de preguntas de impacto que ayudan al planificador a entender el riesgo en términos técnicos y relativos al proyecto.
- B. *Proyección del riesgo*. También denominada *estimación del riesgo*, el planificador lo evalúa de dos formas, la probabilidad de que el riesgo sea real y las consecuencias de los problemas asociados con el riesgo.
- C. *Evaluación del riesgo*. Durante la evaluación se examina además la exactitud de las estimaciones que se han realizado durante la proyección del riesgo; se intenta dar prioridad a los riesgos que no se han cubierto y se inicia a pensar en las formas de controlar y/o prevenir los riesgos que tengan una mayor probabilidad de ocurrir.

2.5.- Técnicas de la Cuarta Generación (4GL)

La herramienta genera automáticamente el código fuente basándose en la especificación del técnico. Para el ingeniero del software existe cierto debate sobre cuánto puede aumentarse la rapidez en la construcción de un programa respecto a cuánto ha de elevarse el nivel en el que se especifique el software para una máquina. El modelo 4GL para la ingeniería del software se orienta hacia la habilidad de especificar software a un nivel que sea más próximo al lenguaje natural o en una notación que proporcione funciones significativas.



Actualmente un entorno para el desarrollo del software que soporte el modelo 4GL incluye algunas o todas de las siguientes herramientas: lenguajes no procedimentales para consulta a bases de datos, generación de informes, manipulación de datos, interacción y definición de pantallas y generación de código; estas herramientas existen, pero sólo para dominios de aplicación muy específicos. No existe hoy un entorno 4GL que pueda aplicarse a todas las categorías de aplicaciones del software.

En la figura 2.4 se describe el modelo 4GL para la ingeniería del software. Inicia con la determinación de requerimientos.

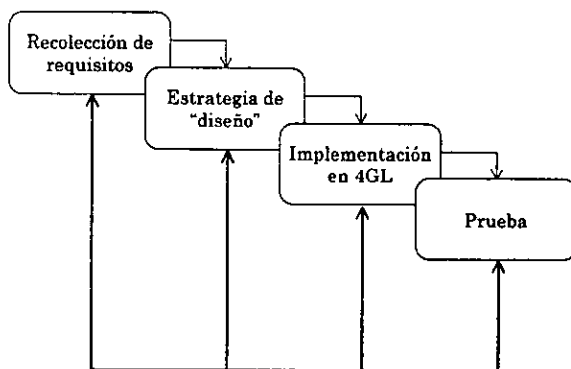


Figura. 2.4 Técnicas de Cuarta Generación (4GL).⁴

2.6.- Selección del Modelo de Ciclo de Vida

Habiendo descrito los diferentes Modelos de Ciclo de Vida del Software (al inicio de este capítulo), seleccionaremos el que se considera adecuado para este proyecto, antes de lo cual resumimos las ventajas y desventajas de cada modelo.

2.6.1.- Ventajas y Desventajas del Ciclo de Vida Clásico

Define claramente las etapas que se tienen que seguir en un orden estricto para realizar un sistema, proporciona una metodología clara, por medio de la etapa del Análisis, una vez concluida ésta, se pueden estimar los recursos que son necesarios para la realización del proyecto, así como el tiempo estimado para su terminación.

Entre los problemas que se presentan algunas veces, cuando se aplica el modelo del ciclo de vida clásico se encuentran:

⁴ Roger S. Pressman. 1995. *Ingeniería del Software*. McGraw-Hill. 3era. Edición. México. Pág. 33



1. Los proyectos reales raramente siguen el flujo secuencial que propone el modelo. Siempre ocurren interacciones y se crean problemas en la aplicación del modelo.
2. Normalmente es difícil para el cliente establecer explícitamente al principio todos los requerimientos. El ciclo de vida clásico requiere esto y tiene dificultades en acomodar posibles incertidumbres que pueden existir al comienzo de muchos proyectos.
3. El cliente debe tener paciencia, ya que una versión funcionando del programa no estará disponible hasta las etapas finales del desarrollo del proyecto. Un error importante no detectado hasta que el programa está funcionando puede ser desastroso.

Es importante mencionar que este modelo proporciona la ventaja de que la documentación del sistema se va obteniendo conforme se completa cada una de las etapas. Además proporciona libertad al analista en cuanto a la selección de la metodología a usar en cada una de las etapas del ciclo, ya que este modelo define cada etapa como un conjunto de entradas y salidas.

2.6.2.- Ventajas y Desventajas del Ciclo de Vida de Prototipos

Es conveniente usar este modelo cuando no se conocen con exactitud los requerimientos del sistema, ya que al presentar un prototipo al usuario cada vez que el analista se entrevistó con él, se obtiene una retroalimentación que permite que el prototipo se acerque cada vez más a lo que se quiere en realidad.

La construcción de prototipos como modelo de ciclo de vida, puede ser problemático por las siguientes razones:

1. El primer prototipo que se muestra al cliente no es funcional al cien por ciento ya que en esto consiste dicho ciclo de vida. Por lo que se tendrán que hacer varios sistemas prototipo hasta que el cliente quede total y completamente satisfecho. En ocasiones el cliente no está consciente de ello.
2. El técnico de desarrollo realiza frecuentemente ciertos compromisos de implementación en orden a obtener un prototipo que funcione rápidamente. Puede utilizarse un sistema operativo o lenguaje de programación inapropiado simplemente porque está disponible y es conocido; un algoritmo ineficiente puede implementarse de forma sencilla para demostrar su capacidad. Después de pasar algún tiempo en el que el técnico estaba familiarizado con estas elecciones, olvida las razones por las que eran inapropiadas. La elección menos ideal forma ahora parte integral del sistema.

Aunque se pueden presentar problemas, la construcción de prototipos es un modelo efectivo para la ingeniería del software. La clave está en definir al comienzo las reglas del juego; esto es, el cliente y técnico deben estar de acuerdo en que el prototipo se construya para servir sólo como un mecanismo de definición de los requerimientos. Posteriormente



ha de ser descartado (al menos en parte) y debe construirse el software real, con los ojos puestos en la calidad y mantenimiento.

En este caso, el proyecto a desarrollar no se adecúa a las características que este ciclo de vida propone para ser empleado.

2.6.3.- Ventajas y Desventajas del Ciclo de Vida en Espiral

El modelo en espiral utiliza un enfoque “evolutivo” (que procede a estar evolucionando), permitiendo al desarrollador y al cliente entender y reaccionar a los riesgos en cada nivel evolutivo. Utiliza la creación de prototipos como un mecanismo de reducción de riesgos, pero permite a quien lo desarrolla aplicar el enfoque de creación de prototipos en cualquier etapa de la evolución del producto. El modelo demanda una consideración directa de riesgos técnicos en todas las etapas del proyecto, si se aplica adecuadamente, debe reducir los riesgos antes que se conviertan en problemáticos.

Algunas desventajas que presenta este modelo son:

1. Puede ser difícil convencer a grandes clientes de que el enfoque evolutivo es controlable.
2. Requiere una considerable habilidad para la valoración del riesgo, y cuenta con ésta habilidad para el éxito.
3. Si no se descubre un riesgo importante, indudablemente surgirán problemas.
4. El modelo en sí mismo es relativamente nuevo y no se ha usado tanto como el ciclo de vida clásico o la creación de prototipos.

En este caso, no consideramos adecuado este Modelo para el sistema, ya que no se cuenta con la experiencia para la valoración del riesgo.

2.6.4.- Ventajas y Desventajas de las Técnicas de Cuarta Generación (4GL)

Idealmente, el cliente debe describir los requerimientos y estos deben traducirse directamente en un prototipo operacional; pero éste no funciona. El cliente puede no estar seguro de lo que necesita, puede ser ambiguo en la especificación de hechos que son conocidos y puede ser incapaz o no desear especificar la información en la forma que una herramienta 4GL puede consumirla.

No se usaron estas técnicas ya que el Instituto Federal Electoral no cuenta con el software necesario (p.e.: CASE, CAD, CAE), para el desarrollo de aplicaciones con técnicas 4GL, debido a factores económicos.



Después de haber evaluado las características de cada Modelo de Ciclo de Vida, y tomando en cuenta las características del Sistema Integral de Oficinas Centrales, se concluye que el modelo que más se apega para resolver el problema es el ciclo de vida clásico, ya que ofrece mayores ventajas sobre los otros.

Para comprender la construcción del modelo de base de datos, se revisará los Fundamentos de Bases de Datos.



CAPÍTULO 3

FUNDAMENTOS DE BASES DE DATOS

En la actualidad los datos son recursos administrables y que con su almacenamiento y empleo adecuado pueden proporcionar una importante ventaja competitiva. Por tanto, mientras mayor sea la cantidad de datos a que tiene acceso una computadora, aumentará el potencial de información que proporcione.

En todas las esferas de la vida como en la industria y el comercio, las bases de datos amplían en gran medida las posibilidades de acción. Algunos de los aspectos sobresalientes del desarrollo industrial de nuestros días, se deben al extraordinario crecimiento de la disponibilidad de información.

Sin embargo, el hecho de que en la actualidad se tenga una gran disponibilidad de información en el momento requerido, no sólo se debe a la innovación tecnológica, sino también al desarrollo de nuevos métodos para el almacenamiento de datos que diariamente se requieren para administrar negocios, bancos, universidades, organismos gubernamentales y otras actividades.

El medio ambiente de base de datos, con su software de apoyo y las técnicas de manejo que le acompañan, proporcionarán el marco para tratar los datos como un recurso estandarizado, administrable y compatible. Compartir archivos de datos entre distintas aplicaciones y emplear descripciones de datos y proposiciones de acceso consistentes en los programas en el medio ambiente de base de datos, ayuda a tener una visión estandarizada de todo el proceso de desarrollo de un sistema de información.

Para describir un método formal asociado con bases de datos; primero, es necesario aclarar algunos elementos relacionados con el objetivo principal de este enfoque: controlar y usar los datos eficazmente.

3.1.- Conceptos básicos

DATOS: Los datos son una representación del mundo real. Únicamente identificarán la porción que ejemplifique un hecho particular de interés para una o más aplicaciones.



INFORMACIÓN: Es el resultado obtenido de la transformación de los datos; para una persona dada, con un fin determinado, o para satisfacer un requerimiento específico. Este punto de vista es muy relativo, dado que, lo que son datos "procesados" para una persona, lo son en crudo para otra. Los mismos datos son objeto de diferentes clases de procesamiento en distintas circunstancias.

ENTIDAD: Una entidad es una persona, un lugar, una cosa, un evento o un concepto acerca del cual se desea registrar información.

ATRIBUTOS: Son las características básicas que identifican a la entidad (son los campos de datos correspondientes a una Tabla).

VALOR DE LOS DATOS: El valor de los datos es la información o los datos mismos contenidos en cada campo de éstos valores que toman los campos de datos, pueden ser cuantitativos, cualitativos o descriptivos, dependiendo de la manera como los campos de datos (atributos) describan a la entidad.

REGISTROS: Un registro es una colección de datos relacionados, referentes a una entidad.

ARCHIVOS DE DATOS: Los registros de datos forman un archivo de datos. Así, un archivo es un conjunto de registros similares, esto es, del mismo tipo.

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

La idea básica en la implantación de una base de datos es que los mismos datos sean aprovechados por tantas aplicaciones como sea posible, permitiendo no sólo la lectura de aquéllos almacenados, sino la continua modificación de los que son necesarios para el control de las aplicaciones. Las bases de datos pueden ser diseñadas para el procesamiento por lotes, en tiempo real, o en línea. En ocasiones, muchos datos se hallan simultáneamente guardados en varios archivos con distintas finalidades y también con diferentes fechas de actualización; en la base de datos se pretende eliminar esta redundancia; sin embargo en algunas bases de datos, se admite cierta redundancia (controlada o mínima) con el objeto de reducir los tiempos de acceso o simplificar los métodos de direccionamiento.

3.2.- Conceptos asociados a bases de datos

REDUNDANCIA: Se habla de redundancia cuando la información se repite en diferentes archivos de la misma base de datos y esto conduce a que existan datos duplicados. Por lo tanto, existe la necesidad de reducir el grado de redundancia con otras características deseables de la base, de modo que es preferible hablar de redundancia



controlada o redundancia mínima en lugar de no redundancia como criterio de diseño. La redundancia no controlada ocasiona varios inconvenientes. En primer lugar se tiene el costo adicional del almacenamiento de copias múltiples de los mismos datos. En segundo término, y esto es mucho más serio, el hecho de que para actualizar por lo menos una parte de las copias redundantes es preciso recurrir a múltiples operaciones de actualización.

En las grandes operaciones de procesamiento, que no cuentan con el apoyo de una base de datos, hay tantos datos redundantes que resulta prácticamente imposible mantenerlos todos en el mismo nivel de actualización.

INCONSISTENCIA: Este problema surge como resultado del anterior, significa que al actualizar los datos almacenados en diversos archivos de la misma base de datos de una manera duplicada existe el riesgo de no realizarlo en todos los archivos y en este momento se genera la información inconsistente.

INTEGRIDAD: Son las medidas de seguridad usadas para mantener correctos los datos en la base de datos y que permiten recuperarse de las contingencias sin daño para éstos. Además de protegerlos contra posibles problemas de software, deben incluirse también procedimientos de verificación que aseguren que los valores de los datos se ajustan a ciertas reglas preestablecidas. Hay diferentes maneras de asegurar la formas de asegurar la integridad de los datos:

VALIDACIÓN DE LOS DATOS.- El contenido de cada elemento de entrada debe coincidir con el tipo de dato descrito previamente.

VALIDACIÓN DEL VALOR DE DATOS.- El contenido de un campo de entrada puede confirmarse para cierto rango de valores.

SEGURIDAD: Para evitar que una persona no autorizada tenga acceso al sistema, es común el uso de claves de acceso. Existen varios aspectos de seguridad relacionados específicamente con los datos, como son impedir que vean los datos aquellas personas que no deban hacerlo y evitar que modifiquen información que no deba alterarse.

CONCURRENCIA: Se dice que hay concurrencia cuando dos elementos u ocurrencias intentan acceder al mismo tiempo a un recurso. Por ejemplo en un sistema de usuarios múltiples, cuando dos usuarios intentan actualizar en forma simultánea el mismo registro, interfieren entre sí, de manera que una de esas dos actualizaciones podrían quedarse esperando.

Una Base de Datos en general, es tanto integrada como compartida.

Por integrada, se entiende que una base de datos puede considerarse como una unificación de varios archivos de datos independientes, donde se eliminan parcial o totalmente cualquier redundancia entre los mismos, por compartida se entiende que las partes individuales de la base de datos pueden compartirse entre varios usuarios



distintos, en el sentido de que cada uno de ellos puede tener acceso a la misma base de datos, y utilizarla con propósitos diferentes. Tal comportamiento, es consecuencia del hecho de que la base de datos es integrada.

La palabra compartida a menudo se amplía para abarcar también al comportamiento concurrente, es decir, la oportunidad de que diversos usuarios accesen la base de datos, tal vez la misma parte de ella e incluso el mismo tiempo, denominándose sistema de usuarios múltiples.

3.3.- Necesidad de las bases de datos

Considerando el manejo de archivos para un grupo de aplicaciones, sin el uso de una base de datos, cada programa de aplicación es considerado como un caso separado y distinto; en consecuencia, no hay coordinación entre los programas de aplicaciones del mismo grupo. Por lo general dos programas de aplicación necesitan recurrir al mismo archivo y no solamente un archivo puede tener múltiples usuarios, sino que además cada usuario puede tener diferentes necesidades con respecto a dicho archivo.

Debido a estos problemas a los que se enfrentan los analistas y diseñadores de sistemas, surge la necesidad de hacer uso de métodos que faciliten el diseño de bases de datos, a fin de ayudar en la conceptualización física y lógica de los datos, el establecimiento de las relaciones entre las mismas bases de datos, así como a determinar sus atributos; optimizándose de esta manera los recursos con que se cuenta (espacio en disco) y evitando la duplicidad de datos (redundancia e inconsistencia).

3.3.1.- Ventajas del modelo de bases de datos

- Disminuye la redundancia.
- Evita la inconsistencia.
- Conserva la integridad de los datos.
- Permite compartir datos en diferentes aplicaciones.
- Facilita el desarrollo de aplicaciones.
- Uniformiza los controles de seguridad, privacidad e integridad.
- Proporciona independencia entre los datos y los programas.
- Reducción en el mantenimiento a programas.
- La información se encuentra centralizada.
- Usando una BD (base de datos) se pueden aplicar restricciones en el acceso a los datos por motivos de seguridad.
- El acceso puede ser en línea¹ o batch².
- Reduce el tiempo de desarrollo y mantenimiento de las aplicaciones.
- Favorece el cumplimiento de normas para representación de los datos.

¹ Acceso en Línea: Acceso a los datos almacenados en forma aleatoria.

² Acceso en Batch: Acceso a los datos almacenados en forma secuencial.



3.4.- Sistema manejador de bases de datos (DBMS)

Los sistemas de información, son empleados por las personas, quienes por tal motivo deben proporcionar datos organizados y métodos de acceso que faciliten la interacción con los usuarios. Los datos almacenados en una base de datos tienen una organización física, la cual es dependiente del medio de almacenamiento empleado (discos o cintas).

Un DBMS es un conjunto de rutinas, funciones, métodos de acceso, áreas de trabajo, almacenamiento y control, requeridas para el tratamiento del manejo de información bajo el concepto de bases de datos.

Un DBMS extrae datos de los programas y permite diferentes vistas lógicas de los mismos datos, los cuales son independientes de la forma en que se distribuya físicamente la información en disco.

Los Sistemas Manejadores de Bases de Datos (DBMS) han evolucionado hasta el punto de conseguir la aceptación general de los usuarios, y aún así, un problema importante que se presenta al usuario es el empleo efectivo de esos sistemas; sin embargo, un elemento vital para lograr una buena utilización y respuesta de un DBMS es el diseño de la base de datos, (en la figura. 3.1, se muestra la estructura flexible para un DBMS).

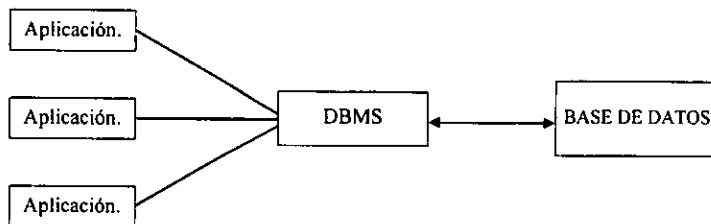


Figura. 3.1 Estructura flexible para un DBMS.³

3.4.1.- Objetivo de un DBMS

En términos generales, algunos de los objetivos principales del DBMS son: eliminar la redundancia, mantener la consistencia de los datos, resolver los problemas de concurrencia y regular el acceso a los datos.

³ Henry F. Korth - Abraham Silberschatz, 1993. *Fundamentos de Bases de Datos*. McGraw-Hill, 2da. Edición, México, Pág. 12



3.4.2.- Características de los DBMS

La administración de las bases de datos combina la necesidad que tiene el usuario por técnicas más complejas para la manipulación de los datos con la creciente tecnología de las computadoras. Un DBMS, hace más fácil la puesta al día y la modificación de las bases de datos existentes, reduce de modo substancial el espacio necesario para almacenarlos y procesarlos, su uso mejora significativamente el manejo de bases de datos en comparación con los métodos comunes de manipulación de archivos. Además:

- ♦ Proporciona independencia entre los programas de aplicación final de los usuarios y la estructura de almacenamiento física de los datos.
- ♦ Contiene programas de utilidad para facilitar la creación, mantenimiento y reestructuración de las bases de datos.
- ♦ Posee capacidad simple para la reorganización de los datos.
- ♦ Atiende de modo efectivo las diferentes necesidades de la organización.
- ♦ Suministra información consistente y a tiempo para la toma de decisiones.
- ♦ Genera habilidad para afectar la seguridad de los datos e imponer límites de acceso a ellos.
- ♦ Tienen capacidad de reinicio (restart) automático en caso de falla del sistema.
- ♦ Brindan la posibilidad de recuperar las operaciones en forma manual con esfuerzo mínimo.

3.5.- Modelos de los DBMS

A continuación se describirán 2 de los modelos existentes de un Sistema Manejador de Base de Datos (DBMS) en forma general y más a detalle el Modelo Relacional.

3.5.1.- Modelo de Red

En el modelo relacional, los datos y las relaciones entre ellos se representan mediante una colección de tablas. El modelo de red difiere del modelo relacional en que los datos se representan por medio de colección de *registros* y las relaciones entre los datos mediante ligas.

Una base de datos en red consta de una colección de registros, los cuales están conectados entre sí por medio de ligas. Cada registro es una colección de campos (atributos), cada uno de los cuales contiene solamente el valor de un dato. Una liga es una asociación entre dos registros exclusivamente. Así pues, una liga puede verse como una forma restringida (binaria).

El ejemplo de la base de datos de la figura 3.2., muestra que Rodríguez tiene la cuenta 101, que Suárez tiene las cuentas 405 y 231, y Olmedo tiene la cuenta 131.

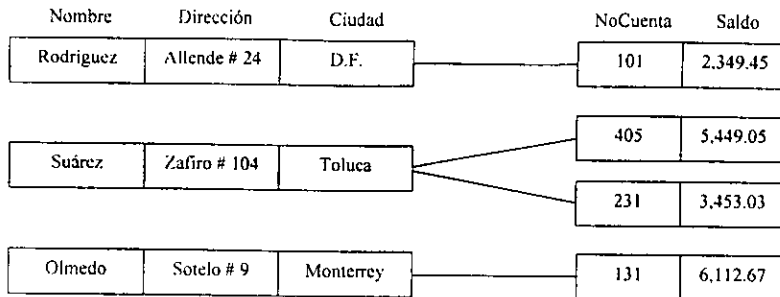


Figura. 3.2 Ejemplo de Diagrama de Estructura de Datos. (Modelo de Red).

3.5.2.- Modelo Jerárquico

En el modelo de red, los datos se representan mediante colecciones de registros y las relaciones entre los datos por medio de ligas. Esta representación también es válida para el modelo jerárquico. La única característica que los diferencia es que en el modelo jerárquico los registros se organizan para formar colecciones de árboles en vez de grafos arbitrarios.

Considere una base de datos que representa una relación *cuenta-cliente* en un sistema bancario. Existen dos tipos de registros, *cliente* y *cuenta*. Estos se definen de la misma manera que en el modelo de red. Consta de tres campos: *nombre*, *calle* y *ciudad*. De manera similar, el registro cuenta consta de dos campos: *número* y *saldo*.

En la figura 3.3, aparece un ejemplo de esta base de datos. Muestra que el cliente Rodríguez tiene la cuenta 101, el cliente Suárez las cuentas 405 y 231, el cliente Olmedo la cuenta 131.

Nótese que el conjunto de todos los clientes y cuentas está organizado en forma de árbol con raíz donde la raíz del árbol es un nodo ficticio. Una base de datos jerárquica es una colección de árboles de este tipo.

El contenido de cada registro específico puede tener que repetirse en varios sitios. Por ejemplo, en el sistema bancario *cliente-cuenta*, una cuenta puede pertenecer a varios clientes. La información correspondiente a esa cuenta, o la correspondiente a los clientes a los que puede pertenecer, tendrá que repetirse. Esa repetición puede ocurrir tanto en el mismo árbol de base de datos como en varios árboles distintos. La repetición de registros tiene dos desventajas principales:



- ♦ Puede producirse una inconsistencia de los datos al llevar a cabo la actualización.
- ♦ El desperdicio de espacio es inevitable.

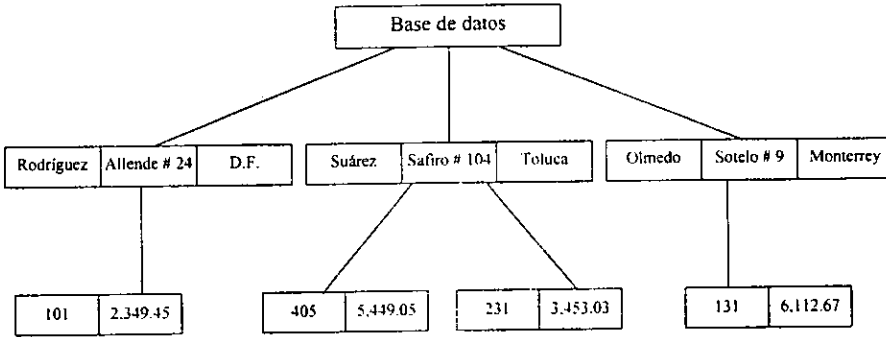


Figura. 3.3 Ejemplo de Modelo Jerárquico.

3.6.- Enfoque Relacional

Cuando hablamos del termino relacional, nos referimos a establecer una relación entre varias personas o cosas; estableciendo como relación una conexión de una cosa con otra, o la correspondencia entre dos personas o cosas que consideran al mismo tiempo y que tienen ciertas características en común. De esta manera en bases de datos también podemos hablar de relaciones, como se vera a continuación

3.6.1.- Definición de un DBMS Relacional

El DBMS relacional consiste en una colección de tablas, cada una de las cuales está asignada con un nombre único, cada una de éstas representa una entidad del mundo real y a su vez se identifica por filas y columnas. Una fila de una tabla comprende una relación entre un conjunto de valores. Las columnas, campos o atributos de una tabla pueden arreglarse en cualquier orden sin afectar el significado de los datos. Una tabla siempre tiene un campo o grupo de campos que sirven como llave única; debido a que dos registros no pueden ser idénticos, aquélla que se elija como llave única de identificación se denomina llave primaria. Si en un conjunto de tablas que constituyen una base de datos relacional un atributo sirve como llave primaria de una tabla y aparece como campo de otra, entonces se define como llave foránea.

3.6.2.- Reglas de Integridad

Las reglas de integridad proporcionan un medio para asegurar que los cambios efectuados en la base de datos por usuarios autorizados no resulten una pérdida de



consistencia de los datos, de esta manera las reglas de integridad protegen la base de datos contra daños accidentales.

A).- Valores Nulos

Cuando un campo no tiene valores, contiene un nulo o en el caso de campos de texto, un nulo o una cadena de longitud cero. Permitir nulos en un campo puede indicar que es información que existe pero se desconoce.

Características de los Valores Nulos:

- ♦ No ocupan espacio.
- ♦ Se pueden realizar operaciones con ellos cambiando de valor nulo a valor cero.
- ♦ Se pueden obtener los registros que contengan valores nulos al realizar una consulta.
- ♦ Una llave primaria no puede tener la propiedad de valor nulo.

En ocasiones es necesario prohibir la entrada de valores nulos por lo que se establecen ciertas restricciones.

B).- Integridad Referencial

A menudo se requiere asegurar que un valor registrado en una tabla para un conjunto de atributos dados, también aparezca para un cierto conjunto de atributos de otra tabla. Esto se llama integridad referencial; es decir, que las actualizaciones (modificación, eliminación, inserción) efectuadas en una tabla que esté asociada con otra tabla, deben ser actualizadas sin que se produzca la inconsistencia.

C).- Afirmaciones

En una afirmación se expresa una condición que siempre satisfaga la base de datos. Por ejemplo, en la *Tabla Inversiones* se necesita que el campo *Número de Cuenta* sea mayor que 1,000,000 para llevar un eficaz control de los clientes. Cuando se hace una afirmación, el sistema prueba su validez. Si la afirmación es válida, entonces cualquier modificación futura de la base de datos está permitida sólo si no provoca que se viole la afirmación.

D).- Triggers (Disparadores)

Un disparador es una sentencia ejecutada por el sistema automáticamente como un efecto secundario de una modificación de la base de datos. Para diseñar un mecanismo de disparador se deben especificar las condiciones bajo las cuales se va a accionar el disparador, y las medidas que se van a tomar cuando se realice dicha acción. Para ilustrarlo, al realizar movimientos dentro de la *Tabla Inversiones*, ésta afectará automáticamente a otra tabla, que registrará el tipo del movimiento realizado (Cancelado, Diferido y Aplicado), que modificará el campo *estado*; la condición puesta en este campo ejecuta el disparador que de manera automática, nos permitirá realizar o no consecuentes modificaciones, actualizaciones o eliminación de registros de la *Tabla Inversiones*.



3.7.- Asociaciones

Una asociación es la unión o enlace entre dos o más entidades (u otras asociaciones), las cuales se encuentran dentro del alcance del sistema, y por ello el sistema debe mantener, correlacionar y desplegar información.

Generalmente las asociaciones requieren de al menos dos entidades (es decir tablas primas), éstas deben estar dentro del alcance del sistema. Existen tres tipos de asociaciones:

Asociación UNO a UNO (1:1)

Las ocurrencias de una entidad se pueden relacionar sólo a una ocurrencia de la otra entidad.

Asociación UNO a MUCHOS (1:M)

Se dice que una relación entre entidades es UNO a MUCHOS, si las ocurrencias de una entidad están relacionadas con diversas ocurrencias de otra entidad.

Asociación MUCHOS a MUCHOS (M:M)

Se presenta cuando se asocian una ocurrencia en una entidad, con muchas ocurrencias en la otra entidad y viceversa.

3.8.- Normalización

En términos generales, el objetivo del diseño de una base de datos relacional es generar un conjunto de esquemas (diseño global de la base de datos) de relaciones que le permitan almacenar datos sin redundancia innecesaria, pero que a la vez faciliten recuperar información de manera sencilla. Para determinar si un esquema de relaciones tiene una de las formas normales, se necesita información adicional sobre la entidad del mundo real que se modelará con la base de datos.

La normalización es una técnica para el diseño de bases de datos, representativa del tipo de métodos que toman de entrada una lista de campos y las asociaciones entre ellos. La normalización de datos constituye una metodología para arreglar campos en tablas, de manera que se elimine la redundancia entre los campos no llave y donde cada una de las tablas resultantes se ocupa de un área de conocimiento.

La entrada requerida por el proceso de normalización es una lista del conjunto de campos de datos y las asociaciones entre ellos. Dado que resulta más fácil explicar el proceso en el contexto de un ejemplo que incluya registros de muestra, a continuación se enseñan los datos en un arreglo estructurado en forma libre, denominado forma no normalizada.

**CLIENTES.**

NoCuenta	NoBanco	NomBanco	Beneficiario	Domicilio	Nacionalidad	NoNacio	Saldo
021	01	Banamex	Saul	Santiago # 32	USA	02	3,459.34
	02	Bancomer					2,300.45
035	03	Bancrecer	Eduardo	Roma # 123	Mex.	01	1,234.09
	05	Bitel					456.96
050	04	Banoro	Rafael	Narvarte # 2	Mex.	01	5,323.98
075	06	Inverlat					32,212.90
	07	Mercantil P.	Hector	Anzures # 4	USA	02	2,323.87

*Figura. 3.4 Datos No Normalizados.***Primera Forma Normal (1nf)**

Los datos en la primera forma normal tienen la propiedad de que cada anotación de datos o valor de campo debe ser indivisible (atómico). Para que una tabla este en primera forma normal es necesario eliminar los atributos o grupos de atributos repetidos. La siguiente figura es la representación de la primera forma normal de los datos de la figura 3.4. Cada anotación de campo de un registro está constituida por un solo elemento de datos no subdivisible. En esencia, los registros no normalizados con campos multivaluados se desglosarán para producir varios registros con algunos datos repetidos, según fue necesario; la representación de los datos en la primera forma normal no es, por sí misma, útil para el control de la redundancia sino únicamente un punto de arranque para continuar trabajando.

CLIENTES.

NoCuenta	NoBanco	NomBanco	Beneficiario	Domicilio	Nacionalidad	NoNacio	Saldo
021	01	Banamex	Saul	Santiago # 32	USA	02	3,459.34
021	02	Bancomer	Eduardo	Roma # 123	Mex.	01	2,300.45
035	03	Bancrecer	Eduardo	Roma # 123	Mex.	01	1,234.09
035	05	Bitel	Eduardo	Roma # 123	Mex.	01	456.96
050	04	Banoro	Rafael	Narvarte # 2	Mex.	01	5,323.98
075	06	Inverlat	Hector	Anzures # 4	USA	02	32,212.90
075	07	Mercantil P.	Hector	Anzures # 4	USA	02	2,323.87

← PK

*Figura. 3.5 Primera Forma Normal.***Segunda Forma Normal (2nf)**

Para que una tabla se encuentre en segunda forma normal es necesario eliminar los atributos dependientes de una parte de la llave primaria.



Tal cual se puede apreciar los datos de la figura 3.5., son muy redundantes. En este punto, la metodología se orienta al problema de cómo buscar en la estructura de datos y qué modificar para disminuir la redundancia.

La combinación de campos NoCuenta (Número de Cuenta) y NoBanco (Número de Banco) es una llave (Primary Key) válida para este archivo, lo cual significa, "que cada campo no llave del archivo depende de la llave primaria", aunque ambas partes de la llave primaria compuesta son necesarias para definir el campo Saldo; sólo una de ellas se necesita para definir cada uno de los otros campos no llave (NomBanco, Beneficiario, Domicilio, Nacionalidad, NoNacio).

En la Figura 3.6, se muestran los mismos datos en la segunda forma normal; los cuales se han dividido en tres tablas, cada una de tablas tiene la propiedad de que su llave primaria define a cada uno de sus campos no llave. Dentro de una tabla específica, ningún campo no llave está determinado solamente por una parte de la misma. Varios campos se duplicaron en este proceso, NoCuenta que sólo aparecía una vez como campo en la primera forma normal, ahora se encuentra tanto en la tabla de *clientes* como en la tabla de *saldos*; este tipo de duplicación de campo es necesario en esta etapa entre aquellos campos que participan como campos llave (PK). En realidad, el número total de ocurrencias de valor de campo ha disminuido, lo cual es indicativo de un decremento en la redundancia. Obsérvese también que todas las entidades identificables de manera individual están representadas en cada una de las tablas.

CLIENTES.

NoCuenta	Beneficiario	Domicilio	Nacionalidad	NoNacio
021	Saul	Santiago # 32	USA	02
035	Eduardo	Roma # 123	Mex.	01
050	Rafael	Narvarte # 2	Mex.	01
075	Hector	Anzures # 4	USA	02

← PK →

BANCOS.

NoBanco	NomBanco
01	Banamex
02	Bancomer
03	Bancrecer
04	Banoro
05	Bital
06	Inverlat
07	Mercantil P.

← PK →

**SALDOS.**

NoCuenta	NoBanco	Saldo
021	01	3.459,34
035	02	2.300,45
035	03	1.234,09
035	05	456,96
050	04	5.323,98
075	06	32.212,90
075	07	2.323,87

PK

Figura. 3.6 Segunda Forma Normal.

Tercera Forma Normal (3nf)

Para que una tabla se encuentre en tercera forma normal se debe eliminar los atributos que dependan de atributos que no son parte de la llave primaria.

La tabla *bancos* y la de *saldos* de la figura 3.6, están libres de redundancia pues todos los campos no llave dependen de su llave primaria. Así, ambas tablas ya están en tercera forma normal. Si se vuelve a dar un vistazo a la tabla de *clientes* se le revelará ciertas redundancias residuales, por ejemplo, los registros 1 y 4 indican el número de nacionalidad 02 con nacionalidad americana y algo similar sucede con los registros 2 y 3, el número de nacionalidad es 01, sin embargo, de acuerdo con la llave primaria de esa tabla; NoCuenta identifica cada uno de sus otros campos, pero dado, que la llave está constituida por un campo, la regla de la segunda forma normal de dependencia de los campos no llave con respecto a la llave primaria no se cumple.

El problema se encuentra relacionado con el campo, NoNacio, que define a Nacionalidad, pero claramente no es una llave primaria para toda la tabla (no define a NoCuenta, Beneficiario o Domicilio). De esta manera, la situación generada es que un campo no llave define a otros del mismo tipo. Esta forma de relación inesperada, es otro indicativo de que se están mezclando tipos fundamentalmente distintos de información en la misma tabla, acción que provoca la redundancia.

CLIENTES.

NoCuenta	Beneficiario	Domicilio	NoNacio
021	Saul	Santiago # 32	02
035	Eduardo	Roma # 123	01
050	Rafael	Narvarte # 2	01
075	Hector	Anzures # 4	02

PK

**NACIONALIDAD.**

NoNacio	Nacionalidad
01	Mex.
02	USA

PK

BANCOS.

NoBanco	NomBanco
01	Bancomer
02	Bancomer
03	Bancrecer
05	Bitel
04	Banoro
06	Inverlat
07	Mercantil P.

PK

SALDOS.

NoCuenta	NoBanco	Saldo
021	01	3,459.34
035	02	2,300.45
035	03	1,234.09
035	05	456.96
050	04	5,323.98
075	06	32,212.90
075	07	2,323.87

PK

Figura. 3.7 Tercera Forma Normal.

En la figura 3.7., se muestra la representación de los datos en la tercera forma normal; obsérvese que se ha creado una nueva tabla, la de nacionalidad, para separar los datos relacionados con el tipo de nacionalidad del cliente, así, al dividir la **Tabla Nacionalidad** se dejó una copia de la columna NoNacio en la **Tabla Clientes**, esto se hace debido a que constituye la única forma de continuar indicando la nacionalidad del cliente. Para concluir se puede señalar que en la tercera forma normal, no existe situación alguna donde un campo no llave defina a otro del mismo tipo en una tabla.

3.9.- Desnormalización

No siempre es posible tener un modelo 100% normalizado, o sea hasta su tercera forma normal, ya que existen situaciones en las cuales es necesario tener redundancia, esto



cuando se quiere mayor respuesta en el sistema por lo que es valido pensar en desnormalizar.

En la figura 3.8, se muestran los datos de *ejemplo en una tabla no normalizada*, los datos de ésta pueden incluir situaciones como un campo multivaluado; por ejemplo, el segundo registro en la figura 3.8, se refiere al número de cuenta 35 y se anota en ese sólo registro, que tiene tres capacidades. Esta forma de datos tiene ciertas desventajas como: la necesidad de alguna forma de registro de longitud variable y complejidad adicional para el programador, lo que invariablemente lleva a un número mayor de errores en la programación. Además, en el caso del registro de *clientes* número 21, se sabe que el número de banco 02 coincide con el nombre del banco "Bancomer" y así sucesivamente, mediante todos los pares de números de capacidad y categorías listados, pero tales coincidencias pueden no siempre existir en todas las situaciones semejantes, lo que provoca una gran confusión.

CLIENTES.

NoCuenta	NoBanco	NomBanco	Beneficiario	Domicilio	Nacionalidad	NoNacio	Saldo
021	01	Banamex	Saul	Santiago # 32	USA	02	3,459.34
	02	Bancomer					2,300.45
035	03	Bancrecer	Eduardo	Roma # 123	Mex.	01	1,234.09
	05	Bitel					456.96
050	04	Banoro	Rafael	Narvarte # 2	Mex.	01	5,323.98
075	06	Inverlat					32,212.90
	07	Mercantil P.	Hector	Anzures # 4	USA	02	2,323.87

Figura. 3.8 Datos No Normalizados.

Después de revisar los fundamentos de las bases de datos, se explicara los puntos más importantes del análisis de un sistema, punto importante para el desarrollo del Sistema Integral de Oficinas Centrales.



CAPÍTULO 4

ANÁLISIS DEL SISTEMA

4.1.- Determinación de los Requerimientos de información

Los propósitos que se tienen al realizar un análisis de requerimientos son:

- Identificar las necesidades del cliente.
- Evaluar la viabilidad del sistema a implementar.
- Realizar un análisis técnico y económico.
- Asignar funciones al software, al hardware, a la gente, a la base de datos y a otros elementos del sistema.
- Establecer restricciones de costo y tiempo.
- Crear una definición del sistema que sea la base para todo el trabajo posterior de ingeniería.

El tamaño del sistema y su complejidad, el área de aplicación, el uso final y las obligaciones del contrato son sólo unas pocas de las muchas variables que afectan al esfuerzo total dedicado al análisis. Una regla usada a menudo es que se debe dedicar entre el 10 y el 20 por ciento de todo el esfuerzo de desarrollo al análisis del sistema y aplicar otro 10 o 20 por ciento del esfuerzo a la ingeniería del software del análisis de los requisitos del software. Todas las tareas han de ser dirigidas por un analista, éste trabaja en contacto con el personal técnico y administrativo, tanto del cliente como del que desarrolla el sistema. Se debe organizar un equipo para cada una de las tareas de análisis, dada la magnitud de algunos programas.

La primera etapa del proceso de análisis del sistema implica la identificación de las necesidades. El analista se entrevista con el cliente. El cliente puede ser un representante de una compañía externa, (cuando se está definiendo un producto) o de otro departamento (cuando se va a desarrollar un sistema interno).

La identificación de las necesidades es el punto de partida en la evolución de un sistema basado en computadora. Para empezar, el analista da asistencia al cliente definiendo los objetivos del sistema: la información que se va a obtener, la información



que se va a suministrar, las funciones y el rendimiento requerido. El analista se encarga de distinguir entre lo que necesita el cliente y lo que el cliente desea.

La siguiente etapa que aborda el analista, es la determinación de los requerimientos de información a partir de los usuarios particularmente involucrados. Para identificar las necesidades de información dentro de la empresa, pueden utilizarse diversos instrumentos, los cuales incluyen: el muestreo, el estudio de los datos y formas usadas por la organización, la entrevista, los cuestionarios; la observación de la conducta de quien toma las decisiones, así como de su ambiente y también el desarrollo de prototipos.

En esta etapa el analista hace todo lo posible por identificar qué información requiere el usuario para desempeñar sus tareas. Puede visualizar los métodos para establecer las necesidades que lo obligan a relacionarse directamente con los usuarios. Esta etapa sirve para elaborar la imagen que el analista tiene de la organización y de sus objetivos. Por consiguiente, el primer paso es comprender la situación, siguiendo con el proceso descrito.

4.1.1.- Técnicas para la recopilación de hechos de estudio

En la fase de análisis del proceso de desarrollo del sistema de información, el analista de sistemas depende de técnicas específicas para recopilar información.

Las técnicas que se presentan, no son una lista exhaustiva de lo que existe disponible o lo que se utiliza. Sin embargo, se hará la identificación de las técnicas principales utilizadas por los analistas de sistemas.

4.1.1.1.- La Entrevista

En muchos casos, la mejor forma de obtener hechos de estudio críticos consiste en realizar una serie de entrevistas. En general, preguntas tales como: "¿éste reporte le proporciona lo que necesita?" y "¿cómo podría hacerse esto mejor?" permiten a quienes responden contribuir al análisis.

Es importante que el analista de sistemas se asegure de que cada persona que responde entienda que el objetivo final del trabajo en sistemas es hacer que el nuevo sistema sea más útil. La obtención de hechos significativos y útiles por parte de las personas que responden, está en función de una actitud positiva de parte de todos los participantes. Los tonos y matices sutiles de parte de quienes responden pueden ser tan significativos como sus respuestas directas a las preguntas. Finalmente, debido a que el sistema final desarrollado descansará en gran medida en los hechos proporcionados por las personas no será mejor que los hechos sobre los cuales se basa.

La entrevista, es la técnica más significativa y productiva de que dispone el analista



para la indagación de hechos, es un intercambio cara-cara de información.

Adicionalmente, la entrevista proporciona una oportunidad excelente para que el analista establezca relaciones con el personal usuario.

Las entrevistas se realizan a todos los niveles de la organización, desde el presidente o el jefe principal de operaciones, hasta el empleado de correspondencia o personal de mantenimiento. En consecuencia, los procedimientos para las entrevistas pueden ir desde aquellos altamente formales hasta los casuales. El éxito de la entrevista depende de lo bien que el analista se adapte a estas variables ambientales.

4.1.1.2.- Método de Análisis en Grupos

Este método se utilizará cuando en el proyecto de sistema, se encuentran diversos grupos de usuarios involucrados en actividades diferentes y con intereses diferentes, o también en un sistema que servirá como una nueva función de la empresa con la cual no existe una experiencia previa.

El analista deberá enfocar correctamente su tema de discusión, en cada sesión frente al grupo, para obtener datos relevantes que lleven a un buen enfoque de la situación a resolver y no estar recibiendo información fuera de lugar; ciertamente cuando se está trabajando con un grupo de personas se puede caer en divagaciones o discusiones fuera de contexto por lo que el analista deberá dejar en claro los puntos a discutir y también los acuerdos a los que llegó el grupo.

Un analistas podrá tener una diversidad de opiniones sobre los puntos específicos que se estén indagando, esto dará un resultado más conveniente y favorable, con lo cual es posible formarse un criterio más amplio de como se están realizando los procesos dentro de la empresa.

Mediante este proceso el analista podrá tener una visión más amplia de las posibles relaciones que tendrán los diferentes departamentos en la empresa, la información que será esencial para cada departamento y los datos que tendrán que compartir, así mismo el flujo de información que se tendrá.

4.1.1.3.- El Cuestionario

El analista de sistemas puede utilizar un cuestionario en varios momentos durante el proceso de desarrollo de sistemas. Los analistas deben identificar lo que desean saber, para estructurar las preguntas que den por resultado las respuestas a estas necesidades para preparar y entregar el cuestionario a la persona que lo va a contestar.

El cuestionario puede utilizarse, como una herramienta para descubrir hechos cuando el entrevistado está alejado físicamente del analista y un viaje es prohibitivo para cualquiera de los dos, en los casos en donde hay muchos receptores potenciales (p. ej.,



una fuerza de ventas) y cuando la información pretende verificar información similar recopilada de otras fuentes.

4.1.1.4.- La Observación

Otra técnica con que cuenta el analista durante la indagación de hechos consiste en observar a las personas en el momento de ejecutar su trabajo. La observación, es una técnica para descubrir hechos que tienen una amplia aceptación por parte de los científicos. El propósito de la observación es múltiple dado que le permite al analista determinar lo que se está haciendo, la forma en que se hace, quién lo realiza, cuándo, cuánto tiempo requiere, donde se hace y por qué.

El analista puede observar e interactuar con las personas que están siendo observadas. Esta interacción puede consistir simplemente en cuestionar una tarea específica, pedir una explicación, etc.

La observación puede emplearse para verificar lo que se reveló en una entrevista o como paso preliminar para esta última. La observación también es una técnica valiosa para recopilar hechos que representan relaciones. La observación tiende a ser más significativa a nivel técnico del procesamiento de datos en donde las tareas se pueden cuantificar más fácilmente. Las actividades técnicas incluyen tareas relacionadas con la recopilación de datos, su acumulación y transformación. Las actividades de toma de decisiones no se prestan fácilmente a la observación, se pueden entender mejor a través del proceso de la entrevista y el uso del análisis a nivel de decisiones.

Son muchos los beneficios que pueden derivarse de una observación hábil. Sin embargo, a medida que los analistas adquieren más experiencia, se vuelven más selectivos en lo que observan y el momento en que lo hacen. Se recomienda mucho que cuando se utilice la observación, ésta se emplee conjuntamente con otras técnicas de indagación de hechos para maximizar su eficacia.

4.2.- Análisis de las necesidades del sistema

La siguiente etapa que se ejecuta consiste en analizar las necesidades propias del sistema. Una vez más, existen herramientas y técnicas especiales que facilitan la realización de las especificaciones requeridas. Estas, incluyen el uso de los diagramas de flujo de datos que cuentan con una técnica estructurada para representar en forma gráfica la entrada de datos de la empresa, los procesos y la salida de la información. A partir del diagrama de flujo de datos se desarrolla un diccionario de datos que contiene todos los elementos que utiliza el sistema, así como las especificaciones del tipo de dato de cada campo, si es *Numérico*, *Texto*, *Lógico*, *Memo*, *Fecha/Hora* y el espacio requerido para cada tipo de dato.



Durante esta fase, el analista de sistemas también analiza las decisiones estructuradas por realizar, que son decisiones donde las condiciones, alternativas, acciones y reglas de acción podrán determinarse. Existen tres métodos para el análisis de las decisiones estructuradas: el lenguaje estructurado (en nuestro caso el español), las tablas de decisiones y los árboles de decisiones.

No todas las decisiones en las empresas se encuentran estructuradas; no obstante, es importante que las comprenda el analista de sistemas. Las decisiones semiestructuradas (decisiones que se toman bajo riesgo), con frecuencia se apoyan en los sistemas de toma de decisiones. Cuando analiza las decisiones semiestructuradas, el analista las examina de acuerdo con el grado de complejidad del problema y con el número de criterios considerados al llevarlas a cabo.

El análisis de decisiones de criterio múltiple (aquellas donde numerosos factores tienen que equilibrarse) también es parte de esta etapa. Se disponen de muchas técnicas para el análisis de decisiones de criterio múltiple; incluyendo, entre otras, el proceso de intercambio y la aplicación de métodos de ponderado.

A esta altura del ciclo de desarrollo del sistema, el analista prepara una propuesta del sistema que resume todo lo que ha encontrado, presenta un análisis costo/beneficio de las alternativas y plantea las recomendaciones (si es que existen) de lo que deberá realizarse. Si la dirección acepta alguna de las recomendaciones, el analista procederá de acuerdo con ellas. En sistemas, cada problema es único y en consecuencia, nunca habrá sólo una solución correcta. La manera como se plantea una recomendación (esto es, una solución) depende de las características individuales de cada analista aunadas a su formación profesional.

4.2.1.- Preparación del Reporte de Terminación del Análisis de Sistemas

Sin embargo, quizás la comunicación más importante de todas es el reporte de terminación del análisis del sistema, el que describe los hallazgos del análisis de sistemas. El formato y contenido de este reporte incluye lo siguiente:

- Una nueva exposición de la razón y alcance del análisis.
- Una lista de los principales problemas identificados.
- Una presentación de todos los requerimientos de los usuarios.
- Un planteamiento de todas las suposiciones críticas hechas por el analista durante el análisis.
- Una proyección de los recursos requeridos y los costos esperados que estarán involucrados en el diseño de cualquier nuevo sistema o en la modificación del sistema actual. Esta proyección incluye la factibilidad de continuar con el trabajo en sistemas.
- Cualquier recomendación referente al sistema propuesto o a sus requerimientos.



En general, el reporte de terminación del análisis de sistemas está dirigido a dos receptores diferentes. Primeramente, el gerente utiliza el reporte para determinar si el analista ha realizado un trabajo competente en la identificación de los requerimientos de los usuarios y en evaluar la forma en que estos requerimientos entraron en cualquier plan maestro o general, para el desarrollo de sistemas en la organización. En segundo lugar, el reporte proporciona a la gerencia general y a la gerencia de los usuarios una oportunidad de determinar si el analista ha considerado todos los requerimientos de la organización.

Para proporcionar un reporte significativo a estas dos partes interesadas, el analista deberá esforzarse por ser conciso pero completo al preparar el reporte. Los requerimientos deberán cuantificarse y explicarse de manera específica.

El analista deberá evitar en el reporte el lenguaje técnico. Deberán anexarse exposiciones y los documentos de trabajo que se utilizaron en el análisis de sistemas.

4.2.2.- Relación costo beneficio

La programación del tiempo y la estimación de costos del software están íntimamente relacionadas. La mayor parte de los costos de un proyecto de software grande son sólo aquellos relativos al pago de la gente que desarrolla el software. El costo del proyecto es directamente proporcional al número de personas-mes requeridas para terminar el trabajo. Hay, desde luego, otros costos, como los de hardware, viajes, capacitación, etc., pero éstos son más fáciles de calcular. No se basan en imponderables como la productividad del programador y la estimación del tamaño del programa fuente.

La técnica más utilizada para calcular los costos del software es estimar el tamaño del sistema de programación que se va a entregar y de ahí calcular el número de programadores-mes requeridos para construir el sistema, por medio de datos históricos de productividad o la intuición. El costo del sistema total se basa en esa cifra más los gastos varios. Si el software se desarrolla para una agencia externa, se añade una cantidad de utilidades a esta estimación.

El tamaño del sistema se calcula por medio de un estudio del sistema preliminar para establecer las partes que componen el sistema. Se hace una estimación del tamaño de cada unidad. Tal estimación suele basarse en la experiencia y la intuición. Después se suman las estimaciones para obtener el tamaño calculado del sistema total. Este método se basa en la suposición de que se puede predecir la productividad del programador para una parte dada del sistema. También, supone que el diseño preliminar no está simplificado en demasía y que refleja con exactitud el sistema que se va a producir.

Ya se ha visto que la productividad del programador se ve afectada por múltiples factores y no se puede estimar con precisión, considerando el tamaño de cada unidad que se va a desarrollar. Otros factores, como la experiencia del programador, la



novedad, la complejidad y las restricciones de la aplicación también se deben tener en cuenta.

Todos los proyectos son realizables (con recursos ilimitados y un tiempo infinito). Desafortunadamente, el desarrollo de un sistema se caracteriza por la escasez de recursos y la dificultad de cumplir los plazos de entrega, es necesario y prudente evaluar la viabilidad del proyecto lo antes posible. Se pueden evitar meses o años de esfuerzo, miles de millones de pesos y una inversión profesional incalculable, si un sistema no mal concebido es reconocido como tal al principio de la etapa de definición.

El análisis de viabilidad está relacionado de varias maneras. Si el riesgo del proyecto es grande, se reduce la probabilidad de producir software de calidad. No será necesario llevar un sistema de viabilidad para sistemas en los que la justificación económica es obvia, se esperan pocos problemas legales y no existe una alternativa razonable. Sin embargo, cuando no se da una de las anteriores condiciones debe efectuarse el estudio.

La justificación económica es normalmente la principal consideración para la mayoría de los sistemas y comprende un amplio rango de aspectos en los que se encuentran el análisis costo-beneficio, las estrategias de ingreso a largo plazo, el impacto de otros productos o en centros de explotación, el costo de los recursos que se necesitan para el desarrollo y el crecimiento potencial del mercado.

El aspecto de factibilidad económica es fundamental para el desarrollo de cualquier sistema, parte del análisis de nuestro sistema, fue decidir *cómo* producir un sistema eficiente (económico) y efectivo (significativo y útil). Para ello, se evalúan los *aspectos de diseño*, determinar las *alternativas factibles* y después se decide por un sólo conjunto de *especificaciones detalladas* para la solución del problema. Los aspectos de metodología y uso de recursos afectan las alternativas que serán consideradas y las decisiones relacionadas con los aspectos de diseño no pueden tomarse antes de estudiar cuidadosamente los recursos económicos de que se dispone.

El propósito de la evaluación económica, es comparar el desarrollo estimado y los costos de operación para identificar los beneficios con un nivel general de precisión que nos posibilite determinar la viabilidad del sistema, los costos de desarrollo serán estimados en base a los requerimientos del equipo y del personal involucrados en el proyecto, en tanto que los costos de presentación variarán de acuerdo al estilo de administración de la organización en la que el sistema será implementado.

4.2.3.- Concepto de redes de área local

Debido a que en el I.F.E. se encuentran operando en una Red local con sistema operativo Netware, describiremos a continuación los conceptos de red de área local y Sistema Operativo Netware, los cuales tendremos que tomar en consideración para llevar a cabo nuestro análisis.



Red de Área Local

Una red de área local, es un sistema que permite a las computadoras personales compartir información y recursos en una área limitada (generalmente menor de una milla desde el servidor de archivos a una estación de trabajo). Una LAN, requiere que las estaciones de trabajo individuales (PC) sean conectadas físicamente, y que el software de la red resida en el servidor de archivos (esto permite la compartición de periféricos, datos, y programas de aplicación).

Una red local de acuerdo con el concepto del proyecto IEEE puede describirse por sus funciones y características, es un sistema de comunicación de datos que permite que un número de dispositivos de procesamiento de información independientes se comuniquen entre ellos, con las siguientes características:

- ♦ Área moderada; por ejemplo, una oficina, un almacén, una universidad.
- ♦ Canal de comunicación de capacidad media-alta
- ♦ Probabilidad de error baja en los mensajes internodo.

Las áreas de aplicación se agrupan en una o más de las siguientes categorías:

- ♦ datos
- ♦ voz
- ♦ gráficos

Los objetivos primordiales de la red local son:

- ♦ Asegurar la competitividad de productos diseñados y fabricados por empresas distintas
- ♦ Disminuir la duplicidad de trabajo .
- ♦ Disminuir el papel de trabajo (como memorándums)
- ♦ Eliminar la pérdida de integridad de datos al trabajar muchos usuarios en el mismo proyecto y con diferentes discos
- ♦ Los recursos de hardware y software deberán ser aprovechados por toda la empresa, aunque cada departamento tenga sus propios equipos, deberán existir recursos compartidos.
- ♦ Facilidades para obtener información actualizada de otros departamentos.

Netware

Netware es un sistema operativo de red que fue creado por Novell Inc. Desde sus orígenes ha sufrido una serie de cambios y adaptaciones hasta convertirse en un sistemas operativo estable y maduro.

Un aspecto fundamental en la aceptación de Netware, se debe a que permite que los programas escritos para MS-DOS y Windows funcionen con el sistema operativo de la



red; por lo tanto no es necesario reescribirlos o recompilarlos; sin embargo se deben tener consideraciones de uso, como lo es el bloqueo de archivos o recursos.

Otro de los factores primordiales para la aceptación de Netware es su facilidad de uso tanto para el administrador de la red así como para el usuario final.

Generalmente, la persona designada como administrador de la red cuenta ya con conocimientos de computación; no sucede lo mismo con un usuario, el cual puede o no ser experimentado; en el caso de los últimos al usar Netware notarán que cuenta con una gran variedad de menús para desarrollar su trabajo, además de ello y debido a que Netware hace uso de MS-DOS, no es necesario prácticamente aprender toda una serie de comandos. El emigrar de MS-DOS a Netware no es un cambio fuerte para un usuario; únicamente se requiere que entienda el concepto global de la red y las seguridades, el usuario seguirá viendo un disco duro, quizá con una letra diferente (F en lugar de C), requerirá un usuario y password para conectarse además de reconocer la necesidad de directorios de trabajo para toda la información que requiera, posiblemente para usar Windows necesite de una capacitación sencilla.

Netware permite que mediante una misma tarjeta circulen varios protocolos como lo son IPX/SPX¹, TCP/IP².

Las primeras versiones de Netware permitían la creación de pequeños grupos de trabajo ya se de 4 u 8 estaciones de trabajo; posteriormente Advanced Netware permitía hasta 100 estaciones de trabajo, con Netware 3.11 tenemos la posibilidad de conectar hasta 250 estaciones de trabajo. Una gran ventaja de Netware 3.11 es que permite que se carguen al sistema operativo módulos de software con lo que se facilitan ciertas labores e inclusive dinámicamente se pueden ejecutar labores que antes no se podían como lo es el dar de alta una impresora. Netware 3.11 permite hasta 16 impresoras compartidas y hasta 5 por servidor.

NetWare está diseñado para redes soportadas por un servidor de archivos. Para entender este concepto, es necesario comprender como un servidor de archivos funciona bajo software Novell. Bajo el modelo OSI³, el software del servidor de archivos Novell reside en la aplicación, mientras el software del sistema operativo en disco (DOS) reside en la estación. En efecto, el software del servidor de archivos forma un shell alrededor de DOS, con la facultad de interceptar comandos desde los programas de aplicación antes de que ellos puedan alcanzar el procesador de comandos de DOS. Como veremos después, el usuario de una estación de trabajo no es consciente de este fenómeno. El usuario simplemente pregunta por una base de datos o un programa, sin importar en dónde esta localizado.

Cuando un programa de aplicación requiere un archivo específico, el shell debe determinar primero si el requerimiento es para un archivo local (residente en los discos propios de la estación de trabajo) o un requerimiento de red para información residente

¹ IPX/SPX: Protocolo de comunicaciones.

² TCP/IP: Protocolo de comunicaciones.

³ Modelo OSI: Modelo de referencia de Interconexión de Sistemas Abiertos.



en el servidor de archivos. Si la información es localizada en los drives de la estación de trabajo, el requerimiento es enviado al procesador de comandos de DOS, en donde es manejado como un operación normal de E/S. Cuando un archivo particular es cargado en la CPU de la estación de trabajo para ser procesado, el usuario puede notar que el dispositivo de disco está trabajando.

Cuando el requerimiento está localizado en el servidor de archivos, el shell genera un requerimiento de lectura para el mismo - que localiza la información y lo transmite a la estación de trabajo en la forma de un paquete de respuesta (REPLY PACKET). El paquete, es recibido por un traductor de respuestas, que convierte esta información en una forma que DOS en la estación de trabajo pueda manejar.

La estación de trabajo es completamente ignorante de los mecanismos internos de esta operación, debido a que el servidor de la red es tan rápido que la respuesta de la estación de trabajo y la red parecerá igualmente rápida excepto en casos del exceso de tráfico en la red.

4.3.- Justificación del Sistema

Esta justificación está fundamentada sobre dos bases importantes:

- A).- El sistema servirá como herramienta para la toma de decisiones de carácter financiero que dentro del IFE se realicen, así como para tener un control total sobre el presupuesto que a éste se le asigna.
- B).- Se abatirán costos, como se menciona más adelante.

4.3.1.- Objetivo del Sistema

Dentro de los objetivos del sistema se contempla el contar con la información más relevante del presupuesto al instante, así como poder obtener información para la toma de decisiones más rápidamente. Se pretende que el sistema realice el registro presupuestal, el registro financiero y automatice el área contable evitando la redundancia de información que hasta el momento se tiene.

4.3.2.- Prioridades para las características del sistema

Se contemplan básicamente tres:

- ♦ Integridad de Información
- ♦ Manipulación sencilla y clara del sistema
- ♦ Unicidad y seguridad de la información
- ♦ Utilización del equipo y software existente dentro del propio Instituto.



4.3.3- Análisis Costo-Beneficio

El Sistema Integral de Oficinas Centrales será rentable en la medida en que sus costos de operación satisfagan las necesidades de los usuarios, a medida que el hardware disminuye su costo, la rentabilidad puede elevarse drásticamente si el sistema no está planeado adecuadamente, esto debido a que los sistemas complejos no pueden ser construidos en un solo paso, frecuentemente se utiliza una estrategia incremental para obtener un producto básico y luego modificarlo y ampliarlo progresivamente hasta lograr la funcionalidad deseada. Entonces, la rentabilidad global requiere que el producto sea fácilmente modificable y ampliado.

Costos beneficios operativos

Para disminuir los costos de instalación y operación del sistema aprovecharemos parte de la infraestructura de cómputo ya instalada dentro del I.F.E. Esto en primera instancia, disminuirá de manera directa los costos operativos, sin embargo; a medida que se aumente el volumen de datos, se integren a la red nuevas aplicaciones y nuevos usuarios, se decida modificar el sistema para incluir nuevos módulos o adicionar programas para nuevos procesos, el rendimiento de la red se verá afectado y esto se reflejará en el rendimiento del Sistema Integral de Oficinas Centrales.

La infraestructura dentro del I.F.E. que se decidió utilizar fue la siguiente:

- Red Novell versión 3.11.
- Servidor 486 a 66 Mhz con 16 Mb en RAM .
- Estaciones de trabajo 386 a 50 Mhz y 8 Mb en RAM.
- Impresoras láser compartidas.
- Impresoras de matriz compartidas.

Esta infraestructura nos permite instalar la herramienta de desarrollo en red, desarrollar el sistema y liberarlo a producción. Estimamos que con esta infraestructura podemos trabajar a corto plazo sin mayor inversión, sin embargo recomendamos una actualización a los equipos a corto mediano como sigue:

- Servidor de red con procesador de 150 Mhz con disco duro de 4 Gb, 48 Mb memoria RAM y CD. *
- Estaciones de trabajo con procesador a 100 Mhz . *
- Una estación de trabajo con procesador a 100 Mhz y CD. *

* Equipos sugeridos al 17 de Octubre de 1996.

En la actualidad el control presupuestal no es confiable al 100 % debido a que no se tiene un mecanismo que administre de manera total los presupuestos asignados, así con el Sistema Integral de Oficinas Centrales controlaremos el presupuesto anual autorizado, desde la etapa de autorización, hasta la aplicación del mismo. Tendremos la

información exacta de las cuentas de cheques o de inversión, teniendo como beneficio que Contabilidad pueda de manera rápida realizar movimientos de control como conciliaciones bancarias y generación de pólizas; así mismo, permitirá de manera inmediata a los usuarios del I.F.E. compartir información de diferentes áreas a través de una base de datos común, concentrando la información en un solo punto para que los usuarios puedan modificar y consultar información de manera rápida, segura y con un alto grado de confiabilidad, además se optimizarán recursos materiales y se eliminarán tiempos perdidos en búsquedas y consultas que en la actualidad no se hacen por medios electrónicos.

Costos y beneficios a futuro

Considerando que todas las organizaciones son dinámicas, y que el I.F.E., al igual que una gran organización, es susceptible a cambios en su estructura y procedimientos a lo largo del tiempo, tenemos que considerar que el Sistema Integral de Oficinas Centrales será modificado varias veces durante su vida útil, los cambios que propicien ciertas modificaciones sobre el sistema no los podemos predecir actualmente, de tal manera que realizamos un diseño flexible e integrado de base de datos que se pueda adaptar de manera fácil a los cambios futuros en la organización a través de tener flexibilidad.

En base a lo anterior se determino que fue la mejor alternativa debido a las ventajas que representa en la actualidad tener un sistema flexible. Como se ilustra en la figura 4.1., el costo total a lo largo de la vida del sistema puede muy bien resultar menor cuando el objetivo del diseño es la flexibilidad.

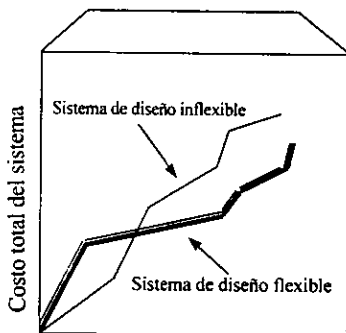


Figura. 4.1 Vida del Sistema.

Costos beneficios intangibles

Estos costos son difíciles de cuantificar, debido a que dependerán de los cambios en los procedimientos internos del I.F.E., si los cambios en los procedimientos son orientados para mejorar la productividad con base en la infraestructura actual, seguramente se obtendrá de manera global un beneficio sin la necesidad de realizar un nuevo sistema.



Costos beneficios del personal

La relación existente entre los módulos del S.I.O.C. permitirá que las diferentes áreas puedan compartir información, sin necesidad de invertir tiempos de espera perdidos cuando una área espera información de otra área para continuar con su trabajo: El Sistema Integral de Oficinas Centrales liberará al personal de tareas rutinarias para que pueda perfeccionar las actividades propias de su área o incluso desarrollar otras actividades.

Costos beneficios del procesamiento de datos

En este momento es difícil estimar el costo - beneficio del procesamiento de datos. Tenemos que considerar que en el futuro los posibles cambios en los equipos de hardware pueden impactar en el incremento de los costos operativos y de proceso de datos si el sistema es implementado en una plataforma con características diferentes al actual. Por otro lado la instalación no planeada de un nuevo software que este interactuando en el mismo ambiente que el Sistema Integral de Oficinas Centrales puede afectar el rendimiento del sistema, aumentando los costos de procesamiento de datos.

Con el diseño se cumple una parte importante para la creación de un nuevo sistema, posterior a esto, debemos pasar a la siguiente etapa que es la de desarrollo, en esta etapa se sigue según lo especificado en el diseño. En el siguiente capítulo se explicara lo relacionado al diseño.



CAPÍTULO 5

DISEÑO Y DOCUMENTACIÓN DEL SISTEMA

5.1.- Técnicas Estructuradas Aplicadas

Para diseñar un sistema, existen técnicas como son los diseños descendentes y los ascendentes. además del enfoque modular de la programación, en este capítulo se describe cada una de estas técnicas.

5.1.1.- Diseño Ascendente.

Tiene un enfoque de “abajo hacia arriba”, esto es, se identifican aquellos procesos que necesitan computarizarse conforme van apareciendo, su análisis como sistemas y su codificación; o bien, la adquisición de paquetes de software para satisfacer el problema inmediato. Los problemas que requieren de la computarización, con mayor frecuencia se encuentran en los niveles inferiores de la organización. Es por ello que los problemas en tales niveles inferiores son los únicos problemas en los cuales el cómputo podría ser costeable. En consecuencia, ese enfoque se denomina ascendente, refiriéndose a que la computarización se implanta desde el nivel más bajo. Con frecuencia, las empresas se apegan a este enfoque del desarrollo de sistemas para iniciarse, adquiriendo, por ejemplo, paquetes de software de contabilidad, otro para la programación de la producción y alguno más para mercadotecnia.

Cuando la programación se realiza internamente y haciendo uso de un enfoque ascendente, es difícil llegar a integrar los subsistemas, a grado tal de que el desempeño global sea fluido. Los problemas de interacción entre los sistemas son sumamente costosos y muchos de ellos no se solucionan hasta que la programación alcanza la fecha límite para la integración total del sistema. En ésta fecha, ya se cuenta con muy poco tiempo, presupuesto o paciencia de los usuarios, como para corregir aquellas delicadas interfaces, que en un principio, se ignoraron.

Aunque cada subsistema parece ofrecer lo que se requiere, cuando se contempla al sistema como una entidad global, adolece de ciertas limitaciones por haber tomado un enfoque ascendente. Uno de ellos es la duplicación de esfuerzos para acceder el software y más aún, al introducir los datos. Otro es, que se introducen al sistema muchos datos carentes de valor. Un tercero; y tal vez el más serio inconveniente del enfoque ascendente, es que los objetivos globales de la organización no fueron considerados y en consecuencia, no se satisfacen.



5.1.2. Diseño Descendente

Implica observar la gran imagen del sistema y luego, desglosarlo en partes más pequeñas o subsistemas, tal y como se muestra en la figura 5.1 el diseño descendente obliga a que los analistas de sistemas se enteren primero de los objetivos globales de la organización, así como el establecimiento de la mejor manera de satisfacerlos dentro de un sistema integral. Luego, el analista se dirigirá a dividir tal sistema en sus subsistemas y sus requerimientos.

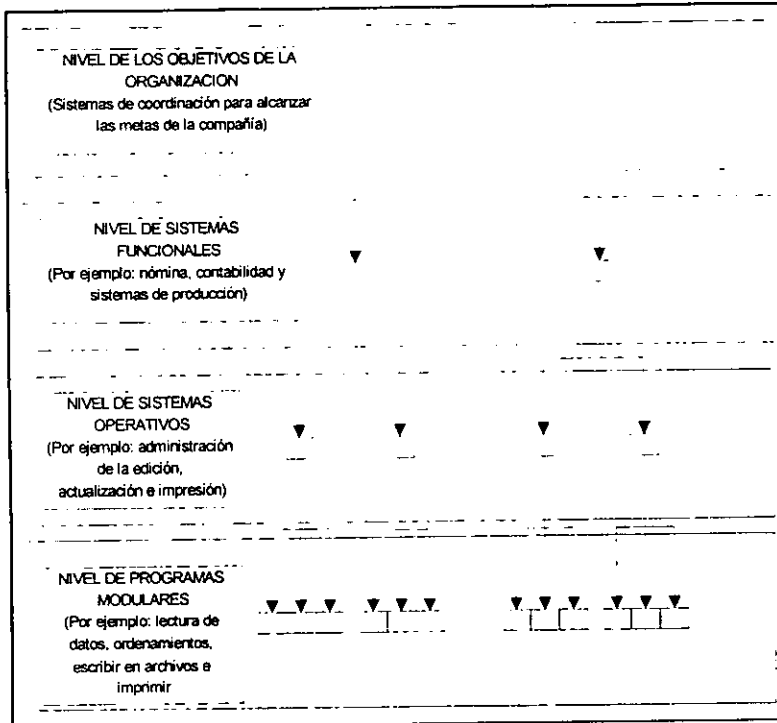


Figura. 5.1 Diseño Descendente.¹

Cuando el analista de sistemas emplea un enfoque descendente, está empleando las interrelaciones e interdependencias de los subsistemas, para apegarse lo mejor posible a las necesidades existentes en la organización.

El enfoque descendente da la importancia debida a las interfaces requeridas por el sistema y los subsistemas; los cuales no existen en el enfoque ascendente.

¹ Kenneth E. Kendall - Julie E. Kendall. 1991. *Análisis y Diseño de Sistemas*. Prentice-Hall, Hispanoamérica, S.A., México. Pág. 741



Dentro de las ventajas de la utilización de un enfoque descendente en el diseño de sistemas, se encuentra el evitar el caos originado al tratar de diseñar el sistema "en un sólo paso". El tratar de integrar a todos los subsistemas y que todos ellos funcionen al unísono, es buscar el fracaso.

La segunda ventaja de hacer uso del enfoque descendente en el diseño, es la posibilidad de contar con grupos de analistas de sistemas trabajando por separado pero simultáneamente en subsistemas independientes, pero necesarios. Esto puede ahorrar una gran cantidad de tiempo. El trabajo de grupos integrados para el diseño de subsistemas es particularmente conveniente para la búsqueda del aseguramiento de la calidad total.

La tercera ventaja estriba en evitar el problema asociado con un enfoque ascendente. Esto es, la utilización de un enfoque descendente previene que el analista de sistemas se adentre en los detalles y dé la pauta para que se pierdan los objetivos centrales del sistema.

Existen ciertos inconvenientes en el diseño descendente que se deben de considerar. Primero, puede suceder que el sistema se divida en subsistemas "incorrectos". Se debe prestar atención a la necesidad de la superposición y la distribución de los recursos, de tal forma que una participación de subsistemas tenga sentido en el esquema global del sistema. Además, es importante que cada subsistema se integre de modo correcto al sistema.

El segundo inconveniente se presenta cuando se realizan las divisiones en subsistemas, sus interfaces pueden descuidarse o simplemente ignorarse. La responsabilidad para lograr la adecuada interrelación debe quedar ampliamente detallada.

Un tercer inconveniente es que los mecanismos para la reintegración de los subsistemas deben planearse desde el principio, para poder llevar a cabo dicha reintegración posteriormente. Para no perder de vista ésta tarea, se puede intercambiar la información entre los equipos de subsistemas de manera regular, se pueden utilizar instrumentos que permitan cierta flexibilidad, si se requirieran cambios de los sistemas interrelacionados.

5.1.3. Desarrollo Modular

Una vez que se ha tomado el enfoque del diseño descendente, también será útil durante la programación, un enfoque de concepción modular. Esto nos conducirá a dividir la programación en fracciones lógicas y manejables. Este tipo de programación se apega bien al diseño descendente porque enfatiza las interfaces entre los módulos, más que mantenerlas ignoradas hasta el final del desarrollo del sistema. De manera ideal, cada módulo debe ser funcionalmente cohesivo, de tal manera que satisfaga sólo una función.



El diseño de programas modulares tiene tres ventajas básicas:

- Los módulos son más fáciles de escribir y de revisar, ya que están virtualmente autocontenidos. La detección de un error dentro de un módulo es menos complicada, ya que los problemas asociados al mismo no llegarán a trascender a los otros.
- El mantenimiento de los módulos es más fácil. Las modificaciones pueden limitarse a unos cuantos módulos y no al programa completo.
- La problemática de los módulos es más fácil de entender, ya que son sistemas autocontenidos. Esto significa que un lector comprenderá la función de un módulo específico, con sólo tomar su listado de código.

Algunos lineamientos para la programación modular incluyen:

1. Mantener cada módulo de un tamaño manejable (de modo ideal incluyendo sólo una función).
2. Prestar atención particular a las interfaces críticas (esto es, a los datos y a las variables de control que pasan entre los módulos).
3. Minimizar el número de módulos que el usuario necesite modificar cuando haga cambios.
4. Mantener las relaciones jerárquicas establecidas en las etapas de descenso.

5.1.4. Diagrama estructural.

Es el instrumento recomendado para el diseño de un sistema modular descendente. Un diagrama estructural simplemente es un diagrama que consta de rectángulos, los cuales, representan a los módulos y se conectan por medio de flechas.

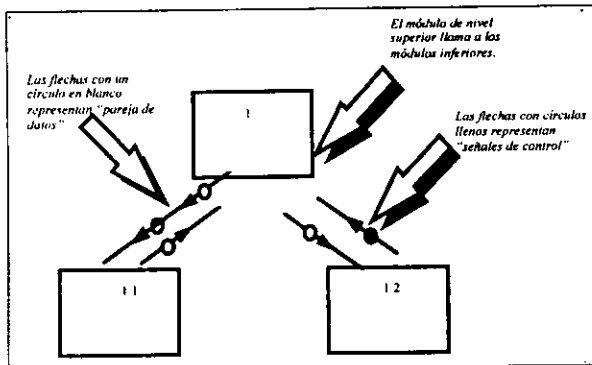


Figura. 5.2 Diagrama Estructural.²

² Kenneth E.Kendall - Julie E.Kendall. 1991. *Análisis y Diseño de Sistemas*. Prentice-Hall, Hispanoamérica, S.A., México. Pág. 746



La figura 5.2 muestra los tres módulos etiquetados 1, 1.1 y 1.2. El número a la derecha al punto en 1.1 y el 1.2 implica que estos son subconjuntos del módulo 1. Para recalcar que el diagrama estructural es una técnica de diseño descendente, las flechas de conexión se dibujan con una dirección de arriba hacia abajo.

A los lados de las flechas de conexión, se dibujan dos flechas más pequeñas. Tales flechas con círculos vacíos se denominan "parejas de datos", mientras que las flechas con los círculos llenos se denominan indicadores de control. Estas flechas indican que algo pasa, ya sea hacia abajo del módulo superior o de regreso del módulo inferior.

De manera ideal, el analista debe mantener el acoplamiento en un mínimo. Cuanto menos parejas de datos e indicadores de control tenga el sistema, más fácil será modificarlo. Cuando estos módulos se encuentran programados es importante pasar el menor número de pares de datos entre los módulos.

Deben evitarse los numerosos indicadores de control. Este se diseña para tenerlo en el descenso, pero en ocasiones, será necesario pasar indicadores de control hacia arriba. Unos ejemplos de indicadores de control son aquellos mensajes que le dicen al módulo superior que se ha alcanzado el final del archivo; que una transacción no es válida o que tal empleado no existe. Otro símbolo que se utiliza en los diagramas estructurales es un bucle o loop, que indica que los procedimientos encontrados dentro del loop se repetirán hasta el final. Otro símbolo que se utiliza en los diagramas estructurales es el de un pequeño diamante. El mismo se coloca en la base de uno de los rectángulos y significa que se ejecutará sólo uno de los módulos debajo del diamante.

La planeación y el control son elementos esenciales de cualquier sistema que aspire a tener éxito. Al desarrollar software para los sistemas, la planeación descendente toma un lugar en el diseño, antes de que se inicie la programación. Se necesitan técnicas que auxilien en la definición de los objetivos de los programas, de tal modo que éstos lleguen a concluirse.

Las técnicas descritas en la sección anterior tienen un significado no sólo para usarse en el diseño inicial del software, sino también para su mantenimiento. Se requiere darles mantenimiento constante. El esfuerzo del aseguramiento de la calidad total, requiere que los programas se documenten de modo adecuado.

El software y los procedimientos deben quedar documentados, de manera tal, que sean codificados en un formato de fácil acceso. El acceso a los procedimientos es necesario tanto para el personal de ingreso reciente como para el que conoce el sistema; y asimismo para recordarles a aquellos que lo utilizan de manera esporádica. La documentación permite a los usuarios, programadores y analistas "ver" el sistema, su software y los procedimientos, sin necesidad de una interacción directa.

En sí, cierta documentación proporciona un panorama del sistema, otra que contiene los procedimientos, detalla lo que debe realizarse para operar el software y una más, detalla el código del programa utilizado.



Existen numerosas razones por las cuales los sistemas y los programas quedan pobremente documentados o, simplemente sin documentar; algunos de los problemas residen en los sistemas y los programas en sí, mientras que otros se asocian a los analistas de sistemas y a los programadores.

Algunos sistemas antiguos fueron escritos antes de que las empresas estandarizaran sus técnicas de documentación; sin embargo, todavía se encuentran en uso. Muchos otros sistemas han tolerado modificaciones mayores y menores a lo largo de los años, pero su documentación no ha sido actualizada de modo paralelo. Algunos sistemas, como los de aplicación, fueron adquiridos por sus aplicaciones relevantes, independientemente de que cuentan con una documentación anexa. Los analistas de sistemas pueden dejar de documentar de manera adecuada los sistemas, por carecer de tiempo o de incentivos económicos o por el esfuerzo requerido para hacer la documentación.

Algunos analistas no documentan los sistemas, porque no les agrada hacerlo o consideran que no es un trabajo real. Además, otros son reticentes a documentar sistemas que no fueron desarrollados por ellos mismos, tal vez por prevenir aquellos comentarios que emerjan por incluir material incorrecto acerca de sistemas de otros.

5.2. Técnicas de diseño y documentación

Actualmente no existe una sola técnica sencilla y estandarizada para la documentación y el diseño. A continuación, se exponen diferentes técnicas que se utilizan actualmente. Cada una tiene sus propias ventajas y desventajas, ya que cada una de ellas tiene propiedades exclusivas. Las técnicas anteriormente mencionadas se califican en base a dos atributos: que sean estructuradas y visuales. Una técnica que es altamente estructurada y además visual, es el método HIPO, el cual se explica a continuación, junto con otras técnicas más.

5.2.1. El Método HIPO

HIPO es un método común para desarrollar software de sistemas. HIPO es una abreviatura del nombre en inglés de (*Hierarchical Input Process, Output*) ó (*Jerarquía en el Proceso de Entrada y Salida*) método desarrollado por IBM.

Primero, es una técnica jerárquica, porque el sistema completo de programación se conforma con pequeños subsistemas. Esto soporta un enfoque de diseño descendente y también reduce la complejidad, ya que cada uno de los subcomponentes puede consultarse de manera separada.

Segundo, el acrónimo hace que se consideren las tres partes principales de cualquier sistema: entrada, proceso y salida (*input, process, y output*). Una vez que el diagrama



jerárquico se completa, se elaboran otros diagramas HIPO en páginas divididas verticalmente entre las secciones, donde la sección izquierda corresponde a la entrada, la del centro corresponde al proceso, y la sección de la derecha corresponde a la salida.

HIPO es una técnica visual. El principal beneficio de una técnica visual se deriva de la facilidad de lectura de símbolos estandarizados, utilizados para ilustrar los diferentes tipos de entradas, almacenamiento de datos y dispositivos de salida. (ver figura 5.3).

Existen tres tipos principales de diagramas en los sistemas HIPO:

- 1.- VTOC o Tabla Visual de Contenido (*Visual Table of Contents*)
- 2.- Diagramas generales HIPO (*input/process/output*)
- 3.- Diagramas detallados HIPO

A continuación se expone cada uno de estos tipos:

- ♦ **El VTOC (Visual Table of Contents)**

El VTOC es el diagrama de jerarquías. Proporciona un mapa que permite al lector localizar un módulo del programa existente dentro del sistema principal. El diagrama jerárquico del VTOC parece ser similar a un típico diagrama de la estructura de una organización; tomando la forma de una pirámide. Debajo del diagrama hay un espacio para una descripción más detallada de los cuadros.

- ♦ **Diagrama general HIPO**

El siguiente tipo de diagrama en el sistema HIPO permite una visión global de la entrada, el proceso y la salida, y en consecuencia se refiere como diagrama panorámico. En este punto, es útil listar todas las entradas, los procesos y salidas en las tres secciones de papel sin dibujar símbolos especializados.

- ♦ **Diagramas detallados HIPO**

Con el fin de contar con algo más útil, los diagramas generales deben descomponerse en cada uno de los módulos autocontenidos en él. En este punto sí conviene utilizar símbolos para los elementos de entrada y de salida. Se utilizan símbolos de disco para indicar como se almacenan los archivos.

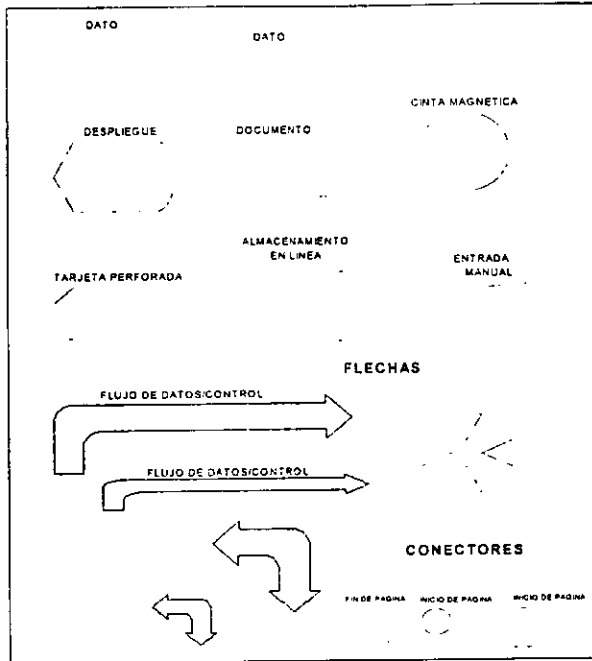


Figura. 5.3 Sismología del Método HIPO.³

HIPO es una técnica estructurada altamente visual para el diseño y la documentación. Una vez que los analistas se familiarizan con los símbolos que utiliza, el HIPO se convierte en un instrumento muy valioso. Sin embargo, pocas veces los demás miembros de la organización están familiarizados con tales símbolos, de tal forma que el HIPO se vuelve un instrumento demasiado especializado para explicar la operación de un sistema o programa.

El HIPO requiere de una considerable cantidad de espacio gráfico. Con el fin de ver todo el programa completo, son necesarias varias páginas. Y muchas páginas hacen que el lector se pierda. Los diferentes niveles de los diagramas también ocupan espacio; y en ocasiones es difícil seguir el flujo del programa.

Sin embargo, el HIPO es muy útil para la documentación de los programas. Sirve como un elemento recordatorio para el autor del programa, a quien ubica con rapidez después de un largo periodo. Otros programadores que entiendan los símbolos estandarizados, también apreciarán el valor de los diagramas HIPO.

³ Kenneth E. Kendall - Julie E. Kendall. 1991. *Análisis y Diseño de Sistemas*. Prentice-Hall, Hispanoamérica, S.A., México. Pág. 755

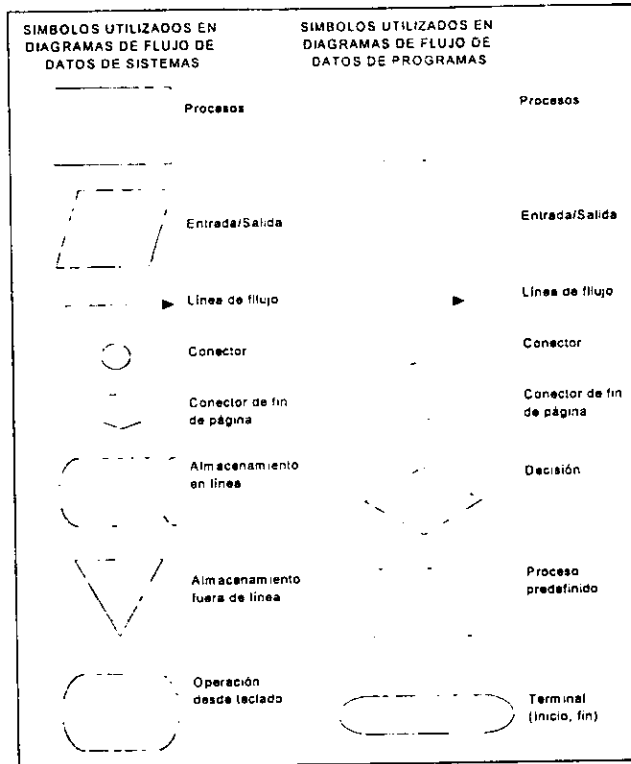


Figura. 5.5 *Sismología de Diagramas de Flujo.*⁵

Es probable que la mejor razón para utilizar un diagrama de flujo es que han sido utilizados históricamente y quienes han sido promovidos dentro de una compañía, con el tiempo llegan a entenderlo mejor que cualquier otra técnica más reciente. Si esto es una consideración importante, los diagramas de flujo ordinarios deberán elegirse como la técnica de documentación más adecuada.

5.2.3.- Diagramas Nassi-Schneiderman

Un enfoque más estructurado, pero tal vez menos visual para el diseño y la documentación es el diagrama Nassi-Schneiderman (N-S). La principal ventaja de un diagrama N-S es que adopta la filosofía de la programación estructurada. Segundo, utiliza un número limitado de símbolos, de tal forma que el diagrama ocupa menos

⁵ Kenneth E. Kendall - Julie E. Kendall. 1991. *Análisis y Diseño de Sistemas*. Prentice-Hall, Hispanoamérica, S.A., México. Pág. 760

5.2.2.- Diagramas De Flujo

Otro enfoque visual, pero más estructurado para el diseño y la documentación de los programas es el diagrama de flujo ordinario. En la figura 5.3 se pueden encontrar ejemplos de símbolos utilizados para documentar tanto los sistemas como los programas.

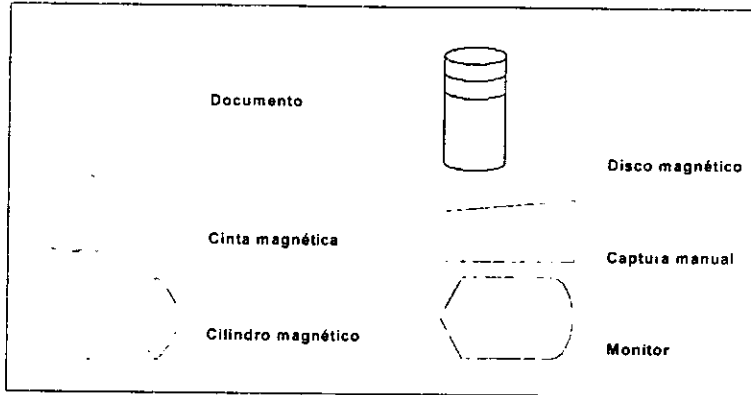


Figura. 5.4 Símbología de los Diagramas de Flujo.⁴

Observe que ciertos símbolos son comunes para sistemas y programas. En la figura 5.4 se utilizan ejemplos más particulares de los símbolos para identificar la forma de entrada (documento en papel, cinta magnética, tambor, disco, captura manual y monitor VDT). El diagrama de flujo es lineal. Además no se les otorga demasiada atención a la entrada y a la salida. Existen numerosas desventajas en el uso de los diagramas ordinarios de flujo: no se elaboran con base en los principios de la programación estructurada, de tal forma que ilustran el flujo del programa, pero no su estructura. De forma similar al HIPO, requieren de un espacio considerable, de tal modo que el lector tiene que pasar por varias páginas para asimilar todo el contenido del programa. Cuenta con demasiadas ramificaciones, cada una de ellas proveniente de cada decisión. En los diagramas de flujo, cada autor utiliza un estilo particular; y en consecuencia, un autor tendrá dificultades para leer el diagrama de flujo de otro autor. También se tiene toda una lista extensa de símbolos que comprender y memorizar, lo cual hace difícil la difusión de los diagramas de flujo.

⁴ Kenneth E. Kendall - Julie E. Kendall, 1991. *Análisis y Diseño de Sistemas*. Prentice-Hall, Hispanoamérica, S.A., México. Pág. 761



espacio y puede leerse con cierta facilidad por la gente poco familiarizada con símbolos ajenos a los de los diagramas de flujo.

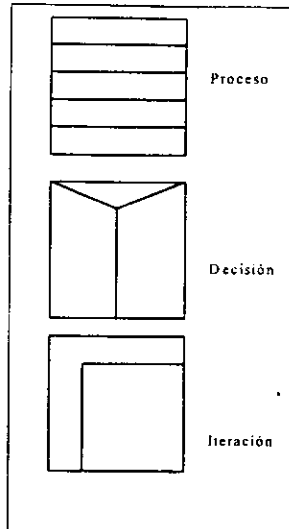


Figura. 5.6 Sismología de los Diagramas Nassi-Schneiderman.⁶

La figura 5.6 muestra los tres símbolos básicos que se utilizan en un diagrama Nassi-Schneiderman. El primero es un cuadro, que sirve para representar cualquier proceso del programa. El segundo símbolo, que es una columna dividida por un triángulo incorporado, representa una decisión. La forma más sencilla de una decisión, "verdadera" o "falsa", se muestra en este ejemplo, pero cualquier forma de una decisión, incluyendo distintas condiciones alternativas, pueden ilustrarse mediante el uso de este símbolo. El tercero es un cuadro dentro de otro, que se utiliza para indicar que se lleva a cabo una interacción. El cuadro dentro del cuadro también aparece como una sangría en el diagrama global.

En la programación estructurada se utiliza un enfoque descendente. El analista comenzará dibujando primero las principales subrutinas y luego hará una sangría para completar más adelante las rutinas internas.

Los diagramas Nassi-Schneiderman deben estar completos y ser muy claros, con el fin de que se entiendan. Esto es una desventaja en comparación con otros métodos, ya que el temor de que queden incompletos podría impedir que el analista y el programador utilicen este tipo de diagramas. Si de manera regular se hacen cambios, los diagramas Nassi-Schneiderman no serán apropiados, ya que deberán dibujarse nuevamente; y en consecuencia, su modificación no es sencilla. Las ventajas de este tipo de diagramas

⁶ Kenneth E. Kendall - Julie E. Kendall, 1991, *Análisis y Diseño de Sistemas*, Prentice-Hall, Hispanoamérica, S.A., México, Pág. 764



son varias, por ejemplo, brindan al programador un instrumento de diseño que es compatible con la programación estructurada; son fáciles de leer porque no poseen símbolos complejos y no ocupan mucho espacio.

5.2.4.- Diagramas Warnier-Orr

Cuentan con un enfoque jerárquico o descendente. Este tipo de diagrama no es tan visual como el de Nassi-Schneiderman, pues las llaves son los únicos símbolos que utiliza. En la Figura 5.7 se muestran las llaves y otras notaciones de estos diagramas. Las llaves se utilizan para representar conjuntos y subconjuntos, y las variables tales como M y N se utilizan para representar el número de casos de una interacción. Cuando hay una condición, ya sea que se cumpla o no, se utiliza la notación (0,1) y un + significa que la lista de artículos son las posibles alternativas. Se utiliza PERFORM para dirigirse a otra parte del programa.

El lado izquierdo del diagrama Warnier-Orr representa una visión general del sistema y el analista se desplaza de izquierda a derecha, conforme se descompone el sistema en pequeños subsistemas. El desarrollo de los diagramas Warnier-Orr es único, porque una vez que se define la estructura general, el analista comienza con la salida y trabaja hacia atrás. A diferencia del diagrama Nassi-Schneiderman, en este diagrama, se debe dejar suficiente espacio para incorporar las modificaciones necesarias.

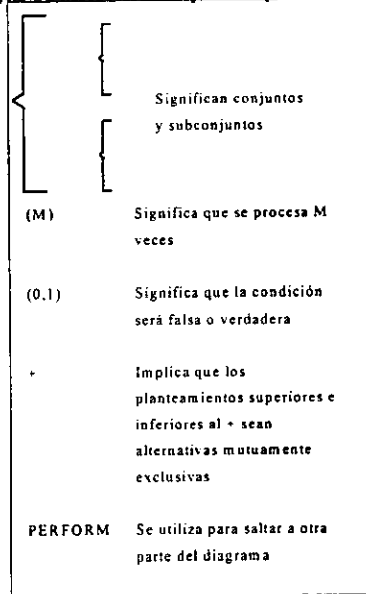


Figura. 5.7 Símbología de los Diagramas Warnier-Orr.⁷

⁷ Kenneth E. Kendall - Julie E. Kendall, 1991. *Análisis y Diseño de Sistemas*. Prentice-Hall, Hispanoamérica, S.A., México. Pág. 764



5.2.5.- Pseudocódigo

Es el español estructurado, no se trata de un tipo particular de código de programación, se utiliza como un paso intermedio en el desarrollo de tal código de programación. Consta de palabras que se asemejan a un lenguaje de programación, solo que en nuestro idioma. Por ejemplo, para formar una instrucción If...Then...Else usamos los mnemónicos Si...Entonces...O sino.

5.2.6.- Manuales de procedimientos

Los manuales de procedimientos son documentos de carácter organizacional muy comunes, con los cuales la mayoría de las personas han tenido contacto. Ellos son el componente en español de la documentación, aunque también contienen códigos de programación, diagramas de flujo, etc. Los manuales se usan para comunicarse con quienes usarán los sistemas. Pueden contener comentarios introductorios; pasos para realizar diferentes operaciones; instrucciones de cómo resolver problemas de operación y qué hacer cuando algo no funciona.

Los manuales deben ser redactados con un enfoque directo y estandarizado, deben estar actualizados para poder ser más útiles y no deben contener textos retóricos referentes a los beneficios de la aplicación, del contenido del manual o de lo que documenta el manual. Además de las anteriores características, se deben perseguir cuatro objetivos en la elaboración de un manual:

- A.- Que posea una buena organización de su contenido.
- B.- Que se pueda encontrar la información requerida de forma sencilla.
- C.- Que resuelva el mayor número de dudas que sea posible.
- D.- Que esté redactado con claridad.

Además de la organización del manual y de su claridad, debe dedicarse especial atención al tipo de gente a quien va dirigido el mismo.

5.2.7.- El Método Folklore

Es una técnica de documentación de sistemas, que ha sido creada para complementar algunas de las técnicas que se han descrito. Aún con la plétora de técnicas disponibles, muchos sistemas quedan escasa e inadecuadamente documentados. La técnica del FOLKLORE recopila la información que comparten los usuarios, pero que rara vez queda plasmada en un escrito.

El FOLKLORE es una técnica sistemática, basada en métodos tradicionales, que han sido utilizados para recopilar las costumbres de las personas. Este enfoque de



documentación de sistemas requiere que el analista entreviste a los usuarios, realice investigación sobre la documentación existente en los archivos, y observe el procesamiento de la información. El objetivo es recopilar aquella información que se encuentre en cualquiera de las siguientes cuatro categorías: costumbres, cuentos, expresiones y elementos artísticos.

Al documentar costumbres y hábitos, se logra que los programas funcionen sin problemas. Como ejemplo de costumbres se tienen estos casos: "normalmente, toma dos días actualizar el registro mensual, porque la tarea es demasiado larga"; "se presentan las cuentas comerciales en un solo día y almacenamos las demás para el día siguiente".

Los cuentos son historias que los usuarios practican con referencias a su manera de ver el sistema. La precisión del cuento, por supuesto, dependerá de la memoria del usuario; y, en el mejor de los casos, no deja de ser una opinión acerca de cómo funciona el programa.

Las anécdotas o historias normalmente tienen un principio, una parte intermedia y un final. En este caso, existe la historia sobre un problema (el principio), una descripción de los efectos (la parte central) y la solución (final).

Las expresiones son breves planteamientos que representan generalizaciones o recomendaciones. A los usuarios les gusta dar recomendaciones y el analista debe tratar de capturar estas sugerencias e incluirlas en la documentación FOLKLORE.

La recopilación de manifestaciones artísticas es otra actividad importante del folklorista tradicional y que el analista de sistemas no debe pasar por alto. Los diagramas de flujo, las figuras y las tablas que desarrollan los usuarios, en ocasiones, llegan a ser mejores o más útiles que los mismos diagramas dibujados por el autor original del sistema. Con frecuencia, los analistas los encontrarán en los tableros de avisos o en los pizarrones; o bien, les pedirán a los usuarios que limpien sus archivos y sólo dejen aquellos diagramas de flujo que consideren que son de utilidad.

El enfoque de FOLKLORE funciona porque puede auxiliar la falta de conocimiento creada cuando un autor se retira. Los contribuyentes del documento FOLKLORE no tienen porque documentar todo el sistema, sólo aquellas partes que conozcan. Finalmente, es conveniente que los usuarios contribuyan, complementando las tareas del analista.

El riesgo de confiar en FOLKLORE es que la información recopilada por los usuarios puede no ser objetiva, parcialmente correcta o peor aún, incorrecta. Sin embargo, al menos que alguien tome la tarea de reescribir la documentación del programa, la descripción de costumbres, cuentos, expresiones o formas artísticas, puede ser la única información escrita referente a como opera el programa.



5.3.- Interfaz con el Usuario

Se han encontrado "interfaces" difíciles de aprender, complicadas de utilizar, confusas y en muchos casos, totalmente frustrantes.

El diseño de la interfaz de usuario es un tema que se ha hecho cada vez más importante, a medida que ha aumentado el uso de las computadoras. Desde el punto de vista del usuario, lo que permite a un piloto manejar un avión moderno, a un radiólogo interpretar la salida de un scanner o a un banquero transferir millones de dólares a través de continentes, es la interfaz. Esta es, en muchos sentidos, el "envoltorio" del software de computadora. Si es fácil de aprender, simple de utilizar, directo y no muy estricto, el usuario podrá hacer buen uso de lo que hay dentro. Si, por el contrario, no tiene ninguna de estas características, aparecerán problemas invariablemente.

De un buen diseño de la interfaz, depende si el usuario se adaptará a un ritmo normal de interacción. El o ella pueden incluso olvidar que se está llevando a cabo una comunicación con la máquina. Pero si no lo es, el usuario lo sabrá inmediatamente y se sentirá a disgusto con un modo de interacción poco amigable.

El diseño de la interfaz de usuario tiene tanto que ver con el estudio de la gente como con aspectos de la tecnología. ¿Quién es el usuario? ¿Cómo aprende el usuario a interaccionar con un sistema nuevo basado en computadora? ¿Qué espera el usuario del sistema? Estas son sólo unas pocas de las muchas preguntas que deben ser planteadas y respondidas como parte del diseño de la interfaz de usuario.

5.3.1.- Factores Humanos

En un nivel fundamental, se debe comprender la percepción visual, la psicología cognitiva de la lectura, la memoria humana y el razonamiento deductivo e inductivo.

En otro nivel, debemos comprender al usuario y su comportamiento. Finalmente, debemos entender las tareas que el sistema basado en software realiza para el usuario y las tareas que pide el usuario como parte de la interacción entre el hombre y la máquina.

La interfaz de usuario es el mecanismo a través del cual se establece un diálogo entre el programa y el humano. Si se han tenido en cuenta los factores humanos, el diálogo será fluido y se establecerá un ritmo entre el usuario y el programa. Si estos factores se han ignorado, el sistema será casi siempre visto como "poco amigable".



5.3.2.- Estilos de interacción entre hombre y máquina

Los estilos de interacción hombre-máquina abarcan una gran variedad de opciones que están íntimamente ligadas a la evolución de las computadoras. Las opciones para el estilo de interacción han aumentado a medida que el hardware se ha ido haciendo más sofisticado. Aún así, en muchos casos, modernos sistemas basados en computadora siguen utilizando estilos de interacción que fueron diseñados originalmente para entornos de hardware que han quedado obsoletos hace 20 años, la interfaz de preguntas y órdenes (comunicación textual).

Aunque estas cadenas de órdenes y preguntas antiguas eran concisas, también eran propensas a errores, muy estrictas (si se cometía un error) y bastante difíciles de aprender.

Una variante (y a menudo una mejora) de la interfaz de órdenes y preguntas es la interfaz de menú simple. Un elemental menú ofrece al usuario un contexto global y es menos dado a errores que el formato de línea de órdenes, pero su uso puede llegar a ser tedioso. Por ejemplo: si en un menú hubiera opciones adicionales (más menús), el usuario no podría ir directamente a otra opción, sino que tiene que pasar por cada nivel de menú hasta alcanzar la opción deseada. Esto puede ser frustrante e ineficiente.

A medida que el hardware se ha hecho más eficiente y los ingenieros de software han aprendido más sobre los factores humanos y su impacto en el diseño de la interfaz, han ido apareciendo en las modernas interfaces orientadas a ventanas con opción de señalar y elegir, lo cual ofrece al usuario un gran número de ventajas:

1. Se pueden visualizar diferentes tipos de información simultáneamente, permitiendo al usuario cambiar de contexto (p. ej.: para realizar escritos en una ventana, hacer cálculos en otra, correr un programa en una tercera) sin perder conexión visual con otros trabajos. Las ventanas permiten al usuario, realizar muchas tareas cognitivas y de comunicación.
2. El esquema de menús desplegados permite realizar muchas tareas interactivas diferentes. Estos menús permiten al usuario realizar tareas de control y diálogo de forma sencilla.
3. La utilización de iconos gráficos, menús desplegados, botones y técnicas de presentación continua reducen el número de pulsaciones en el teclado. Esto puede suponer un aumento en la eficiencia de aquellos con poca experiencia en mecanografía.

La generación actual de IHM, une todos los atributos de las interfaces de tercera generación con el hipertexto y con la multitarea -la habilidad para realizar diferentes tareas simultáneamente (desde el punto de vista del usuario).



Cada una de las interfaces que se han descrito pueden encontrarse en cualquier área de aplicación. No hay duda de que la tendencia actual son las interfaces con multitarea, ventanas, iconos, menús y dispositivo de señalización.

5.3.3.- Diseño de la interfaz hombre-máquina

El diseño de la interfaz hombre-máquina es un elemento de un tema más amplio que se ha denominado diseño de software. Los métodos de diseño para la IHM no se usan extensivamente, aunque existen técnicas de especificación gráficas y basadas en lenguajes para interfaces de usuario.

El proceso global de diseño de una interfaz de usuario comienza con la creación de diferentes modelos de funcionamiento del sistema (tal y como se percibe exteriormente). A continuación, se especifican las tareas humanas y las orientadas a la computadora, necesarias para el funcionamiento del sistema, se consideran los aspectos de diseño que han de aplicarse a todos los diseños de interfaces, se utilizan herramientas para realizar prototipos y finalmente, se implementa el modelo de diseño evaluándose el resultado bajo criterios de calidad.

5.3.4.- Aspectos de Diseño

Según evoluciona el diseño de una interfaz de usuario, casi siempre aparecen cuatro aspectos de diseño comunes: tiempo de respuesta del sistema, facilidades de ayuda al usuario, manejo de la información de error y asignación de nombres a las órdenes. Desafortunadamente, muchos diseñadores no tienen en cuenta estos aspectos hasta que es demasiado tarde en el proceso de diseño. Esto suele dar lugar a revisiones innecesarias, retrasos del proyecto y frustración del cliente. Es mucho mejor definir estos puntos como aspectos del diseño y considerarlos desde el comienzo del mismo, cuando todavía no es difícil hacer cambios y los costos son bajos.

El tiempo de respuesta del sistema es el punto más crítico de muchos sistemas interactivos (particularmente en aplicaciones de red, ya que tienen tiempo compartido). En general, el tiempo de respuesta del sistema se mide desde que el usuario realiza una acción de control hasta que el programa responde con la acción o salida deseada.

El tiempo de respuesta del sistema tiene dos características importantes: *el retardo* y *la variabilidad*. Si el retardo de la respuesta es demasiado grande los resultados son, inevitablemente, la frustración y el stress del usuario. Sin embargo, un retardo muy pequeño puede ser perjudicial cuando el usuario es conducido por la interfaz. Una respuesta rápida puede forzar al usuario a ir muy aprisa y cometer errores.

La "variabilidad" se refiere a la desviación del tiempo de respuesta medio y en muchos sentidos supone la característica más importante de todas las relacionadas con el



tiempo de respuesta. Una baja variabilidad permite establecer un ritmo propio de interacción pero con un retardo alto, el usuario puede estar desconcertado preguntándose si ha ocurrido algo "diferente" a lo esperado.

Casi todos los usuarios de un sistema interactivo basado en computadora, requieren ayuda de vez en cuando. En muchos casos, los sistemas interactivos modernos ofrecen facilidades de ayuda interactiva que permiten al usuario obtener una respuesta sin abandonar la interfaz.

Normalmente, se encuentran dos tipos diferentes de facilidades de ayuda: *integrada* y *añadida*. La *ayuda integrada* se diseña en el software desde el principio. A menudo es sensible al contexto, permitiendo al usuario seleccionar aquellos temas relacionados con las acciones que está ejecutando en ese momento. Obviamente, esto reduce el tiempo requerido por el usuario para obtener ayuda e incrementa la "amigabilidad" de la interfaz. Una *ayuda añadida* es aquella que se incorpora al sistema después de haber sido construido. En muchos casos, es un manual de usuario interactivo con un conjunto limitado de tipos de preguntas. El usuario puede tener que buscar a través de una lista de cientos de temas para encontrar la respuesta apropiada, cometiendo frecuentes errores de elección y recibiendo mucha información irrelevante. En general la ayuda integrada es preferible a la añadida.

Cuando se considera un sistema de ayuda para el usuario, deben tenerse en cuenta varios aspectos de diseño:

- ¿Estará la ayuda disponible en todas las funciones del sistema y en todo momento durante la interacción con el sistema? Las opciones son: ayuda únicamente para un subconjunto de todas las funciones y acciones o ayuda para todas las funciones.
- ¿Cómo podrá solicitar ayuda el usuario? Las opciones son: un menú de ayuda, una función especial de ayuda o una orden AYUDA.
- ¿Cómo se presentará la ayuda? Las opciones son: una ventana separada, una referencia a un documento impreso (poco recomendable), una sugerencia de una o dos líneas generadas en una posición fija de la pantalla.
- ¿Cómo volverá el usuario a la interacción normal? Las opciones son: una tecla de retorno que aparece en la pantalla, una tecla de función o una secuencia de control.
- ¿Cómo se estructurará la información de ayuda? Las opciones son: una estructura "plana" en la cual la información es accesible mediante una palabra clave, una jerarquía en capas que suministra información más detallada a medida que el usuario avanza por la estructura o utilización de hipertexto.

Los mensajes de error y avisos constituyen "malas noticias" para los usuarios de sistemas interactivos que indican que algo ha ido mal.

En general, todos los mensajes de error o de aviso producidos por un sistema interactivo deberían tener las siguientes características:

- El mensaje debe describir el problema en un lenguaje que comprenda el usuario.



- El mensaje debe proporcionar una información constructiva para poder solventar el problema.
- El mensaje debe indicar las consecuencias negativas del error.
- El mensaje debe ir acompañado de una clave visual o audible. Esto es, puede generarse un pitido acompañando la visualización de un mensaje, o bien éste puede parpadear momentáneamente y ser visualizado en un color fácilmente reconocible como "color de error".
- El mensaje no debe aportar un juicio sobre lo ocurrido. Esto es, no debe culpar al usuario.

Ya que a nadie le gustan las malas noticias, pocos usuarios disfrutarán de un mensaje de error. Pero una filosofía efectiva de mensajes de error puede hacer mucho para mejorar la calidad de un sistema interactivo y reducirá significativamente la frustración del usuario cuando aparezcan problemas.

Uno de los modos más usados de interacción entre el usuario y el sistema ha sido la introducción de órdenes por medio del teclado. Hoy en día, la utilización de interfaces orientadas a ventanas con utilización de dispositivos de señalización y elección, ha reducido la dependencia del teclado, pero muchos usuarios siguen prefiriendo un modo de interacción orientado a órdenes.

Cuando el sistema va a proporcionar un modo de interacción basado en órdenes tecladas, aparece una serie de aspectos de diseño que se deben tener en cuenta:

- ¿Debe disponer cada opción de menú de una orden teclada correspondiente?
- ¿Qué formato deben tener las órdenes? Las opciones son: una secuencia de control (p.ej.: ^P), teclas de función, una palabra completa.
- ¿Será complicado aprender y recordar las órdenes? ¿Qué puede hacerse si se olvida una orden?
- ¿Puede el usuario personalizar o abreviar las órdenes?

En un número cada vez mayor de aplicaciones, en lugar de escribir cada orden individualmente (y repetidamente), se escribe la macro de órdenes y todas las órdenes asociadas a ella se ejecutan en secuencia. Todas las aplicaciones deberían seguir los mismos convenios sobre el uso de órdenes. Por ejemplo, en una aplicación determinada se pulse (^P) para duplicar un objeto gráfico y que en otra aplicación distinta se use (^P) para borrar un objeto gráfico. La posibilidad de que ocurra un error es obvia.

5.3.5.- Evaluación del Diseño

Una vez que se ha creado una versión operativa de la interfaz, ésta debe ser evaluada para determinar si satisface las necesidades del usuario.



El ciclo de evaluación de la interfaz del usuario tiene la forma que se muestra en la figura:

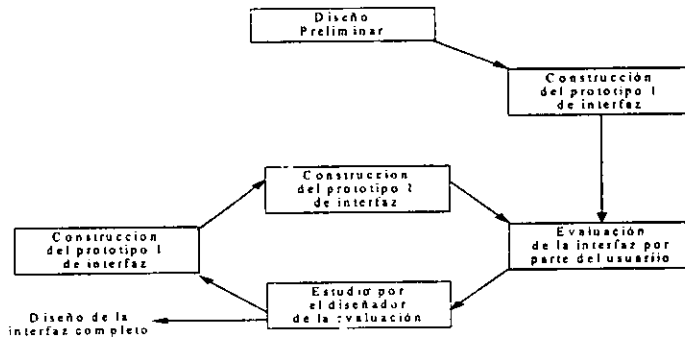


Figura. 5.8 Ciclo de Evolución de la Interfaz del usuario.⁸

Después de completar un diseño preliminar, se crea un prototipo de primer nivel. El prototipo es evaluado por el usuario, quien proporciona al diseñador comentarios directos acerca de la eficiencia de la interfaz. Además, si se utilizan técnicas de evaluación formales (p. ej.: cuestionarios, hojas de evaluación), el diseñador puede extraer información de esos cuestionarios (p. ej.: al 80% de los usuarios no le gustó el mecanismo para almacenar la información). Las modificaciones del diseño se realizan en base a la información del usuario, tras lo que se desarrolla el siguiente prototipo. El ciclo de evaluación continúa hasta que no es necesario hacer más modificaciones sobre el diseño de la interfaz. Pero, ¿es posible evaluar la calidad de una interfaz de usuario antes de que esté construido el prototipo? Si pueden descubrirse y corregirse problemas potenciales a tiempo, se conseguirá reducir el número de veces que habrá que recorrer el ciclo de evaluación y el tiempo de desarrollo del prototipo.

Si se ha creado un modelo de diseño de la interfaz se pueden aplicar en las primeras revisiones, varios criterios de evaluación:

1. La longitud y complejidad de la especificación escrita del sistema y de su interfaz proporciona una indicación de la cantidad de aprendizaje requerido por los usuarios.
2. El número de órdenes especificado y el número medio de argumentos por orden, proporcionan una indicación del tiempo de interacción y de la eficiencia global del sistema.

⁸ Charles I. Biggs - Evan G Birks. 1980. *Managing the System Development Process*. Prentice-Hall Inc., Englewood Cliffs, N.J. pag. 113



3. El número de acciones, órdenes y estados del sistema, indicados en el modelo de diseño, indican la carga de memorización de los usuarios del sistema.
4. El estilo de la interfaz, las facilidades de ayuda y el protocolo de manejo de errores proporcionan una indicación general de la complejidad de la interfaz y del grado de aceptación por parte de los usuarios.

Una vez que se ha construido el primer prototipo, se puede recoger una gran variedad de datos cuantitativos y cualitativos que ayudarán en la evaluación de la interfaz.

Si se desea obtener datos cuantitativos, se puede utilizar un tipo de análisis basado en la observación del usuario durante un cierto tiempo. Estos datos pueden utilizarse como guía para la modificación de la interfaz.

5.3.6. Directrices para el diseño de interfaces

El diseño de interfaces de usuario recae fuertemente en la experiencia del diseñador y en la experiencia anecdótica que aparece en cientos de artículos técnicos y docenas de libros. Se sugieren tres categorías de directrices para el diseño de IHM: interacción general, visualización de información y entrada de datos.

5.3.6.1.- Interacción General

Las directrices para la interacción general sobrepasan a menudo las fronteras que las separan de la visualización de información, la entrada de datos y el control global del sistema. Son por lo tanto, complementarias y se corre un gran riesgo ignorándolas. Las siguientes directrices se centran en la interacción general:

Ser consistente. Se debe utilizar un formato consistente para la selección de menús, la entrada de órdenes, la visualización de datos y cientos de otras funciones que incorpora una IHM.

Ofrecer una realimentación significativa. Se debe proporcionar al usuario una realimentación visual y auditiva para asegurar que se establece una comunicación bidireccional (entre el usuario y la interfaz).

Preguntar por la verificación de cualquier acción destructiva no trivial. Si un usuario pide que se borre un archivo, indica que se escriba sobre información ya existente o desea concluir la ejecución de un programa, debe aparecer un mensaje del tipo "¿Está seguro...?".

Permitir una vuelta atrás fácil en la ejecución de la mayoría de las acciones. Las funciones DESHACER o INVERTIR han salvado a decenas de miles de usuarios finales de millones de horas de frustración. La vuelta atrás debe estar disponible en todas las aplicaciones interactivas.

Reducir la cantidad de información que debe ser memorizada entre acciones. No se debe esperar que el usuario recuerde una lista de números o nombres para que pueda



reutilizarlos como entrada a la siguiente función. La carga de memorización debe ser minimizada.

Buscar la eficiencia en el diálogo el movimiento y el pensamiento. El número de pulsaciones debe ser minimizado, la distancia que un ratón debe recorrer entre dos pulsaciones también debe ser tomada en cuenta al diseñar el formato de presentación y el usuario debe encontrar pocas veces una situación en la que tenga que preguntar "¿Qué significa esto?".

Perdonar los errores. El sistema debe protegerse de los errores del usuario que pudiesen afectarle causándole un fallo.

Categorizar las actividades en base a su función y organizar la geografía de la pantalla convenientemente. Uno de los beneficios claves del menú desplegable es su habilidad para organizar las órdenes de acuerdo a su tipo. En esencia, el diseñador debe hacer hincapié en la ubicación coherente de órdenes y acciones.

Proporcionar facilidades de ayuda sensibles al contexto.

Utilizar verbos de acción simples o frases verbales cortas para nombrar las órdenes. Un nombre de orden largo es más difícil de reconocer y de recordar. También ocupa un espacio innecesario en las listas de menús.

5.3.6.2.- Visualización de la información

Si la información que presenta la IHM es incompleta, ambigua o incomprensible, la aplicación no satisfará las necesidades del usuario. Las siguientes directrices se centran en la visualización de la información:

Mostrar sólo aquella información que sea relevante en el contexto actual. El usuario no debe tener que buscar a través de datos extraños, menús y gráficos para obtener información relevante a una función concreta del sistema.

No abrumar al usuario con datos, utilizar un formato de presentación que permita una asimilación rápida de la información. Los gráficos o esquemas deben reemplazar a las tablas de datos.

Utilizar etiquetas consistentes, abreviaciones estándar y colores predecibles. El significado de una pantalla debe ser obvio, evitando referencias a otras fuentes externas de información.

Permitir al usuario mantener el contexto visual. Si se están reescalando gráficos en una pantalla, debe mantenerse la imagen original (en forma reducida en la esquina de la pantalla) de tal forma que el usuario no pierda la posición relativa de la porción de imagen que está observando.

Producir mensajes de error significativos.



Utilizar mayúsculas y minúsculas, tabulaciones y agrupaciones de texto para ayudar a la comprensión. La mayoría de la información visualizada en una IHM es textual, por lo que la presentación y formato del texto tiene una incidencia importante en la facilidad con que la información es asimilada por el usuario.

Utilizar ventanas (si están disponibles) para modularizar los diferentes tipos de información. Las ventanas mantienen accesibles al usuario muchos tipos diferentes de información.

Utilizar representaciones "analógicas" para mostrar la información que es más fácil de asimilar bajo este tipo de representación.

Considerar la geografía disponible en la pantalla y utilizarla eficientemente. Cuando se utilizan muchas ventanas, debe haber espacio disponible para mostrar al menos, una porción de cada una.

5.3.6.3.- Entrada de datos

El usuario emplea la mayor parte de su tiempo seleccionando órdenes e introduciendo datos. El teclado sigue siendo el medio de entrada de datos más importante, pero el ratón, los scanners e incluso los sistemas de reconocimiento de voz ya son alternativas eficientes. Los siguientes puntos se centran en la entrada de datos:

Minimizar el número de acciones de entrada de datos que debe realizar el usuario. Esto se puede conseguir utilizando el ratón para seleccionar un elemento de entre un conjunto predefinido utilizando listas de datos, utilizando macros.

Mantener la consistencia entre la información visualizada y los datos de entrada.

Permitir al usuario personalizar la entrada de datos. Un usuario experto podría decidir crear órdenes personalizadas y respuestas con determinados mensajes.

La interacción también debe ser flexible y estar ajustada al modelo de entrada preferido por el usuario. Un usuario puede encontrarse cómodo utilizando el teclado, mientras otro prefiere el ratón.

Desactivar órdenes que sean inapropiadas en el contexto actual. Esto evita que el usuario realice acciones que podrían producir un error.

Permitir al usuario controlar el flujo interactivo. El usuario debe poder evitar acciones innecesarias, cambiar el orden de acciones requeridas y salir de situaciones de error sin tener que abandonar el programa.



Proporcionar ayuda en todas las acciones de entrada de datos.

Eliminar las entradas innecesarias. Proporcionar valores por default y en lo posible ayudar al usuario con la información calculada.

5.3.7. Estándares de interfaz

Hoy en día las aplicaciones de software tienden a desarrollar interfaces con ventanas y dispositivos de señalización y selección. La creación de estas interfaces no es una tarea sencilla. Por esta razón, se están adoptando estándares de interfaces.

Se pueden utilizar las herramientas SDIU⁹ para crear un prototipo de interfaz de usuario en mucho menos tiempo del requerido para desarrollarlo partiendo de cero. El usuario se familiariza con la presentación y el ritmo de la IHM y de esta forma, aprende mucho antes cualquier aplicación nueva que utilice la misma interfaz estándar. Después de un tiempo, la utilización de la interfaz se hace intuitiva y mucho más productiva para el usuario final.

Aunque hay en desarrollo varios estándares de interfaz, el que más se sigue es el "Sistema X Windows". El sistema Windows define una sintaxis y una semántica para el diseño de IHM y proporciona herramientas para la creación de pantallas, ventanas y gráficos, así como un protocolo para manejar recursos, interacciones con dispositivos y manejo de excepciones. Se han desarrollado varias extensiones y variaciones del sistema X Windows estándar que se pueden utilizar sobre PC y estaciones de trabajo bajo UNIX y otros sistemas operativos.

En resumen la interfaz hombre-máquina, comúnmente denominada "interfaz de usuario", es la puerta hacia aplicaciones de software interactivas. El diseño de interfaces hombre-máquina requiere el conocimiento tanto de factores humanos como tecnológicos. La percepción humana, el nivel de preparación y el perfil de comportamiento del usuario, así como el resto de las tareas que realiza el usuario, son factores que influyen en el diseño de una interfaz. Además, el estilo de la interfaz, la disponibilidad del hardware, la tecnología del software y la aplicación misma, son piezas clave del resultado final.

El diseño de IHM supone la creación de cuatro tipos diferentes de modelos: el modelo de diseño, el modelo de usuario, la percepción del sistema y la imagen del sistema. Cada uno representa la interfaz desde una perspectiva diferente y todos deben ser utilizados en el proceso de diseño. Para desarrollar una interfaz satisfactoria, todos deben converger en una única representación del sistema.

5.4.- Definición de Módulos

El concepto de modularidad es la división de los componentes del sistema y son los nombres y ubicaciones determinados, que se denominan módulos y que se integran

⁹SDIU: Sistemas de Desarrollo de Interfaces de Usuario. Véase 6.3.6.4.



para satisfacer los requerimientos del problema. Se dice que la "modularidad es el atributo individual que permite a un programa ser intelectualmente manejable". Esto nos lleva a una conclusión del tipo "divide y vencerás" es más fácil resolver un problema complejo cuando se divide en trozos donde se pueden manipular en forma más factible. Hay situaciones (p. ej.: software de tiempo real, software de microprocesadores) en las que son inaceptables la menor velocidad relativa y el exceso de memoria derivados del uso de subprogramas (p. ej.: subrutinas procedimiento). En tales situaciones, el software puede y debe ser diseñado considerando la modularidad como filosofía principal.

Un diseño modular reduce la complejidad, facilita los cambios (un aspecto crítico de la facilidad del mantenimiento del software) y produce como resultado una implementación más sencilla, permitiendo el desarrollo de las distintas partes de un sistema.

Tipos de Módulos

Para definir módulos en una arquitectura de software se utiliza la abstracción y el ocultamiento de información. Ambos atributos deben ser traducidos a características operativas del módulo, caracterizadas por el historial de incorporación, el mecanismo de activación y el camino de control.

El *historial de incorporación* se refiere al momento en el que se incluye el módulo en la descripción del software en lenguaje fuente. Por ejemplo, un módulo definido como macro de tiempo de compilación es incluido por el compilador, mediante la inserción de su código, al encontrar una referencia en el código creado por el desarrollador. Un subprograma convencional (p. ej: una subrutina o un procedimiento) es incluido mediante la generación de código de bifurcación y enlace.

Existen dos *mecanismos de activación*. Convencionalmente, un módulo es invocado mediante *referencia* (p. ej: una sentencia "de llamada"). Sin embargo, en las aplicaciones de tiempo real, un módulo puede ser invocado mediante una *interrupción*, esto es, un suceso exterior produce una discontinuidad en el procesamiento que da como resultado el paso del control a otro módulo. Los mecanismos de activación son importantes porque pueden afectar a la estructura del programa.

El camino de control de un módulo describe la forma en la que se ejecuta internamente. Los módulos convencionales tienen una única entrada y una salida y ejecutan secuencialmente una tarea. Algunas veces se necesitan caminos de control más sofisticados. Por ejemplo, un módulo puede ser *reentrante*. Esto es, un módulo se diseña de forma que de ninguna manera pueda modificarse a sí mismo a las direcciones que referencia localmente. Así el módulo puede ser usado para más de una tarea concurrentemente.

Dentro de una estructura de programa, un módulo puede ser clasificado como:



- Un módulo *secuencial* que se referencia y se ejecuta sin interrupción aparente por parte del software de la aplicación.
- Un módulo *incrementa* que puede ser interrumpido antes de que termine, por el software de la aplicación y posteriormente, se restablece su ejecución en el punto que en que se interrumpió.
- Un módulo *paralelo* que se ejecuta a la vez que otro módulo, en entornos de multiprocesadores concurrentes.

Los módulos *secuencias* son los que se encuentran más frecuentemente y están caracterizados como macros de tiempo de compilación y como subprogramas convencionales - subrutinas, funciones o procedimientos -. Los módulos *incrementales*, denominados frecuentemente *corrutinas*, mantienen un puntero de entrada que permite volver a ejecutar el módulo desde el punto de interrupción. Dichos módulos son extremadamente útiles en sistemas conducidos por interrupciones. Los módulos *paralelos*

Una vez que se tienen determinadas las entidades y sus relaciones, podemos centrarnos 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.

La íntima relación que existe entre el software y los datos se remonta hasta los orígenes de la informática. El concepto original que trasciende al de computadora con programa almacenado, es que los programas pueden ser vistos como datos y los datos interpretados como programas. La estructura de la información, denominada *estructura de datos*, ha demostrado que tiene un importante impacto sobre la complejidad y la eficiencia de los algoritmos diseñados para procesar información.

La estructura de los datos afecta al diseño, tanto en el aspecto estructural del software, como en un aspecto procedimental. Los datos repetitivos siempre se procesan con un software que tenga facilidades de control para la repetición, los datos alternativos (es decir, información que pueda estar o no estar presente) requieren software con elementos condicionales de procedimientos, una organización jerárquica de los datos tiene frecuentemente una extraordinaria semejanza con la estructura del programa que utilizan los datos. De hecho, la estructura de la información precede muy bien la estructura del programa.

La *construcción lógica de programas* (CLP), desarrollada por J.D. Warnier, es un método riguroso para el diseño de software. Sobre el esquema de la relación entre la estructura de datos y la estructura procedimental, Warnier desarrolla un conjunto de técnicas que realizan una transformación de la estructura de datos de entrada/salida en una representación procedimental detallada del software.

El *desarrollo de sistemas estructurados en datos* (DSED), también denominado *metodología de Warnier-Orr*, es una aplicación de CLP que potencia el análisis y los aspectos del diseño. El método DSED proporciona una notación y unos procedimientos para obtener la estructura de datos, la estructura del programa y el diseño procedimental detallado de los componentes del programa (*módulos*). Además, DSED



proporciona una notación que facilita al diseñador examinar el flujo de datos entre las fuentes y los sumideros de la información, así como los procesos que transforman la información.

El concepto de estructura modular de programa no se considera explícitamente. Los módulos se consideran un subproducto del procedimiento, a la idea de independencia de módulos se le da poca importancia.

El análisis de requisitos del software sigue siendo la base del diseño orientado a la estructura de datos. La descripción del ámbito de la información (estructura contenido y flujo de datos) contenida en la *especificación de requisitos de software* perfila la arquitectura del software que ha de desarrollarse durante el diseño. Cada método de diseño proporciona un conjunto de "reglas" que facilitan al diseñador la transformación de la estructura de datos en una representación del software.

Cada método orientado a la estructura de datos tiene su propio conjunto de reglas. Sin embargo, deben realizarse siempre las siguientes tareas de diseño:

- 1).- Evaluar las características de la estructura de datos.
- 2).- Representar los datos en términos de formas elementales, tales como la secuencia, la selección y la repetición.
- 3).- Transformar la representación de la estructura de datos en una jerarquía de control para el software.
- 4).- Refinar la jerarquía del software usando los criterios definidos como parte de un método.
- 5).- Desarrollar finalmente una descripción procedimental del software.

5.5.- Diseño de la ayuda en línea

Para crear un sistema de ayuda en línea, el programador de ésta primero debe reunir información sobre el tema de ayuda, posteriormente se debe planear el sistema de ayuda. Para lo cual es necesario tomar en cuenta las siguientes consideraciones:

- Definición del tipo de usuario.
- Planear el contenido del sistema de ayuda.
- Planear la estructura de los tópicos¹⁰ de ayuda.
- Planear la Sensibilidad al contexto.
- Determinar la estructura de archivos de tópicos.
- Diseñar los tópicos¹⁰ de ayuda.

¹⁰ Un tópico se refiere a un tema en particular del sistema de ayuda.



5.5.1.- Definición del tipo de Usuario

El tipo de usuarios determina el tipo de información que estará disponible en el sistema de ayuda y cómo se presentará tal información. Esta es una clasificación general de los usuarios de sistemas de ayuda :

Usuario	Antecedentes
<i>Novato en computación</i>	<i>Completamente nuevo en el campo de la computación.</i>
<i>Novato en la aplicación¹¹</i>	<i>Algún conocimiento de computación, pero nuevo en el tipo de aplicación que se desarrolla. Por ejemplo, si se está diseñando ayuda para un programa de Hoja de Cálculo y el usuario novato en la aplicación solo está familiarizado con procesadores de texto.</i>
<i>Intermedio en la aplicación</i>	<i>Posee conocimiento acerca de la aplicación.</i>
<i>Experto en la aplicación</i>	<i>Vasta experiencia con el tipo de aplicación.</i>

Los usuarios novatos necesitan ayuda en tareas de aprendizaje y obtención de familiaridad con la interface. Estos usuarios requieren tópicos que definen terminología nueva, específica de la aplicación, o que ofrece información conceptual e información orientada a las tareas que se realizan dentro del sistema.

Los usuarios sofisticados ocasionalmente buscan ayuda con un procedimiento o término, pero más frecuentemente necesitan refrescar su memoria con funciones o declaraciones de las cuales no recuerdan su sintaxis o estructura. El usuario experto tiende o a buscar información de referencia o buscar ayuda sólo con sintaxis de funciones o declaraciones, equivalencias en el teclado, y teclas rápidas.

Realmente no existen reglas para determinar el contenido total del sistema de ayuda. Si se quiere lograr un sistema de ayuda para todos los tipos de usuarios, se podría querer documentar los comandos de menú, procedimientos, definiciones de términos nuevos, características, funciones y todos demás aspectos relevantes de la aplicación.

Se debe considerar cuidadosamente la decisión de implementar ayuda sensible al contexto. Esto debido a que la ayuda sensible al contexto demanda una relación de trabajo estrecha entre el programador de la ayuda y el programador de la aplicación,

¹¹ Por "aplicación" se hace referencia al sistema al cual se pretende incorporar ayuda en línea.



el tiempo de desarrollo requerido para crear un sistema exitoso de ayuda se incrementa significativamente.

5.5.2.- Planeación del Contenido del Sistema de Ayuda

Se deben crear tópicos que sean suficientes para suministrar a los usuarios toda la ayuda que necesiten. Los tópicos deben ser concisos, pero además deben cubrir la profundidad de la aplicación. Siempre se debe recordar, también, que leer en una pantalla cansa los ojos del lector más rápido que leer la misma información de una página impresa. Así que los tópicos deben ser lo más breves posible.

Los sistemas de ayuda deberían incluir:

- Un menú de ayuda
- Pantalla de Contenido
- Lista de palabras clave e índices
- Tópicos individuales de ayuda

Planeación del Menú de Ayuda Para la Aplicación

Un buen menú de ayuda incluye las siguientes opciones:

Contenido
Búsqueda
Acerca de<nombre de la aplicación>

Otras opciones de menú de ayuda proveen beneficios adicionales para el usuario de la aplicación. Si se incluye cualquiera de estas características, se deben colocar entre la opción de búsqueda y Acerca de...:

- Índice (Totalmente alfabético)
- Guía del teclado
- Como usar la ayuda

Planeación de las pantallas de Contenido

El propósito de la pantalla principal de contenido es proveer una visión de el contenido del sistema de ayuda y dar rutas lógicas y claras hacia la información. Esta debería trabajar como el índice en un libro.

Pantallas de Contenido de Subniveles

Las pantallas de subcontenidos abren rutas desde la pantalla principal hacia tópicos individuales. Las entradas por categorías en la pantalla principal de contenido saltan a pantallas de subniveles que o listan tópicos en determinada categoría o despliegan pantallas de contenido de sub-subniveles.



Existen algunas sugerencias que se deberían seguir cuando se planean las pantallas de contenido:

- ♦ Las entradas pueden ser listadas bajo diferentes encabezados de categorías. Por ejemplo, las entradas de referencias y las entradas procedurales pueden estar separadas.
- ♦ Las entradas son presentadas por categoría, como por saltos de textos o ligas gráficas. Se deben enumerar las entradas en un orden lógico (por ejemplo, por ruta de conocimiento, por frecuencia de uso, o alfabéticamente).
- ♦ La lista debería incluir no más de 10 entradas. Hay que evitar tener más de 15 entradas si es posible. Si la lista es más larga de 15 entradas, se debe intentar seccionarla en encabezados o en dos o más de una pantallas de contenido.
- ♦ Evitar subniveles profundamente anidados de tal forma que el usuario no tenga que saltar mas que dos o tres veces para desplegar un tópico de ayuda.
- ♦ En pantallas de contenido con entradas que salten hacia tópicos individuales, evitar listas desplegables de entradas.

Las listas de palabras clave y los tópicos de índice son ayudas navegacionales usadas con pantallas de contenido. La lista de palabras clave y el índice de los tópicos deberían de ser similar uno con otro y trabajar como el índice de palabras de un libro.

5.5.3.- Planeación de la Estructura de los Tópicos de Ayuda.

Una de las primeras cosas que se deberían hacer cuando se planea un sistema de ayuda es identificar los tópicos que se desea mostrar en el sistema, y entonces decidir como se desea organizar esos tópicos en una estructura útil.

Cualquier estructura que se decida usar, se debe intentar minimizar el número de listas a través de las que los usuarios deben saltar para obtener información. Además, se debe evitar que los usuarios se muevan a través de múltiples niveles para alcanzar un tópico. Muchos sistemas funcionan bastante bien con sólo 2 ó 3 niveles.

Estructuración de Tópicos Dentro de una Jerarquía.

Muchos sistemas de ayuda estructuran los tópicos jerárquicamente, esto es, los tópicos son dispuestos dentro de una estructura de rangos en árbol. En la parte superior de la jerarquía se encuentra un índice, o una tabla de contenidos, o ambos. El índice y la tabla de contenidos listan tópicos individuales o tópicos en categorías de tópicos disponibles al usuario.



Los tópicos mismos pueden ser relacionados jerárquicamente. Cada paso sucesivo lleva al usuario un nivel hacia abajo en la jerarquía del sistema de ayuda hasta que el usuario alcanza tópicos de información discreta y específica. La relación jerárquica de los tópicos de ayuda determina en parte cómo navega el usuario a través del sistema de ayuda. La siguiente figura ilustra una posible jerarquía

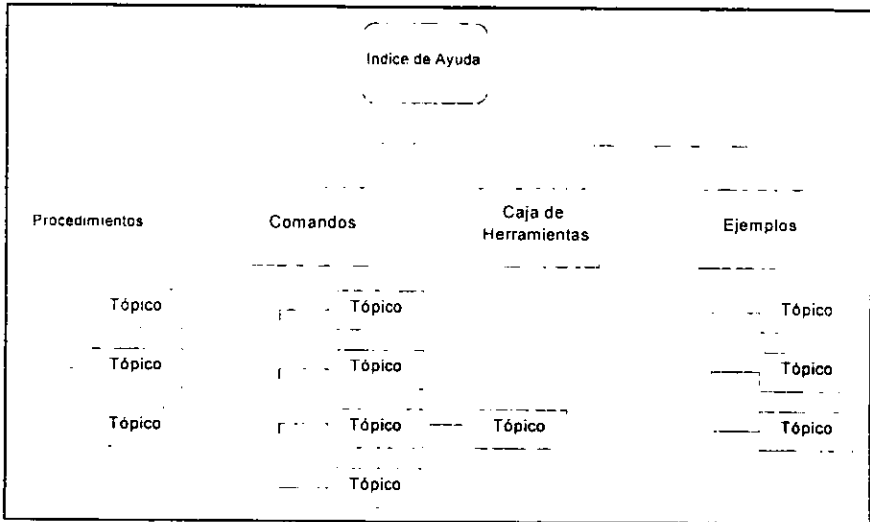


Figura. 5.9 Estructuración de Tópicos Dentro de una Jerarquía.

Estructuración de Tópicos para Acceso de Búsqueda

Los usuarios no están restringidos a navegar hacia arriba o abajo de la jerarquía; pueden saltar lateralmente de un tópico a otro, moviéndose a través de categorías de tópicos. El efecto de los saltos es atravesar lateralmente la estructura jerárquica.

Por ejemplo, la aplicación WinHelp (WINHELP.EXE) contiene una característica de búsqueda que permite al usuario teclear una palabra clave dentro de una caja de diálogo para buscar por tópicos asociados con esa palabra clave. WinHelp entonces despliega una lista de tópicos de los cuales escoger para acceder información que se relacione con la palabra clave.

Si el usuario conoce la característica acerca de la cual necesita ayuda, puede encontrar la información más rápidamente usando la característica de búsqueda, más que moviéndose a través de la estructura jerárquica.



Estructuración de los Tópicos en Secuencias

Además de ordenar los tópicos jerárquicamente, se puede ordenarlos en una secuencia lógica que satisfaga al usuario. La secuencia lógica, o secuencia de "hojeado", permite al usuario escoger los botones de "hojear" para moverse entre tópico y tópico. Las secuencias de hojeado son especialmente importantes para los usuarios a los que les gusta leer varios tópicos relacionados a la vez, tales como los tópicos que cubren los comandos en un menú "Archivo", o series de pasos secuenciales que guían a completar una tarea, tal como copiar un archivo.

5.5.4.- Planeación de la Sensibilidad al Contexto

Cuando se escribe a la par de la aplicación, la ayuda sensible al contexto permite al usuario presionar F1 para ayuda en la parte de la pantalla que tiene el foco actualmente.

Para desarrollar ayuda sensible al contexto se requiere que el programador de la ayuda y el programador de la aplicación se pongan de acuerdo en una lista de referencias de contexto de tal forma que la aplicación y el sistema de ayuda se transfieran la información correcta entre sí. Una referencia de contexto es un número único o una cadena que corresponde a un objeto particular en la aplicación; por ejemplo, para un comando de un menú, formulario, control, o región de la pantalla. Se pueden asignar referencias de contexto arbitrariamente, pero no se debería cambiarlas posteriormente. se usan referencias de contexto para crear ligas entre la aplicación y los tópicos correspondientes de la ayuda.

Se puede lograr ayuda sensible al contexto para un objeto asignándole una referencia de contexto. El cómo se asignen las referencias de contexto, los objetos a los cuales se les puede asignarlas, y si se usa números o cadenas depende del producto que se usó para desarrollar la aplicación.

Las referencias de contexto asignadas en el archivo de proyecto de Ayuda deben corresponder a las referencias de contexto que la aplicación envía en tiempo de ejecución para invocar un tópico en particular.

También se puede crear un archivo guía para manejar las referencias de contexto y la información de archivos.

5.5.5.- Determinación de la Estructura del Archivo de Tópicos

Los archivos de tópicos son archivos que contienen el texto, gráficos, y códigos ocultos que son compilados dentro del sistema de ayuda. Si bien el número y contenido de



archivos de tópicos difiere entre sistemas de ayuda, su estructura es consistente. Cada archivo mantiene uno o más tópicos separados por saltos de página.

La siguiente figura muestra esta estructura básica para la jerarquía mostrada en la figura 5.9.

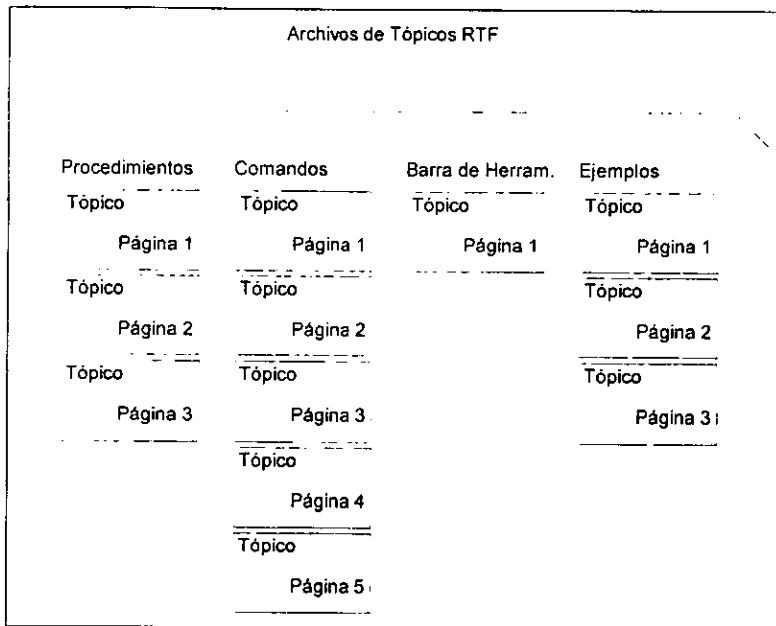


Figura. 5.10 Estructura del Archivo de Tópicos.

Cuando se escoge una estructura de archivos, se debe considerar el alcance y el contenido del sistema de ayuda que se está planeando. Por ejemplo, se podría colocar todos los tópicos de ayuda en un sólo archivo de tópicos, o se podría colocar cada tópico de ayuda en un archivo separado. Para proyectos grandes, estos extremos pueden crear problemas de seguimiento y chequeo. Muchos proyectos usan varios archivos de tópicos, dividiendo el contenido por materia o autor.



El número de tópicos va relacionado directamente con las características cubiertas por el sistema de ayuda. No se puede hacer cambios extensivos a la aplicación sin hacer los debidos cambios al sistema de ayuda. Por ejemplo, si se agregan características adicionales a la aplicación, entonces los tópicos adicionales deben ser creados en el sistema de ayuda para describirlos.

5.5.6.- Diseño de los Tópicos de Ayuda

El cómo la información en la ventana de ayuda aparece al usuario es primariamente una función del diseño del tópico de ayuda.

Optimamente, un tópico de ayuda tiene una o a lo más, dos pantallas de longitud. Una pantalla es definida como aproximadamente una mitad de el ancho de una ventana maximizada y 15-20 líneas de longitud.

Se puede minimizar la longitud del tópico partiendo la información en categorías pequeñas, usando listas de tópicos de múltiples columnas, o creando ventanas "pop-up" y ventanas secundarias para subordinar la información. Aunque no es bueno abusar de las ventanas "pop-up", ya que si existen demasiadas en un tópico la pantalla es difícil de leer. Por ejemplo, si se usan ventanas pop-up para definiciones, cada definición debería saltar sólo una vez, la primera vez que la palabra aparezca dentro del texto. Ocurre una excepción a lo anterior en tópicos muy largos, donde un autor incluye una segunda definición para usuarios que se han saltado y pasado la primera liga.

5.5.7.- Diseño del texto de los tópicos de ayuda

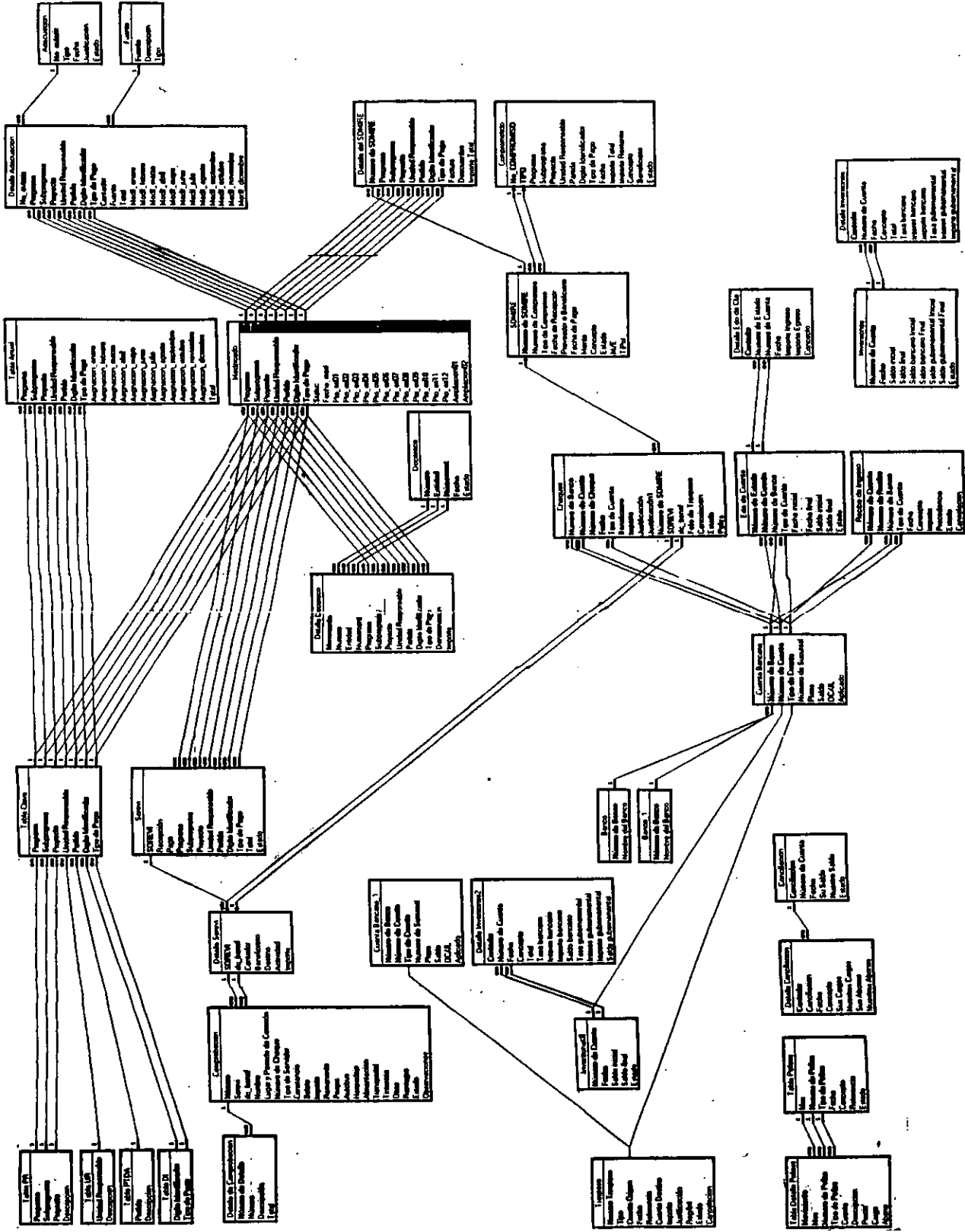
Los archivos de texto no se limitan a texto plano y sin formatear. Se pueden usar diferentes tipos de letra y tamaños, incluir gráficos de color para enfatizar puntos, párrafos con sangría para presentar información compleja, y usar una variedad de otros dispositivos visuales para presentar la información. La investigación en formatos de pantalla y sistemas de ayuda ha producido los siguientes criterios para presentar información a los usuarios:

- **Consistencia.**- Cuando se planeé el sistema de ayuda, hay que mantener en mente el sistema ya terminado. Por ejemplo, desarrollar una estrategia para usar ventanas secundarias que puedan ser usadas consistentemente a través de el sistema de ayuda. Decidir si se usarán ventanas "pop-up" sólo para definiciones de términos o para notas o tips también. Estas decisiones se toman basándose en las necesidades de los usuarios, pero siendo rigurosamente consistente en el diseño. Los usuarios esperan que los tópicos de ayuda tengan la misma apariencia, no importa la información que presenten. La titulación, resaltado, fuentes y posicionamiento del texto consistente es esencial para un sistema de ayuda efectivo.



- ♦ **Lenguaje.**- Se usa un lenguaje que sea apropiado para los usuarios del sistema. Un lenguaje que es demasiado sofisticado puede frustrar a los usuarios requiriendo de ellos que conozcan las definiciones de términos no familiares y conceptos. Hay que considerar los cambios que se podría tener que hacer si la aplicación tiene una difusión internacional. Ciertos elementos visuales que sirven para transmitir información en un país podrían no servir en otros países.
- ♦ **Párrafos y Longitud del texto.**- Hay que usar párrafos cortos. Los usuarios en línea llegan a ser sobrecargados con texto más fácilmente que los lectores de material impreso. Partir el texto en párrafos cortos ayuda a evitar este problema. Se debe usar un mínimo de texto. La velocidad de la lectura de texto se decrementa en un 30% comparada con el texto impreso. El texto conciso ayuda a compensar este decremento de la velocidad de lectura
- ♦ **Espacio en Blanco.**- El espacio en blanco se usa para aislar información visualmente y para hacer el texto en línea más legible. Se puede usarlo libremente, pero no se debe perder de vista la consideración total de espacio que un tópico ocupará en la pantalla. Los usuarios tienden a pensar que hay más información en una pantalla de la que existe en realidad. Por ejemplo, si la proporción de espacio blanco con respecto al texto es de 50:50, los usuarios lo perciben como 40:60. Siempre que sea posible, se recomienda trabajar con un diseñador gráfico para planear el espacio en blanco, color, encabezados, y otros elementos visuales.
- ♦ **Resaltado y tipos de letra.**- Las técnicas de resaltado se deben usar juiciosamente. Usar demasiadas puede llegar a confundir al usuario. Por otra parte; sólo se deben usar una o dos tipos de letras a la vez.
- ♦ **Diseño de gráficos.**- Los gráficos se usan para ayudar a explicar eventos visuales, pero hay que recordar que los gráficos atraen los ojos del usuario más rápido que el texto. Se debe procurar recortar el número de las imágenes que se usen para evitar perder enfoque hacia la información. Los gráficos son más efectivos cuando contribuyen a entender. Los gráficos no limitados a la información pueden distraer más que ayudar, y deberían ser evitados.
- ♦ **Uso del Color.**- Cuando se diseña la apariencia de el sistema de ayuda, se debería considerar el impacto visual del color. Por ejemplo, el color atrae el ojo y agrega énfasis. Los artículos del mismo color aparecen relacionados, mientras los artículos en colores diferentes se usan para pertenecer a diferentes grupos. Los colores diferentes tienen asociaciones diferentes para la gente y las culturas. No se debe confiar en el color para comunicar un significado en particular. El rojo, por ejemplo, implica "advertencia" en una cultura y "felicidad" en otra.

Después de haber especificado algunos aspectos importantes en el diseño y desarrollo del sistema, estamos en posibilidades para iniciar con la creación del Sistema Integral de Oficinas Centrales, abarcando los desde el aspecto de diseño hasta la puesta en marcha.





6.8.2.- Diagramas de Transición de Estados

Los diagramas de transición de estados son una herramienta poderosa de modelado para describir el comportamiento requerido del sistema SIOC_2, así como enfatizar las funciones del mismo. Los DTE se utilizarán en un sistema pasivo, en el sentido de que no busca controlar el ambiente que lo rodea, sino más bien reacciona a él o captura datos que le atañen. Un sistema siempre estará en espera de que llegue a ocurrir algún cambio, no se expresa en términos de que la computadora esté haciendo algo. Esto se debe a que el diagrama de transición de estados se usa para desarrollar un modelo esencial del sistema, es decir, un modelo de cómo se comportaría éste en los casos de que llegue a realizarse una acción.

Esto no significa que el sistema sea incapaz de ejecutar acciones o que no pretendamos mostrarlas, sino sólo que las acciones que ocurran en nuestro modelado no son los mismos estados, que representan condiciones observables en las que el sistema se pueda encontrar. Un estado representa algún comportamiento del sistema que es observable y que perdura durante algún período finito.

Un sistema sólo puede tener un estado inicial, sin embargo puede tener múltiples estados finales. Los diversos estados finales son mutuamente excluyentes, lo que significa que sólo uno de ellos puede ocurrir durante alguna ejecución del sistema.

Para completar el diagrama tendremos las *acciones* del sistema que se muestran junto a la flecha que conecta dos estados, estas *acciones* causan un cambio de estado, normalmente el sistema hará una o más acciones: producirá una salida, desplegará una señal en la terminal del usuario, llevará a cabo un cálculo, etc. Así que las acciones que se muestran en un DTE son respuestas regresadas al ambiente externo o bien cálculos cuyos resultados el sistema recuerda.

En un sistema complejo puede haber docenas de estados distintos del sistema, tratar de ponerlos todos en un mismo diagrama sería difícil. Por tanto, particionaremos los diagramas para elaborarlos y los realizaremos en un primer nivel.

Tenemos el DTE dentro del **Departamento de Presupuesto (Ver Apéndice C)**.

6.8.3.- Definición de Módulos

Debido a las ventajas que proporciona la programación modular, y que fueron expuestas anteriormente, el Sistema Integral de Oficinas Centrales está basado en esta técnica, de tal forma que pueda ser manejable en el mantenimiento.

El Sistema Integral de Oficinas Centrales es un sistema modular, gracias a que Access es un sistema orientado a objetos y a eventos, para lograr la modularidad hicimos uso de los formularios⁶ de Access. En nuestro sistema cada módulo de los que se muestran

⁶ Para una definición de Formulario ver el Glosario



en las páginas subsecuentes de esta tesis es representado por un formulario, cada formulario posee un módulo de código adjunto en el cual se guarda la programación asociada a cada evento de cada objeto del formulario.

Otra herramienta que nos es útil en Access para lograr la modularidad son los procedimientos de evento⁷, los cuales se ejecutan automáticamente como respuesta a un evento iniciado por el usuario o por el código del programa, o bien activado por el sistema.

De esta forma se logra aprovechar las ventajas que posee el desarrollo modular: la escritura por módulos facilita la programación, la detección de errores dentro de los módulos se vuelve más sencilla, los errores de un módulo no trascienden a otros, las modificaciones en caso de ser necesarias solo se hacen a los módulos implicados y si otro programador lee el código de un módulo le será más fácil de entender.

A continuación se muestra el menú principal del sistema, el cual contiene los siguientes módulos:

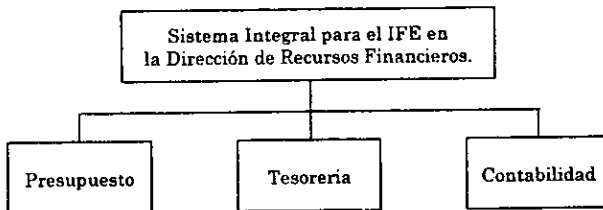


Figura. 6.2. Módulo general del Sistema

Dentro del Departamento de Presupuesto, tendremos las tablas que se ocuparán para elaborar cada uno de los módulos correspondientes.

- Adecuación.
- Comprometido.
- Detalle Adecuación.
- Detalle del Somire.
- Detalle Sorevi.
- Fuente.
- Mes.
- Modificado.
- Seguimiento Presupuestal.
- Somire.
- Sorevi.
- Tabla Anual.
- Tabla Clave.
- Tabla DI.
- Tabla PR.
- Tabla PTDA.
- Tabla UR.

⁷ Para una definición de Evento ver el Glosario.

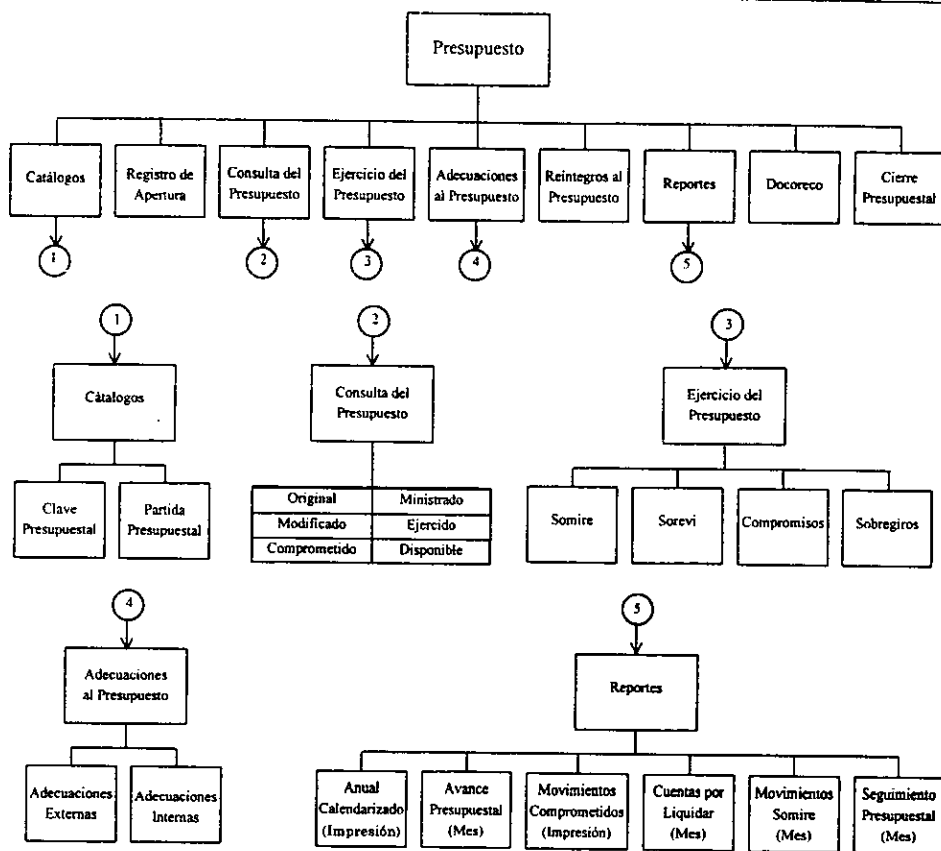


Figura. 6.3. Módulo de Presupuesto

Dentro del Departamento de Tesorería, tendremos las tablas que se ocuparán para elaborar cada uno de los módulos correspondientes.

- Banco.
- Comprobación.
- Conciliación.
- Corrección.
- Cuenta Bancaria.
- Cheques.
- Detalle Inversiones.
- Inversiones.
- Movimientos.
- Recibo de Ingreso.
- Reintegro.
- Saldo.
- Tabla
- Traspaso.
- Uso Reporte Avance.
- Uso Reporte Posición.

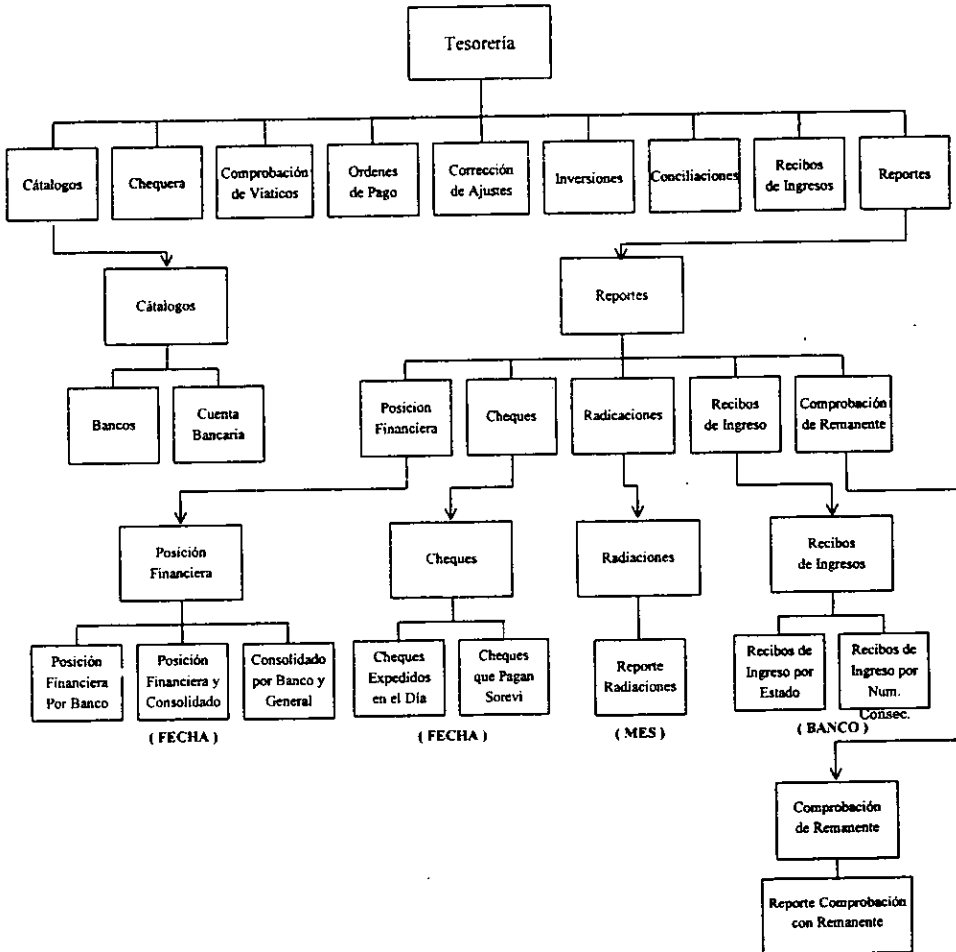


Figura. 6.4. Módulo de Tesorería

Dentro del Departamento de Contabilidad, tendremos las tablas que se ocuparán para elaborar cada uno de los módulos correspondientes.

- Beneficiario.
- Catálogo de Cuentas Contables.
- Detalle Docoreco.
- Docoreco.
- Tabla Detalle Pólizas.

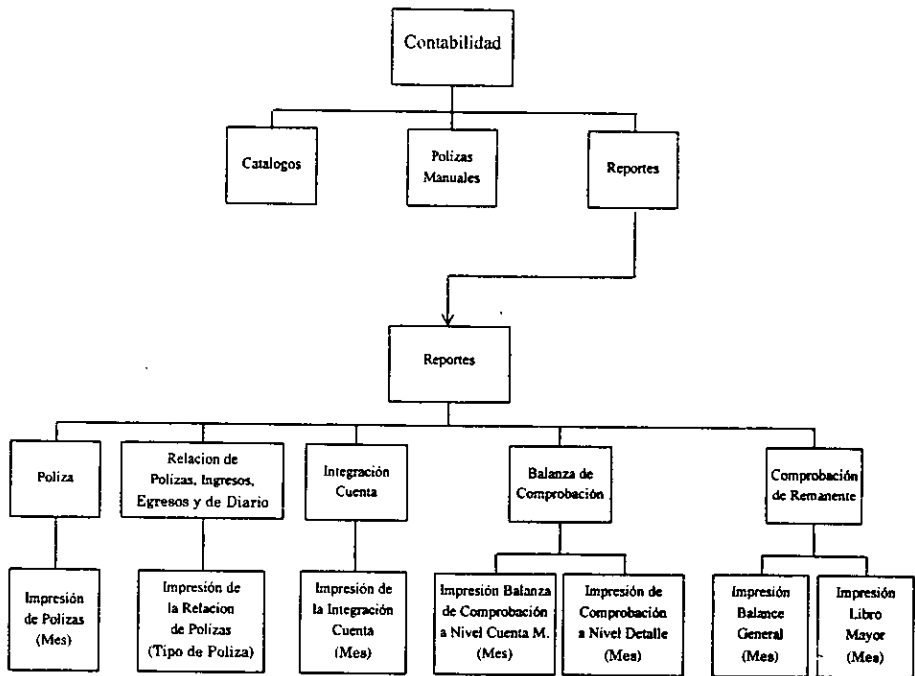


Figura. 6.5. Módulo de Contabilidad

Descripción de cada uno de los módulos que conforman el SIOC.

Presupuesto

Comprende las siguientes opciones:

- ♦ Catálogos
- ♦ Registro de Apertura
- ♦ Consulta del Presupuesto
- ♦ Ejercicio del Presupuesto
- ♦ Adecuaciones al Presupuesto
- ♦ Reintegros al Presupuesto
- ♦ Reportes
- ♦ Docoreco
- ♦ Cierre Presupuestal



Contabilidad

Comprende las siguientes opciones:

- ♦ Catálogos
- ♦ Pólizas Manuales
- ♦ Reportes

Tesorería

Comprende las siguientes opciones:

- ♦ Catálogos
- ♦ Chequera
- ♦ Comprobación de Viáticos
- ♦ Ordenes de Pago
- ♦ Correcciones de Ajuste
- ♦ Inversiones
- ♦ Conciliaciones
- ♦ Recibos de Ingreso
- ♦ Reportes

No se describen a detalle la función de cada módulo ya que esta se encuentra en la ayuda en línea con la que cuenta el sistema.

Después de explicar el proceso de desarrollo del Sistema Integral de Oficinas Centrales, explicare como se desarrollo la implementación y las pruebas de evaluación del sistema.

CAPÍTULO 8

SEGURIDAD Y ACCESO AL SISTEMA

8.1.- Seguridad : Consideraciones Generales



El problema de la seguridad tiene muchos aspectos, entre ellos los siguientes :

- Aspectos legales, sociales y éticos (por ejemplo, ¿tiene la persona que solicita el crédito de un cliente, por ejemplo, derecho legal a obtener la información solicitada ?)
- Controles físicos (por ejemplo, ¿deberá estar cerrado o resguardado de alguna otra manera el cuarto de computadoras o terminales ?)
- Cuestiones de política interna (por ejemplo, ¿cómo decide la empresa propietaria del sistema quiénes pueden tener acceso a qué ?)
- Problemas de operación (por ejemplo, si se utiliza un sistema de contraseñas, ¿cómo se mantiene en secreto las contraseñas ?, ¿Con qué frecuencia se cambian ?)
- Controles del equipo (por ejemplo, ¿posee la CPU características de seguridad tales como claves para la protección de las áreas de almacenamiento o un modo de operación privilegiado ?)
- Seguridad del sistema operativo (por ejemplo, ¿borra el sistema operativo subyacente el contenido de las áreas de almacenamiento y los archivos de datos cuando ya no se necesitan ?)



- ♦ Y por último, Materias de relevancia específica para el sistema mismo de bases de datos (por ejemplo, ¿tiene el sistema de bases de datos un concepto de propiedad de los datos?).

Ahora, habrá que considerar los aspectos incluidos sólo en esta última categoría.

La unidad de información para propósitos de seguridad - es decir, el "objeto de datos" que pudiera requerir protección individual - puede ser desde una base de datos o conjunto de tablas completos, por un lado, hasta un valor específico en una posición específica de fila y columna dentro de una tabla específica, por el otro. Un usuario dado, tendrá por lo regular diferentes derechos de acceso o autorizaciones sobre diferentes objetos de información (por ejemplo, autorización para usar SELECT (seleccionar) sólo con una tabla, autorización para emplear SELECT y UPDATE (modificar) con otra, etcétera). Además, diferentes usuarios pueden tener ciertos derechos sobre el mismo objeto; por ejemplo, el usuario A podría tener autorización para usar (sólo) SELECT con alguna tabla dada, en tanto que otro usuario B podría tener al mismo tiempo autorización para usar tanto SELECT como UPDATE con esa misma tabla.

En el caso específico de SQL, el sistema cuenta con dos características más o menos independientes implicadas en el mantenimiento de la seguridad: 1) el mecanismo de vistas, el cual, puede servir para ocultar datos confidenciales a usuarios no autorizados, y 2) el subsistema de autorización, mediante el cual usuarios con derechos específicos pueden conceder de manera selectiva y dinámica esos derechos a otros usuarios, y después revocar esos derechos, si lo desean.

Por supuesto, todas las decisiones sobre cuáles derechos deben concederse a cuáles usuarios son decisiones de política, no técnicas. Como tales, quedan a todas luces fuera de la jurisdicción del DBMS en sí; lo único que puede hacer el DBMS es obligar al cumplimiento de esas decisiones una vez tomadas.

8.2.- Seguridad en SQL



Para ilustrar el empleo de las vistas con propósitos de seguridad, se presenta un ejemplo basado en una base de datos de proveedores y partes.

1. Un usuario al cual se permite tener acceso a registros completos de proveedores, pero sólo de los proveedores situados en París:

```
CREATE VIEW PROVEEDORES_PARISINOS  
AS SELECT S#, SNOMBRE, SITUACION, CIUDAD  
FROM S  
WHERE CIUDAD = 'Paris';
```

Los usuarios de ésta vista ven un “subconjunto horizontal” - o (mejor dicho) un subconjunto de filas o subconjunto dependiente del valor - de la tabla base S. Existen distintos tipos de vista, por lo que se puede ver : subconjunto vertical, subconjunto de filas, subconjunto de columnas, subconjunto horizontal dependiente del contexto (es decir, los diferentes usuarios verán diferentes subconjuntos, dependiendo de quien emita la proposición).



Las vistas ofrecen en forma “gratuita” una importante medida de seguridad (“gratuita” porque el mecanismo de vistas ya se incluye en el sistema de todos modos para otros propósitos). Sin embargo, la estrategia de seguridad basada en vista resulta un poco torpe en ciertas situaciones sobre todo si algún usuario en particular necesita diferentes derechos sobre ciertos subconjuntos de la misma tabla simultáneamente.

8.3.- Otros Aspectos de Seguridad

1.- Es importante no dar por sentado que el sistema de seguridad es perfecto. Una persona con la intención de infiltrarlo y con la suficiente determinación casi siempre encontrará una manera de sortear los controles, sobre todo si el aliciente para hacerlo es considerable. Por tanto, en situaciones en las cuales los datos son muy delicados, o donde el procesamiento realizado con ellos es crítico, se hace indispensable un seguimiento de auditoría. Si, por ejemplo, las discrepancias en los datos hacen sospechar que los datos se han alterado, el seguimiento de auditoría puede servir para examinar lo sucedido y verificar que las cosas estén bajo control (o, si no lo están, ayudar a descubrir al culpable).



Puede pensarse en un seguimiento de auditoría como un archivo o base de datos especial en el cual el sistema registra en forma automática todas las operaciones realizadas por los usuarios sobre la base de datos normal. Una entrada representativa de el seguimiento de auditoría podría contener la siguiente información :



- ◆ Operación (por ejemplo, UPDATE).
- ◆ Terminal desde la cual se invocó la operación.
- ◆ Usuario que invocó la operación.
- ◆ Fecha y hora de la operación.
- ◆ Base de datos, tabla, registro y campo afectados.
- ◆ Valor anterior del campo.
- ◆ Valor nuevo del campo.

En algunos casos, el simple hecho de mantenerse un seguimiento de auditoría es suficiente para desanimar a un posible infiltrador.

2.- Es posible añadir un nivel más de seguridad mediante el cifrado de los datos. La idea básica aquí es que los datos pueden almacenarse físicamente en el disco, o transmitirse a través de las líneas de comunicación, en forma codificada o cifrada, y cualquier persona que intente obtener acceso a ella a través de canales distintos de los oficiales sólo detectará una mezcla ininteligible de bits. En condiciones ideales, el esfuerzo requerido por parte del intruso para descifrar esa mezcla de datos deberá exceder por mucho la ventaja que pudiera obtener al hacerlo.

Es posible entonces concluir que la seguridad significa proteger la base de datos contra usuarios no autorizados.

La seguridad implica : a) la definición de restricciones apropiadas, b) una especificación de las medidas que se han de tomar si se violan esas restricciones, y c) una supervisión por parte del sistema de las operaciones de los usuarios para detectar tales violaciones.

8.4.- Seguridad dentro del Sistema Integral de Oficinas Centrales



Para proteger la base de datos en el Sistema Integral de Oficinas Centrales fue necesario adoptar medidas de seguridad en varios niveles:

- ◆ Físico: La única manera de seguridad física que existe en el I.F.E. es no permitir la entrada a las instalaciones a toda persona ajena a ellas. Aún no existe una localidad específica que contenga a las computadoras y por lo tanto los sistemas.
- ◆ Humano: Se tiene restringido el uso de una computadora en las instalaciones. Por lo tanto hay poca probabilidad de que un usuario no autorizado permita el acceso a un intruso a cambio de sobornos u otros favores.



- Sistema operativo: Aunque el sistema de base de datos esté bien protegido, si no se protege de forma adecuada el sistema operativo éste puede servir para obtener acceso sin autorización a la base de datos. dado que casi todos los sistemas de base de datos permiten acceso remoto a través de terminales o redes, la seguridad a nivel software dentro del sistema operativo es tan importante como la seguridad física. Actualmente en el I.F.E. para poder arrancar el sistema operativo, todas las máquinas tienen password el cual se cambia regularmente.
- Sistema de base de datos: Una característica que ofrecen los sistemas bajo un entorno de bases de datos es la seguridad de su información. Todos los usuarios del Sistema Integral de Oficinas Centrales pueden hacer consultas sobre las bases de datos únicamente pero no podrán modificar dichas bases. Se tendrá acceso a la información pero por vistas, si es necesario hacer alguna modificación será a través de alguna de las pantallas del sistema.

La seguridad en todos los niveles anteriores debe mantenerse para garantizar la confiabilidad de la base de datos. Un punto débil en un nivel bajo de seguridad permitiría el acceso a las bases de datos. La base de datos del I.F.E. contiene información sobre los recursos financieros que se manejan dentro del mismo Instituto (como nominas, viáticos, pagos, etc), son un blanco tentador para poder violar la información contenida en la base de datos.

La seguridad es una de las tareas más importantes cuando se trabaja en una red local. La posibilidad de trabajar muchos usuarios con las mismas bases de datos, compartiendo aplicaciones y archivos, dispositivos de almacenamiento y otras operaciones, hacen que los usuarios tomen una serie de precauciones que aseguren sus datos y que el trabajo de horas, días o semanas no quede borrado por el simple descuido de un usuario.

Las funciones de seguridad, permiten proteger las bases de datos contra otros usuarios del entorno. Puede permitir el acceso a la información a grupos o usuarios individuales y restringir la posibilidad de ver o modificar todas o partes de las tablas contenidas en la base de datos.

Las funciones de seguridad multiusuario han sido mejoradas . El nuevo administrador de grupos de trabajo (WRKGADM.EXE) permite adjuntar un grupo de trabajo ya existente o crear un archivo SYSTEM.MDA para un grupo de trabajo. Las nuevas autorizaciones del administrador, actualizar datos, insertar datos y eliminar datos, permiten establecer autorizaciones para grupos de trabajo o usuarios individuales.

Las bases de datos utilizadas por varios usuarios precisan mucha más seguridad que la propia red local, es necesario asegurar que cada usuario sólo pueda visualizar y modificar lo que se haya establecido como permitido en su terminal, esto evitará que subdirecciones como Presupuesto puedan acceder a informes de Tesorería o a la tabla Cheques y, sin embargo puedan visualizar informes de consolidado y estado de posición financiera.



8.5.- Seguridad dentro del Entorno de Red

Considerando que las tres subdirecciones de la Dirección de Recursos Financieros comparten los servicios de una red Novell 3.11, se aprovechará dicha infraestructura que nos ofrece Access y NetWare.

NetWare ofrece elementos de seguridad avanzados que aseguran que los archivos van a estar protegidos de usuarios sin autorización.

La seguridad en disco duro se puede utilizar para evitar que extraigan datos importantes mediante discos flexibles. Los responsables pueden evitar que los usuarios trabajen fuera de unos directorios asignados, pudiendo aplicar restricciones de conexión. Por ejemplo, el supervisor de la red puede asignar una restricción de tiempo específico. Esto evita que los usuarios puedan conectarse en un nodo sin control en un momento no autorizado.

La seguridad es importante en el entorno multiusuario de una red local Netware, para evitar que usuarios no autorizados destruyan o deterioren archivos de datos importantes, como la base de datos del sistema Sistema Integral de Oficinas Centrales. Las restricciones de conexión evitan que usuarios sin derecho a acceder al sistema Sistema Integral de Oficinas Centrales o a cualquier otro nivel o aplicación de la red, ande navegando sin autorización del supervisor de la red.

El acceso a los archivos y directorios de un servidor NetWare se asigna o se restringe utilizando un conjunto de derechos. Estos derechos o privilegios permiten a un usuario realizar una determinada tarea, tal como crear un archivo, editarlo o borrarlo. Cada derecho tiene que ser asignado de forma individual. por ejemplo, un usuario podría recibir el derecho de crear y editar un archivo, pero no de borrarlo. Generalmente a los usuarios se les asignan derechos por directorio, pero también pueden aplicarse a los archivos. De este modo, un usuario puede tener derechos para trabajar en un directorio, pero puede tener restricciones sobre el uso de un determinado archivo de ese directorio.

La red Netware de Novell la cual servirá como plataforma para soportar la base de datos del sistema Sistema Integral de Oficinas Centrales tiene diferentes niveles de seguridad:

Seguridad a nivel de Conexión y Clave de Acceso.- Todos los usuarios que tienen una cuenta, generalmente con un nombre relativo al usuario (Login). A dicha cuenta se le asigna una clave de acceso confidencial (password), que tendrá que ser introducida cada vez que el usuario quiere entrar al sistema. Se pueden aplicar restricciones de acceso sobre cualquier cuenta de usuario.

Derecho de Acceso a Directorios.- Los derechos de acceso son los derechos asignados a los usuarios sobre directorios. Cada usuario tiene que recibir estos derechos de forma individual, excepto cuando forma parte de un grupo. Los usuarios



autorizados a realizar tareas cuando reciben derechos y pierden dicha autorización cuando se les retiran.

Seguridad a Nivel de Directorios.- Cuando un usuario recibe derechos sobre un directorio, estos se heredan sobre todos los subdirectorios correspondientes. Existe una prestación denominada filtro de derechos heredados que permite a los administradores o responsables bloquear los derechos que podría heredar un usuario sobre un subdirectorio del directorio padre. El filtro de derechos se aplica individualmente a cada subdirectorio.

Los derechos de un usuario sobre un subdirectorio se obtiene de la combinación de los derechos heredados del directorio padre y los bloqueados por un filtro de derechos heredados. A estos derechos se les denomina derechos efectivos de un usuario.

Atributos de Archivos y Directorios.- Los archivos pueden ser marcados con varios atributos por el supervisor de la red. Los atributos de archivo permiten evitar que un archivo sea modificado o borrado, incluso por usuarios que puedan tener tales derechos sobre el directorio.

La posibilidad de acceso de un usuario de la red a los directorios y archivos de Netware se establecen mediante derechos de acceso. Gracias a éstos, los supervisores o responsables pueden controlar quien o quienes pueden listar los archivos de un subdirectorio, leer de un archivo, modificar un archivo o crear uno nuevo. Si es necesario se le puede dar a los usuarios el derecho de borrado. Los derechos de acceso se pueden aplicar a un directorio completo, o a unos archivos específicos.

Derechos de Acceso.

A	(Access Control) (Control de Acceso)	Ofrece al usuario un control completo sobre un directorio.
C	(Create) (Creación)	Permite a un usuario crear archivos.
E	(Erease) (Borrado)	Permite a un usuario borrar archivos.
F	(File Scan) (Listar Archivos)	Permite a un usuario listar los archivos de un directorio.
M	(Modify) (Modificar)	Permite que el usuario que lo posee modifique el nombre del archivo que tiene abierto.
R	(Read) (Lectura)	Los usuarios con este derecho pueden abrir y leer archivos en un directorio.
S	(Supervisor)	Tiene un control completo sobre todo el directorio raíz.
W	(Write) (Escritura)	Los usuarios con pueden abrir y escribir archivos en el directorio.

El sistema de seguridad ofrece las siguientes ventajas:



- Cada usuario puede recibir un directorio seguro del servidor para su propio uso. Los usuarios que confían sus archivos a un sistema de archivos compartidos se sentirán seguros sabiendo que los restantes usuarios no podrán tocarlos.
- Las prestaciones de seguridad mantienen a los intrusos fuera del sistema y ofrecen una barrera efectiva frente a infecciones de virus informáticos.
- La protección por clave de acceso asegura que sólo se conectarán al sistema los usuarios autorizados. También asegura que cada usuario sea identificado adecuadamente, para ejecutar las secuencias de conexión y asignaciones de unidades adecuadas.
- Se puede restringir la conexión de usuarios a estaciones especificadas. esto evita que los usuarios se conecten desde áreas sin vigilancia.
- Se puede restringir el acceso de los usuarios a períodos de tiempo determinados, evitando que accedan al servidor durante períodos no autorizados o sin control. Los supervisores podrían desear restringir la conexión durante ciertas horas para poder llevar a cabo procedimientos de mantenimiento y copias de seguridad.
- Se puede evitar la carga de archivos en el sistema, evitando que los usuarios saturen el sistema de archivos con archivos innecesarios.

8.6.- Seguridad con Access

La seguridad de Microsoft Access está formada por varias piezas que deben encajar para que sea completa: los grupos de trabajo, las cuentas de usuario y de grupo, la propiedad y la asignación de autorizaciones. A continuación vamos a definir cada uno de estos conceptos:

Grupo de trabajo: Un conjunto de usuarios en un entorno multiusuario que comparten datos y la misma base de datos del sistema (en Access esta información se guarda en el archivo SYSTEM.MDA)

Cuenta de usuario: Una cuenta identificada por un nombre de usuario y un número de identificación personal (PID) que se crea para controlar el acceso a los objetos de la base de datos en un grupo de trabajo seguro.

Grupo: Conjunto de cuentas de usuarios en un sistema seguro; se identifica por el nombre de grupo y el número personal de identificación (PID). Las autorizaciones asignadas a un grupo son válidas para todos los usuarios pertenecientes al grupo.

Autorizaciones: Un conjunto de atributos que especifican que tipo de acceso tendrá un usuario a los datos u objetos de una base de datos. Por ejemplo, una tabla o consulta



con la autorización leer datos permite a un usuario ver pero no editar los datos de dicha tabla o consulta.

Cuando se trata la seguridad para una Base de Datos, se deben tener una cosa en mente:

Planificar el sistema de seguridad desde el principio, a medida que se diseña la base de datos, ya que si se planifica desde el principio un buen sistema de seguridad, se ahorrará mucho trabajo.



CONCLUSIONES

Una de las necesidades principales del Instituto Federal Electoral, es lograr la automatización de los procesos de la Dirección de Recursos Financieros para que sean más eficientes, lo importante en la actualidad es contar con un sistema que no sólo resuelva algún problema específico, sino que además lo haga optimizando recursos y por lo tanto incrementando la productividad y eficiencia en el control financiero del Instituto.

Por esta razón es que hoy en día el desarrollo de sistemas resulta ser de gran utilidad para la solución de cualquier problema, para ello sólo basta con contar con una serie de necesidades resumidas por el usuario que deberán ser analizadas por el diseñador, para encontrar la solución óptima, además de definir el equipo de cómputo necesario.

Con respecto al sistema desarrollado en el presente trabajo, cabe mencionar que cumple con el objetivo planteado y aunque en la fase de análisis se presentaron diversos problemas como la falta de coordinación entre las áreas involucradas, esta situación fue resuelta.

Se eliminó el problema que se tenía de redundancia de datos, construyendo el sistema bajo un ambiente de base de datos, además se vio, un beneficio adicional ya que se puede compartir la misma información en las distintas áreas.

Durante la etapa de desarrollo hubieron diversos problemas que dieron como resultado la experiencia para prever las cargas de trabajo, calcular tiempos estimados para cada fase del desarrollo, considerando retrasos por problemas fuera de nuestro alcance y que no se contemplaron en la planeación inicial.

Una vez terminado el sistema se procedió a la etapa de pruebas durante la cual se realizaron adaptaciones al sistema de acuerdo a las necesidades que en ese momento manifestaron los usuarios.

Ya cumplidas las expectativas del usuario se tomó el primer mes del ejercicio presupuestal 1997 para comparar los resultados del sistema, obteniendo el visto bueno del usuario. Sin embargo, para su total aceptación se solicitaron pruebas en paralelo para los siguientes dos meses.



Se considera que éstas pruebas no tendrán ningún problema en cuanto a resultados, dado que éstos se cotejaron con anterioridad durante las pruebas preliminares y los ajustes finales.

Como mencione en el capítulo siete, debido al constante cambio que toda organización por su crecimiento tiene, se debe de considerar que este sistema deberá crecer paralelamente con la entidad para mantener su funcionalidad y por lo tanto su credibilidad para la toma de decisiones.



BIBLIOGRAFÍA

Alice Y. H. Tsai.

- 1990 *Sistemas de Base de Datos (Administración y Uso).*
Prentice-Hall. Hispanoamérica, S.A., México

Antonio García Orozco.

- 1988 *Legislación Electoral Mexicana 1812 - 1988.*
Adeo Editores, S.A. Tercera Edición, México

C. J. Date.

- 1985 *Introducción a los Sistemas de bases de Datos.*
Ed. Addison-Wesley Iberoamericana. Quinta Edición.
Volúmen 1.

Charles L. Biggs - Evan G. Birks - William Atkins.

- 1996 *Managing The Systems Development Process.*
Prentice-Hall. International, Inc.

Donald H. Sanders

- 1992 *Informática Presente y Futuro.*
McGraw-Hill. Tercera Edición, México

Enid Squire.

- 1984 *Introducción al Diseño de Sistemas.*
Fondo Educativo Interamericano, México
Representación y Servicios de Ingeniería, S.A. de C.V.



Gane Chris and Trish Sarson

- 1993 *Structured Systems Analysis: tools and techniques.*
Prentice-Hall, Inc., Englewood Cliffs, New Jersey

Henry F. Korth - Abraham Silberschatz.

- 1993 *Fundamentos de Bases de Datos.*
McGraw-Hill, 2da. Edición, México

John G. Burch - Gary Grudnitski.

- 1994 *Diseño de Sistemas de Información.*
Grupo Noriega Editores. Editorial Limusa, S.A. de C.V.

James C. Emery.

- 1988 *Sistemas de Información para la Dirección.*
(El Recurso Estratégico Crítico).
Ediciones Díaz de Santos, S.A. México.

James Martín.

- 1987 *Organización de las Bases de Datos.*
IBM System Research Institute.
Prentice-Hall. Hispanoamérica, S.A. México, D.F.

John L. Viescas.

- 1995 *Guía Completa de Microsoft Access 2 para Windows.*
McGraw-Hill, 1era. Edición, México

Martin L. Shooman

- 1991 *Software Engineering Design, Reliability and Managment*
McGraww-Hill International Editions, Singapore



Nestor González Sainz.

- 1993 *Comunicaciones y Redes de Procesamiento de Datos.*
McGraw-Hill, 1era. Edición, México

Kenneth E. Kendall - Julie E. Kendall.

- 1991 *Análisis y Diseño de Sistemas.*
Prentice-Hall. Hispanoamérica, S.A., México

Roger S. Pressman.

- 1995 *Ingeniería del Software.*
McGraw-Hill. 3era. Edición. México

Richard E. Fairley.

- 1987 *Ingeniería del Software.*
McGraw-Hill de México, S.A. de C.V. Derechos Reservados, México

Roger Jennings.

- 1994 *El Libro de Access para Windows.*
Ediciones Anaya Multimedia, S.A.
Edición Oficial. Anaya Multimedia - Temas Microsoft.
1era. Edición. México

Ricardo O. Zurita - Norma González Ehrlich

Benito Guillen Niemeyer. - Jorge Molina Avilés.

- 1993 *Instituto Federal Electoral. Curso de Inducción
(Programa de Formación y Desarrollo)*
Impreso en México por I.F.E., México

Stallings William

- 1990 *Local Networks*
Macmillan Publishing Company, New York



Stan Schatt

1992 *Understanding Local Area Networks*
SAMS, Third Edition. Carmel, Indiana USA.

1992 *PC Magazine en Español.*
Volumen : 4. Número : 4. Artículo : Microsoft Access.

Manual de los Sistemas de Información.

2da. Parte Volumen: 6 Técnicas y Estándares.

(Análisis, Requisitos y su Determinación Diseño y Desarrollo).

(Implantación y su Evaluación). Biblioteca Técnica Philips.

Paraninfo (Madrid).

1993 *NetWare 3.12.*
Novell, Inc. San Jose, CA 95131 U.S.A.

Toby J. Teory

1990 *Database Modeling and Design*
Morgan Kaufmann Publishers, Inc. San Mateo, California



GLOSARIO

A

ACRÓNIMO.

Palabra compuesta por la letra o sílaba inicial de un nombre compuesto: I.F.E.

ADECUACIÓN COMPENSADA.

Ampliación, reducción, adición o cancelación a las asignaciones o claves presupuestarias originales de una entidad, que no implica alteración en el monto total de su presupuesto.

ADECUACIÓN LIQUIDA.

Ampliación, reducción, adición o cancelación a las asignaciones o claves presupuestarias originales de una entidad, que modifican el monto total de su presupuesto.

ADECUACIÓN PRESUPUESTARIA.

Modificación de la estructura financiera de los programas presupuestales aprobados, o ajuste a los calendarios financieros y metas del presupuesto autorizado por la H. Cámara de Diputados que se realiza a través de un documento denominado oficio de afectación presupuestaria.

La asignación original la hace la SHCP en forma calendarizada y todas las adecuaciones al presupuesto se presentan en forma calendarizada.

TIPOS DE MOVIMIENTO DE ADECUACIONES AL PRESUPUESTO.

Ampliaciones		Reducciones	
30	Adición Compensada	35	Reducción Compensada
31	Adición Líquida	36	Reducción Líquida
32	Ampliación Compensada		
33	Ampliación Líquida		

NOTA: Siempre van de la mano una Ampliación Compensada y una Reducción Compensada.



ADICIÓN COMPENSADA(O REDUCCIÓN COMPENSADA).

Es la suma a la asignación presupuestaria proveniente de una reducción o aumento por igual suma a la asignación de una o varias claves presupuestarias del mismo ramo, que no altera el total de su presupuesto.

ADICIÓN LIQUIDA.

Es la creación de una clave presupuestaria no considerada en el presupuesto originalmente aprobado, cuya asignación amplía el monto presupuestal autorizado del ramo.

ADICIÓN PRESUPUESTARIA.

Es la clave presupuestaria con su respectiva asignación, creada con posterioridad a la autorización del presupuesto de egresos de la federación. Existen dos tipos de adición: compensada y líquida.

AFECTACIÓN PRESUPUESTAL.

Es el movimiento que permite adecuar o modificar el presupuesto original autorizado por la H. Cámara de Diputados, el cual se realiza a través de un documento denominado "oficio de afectación presupuestaria". Según el tipo de clave presupuestaria que afecte puede ser: automática, es decir, cuando el traspaso por la entidad es interno; o no automática cuando se requiere de autorización de la Secretaría de Hacienda y Crédito Público. Según el movimiento que produzca, puede ser: ampliación, reducción o movimiento compensado.

AMPLIACIÓN COMPENSADA.

Aumento a la asignación original de una clave presupuestaria, proveniente de una reducción por igual suma a la asignación de una o varias claves presupuestarias de la misma entidad y que no altera el total de su presupuesto.

AMPLIACIÓN LIQUIDA.

Aumento a la asignación original de una clave presupuestaria que incrementa el total del presupuesto de una entidad; puede provenir de una reducción a la asignación de una o varias claves presupuestarias de otras entidades o de un aumento en los ingresos; en este último caso implica también un incremento al presupuesto de egresos de la federación.

AMPLIACIÓN PRESUPUESTARIA.

Es la modificación en aumento a la asignación de una clave presupuestaria ya existente.

ANTEPROYECTO DE PRESUPUESTO.

Estimación de los gastos a efectuar para el desarrollo de los programas sustantivos y de apoyo de dependencias y entidades de la administración pública federal; para su elaboración se deben observar las normas, lineamientos y políticas de gasto que fije la Secretaría de Hacienda y Crédito Público; la información permite a esta dependencia integrar el proyecto de presupuesto de egresos de la federación.

**AÑO FISCAL.**

Año presupuestario y contable para los cuales se presupuestan los ingresos y gastos brutos, y para los que se presentan cuentas, sin incluir ningún periodo complementario durante el cual puedan mantenerse abiertos los libros contables después del comienzo del periodo fiscal siguiente. En México el año fiscal abarca del 1ro. de enero al 31 de diciembre.

AÑO PRESUPUESTAL.

Período para el cual se aprueba, por parte de las cámaras de diputados y senadores que integran el H. Congreso de la Unión, las asignaciones, compromisos, aprovisionamientos y pagos destinados para la ejecución de los programas previstos en el presupuesto de egresos de la federación.

AVISO DE REINTEGRO (DOCUMENTO MÚLTIPLE).

Modalidad del documento múltiple, por medio del cual las dependencias reintegran recursos presupuestarios a la tesorería de la Federación, o a las instituciones bancarias con abono al presupuesto de egresos de la Federación, por concepto de pagos en demasía o indebidos.

B**BENEFICIARIO.**

La persona a cuyo favor se expide o cede un título de crédito. El que adquiere una utilidad, beneficio o ventaja que se origina en un contrato o en una sucesión hereditaria. El que goza de alguna manera de un bien o usufructo.

BIEN.

Todo aquello que puede ser objeto de apropiación, empleado para satisfacer alguna necesidad. Bienes o derechos susceptibles de producir beneficios de carácter patrimonial.

C**CALENDARIZACION PRESUPUESTAL.**

Es la dosificación temporal de los gastos, que realizará la Administración Pública Federal para la ejecución de sus programas. Los gastos recurrentes o fijos constituyen erogaciones relativamente estables, pero existen otros gastos como los destinados a la inversión que se asignan de acuerdo al grado de avance en la ejecución del proyecto, o se asignan en temporadas determinadas. "acuerdos de secas", con el fin de evitar liquidez ociosa y aprovechar condiciones climatológicas favorables.



CANCELACIÓN COMPENSADA.

Supresión total de recursos asignados a una clave presupuestaria, por el traspaso de dichos recursos a otra u otras claves presupuestarias de la misma entidad.

CANCELACIÓN LIQUIDA.

Supresión total de los recursos asignados a una o varias claves presupuestarias que afectan el total del presupuesto de egresos de una entidad. Por su procedencia puede ser derivada de un incremento a la asignación de una o varias claves presupuestarias de otras entidades o una disminución en los ingresos de la federación.

CANCELACIÓN PRESUPUESTARIA.

Acto de suprimir todos los elementos de una clave presupuestaria, y su correspondiente asignación del gasto.

CAPÍTULOS DE GASTO.

Son elementos de la clasificación por objeto del gasto que constituyen un conjunto homogéneo, claro y ordenado de los bienes y servicios que el gobierno federal y las entidades paraestatales adquieren para la consecución de sus objetivos y metas. La unidad básica de registro que conforma un capítulo presupuestario es la "partida", un conjunto de partidas forman un "concepto" y un grupo de conceptos integran un "capítulo". Este nivel de agregación hace posible el análisis retrospectivo y prospectivo de los planes o programas de acuerdo con la naturaleza del gasto a realizar; el IFE maneja los siguientes Capítulos de Gasto:

<i>CAPITULO DE GASTO</i>	<i>DESCRIPCION</i>
1000	SERVICIOS PERSONALES
2000	MATERIALES Y SUMINISTROS
3000	SERVICIOS GENERALES
4000	TRANSFERENCIAS
5000	BIENES MUEBLES E INMUEBLES
6000	OBRAS PUBLICAS
7000	INVERSIONES FINANCIERAS
8000	EROGACIONES EXTRAORDINARIAS
9000	DEUDA PUBLICA

CATALOGO DE CUENTAS.

Lista ordenada y codificada de las cuentas empleadas en el sistema contable de una entidad con el fin de identificar sus nombres y/o números correspondientes; regularmente sirve para sistematizar la contabilidad de una empresa.

**CHEQUE.**

Título de crédito expedido a cargo de una institución de crédito, por quien esté autorizado por ella al efecto, conteniendo la orden incondicional de pagar una suma de dinero a la vista, al portador o a la orden de una persona determinada. Orden de pago dirigida a un banco, contra los fondos poseídos por el girador. En dicha orden de pago puede ser nominativa o al portador.

CHEQUE DE CAJA.

Los que expiden las instituciones de crédito a cargo de sus propias dependencias. Estos cheques deben ser siempre nominativos y no son negociables.

CIERRE PRESUPUESTAL.

Es la revisión mensual del ejercido en una Partida Presupuestal, si existe un remanente en ésta, se transfiere al presupuesto del mes próximo de dicha partida.

CLAVE PRESUPUESTARIA.

Es la representación alfanumérica de los distintos elementos presupuestarios, cuyo objeto es identificar el destino del gasto, y se constituye por los siguientes elementos:

Programa, Subprograma, Proyecto, Unidad Responsable (UR), Partida, Dígito Identificador, y Tipo de Pago (PR - SP - PY - UR - PT - DI - TP)

Ejemplo: AA - 11 - 00 - 101 - 3702 - 11 - 2

ELEMENTO	CLAVE	EN EL EJEMPLO ANTERIOR
Programa	PR	AA
Subprograma	SP	11
Proyecto	PY	00
Unidad Responsable	UR	101
Partida de Gasto	PT	3702
Dígito Identificador	DI	11
Tipo de pago	TP	6

A cada Programa le corresponden determinados Subprogramas y un Proyecto como se muestra a continuación:

Programa	Subprograma(s)	Proyecto
AA	11	00
3W	05	00
ZZ	01,02,03,04	00
5K	01,02	00

Así es posible observar por ejemplo que al Programa AA le corresponde el Subprograma 11 y el Proyecto 00.



Cada Capítulo tiene un Tipo de Pago:

CAPITULO	TIPO DE PAGO
1000	2
1404	3
2000	4
3000	6
4000	4
5000	4
6000	4
8000	4

COMPROBANTE.

Documento que sirve de prueba de un desembolso en efectivo; Ejem. una factura pagada, un cheque cancelado, un recibo de caja chica, una copia al carbón de un cheque. Documento que sirve como evidencia de la autoridad que se tiene para autorizar un desembolso en efectivo. Ejem. Una factura aprobada de un proveedor; una nómina.

COMPROMETIDO(PRESUPUESTO)

Es el monto de bienes y servicios devengados y comprometidos, previamente a su pago, mediante documentos presupuestarios (pedidos, contratos, etc.)

COMPROMISO.

Son los recursos monetarios destinados al principio de un ejercicio presupuestal para los futuros pagos de bienes o servicios que de antemano se sabe se efectuarán durante dicho ejercicio.

CONTRATO

Se refiere a un servicio de una empresa o persona física.

CUENTA POR LIQUIDAR CERTIFICADA.

Documento comprobatorio único utilizado por el gobierno federal para liquidar la adquisición de bienes, servicios, obras, servicios personales, pago de deuda pública, etc. a los proveedores, contratistas y demás beneficiarios. Permite la captación oportuna de la información referente al gasto previo al pago y a su liquidación. Su propósito es sustituir la presentación de los diversos documentos comprobatorios (facturas, notas de remisión, etc.) en la Tesorería de la Federación y oficinas pagadoras facilitándole de esta manera la liquidación oportuno al beneficiario.

**D****DÍGITO IDENTIFICADOR.**

Elemento de la clave presupuestaria que permite conocer la naturaleza económica del gasto público.

Cada capítulo posee un Dígito Identificador:

<i>CAPÍTULO DE GASTO</i>	<i>DÍGITO IDENTIFICADOR</i>
1000	11
2000	10
3000	10
3100	11
3400	11
3404	11
3600	11
3800	11
3806	11
4000	11
5000	20
6000	60
8000	11

DOCORECO (DOCUMENTO DE CORRESPONSABILIDAD, REGISTRO Y CONTROL).

Es el documento a través del cual las distintas Juntas Locales Ejecutivas comprueban sus egresos para justificar la ministración de un mes.

E**EJERCICIO CONTABLE.**

Año económico; período de tiempo comprendido entre dos balances anuales sucesivos.

EJERCICIO FISCAL.

Es el período comprendido entre el 1o. de enero y el 31 de diciembre de cada año para los propósitos fiscales.

EJERCICIO PRESUPUESTARIO.

Período de tiempo durante el cual se han de ejecutar o realizar los ingresos y gastos presupuestados. Normalmente coincide con el año natural.



EJERCIDO COMPROMETIDO.

Monto de los bienes o servicios devengados y comprometidos previamente mediante documentos presupuestarios (pedidos, contratos, etc.)

EJERCIDO NO COMPROMETIDO.

Monto de los bienes o servicios devengados, que no están registrados en el segmento del comprometido.

EJERCIDO O DEVENGADO

Representa el monto de la documentación comprobatoria (facturas, notas, estimaciones, etc) presentadas a la entidad, y que han sido autorizadas para su pago.

EROGACIONES.

Se consideran erogaciones tanto los gastos como las inversiones que realice el contribuyente en un año de calendario sea cual fuere el nombre con que se les designe. Desembolsos en efectivo.

ESTRUCTURA PROGRAMATICA

Conjunto de programas y subprogramas ordenados en forma coherente; define las acciones que efectúan las dependencias y entidades de la Administración Pública Federal, para alcanzar sus objetivos y metas de acuerdo con las políticas definidas por el plan y los lineamientos que sobre aperturas programáticas establece la Secretaría de Hacienda y Crédito Público.

EVENTO.

Una acción reconocida por un objeto, como hacer click con el mouse o presionar una tecla, para la que puede definirse una respuesta. Un evento puede ser causado por una acción del usuario, por una instrucción de Access Basic o por el sistema. Usando propiedades asociadas a eventos se puede indicar a Microsoft Access que ejecute una macro, llame a una función de Access Basic o ejecute un procedimiento de evento como respuesta a un evento.

F

FACTURA.

Documento que se expide para hacer constar una venta, en el que aparece la fecha de la operación, el nombre del comprador, del vendedor, las condiciones convenidas, la cantidad, descripción, precio e importe total, de lo vendido. Se hace constar también el número de la factura, el nombre del comisionista, o agente vendedor, la forma del embarque y otros datos adicionales relativos a cada operación.

**FONDO FIJO DE CAJA.**

La suma de dinero que se pone en manos de una persona, para efectuar ciertos pagos. El encargado de su manejo, rinde cuentas periódicamente, y se le hacen nuevas entregas de dinero para reponer el importe de los pagos hechos. El fondo se limita a una cantidad constante, aunque puede aumentarse o disminuirse según lo indiquen de tiempo en tiempo las necesidades de operación. En cualquier momento, el efectivo, más el importe de los comprobantes de pago, debe ser igual al monto nominal del fondo fijo.

FORMULARIO

Objeto de Microsoft Access que se utiliza para ver y editar información en una base de datos, registro a registro. Los formularios pueden presentar registros de una o varias tablas que se deseen, dispuestos de la forma que se quiera. Además proporcionan la facilidad de usar controles conocidos, como por ejemplo cuadros de texto y casillas de verificación que son usados dentro del ambiente Windows en general, reduciendo errores en el ingreso de ciertos datos, además se pueden agregar otros controles como botones que responden a eventos.

Access permite programar totalmente cada uno de los eventos que ocurren en cada control de los formularios.

G**GASTO.**

Es toda aquella erogación que llevan a cabo los entes económicos para adquirir los medios necesarios en la realización de sus actividades de producción de bienes o servicios, ya sean públicos o privados.

H**HONORARIOS (INGRESOS POR).**

Modalidad de pago o remuneración que recibe un profesionista o trabajador independiente, que es contratado temporalmente, lo cual no implica pago de cuota de tipo sindical, ni prestaciones por parte del derecho de antigüedad, IMSS o ISSSTE.

Remuneración que se cobra o se paga, en compensación de uno o varios servicios profesionales.

Recursos percibidos por la prestación de un servicio personal independiente; los ingresos por la prestación de un servicio personal independiente los obtiene en su totalidad quien presta el servicio.

I**IMPUESTO AL VALOR AGREGADO (IVA).**



Tributo que se causa por el porcentaje sobre el valor adicionado o valor agregado a una mercancía o un servicio, conforme se completa cada etapa de su producción o distribución.

IMPUESTO SOBRE LA RENTA.

Aquél que se causa sobre las utilidades de una empresa o persona.

INSACULAR.

Poner en un saco cédulas o boletos para un sorteo o votación y seleccionar el personal idóneo para dichos fines con autorización oficial.

J

JUNTAS LOCALES EJECUTIVAS.

Son órganos permanentes de ejecución de las acciones del I.F.E. en cada entidad federativa.

M

MANUAL DE PROGRAMACION-PRESUPUESTACION.

Documento básico del proceso de Programación Presupuestación que guía la formulación e integración del proyecto de presupuesto de egresos de la federación, a través de formatos, instructivos e indicaciones técnicas para su llenado, a fin de facilitar el análisis y procesamiento computarizado de las asignaciones presupuestales.

MINISTRACION DE FONDOS.

Recursos presupuestarios que la Tesorería de la Federación directamente o a través de la red bancaria entrega a las dependencias de la administración pública federal, con base en la programación del ejercicio especificada en los calendarios autorizados de ministración de fondos y en la autorización de las órdenes de pago.

Recursos presupuestarios autorizados por la Secretaría de Hacienda y Crédito Público, que la Tesorería de la Federación radica a través de líneas de crédito en los corresponsales del Banco de México, para su ejercicio por parte de las dependencias y entidades de la administración pública federal, con base en la programación del ejercicio especificada en los calendarios de ministración de fondos respectivos.

MODIFICACIÓN PRESUPUESTAL.

Es el cambio hecho a las claves, descripciones o asignaciones del presupuesto de egresos de la federación. Según el movimiento que produzca puede ser, ampliación o reducción.

MODIFICACIÓN PROGRAMATICA.



Cambio en la magnitud y/o calendario de las metas de los programas, subprogramas y proyectos, como consecuencia de una reprogramación o planeación estratégica que se reflejará en la estructura programática.

N

NOMINA.

Listado general de los trabajadores de una institución, en el cual se asientan las percepciones brutas, deducciones y alcance neto de las mismas; la nómina es utilizada para efectuar los pagos periódicos (semanales, quincenales o mensuales) a los trabajadores por concepto de sueldos y salarios.

NORMA PRESUPUESTARIA

Disposición administrativa emitida por la Secretaría de Hacienda y Crédito Público para regular la formulación, ejecución y control del presupuesto de las dependencias y entidades de la administración pública federal.

NORMATIVIDAD PRESUPUESTARIA.

Conjunto de normas, lineamientos, metodologías, procedimientos y sistemas emitidos por la Secretaría de Hacienda y Crédito Público, con la finalidad de dirigirla hacia los objetivos deseados la formulación, ejercicio, control y evaluación del presupuesto, estableciendo el marco al que deben ajustarse las entidades, organismos y empresas que constituyen la administración pública federal.

NORMATIVIDAD PROGRAMATICA.

Conjunto de normas, lineamientos, metodologías, procedimientos y sistemas emitidos por la Secretaría de Hacienda y Crédito Público, para que las dependencias y entidades de la Administración Pública Federal formulen, ejecuten, controlen y evalúen sus programas de mediano plazo, programas operativos anuales y programas de presupuesto.

O

OFICIO DE AFECTACIÓN PRESUPUESTARIA.

Documento que sirve como instrumento para adecuar o modificar las metas y las asignaciones del presupuesto de egresos de la federación, según el movimiento que se produzca, puede generar una ampliación, y/o reducción de partidas de gasto.

OPERACIONES COMPENSADAS.

Son aquellas, que si bien constituyen un ingreso monetario, a ellas corresponde un egreso por el mismo monto, estableciéndose una relación compensatoria.

**ORGANISMO DESCENTRALIZADO.**

Institución definida por la ley orgánica de la Administración Pública Federal con personalidad jurídica y patrimonio propio, constituida con fondos o bienes provenientes de la administración pública federal; su objetivo es la prestación de un servicio público o social, la explotación de bienes o recursos propiedad de la nación, la investigación científica y tecnológica y la obtención o aplicación de recursos para fines de asistencia o seguridad social.

OSI (INTERCONEXIÓN DE SISTEMAS ABIERTOS).

La Organización Internacional de Estandarización (ISO), desarrolló un modelo de referencia para la arquitectura de sistemas. Le llamó OSI (Open System Interconnection). Es el concepto de sistema en un conjunto de una o más computadoras, el software asociados, los periféricos, las terminales, los operadores humanos, los procesos físicos, los medios de transferencia de información, etc., que forman un ente autónomo con capacidad de realizar el procesamiento de la información. OSI pone atención al intercambio de información entre sistemas y no al funcionamiento interno de cada sistema en particular.

El objetivo, a largo plazo, de ISO es desarrollar una compatibilidad total inter-sistemas, entre los muchos productos y servicios ofrecidos por los proveedores y las redes transportadoras alrededor del mundo.

P**PARTIDA COMPENSADA.**

Reducción de una partida autorizada y transferida a otra o varias por el mismo importe, éste movimiento no altera el presupuesto asignado a los programas de la entidad.

PARTIDA DE GASTO(PARTIDA PRESUPUESTAL).

Constituyen elementos afines, integrantes de cada concepto, y representan expresiones concretas y detalladas del bien o servicio que se adquiere, permitiendo la cuantificación monetaria y contable, de los mismos.

Elemento presupuestario en que se subdividen los conceptos y que clasifican las erogaciones de acuerdo con el objeto específico del gasto.

PEDIDO.

Se refiere a bienes muebles, suministros, materiales, o cualquier otro elemento físico requerido para la funcionalidad de los organismos y dependencias oficiales.

PÓLIZA.

Es el documento de comprobación de un ingreso o de un egreso.

PRESUPUESTACION.



Proceso de consolidación de las acciones encaminadas a cuantificar monetariamente los recursos humanos, materiales y financieros, necesarios para cumplir con los programas establecidos en un determinado periodo; comprende las tareas de formulación, discusión, aprobación, ejecución, control y evaluación del presupuesto.

PRESUPUESTO.

Estimación financiera anticipada, generalmente anual, de los egresos e ingresos del sector público federal, necesarios para cumplir con las metas de los programas establecidos. Asimismo, constituye el instrumento operativo básico que expresa las decisiones en materia de política económica y de planeación.

PRESUPUESTO AUTORIZADO DEFINITIVO.

Es la asignación original y sus modificaciones al cierre de la Cuenta de la Hacienda Pública Federal, que deberá coincidir con el presupuesto otorgado.

PRESUPUESTO COMPROMETIDO.

Estimación de las obligaciones contraídas pendientes de ejercerse con cargo a las partidas del presupuesto aprobado mientras no prescriba la acción para exigir el crédito, siempre que hubiesen sido debidamente autorizados y registrados.

PRESUPUESTO DE EGRESOS DE LA FEDERACIÓN.

Es el documento jurídico, contable y de política económica, aprobado por la H. Cámara de Diputados del Congreso de la Unión a iniciativa del Presidente de la República, en el cual se consigna el gasto público, de acuerdo con su naturaleza y cuantía, que debe realizar el gobierno federal y el sector paraestatal de control directo, en el desempeño de sus funciones en un ejercicio fiscal.

PRESUPUESTO EJERCIDO.

Importe de las erogaciones realizadas; respaldado por los documentos comprobatorios (facturas, notas, nóminas, etc.) presentados a la dependencia o entidad una vez autorizadas para su pago, con cargo al presupuesto autorizado.

PRESUPUESTO MODIFICADO.

Es la asignación original consignada en el presupuesto, más las ampliaciones líquidas, menos las reducciones a la fecha. Comprende las variaciones que afectan al presupuesto autorizado durante su ejercicio, las cuales se sustentan en un proceso de modificaciones programático presupuestarias.

PRESUPUESTO ORIGINAL AUTORIZADO.

Estimaciones de gasto autorizadas por la H. Cámara de Diputados en el Presupuesto de Egresos de la Federación y que son el resultado de la iniciativa que el Ejecutivo envía a través de la Secretaría de Hacienda y Crédito Público para desarrollar las actividades de la Administración Pública Federal, durante el período de un año a partir del primero de enero.



PROCESO DE PROGRAMACION-PRESUPUESTACION.

Etapas interrelacionadas que forman parte del proceso de control administrativo en las que se definen objetivos y metas, así como los recursos que se requieren para alcanzarlos. Constituye la estrategia de acción de corto plazo que la Secretaría de Hacienda y Crédito Público utiliza para definir competencias, responsabilidades y actividades a realizar tanto por las entidades globalizadoras como por las coordinadoras de sector y las entidades coordinadas sectorialmente en la formulación del proyecto de presupuesto de egresos de la federación. Las etapas que lo integran son formulación, ejecución, control y evaluación.

PROGRAMA.

Instrumento normativo del sistema nacional de planeación democrática cuya finalidad consiste en desagregar y detallar los planteamientos y orientaciones generales del plan nacional, mediante la identificación de objetivos y metas. Según el nivel en que se elabora puede ser global, sectorial e institucional, de acuerdo a su temporalidad y al ámbito territorial que comprende puede ser nacional o regional y de mediano y corto plazo respectivamente.

Conjunto homogéneo y organizado de actividades a realizar para alcanzar una o varias metas, con recursos previamente determinados y a cargo de una unidad responsable.

PROGRAMA OPERATIVO ANUAL (POA).

Instrumento que traduce los lineamientos generales de la planeación nacional del desarrollo económico y social del país, en objetivos y metas concretas a desarrollar en el corto plazo, definiendo responsabilidades, temporalidad y especialidad de las acciones, para lo cual se asignan recursos en función de las disponibilidades y necesidades contenidas en los balances de recursos humanos, materiales y financieros.

PROGRAMACIÓN.

Proceso a través del cual se definen estructuras programáticas, metas, tiempos, responsables, instrumentos de acción y recursos necesarios para el logro de los objetivos de largo y mediano plazo fijados en los planes, y que se irán concretando mediante la programación económica y social, considerando las limitaciones y posibilidades reales de la Economía Nacional.

PROYECTO.

Conjunto de obras que incluyen las acciones del Sector Público necesarias para alcanzar los objetivos y metas en un programa o subprograma de inversión, tendientes a la creación, ampliación y/o conservación de una entidad productiva perteneciente al patrimonio nacional.

Permite identificar el origen de los recursos que requiere la ejecución de los proyectos de inversión física, necesarios para la construcción, ampliación o remodelación de inmuebles y los estudios de inversión, ya sean por contrato o por administración.

**R****RADICACIÓN DE FONDOS.**

Situación de fondos que la Tesorería de la Federación hace a las oficinas pagadoras, con la finalidad de ejercer el presupuesto de egresos correspondiente.

RAMO PRESUPUESTAL.

Elemento que identifica y clasifica el Gasto Público Federal por entidades administrativas del Poder Ejecutivo (Secretarías, Departamentos de Estado, Presidencia de la República, Procuraduría General de la República), Poderes Legislativo y Judicial y rubros generales de gasto como erogaciones adicionales, deuda pública y desarrollo regional.

RECLASIFICABLE (GASTO).

Monto que se debe transferir de una clave a otra, debido a que la entidad asignó equivocadamente el gasto, debido a una reprogramación o reorientación planteada por alternativas de política económica.

REDUCCIÓN COMPENSADA.

Disminución al monto de una clave presupuestaria, que puede generar una adición y/o ampliación compensada de una o varias claves sin modificar el presupuesto de la entidad

REDUCCIÓN LIQUIDA.

Disminución al monto de una clave presupuestaria que modifica el presupuesto de la entidad o dependencia de que se trate.

REDUCCIONES PRESUPUESTARIAS.

Son los recursos que se disminuyen a una clave presupuestaria, y que pueden tener un efecto líquido o compensado. Modificación que disminuye la asignación de una clave presupuestaria ya existente.

REGISTRO PRESUPUESTARIO.

Asiento de erogaciones realizadas por las dependencias y entidades en relación a la asignación, modificación y ejercicio de los recursos presupuestarios que les hayan autorizado.

REINTEGROS.

Bonificaciones a la clave presupuestaria que realiza la Tesorería de la Federación a las sociedades nacionales de crédito mediante un aviso de reintegro enviado por las entidades de la Administración Pública Federal.

REMANENTE O SUPERAVIT PRESUPUESTAL.

Es la variación positiva de disponibilidades del Gobierno Federal y de los organismos y empresas controlados presupuestalmente, que resulta después de considerar el endeudamiento neto, y que se registra en la cuenta pública con este título.



S

SALDO

La diferencia entre la suma de movimientos deudores y la suma de movimientos acreedores de una cuenta. Si el movimiento deudor es mayor, la cuenta tendrá un saldo deudor; contrariamente, si el movimiento acreedor es mayor, el saldo será acreedor.

SOMIRE. (SOLICITUD DE MINISTRACIÓN DE RECURSOS).

Es un formato de pago que deben enviar todas las Unidades Responsables (UR) a Presupuesto para el pago de sus proveedores.

SOREVI. (SOLICITUD DE RECURSOS DE VIÁTICOS).

Es un formato de pago que deben enviar todas las Unidades Responsables (UR) a Presupuesto, para el pago de viáticos del personal que sale de comisión.

SUBPROGRAMA.

Segmentación del programa en donde se establecen objetivos; metas, recursos y responsables para su ejecución, tiene como finalidad facilitar la ejecución y el control de acciones homogéneas.

STOCK.

Cantidad de mercancías disponibles, existencias, provisión.

SUFRAGÁNEO.

Dícese de lo que depende de la Jurisdicción de otro.

SUPLETORIA.

Que suple una falta.

T

TECHO FINANCIERO.

Límite máximo del presupuesto que se asigna en un período determinado, generalmente un año, a una dependencia o entidad de la Administración Pública Federal el cual incluye gasto corriente y gasto de inversión.

TCP/IP

Protocolo de comunicaciones que define los formatos y la reglas para la transmisión y recepción de información, independientemente de cualquier configuración de red o hardware. TCP/IP, define el formato de los paquetes de transmisión, incluyendo el origen del paquete, el destino, la longitud y el tipo de paquete.

**TRASPASOS DE RECURSOS PRESUPUESTARIOS.**

Es el movimiento que consiste en trasladar el importe total o parcial de la asignación de una clave presupuestaria a otra, previa autorización del sector y de la Secretaría de Hacienda y Crédito Público, en su caso.

U**UNIDAD ADMINISTRATIVA.**

Elemento de la clave presupuestaria que identifica y clasifica el gasto público por entidades, según la organización interna de cada institución.

UNIDAD RESPONSABLE.

Tipo especial de sociedad anónima que tiene por finalidad principal facilitar el uso de crédito a sus miembros. Son consideradas organizaciones auxiliares de crédito y para formarlas se requiere de la Comisión Nacional Bancaria, a cuya vigilancia están sujetas.

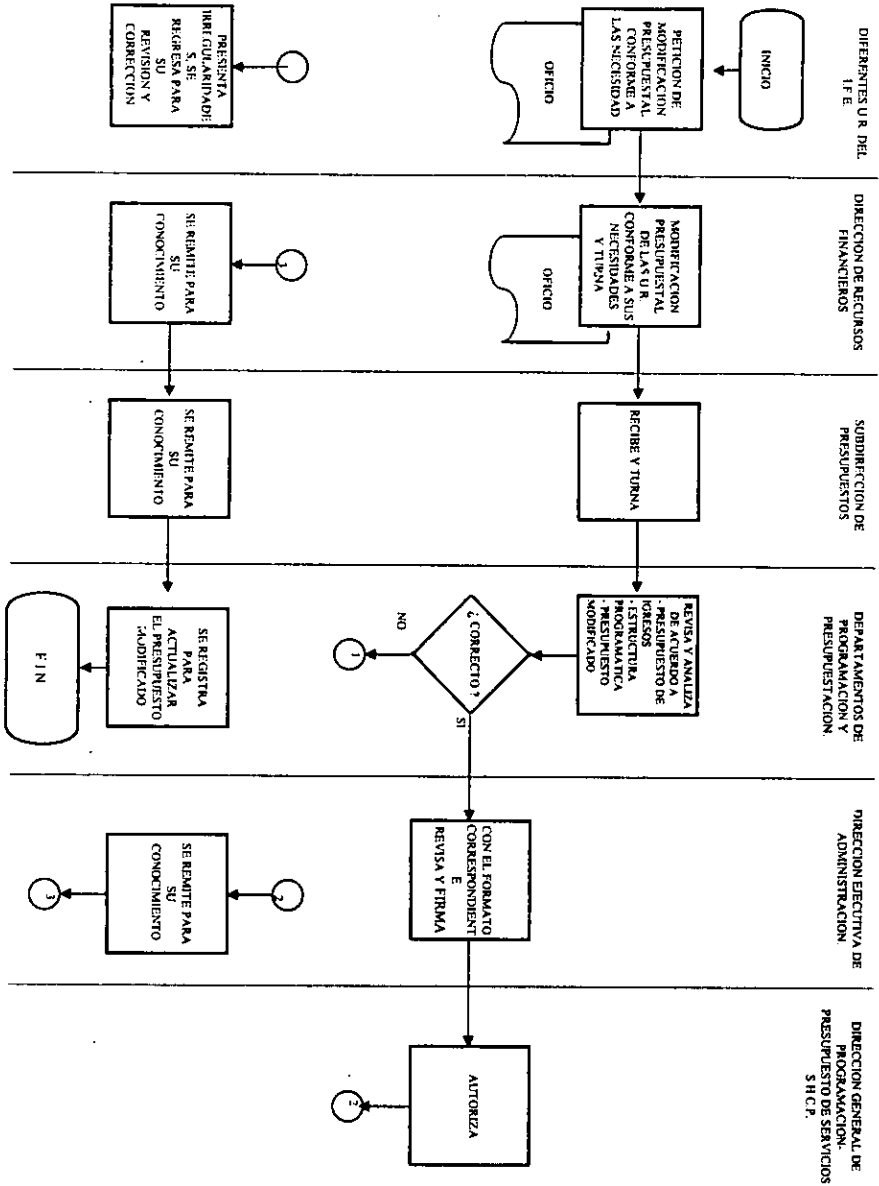
CLAVE	UNIDAD RESPONSABLE
101	Dirección General
103	Secretaría General
111	Registro Federal de Electores
112	Prerrogativas y Partidos Políticos
113	Organización Electoral
114	Servicio Profesional Electoral
115	Capacitación Electoral y F. Civ.
116	Dirección Ejecutiva de Administración.
200	Juntas Locales
300	Juntas Distritales.



Apendice **A** | Diagramas de Flujo de datos



AFECCION PRESUPUESTAL





PROGRAMA OPERATIVO ANUAL POA

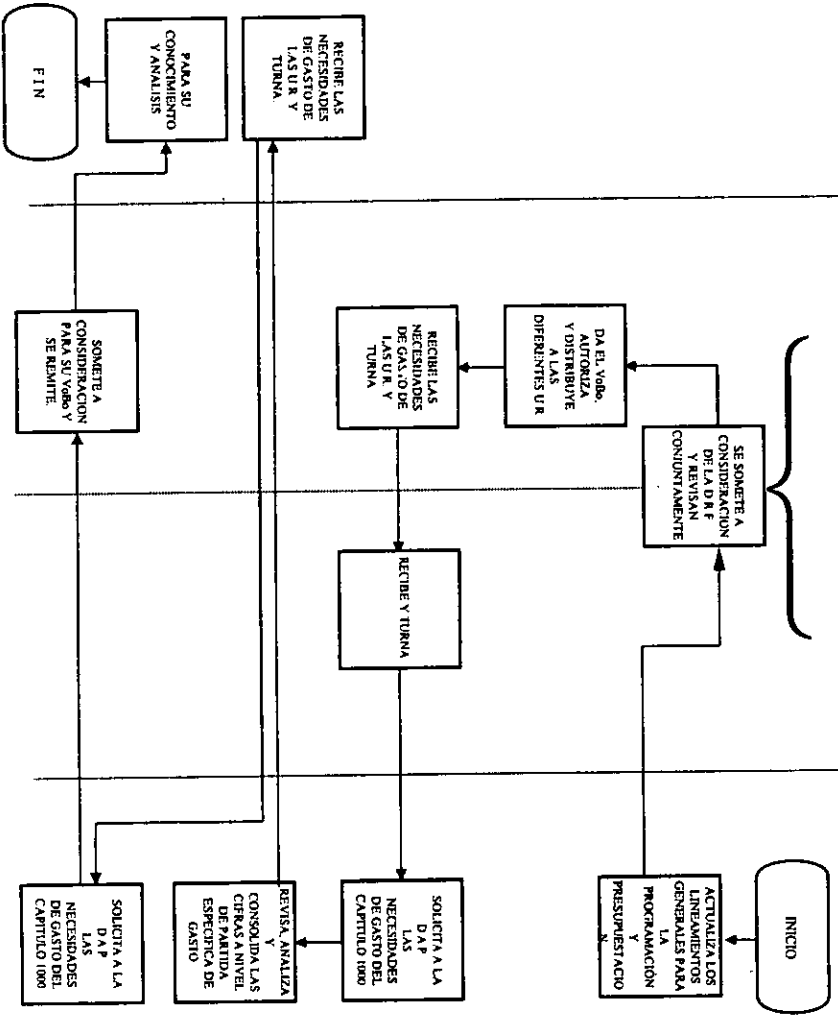
S.H.C.P.

DIRECCION EJECUTIVA DE ADMINISTRACION

DIRECCION DE RECURSOS FINANCIEROS

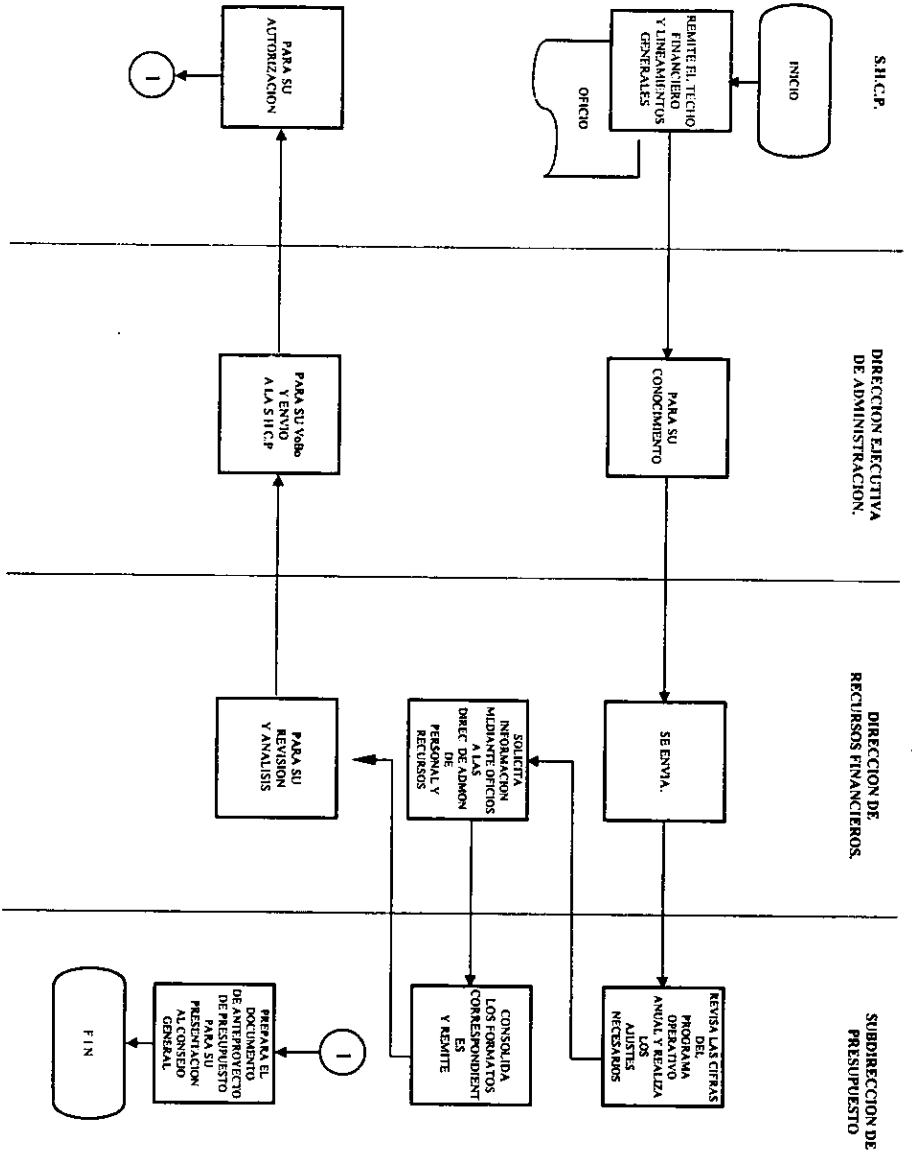
SUBDIRECCION DE PRESUPUESTO

DEPARTAMENTO DE PROGRAMACION Y PRESUPUESTACION



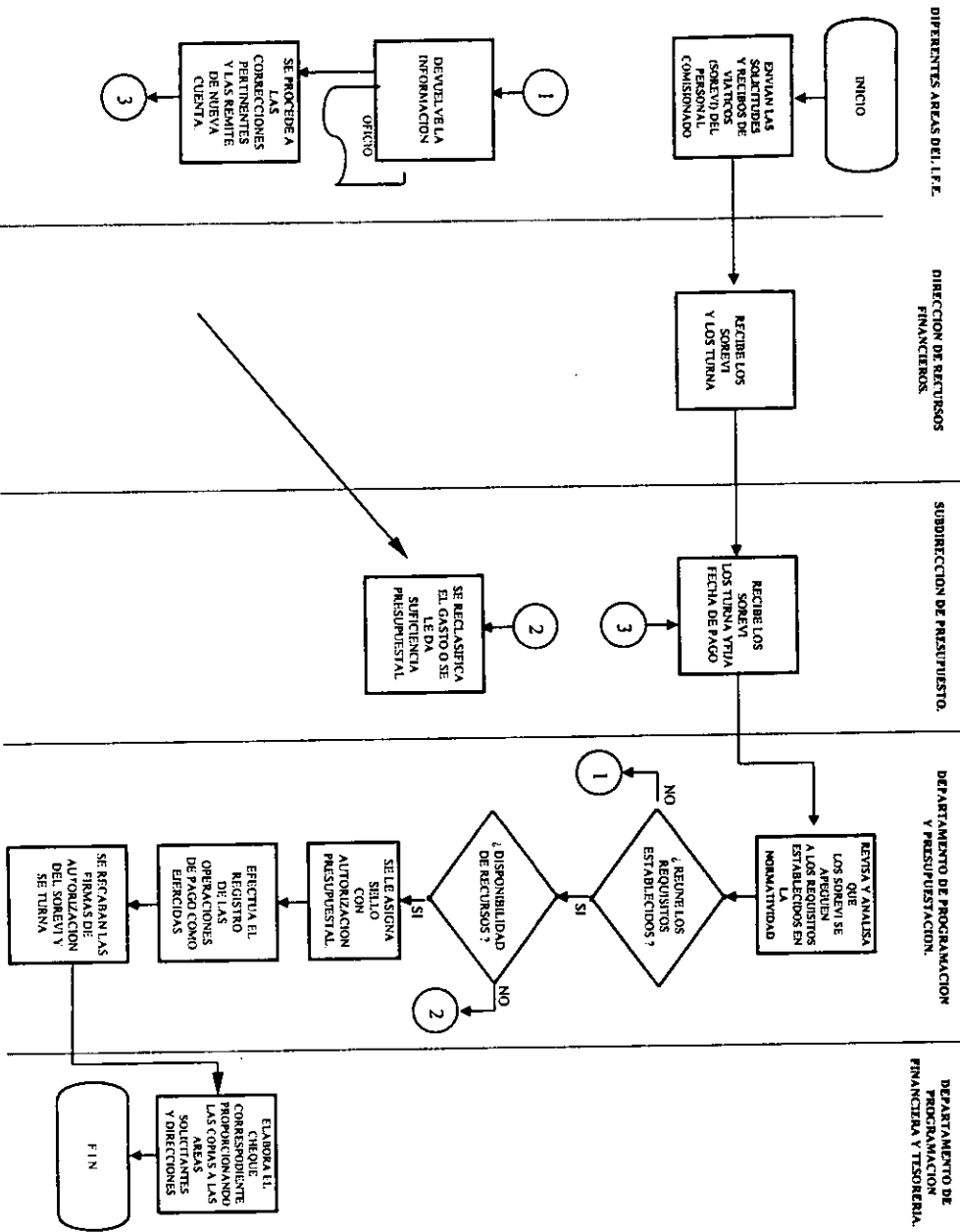


ANTEPROYECTO PRESUPUESTO





SOLICITUD Y RECIBOS DE VIATICOS (SOREVI)





SOLICITUD Y PAGO DEL FONDO FIJO

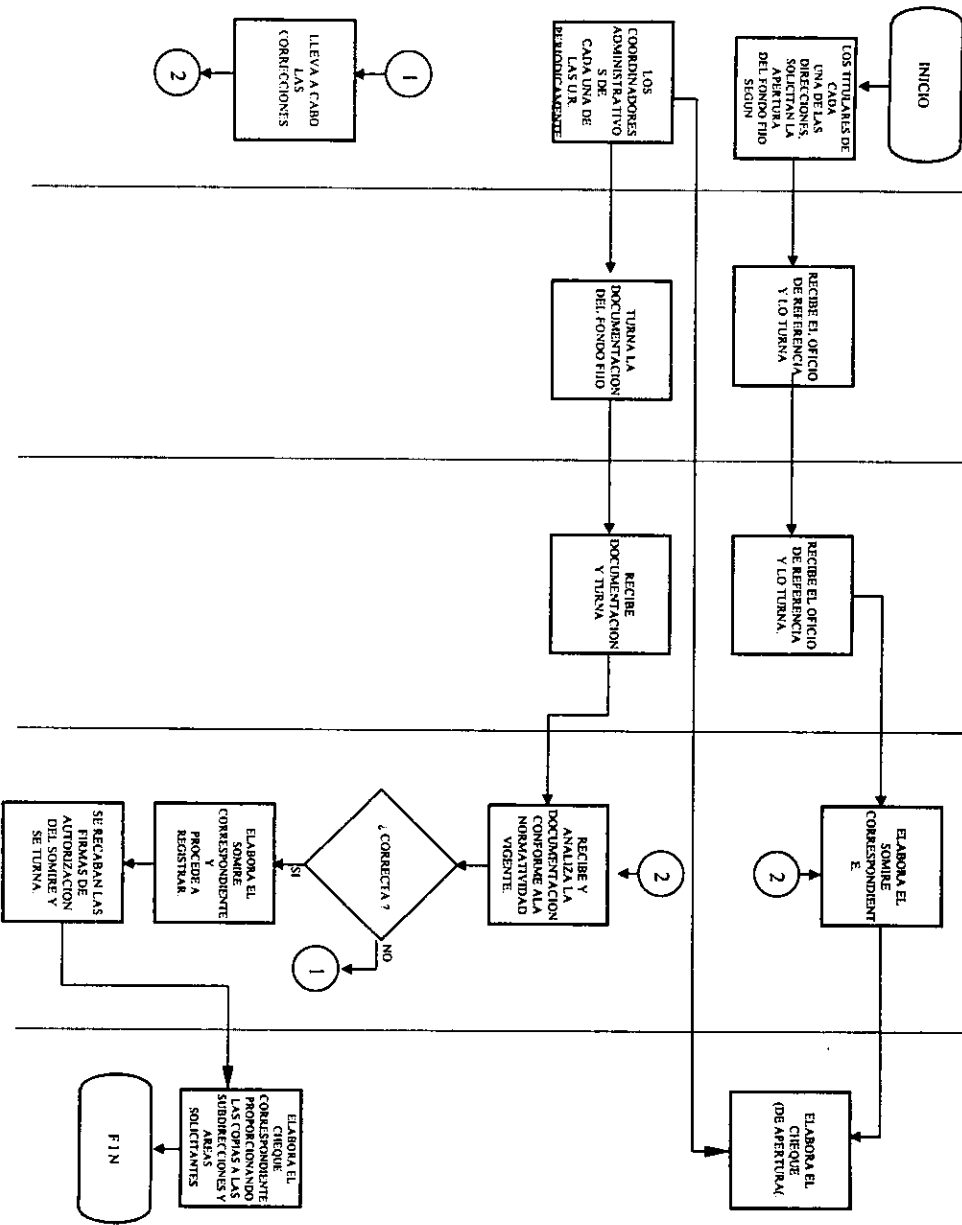
DIRECCIONES EJECUTIVAS DEL I.F.E.

DIRECCIONES DE RECURSOS FINANCIEROS.

SUBDIRECCION DE PASIVOS FIJOS.

DEPARTAMENTO DE PROGRAMACION Y PRESUPUESTACION.

DEPARTAMENTO DE PROGRAMACION FINANCIERA Y TESORERIA.





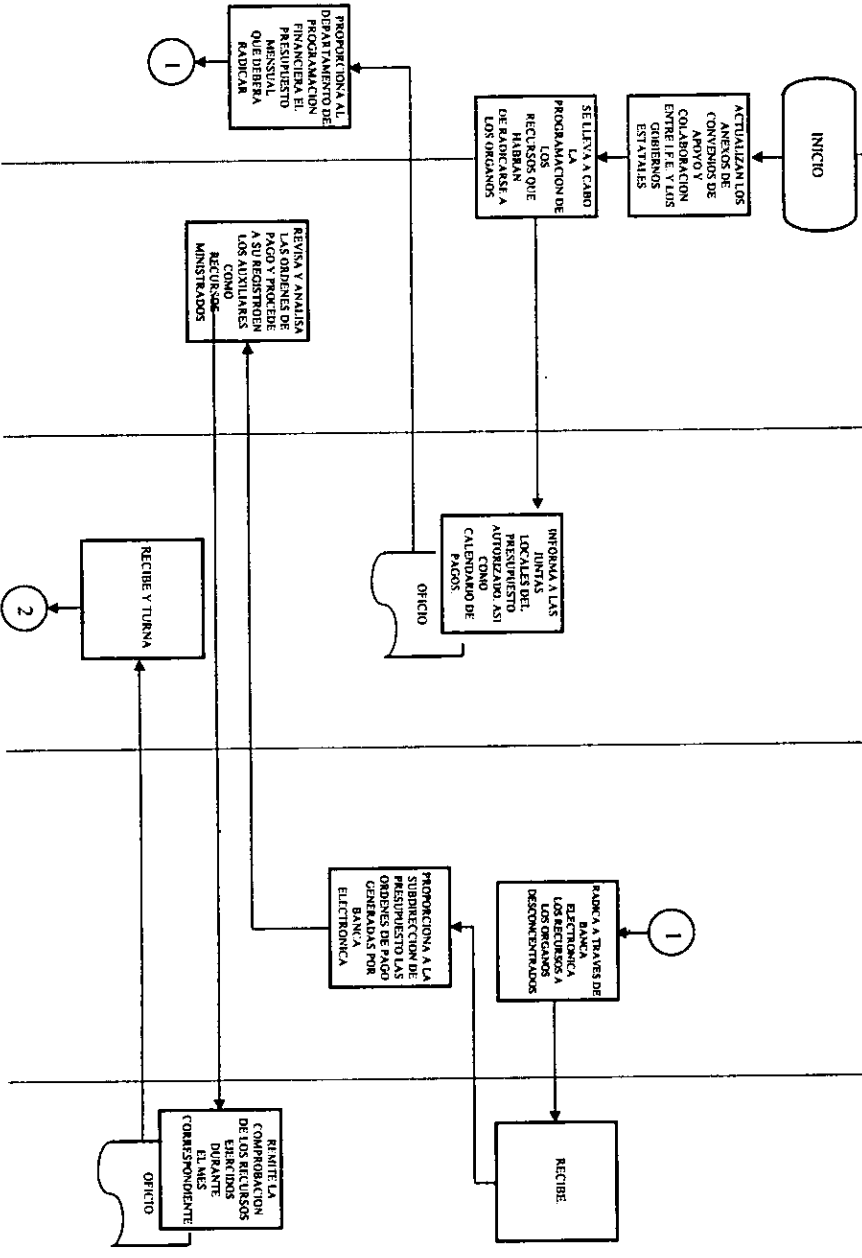
ORGANOS DESCONCENTRADOS JUNTAS LOCALES EJECUTIVAS

SUBDIRECCION DE PRESUPUESTO. DEPARTAMENTO DE CONTROL Y SEGUIMIENTO FINANCIERO.

HOJA : 1/2
DIRECCION DE RECURSOS FINANCIEROS.

DEPARTAMENTO DE PROGRAMACION FINANCIERA Y TESORERIA.

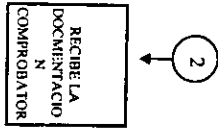
ORGANOS DESCONCENTRADOS.



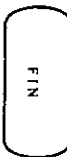
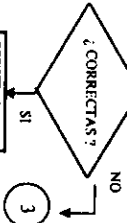
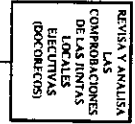


ORGANOS DESCONCENTRADOS JUNTAS LOCALES EJECUTIVAS.

SUBDIRECCION DE PRESUPUESTO.

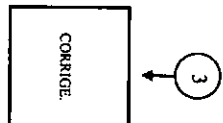


DEPARTAMENTO DE CONTROL Y SEGUIMIENTO FINANCIERO.



HOJA : 2/2
DIRECCION DE RECURSOS FINANCIEROS.

DEPARTAMENTO DE PROGRAMACION FINANCIERA Y TESORERIA.



ORGANOS DESCONCENTRADOS.

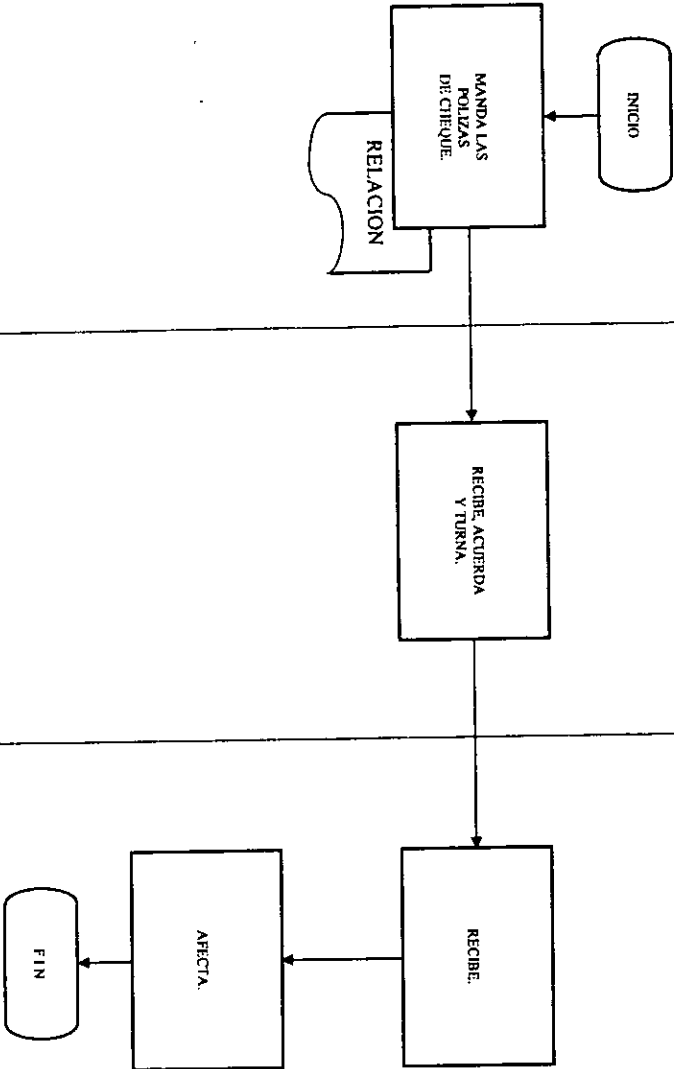


POLIZA - CHEQUE

DEPARTAMENTO DE PROGRAMACION
FINANCIERA Y TESORERIA.

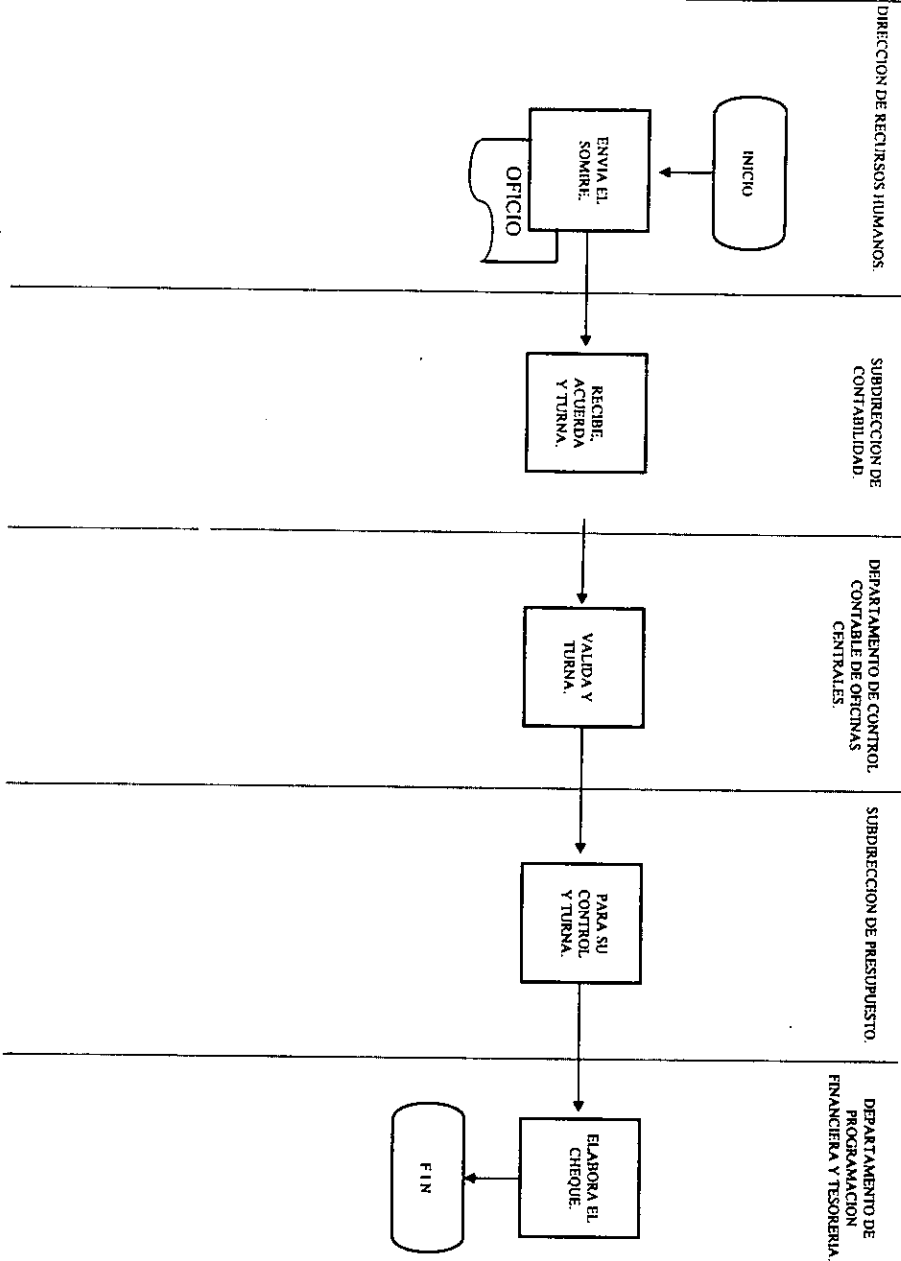
SUBDIRECCION DE CONTABILIDAD.

DEPARTAMENTO DE CONTROL
CONTABLE DE OFICINAS CENTRALES.



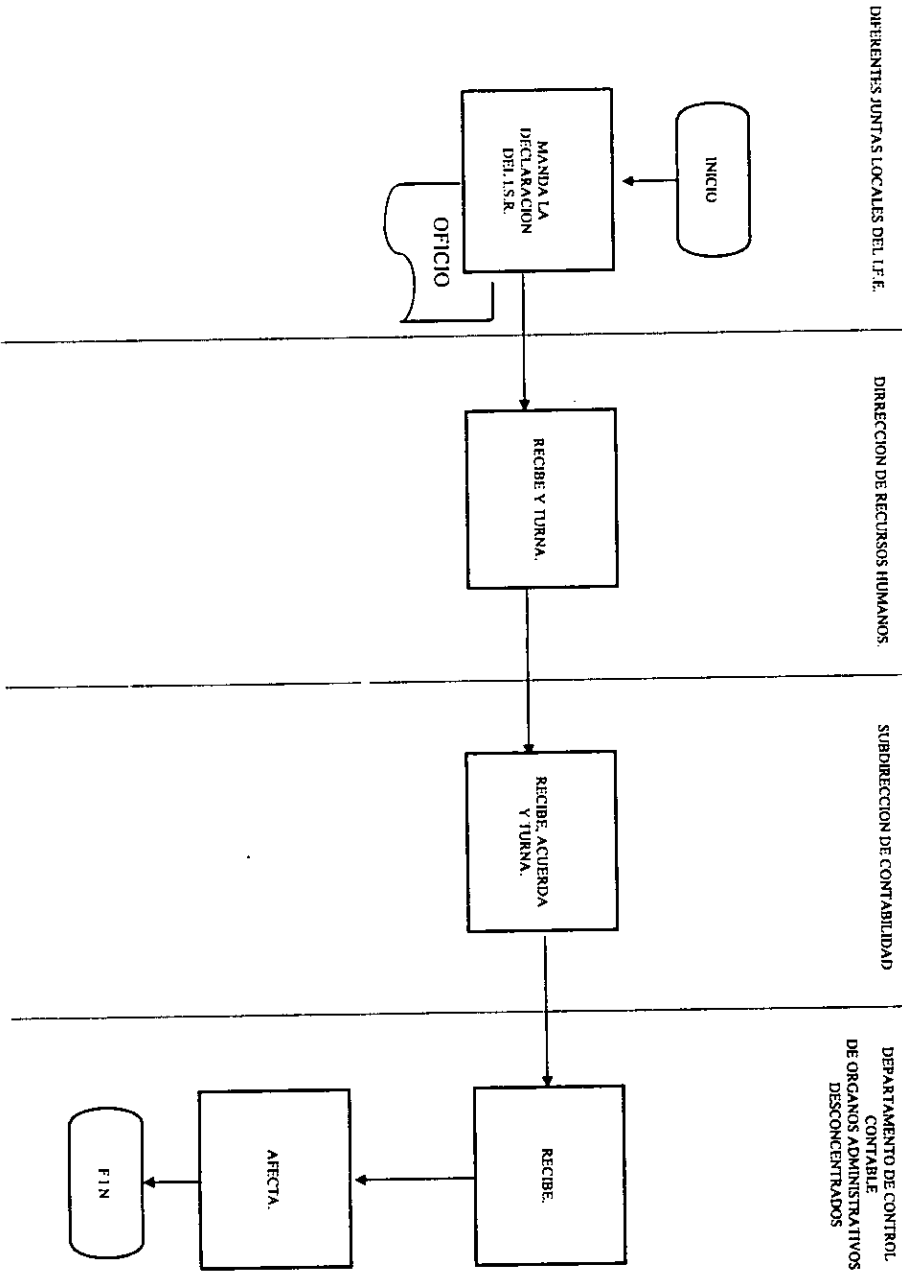


PAGO POR DESCUENTOS Y PERCEPCIONES A FAVOR DE TERCEROS



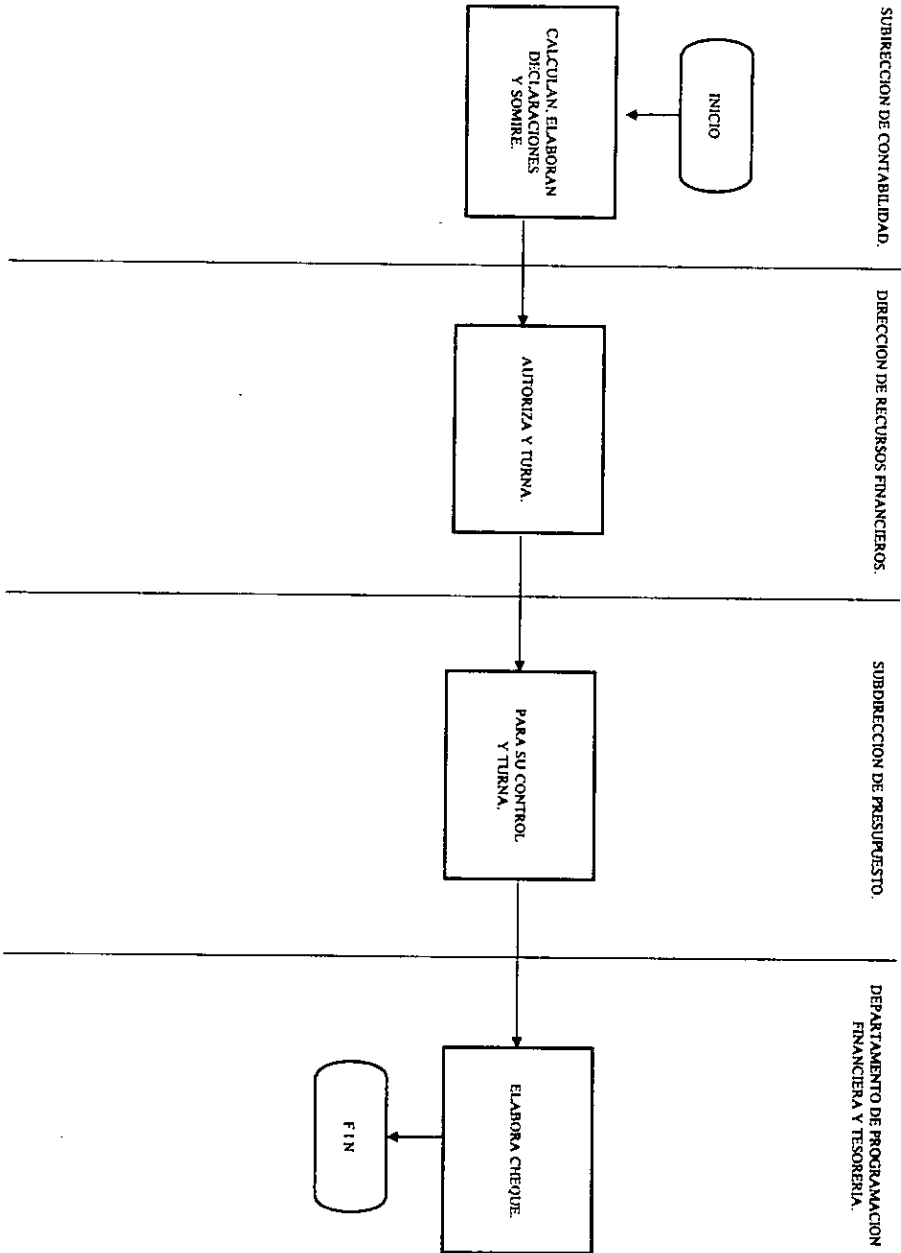


IMPUESTOS SOBRE LA RENTA JUNTAS LOCALES



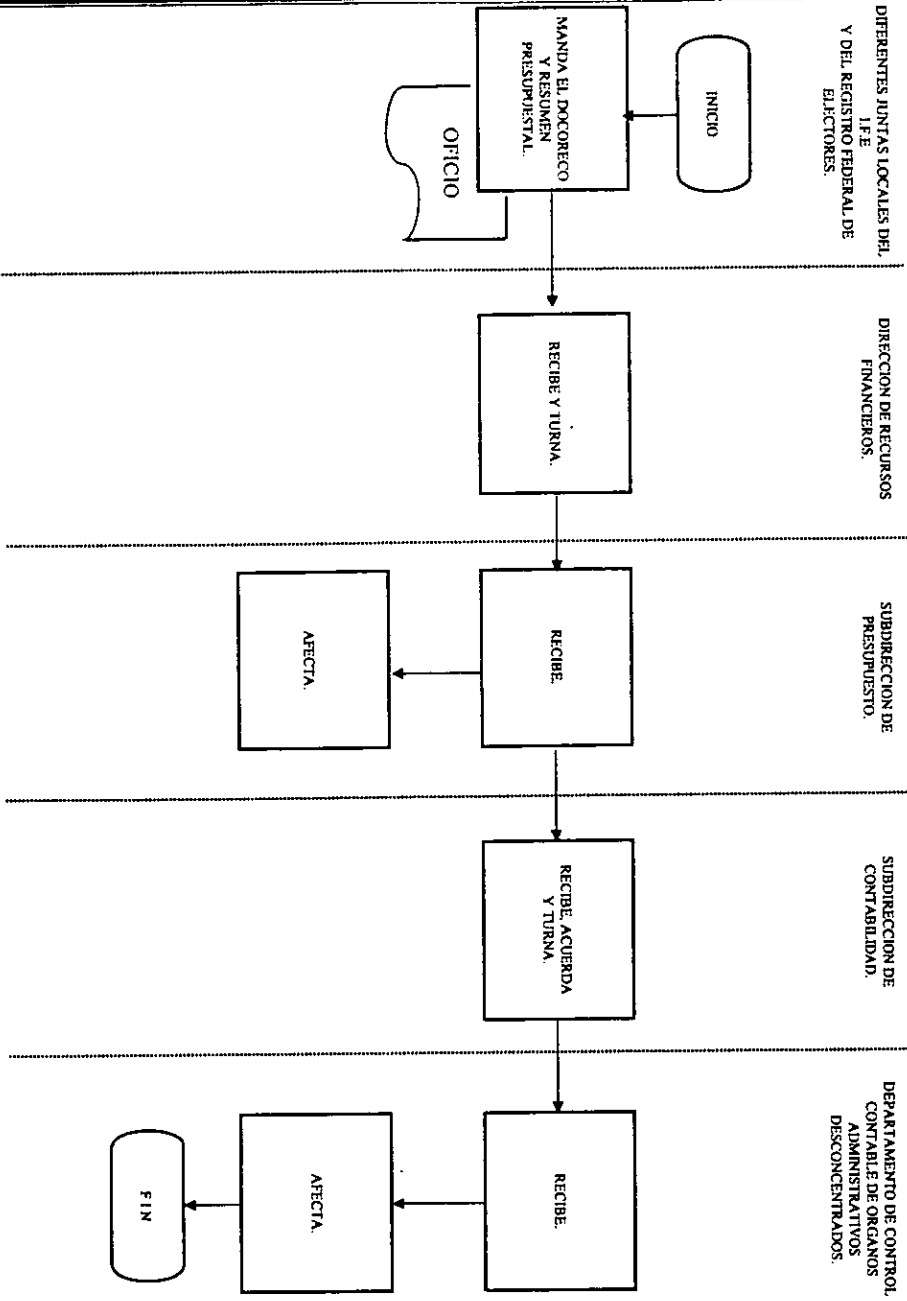


IMPUESTO SOBRE LA RENTA OFICINAS CENTRALES





DOCORECOS



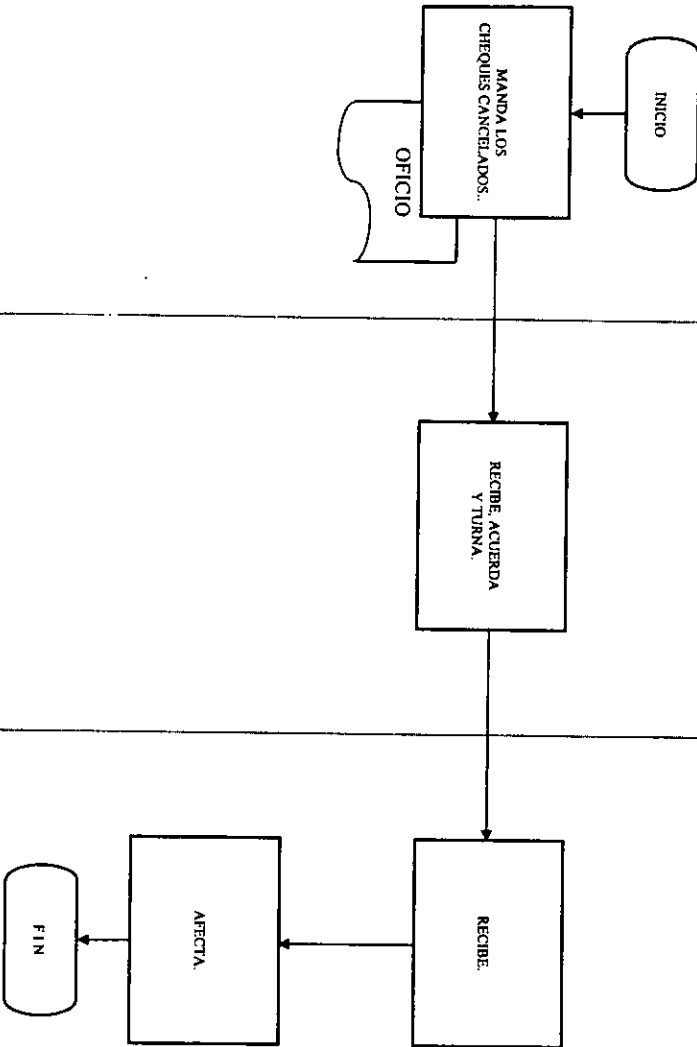


CHEQUES CANCELADOS

DEPARTAMENTO DE PROGRAMACION
FINANCERA Y TESORERIA.

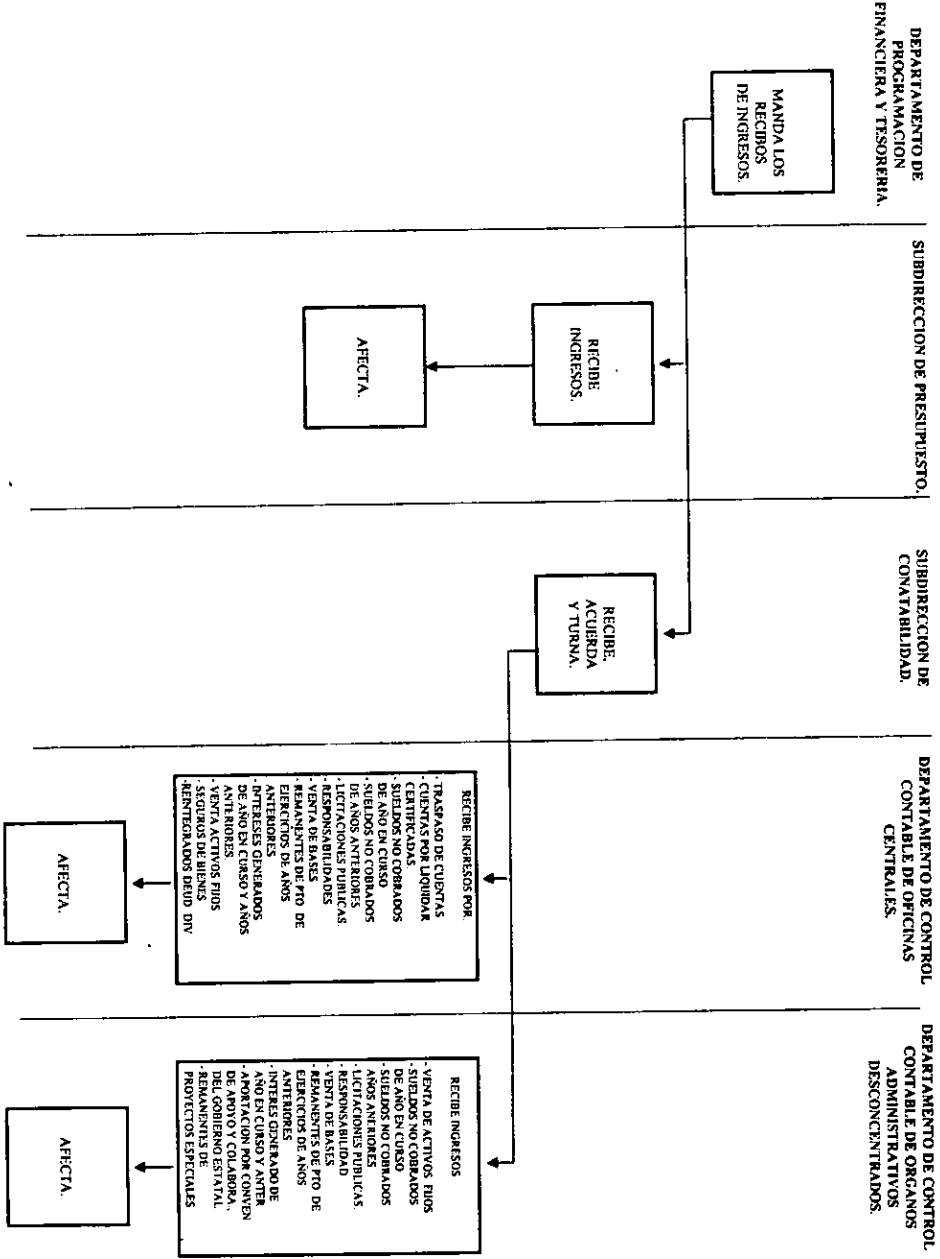
SUBDIRECCION DE CONTABILIDAD.

DEPARTAMENTO DE CONTROL.
CONTABLE DE OFICINAS CENTRALES.



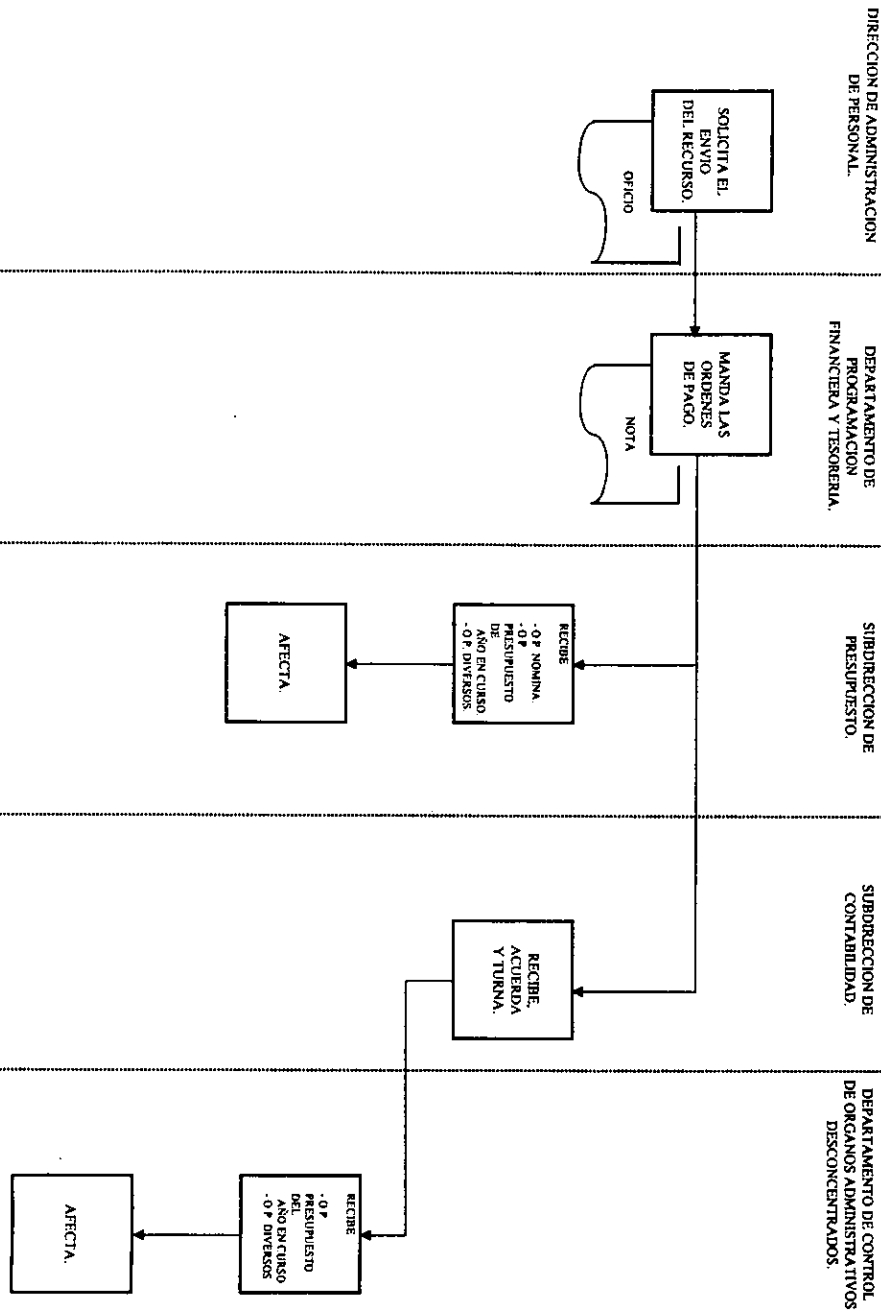


RECIBOS DE INGRESOS



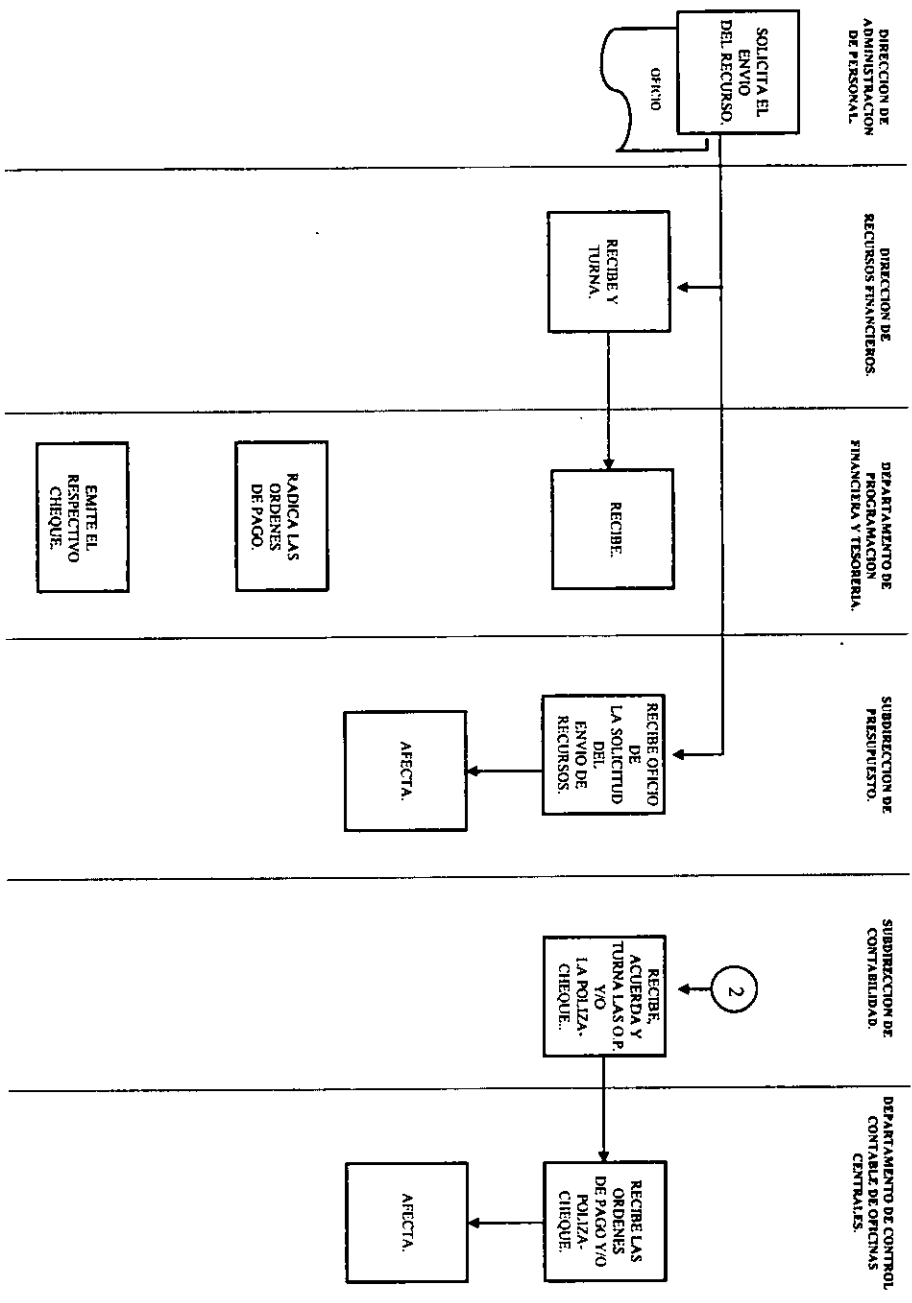


ORDENES DE PAGO



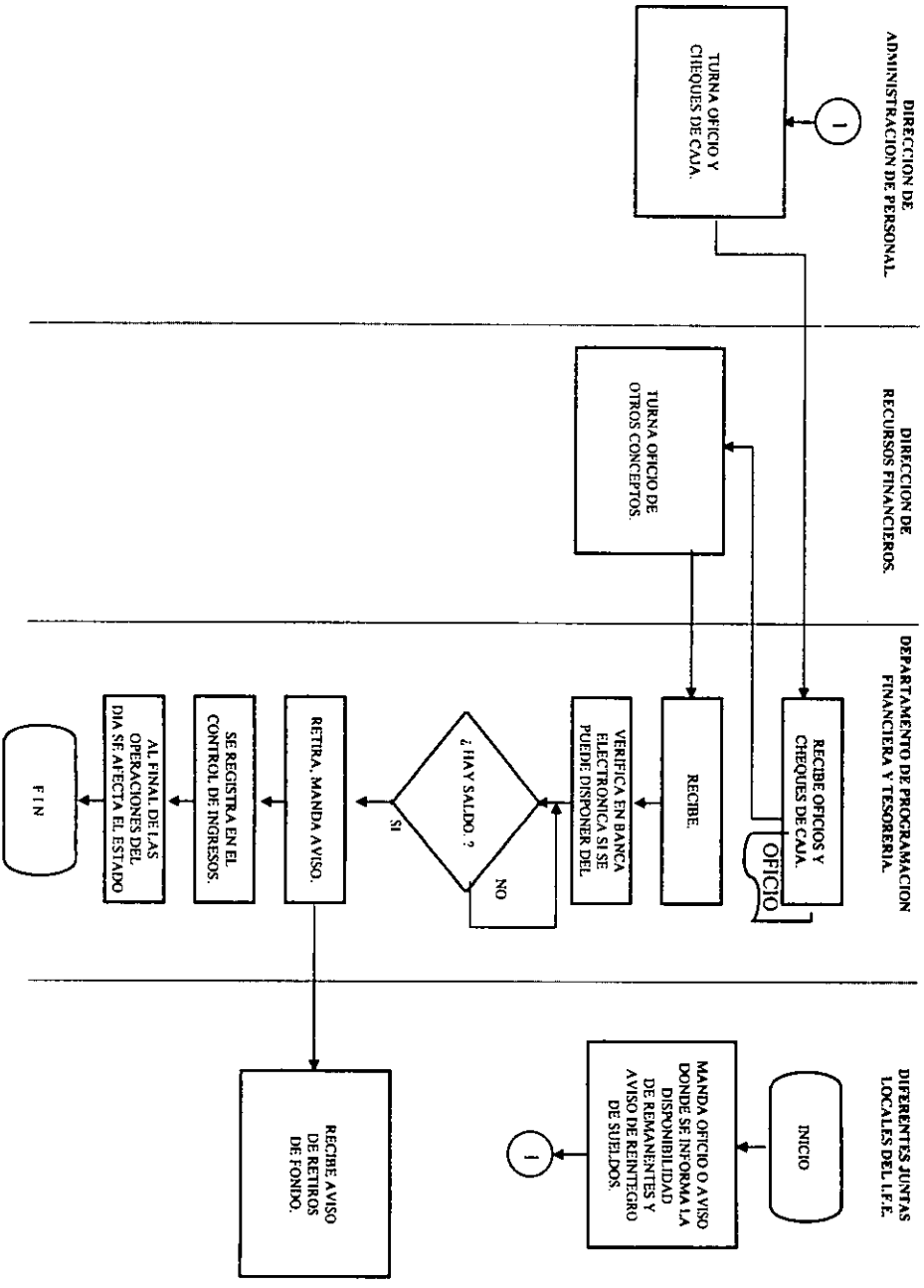


NOMINA





RECUPERACION DE REMANENTES EN OFICINAS CENTRALES Y JUNTAS LOCALES EJECUTIVAS

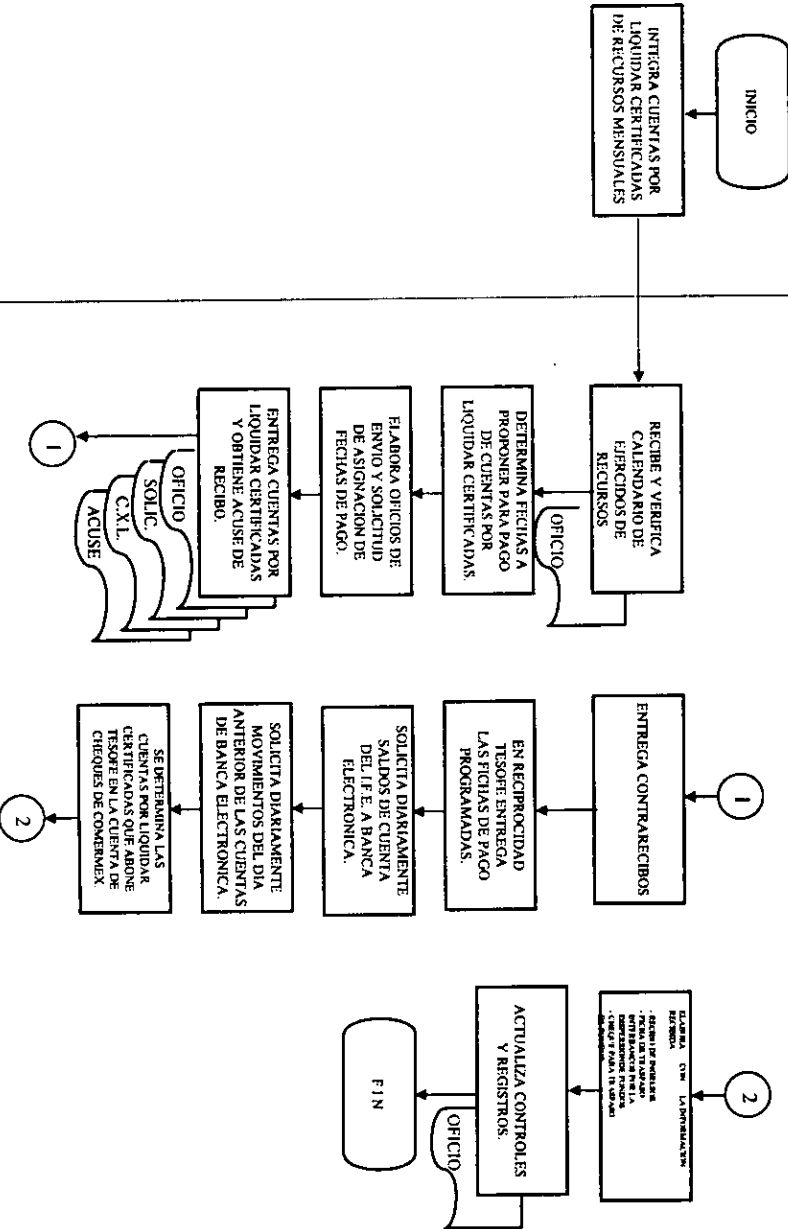




TRAMITACION Y COBRO DE LAS CUENTAS POR LIQUIDAR CERTIFICADAS EN LA TESORERIA DE LA FEDERACION

SUBDIRECCION DE PRESUPUESTO.

DEPARTAMENTO DE PROGRAMACION FINANCIERA.





RECEPCION DE SOLICITUDES DE PAGO DE SOREVI Y SOMIRE

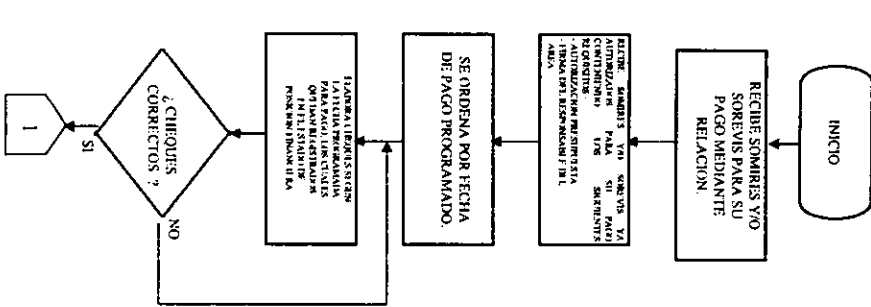
HOJA : 1/2

DEPARTAMENTO DE PROGRAMACION FINANCIERA Y TESORERIA

SUBDIRECCION DE CONTABILIDAD

SUBDIRECCION DE PRESUPUESTO

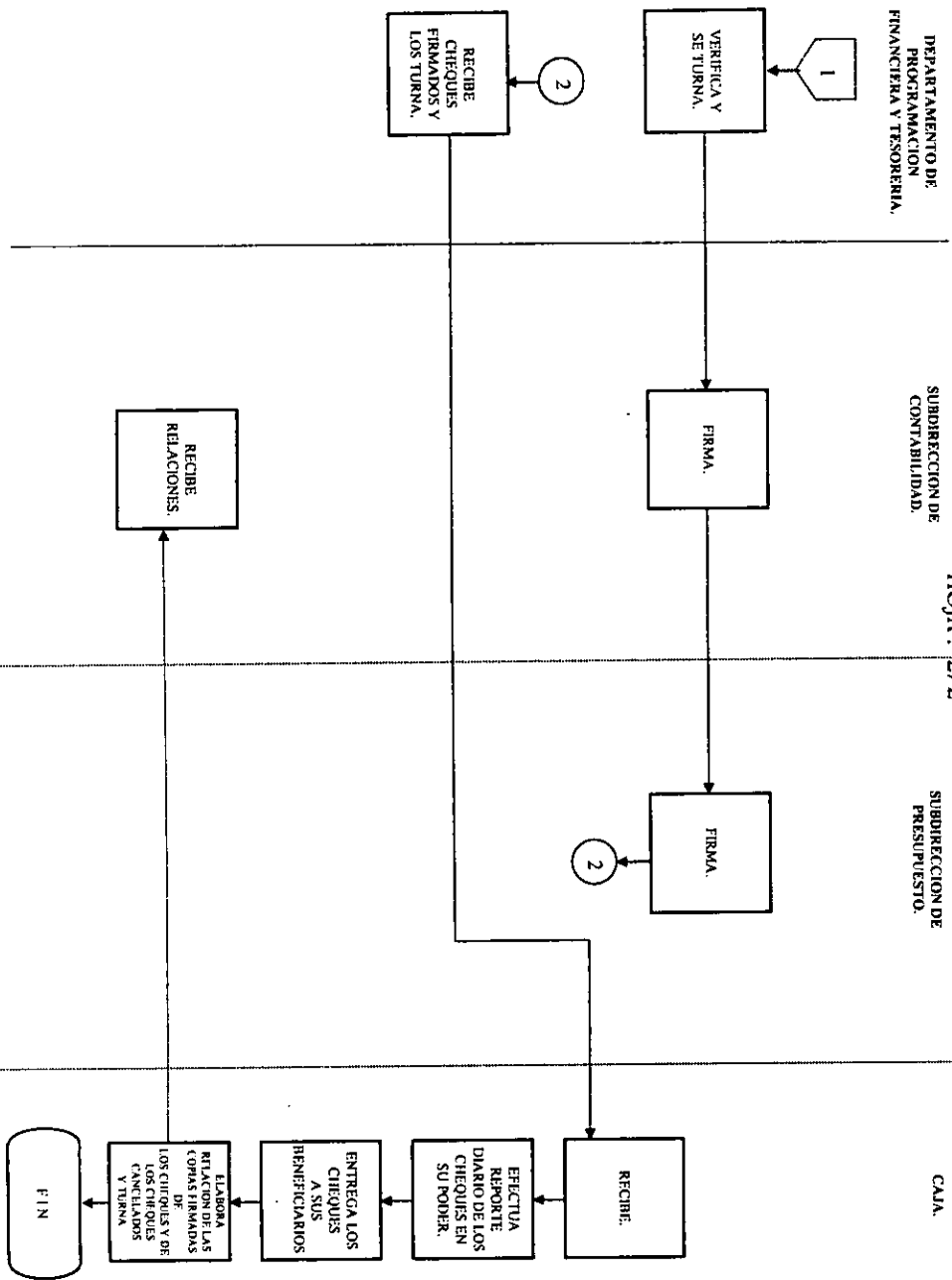
CAJA





RECEPCION DE SOLICITUDES DE PAGO DE SOREVI Y SOMIRE

HOJA : 2/2





PROCESO DE NOMINA

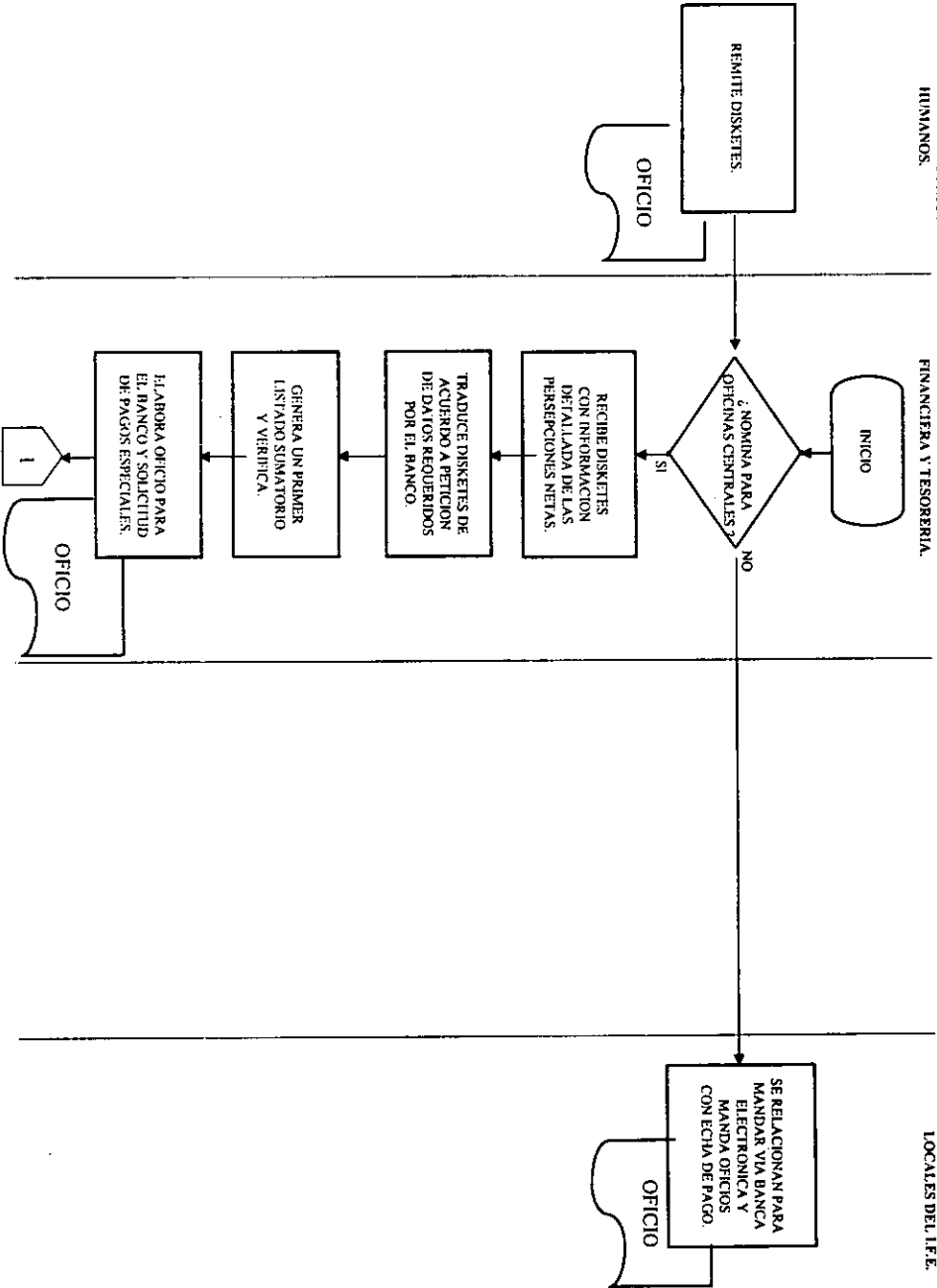
HOJA : 1/2

DIRECCION DE RECURSOS HUMANOS.

DEPARTAMENTO DE PROGRAMACION FINANCIERA Y TESORERIA.

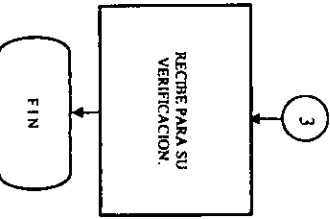
BANCO.

DIFERENTES JUNTAS LOCALES DEL I.F.E.

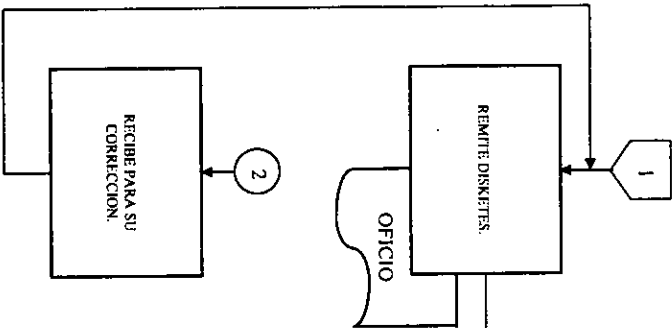




DIRECCION DE RECURSOS
HUMANOS.



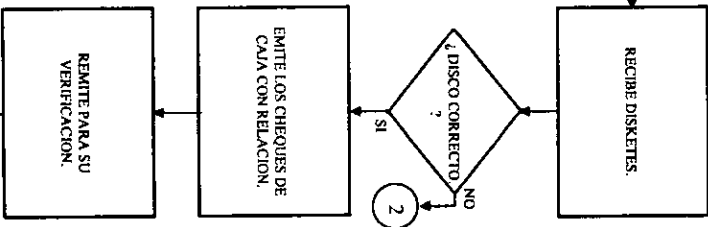
DEPARTAMENTO DE PROGRAMACION
FINANCERA Y TESORERIA.



PROCESO DE NOMINA

HOJA : 2/2

BANCO.

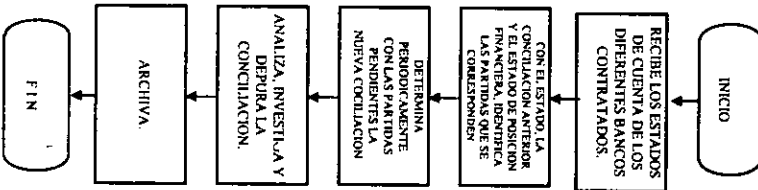


DIFERENTES JUNTAS
LOCALES DEL I.F.E.



CONCILIACIONES BANCARIAS

DEPARTAMENTO DE PROGRAMACION
FINANCIERA Y TESORERIA.



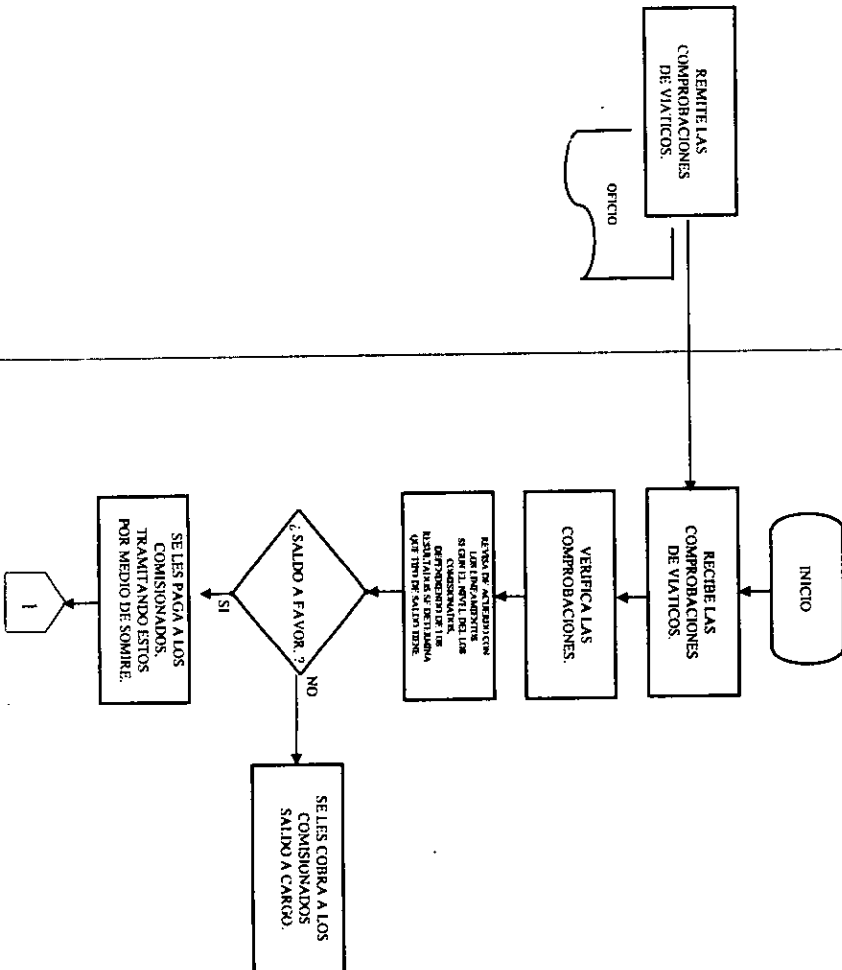


RECEPCION DE COMPROBACIONES DE VIATICOS

DIFERENTES AREAS DEL I.F.E.

HOJA : 1/2

DEPARTAMENTO DE PROGRAMACION
FINANCIERA Y TESORERIA.



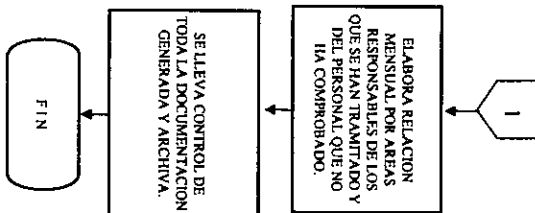


RECEPCION DE COMPROBACIONES DE VIATICOS

DIFERENTES AREAS DEL I.F.E.

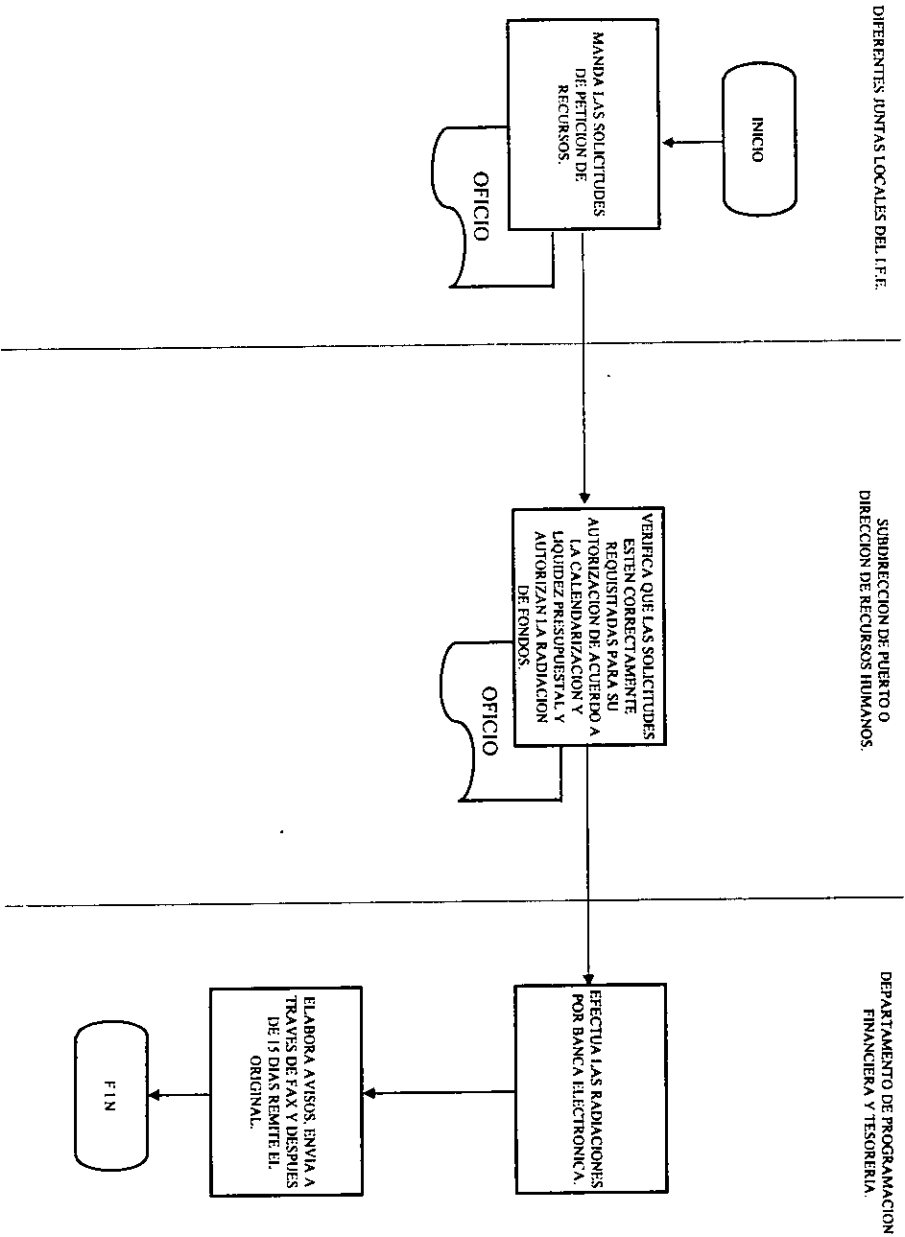
HOJA : 2/2

DEPARTAMENTO DE PROGRAMACION
FINANCIERA Y TESORERIA.





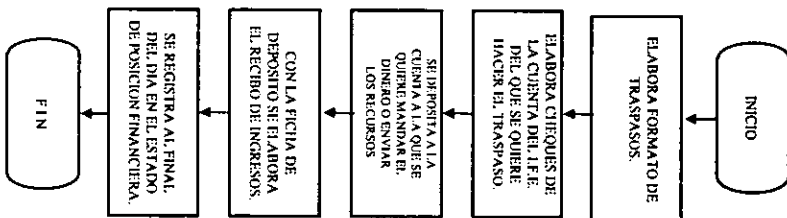
RADICACION DE FONDOS A LAS JUNTAS LOCALES EJECUTIVAS





TRASPASOS

DEPARTAMENTO DE PROGRAMACION
FINANCIERA Y TESORERIA



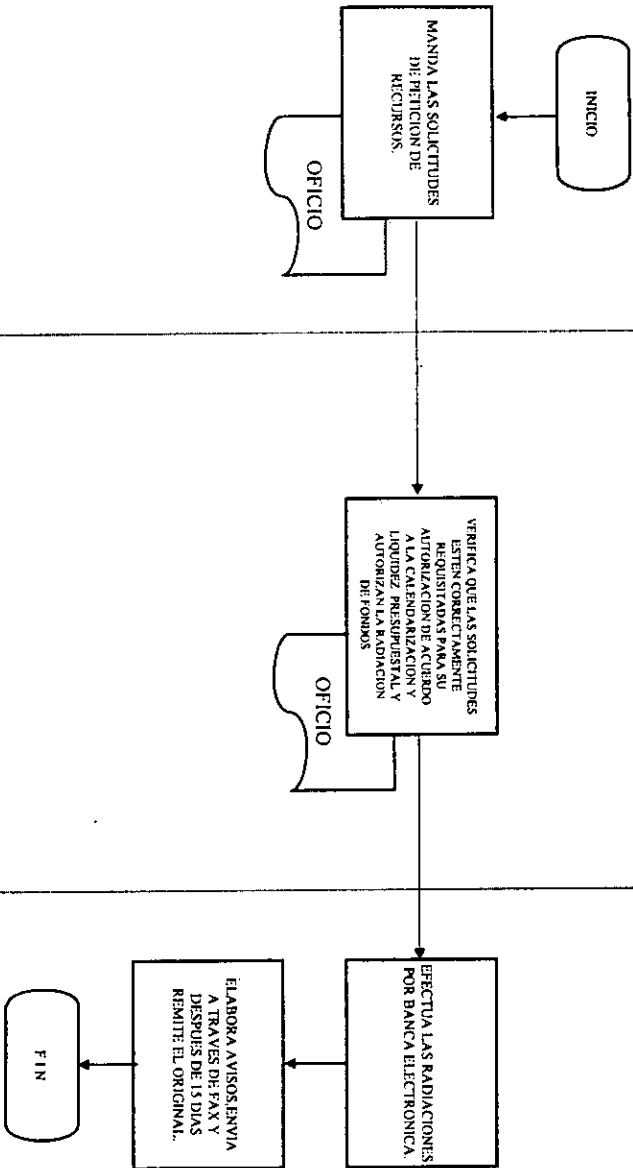


RADICACION DE FONDOS A LAS JUNTAS LOCALES EJECUTIVAS

DIFERENTES JUNTAS LOCALES DEL I.F.E.

SUBDIRECCION DE PUERTO O DIRECCION DE RECURSOS HUMANOS.

DEPARTAMENTO DE PROGRAMACION FINANCIERA Y TESORERIA.





Apendice **B** | Diccionario de Datos

Con base en el diagrama Entidad-Relación que se obtuvo, se describe a continuación la definición de cada una de las entidades que lo conforman

Atributos	Tipo de Datos	Descripción
ADECUACIÓN.		
No. autoriz.	N Numérico (Simple).	Número del documento con que se realiza la Adecuación.
Tipo.	T Texto (3).	La operación es Interna o Externa.
Fecha.	F Fecha/Hora (Fecha Corta).	Fecha en la que se realizó la operación.
Justificación.	M Memo.	Justificación de la Adecuación
Estado.	N Numérico (Simple).	0:Cancelado; 1:Diferido; 2:Aplicado.
BANCO.		
Número de Banco.	C Contador.	Número Consecutivo del Banco.
Nombre del Banco.	T Texto (50).	Razón Social del Banco.
BENEFICIARIO.		
Beneficiario.	T Texto (150).	Nombre del Beneficiario.
ID.	C Contador.	Número Consecutivo del Beneficiario.



Atributos	Tipo de Datos	Descripción
CATALOGO CUENTAS CONTABLES.		
Cuenta.	Texto (16).	Cuenta de Mayor, Subcuenta, SubSubcuenta ó SubSubSub.
Descripcion.	Texto (50).	Nombre de la Cuenta en el Último Nivel.
Naturaleza.	Texto (1).	Deudora ó Acreedora.
Movimiento Actual.	Fecha/Hora (Fecha Corta).	Último Movimiento.
Saldo Inicio.	Moneda.	Saldo Inicial de la Cuenta.
Cargo01.	Moneda.	Cargo Enero.
Abono01.	Moneda.	Abono Enero.
Cargo02.	Moneda.	Cargo Febrero.
Abono02.	Moneda.	Abono Febrero.
Cargo03.	Moneda.	Cargo Marzo.
Abono03.	Moneda.	Abono Marzo.
Cargo04.	Moneda.	Cargo Abril.
Abono04.	Moneda.	Abono Abril.
Cargo05.	Moneda.	Cargo Mayo.
Abono05.	Moneda.	Abono Mayo.
Cargo06.	Moneda.	Cargo Junio.
Abono06.	Moneda.	Abono Junio.
Cargo07.	Moneda.	Cargo Julio.
Abono07.	Moneda.	Abono Julio.
Cargo08.	Moneda.	Cargo Agosto.
Abono08.	Moneda.	Abono Agosto.
Cargo09.	Moneda.	Cargo Septiembre.
Abono09.	Moneda.	Abono Septiembre.
Cargo10.	Moneda.	Cargo Octubre.
Abono10.	Moneda.	Abono Octubre.
Cargo11.	Moneda.	Cargo Noviembre.
Abono11.	Moneda.	Abono Noviembre.
Cargo12.	Moneda.	Cargo Diciembre.
Abono12.	Moneda.	Abono Diciembre.
COMPROBACION.		
Número.	Contador.	Número de la Comprobación.
SOREVI.	Númérico (Entero).	Folio asignado al SOREVI.
rfc benef.	Texto (13).	RFC del Beneficiario.
Nombre.	Texto (50).	Beneficiario.
Lugar y Periodo de Comision.	Texto (150).	Lugar y Periodo de Comisión.
Número de Cheque.	Númérico (Entero).	Número del Cheque.
Tipo de Servidor.	Númérico (Simple).	0:Mando Medio; 1:Operativo.
Constancia.	SI/NO.	SI:Comprobado; No:Sin Comprobar.
Boleto.	Texto (25).	Número del Boleto de Avión.
Importe.	Moneda.	Importe del Sorevi del Comisionado.
Remanente.	Moneda.	Remanente del Importe del Sorevi.
Peajes.	Moneda.	Peajes que se pagaron por medio de un SOMIRE.
Autobus.	Moneda.	Boleto de Autobus que se pagan por medio de un SOMIRE.
Hospedaje.	Moneda.	Hospedaje que se paga por medio de un SOMIRE.
Alimentación.	Moneda.	Alimentación que se paga por medio de un SOMIRE.
Transporte.	Moneda.	Otro tipo de transporte pagado por medio de un SOMIRE.
Tintoreria.	Moneda.	Servicio de Tintoreria pagado por medio del SOMIRE.
Otros.	Moneda.	Otros pagos hechos, pagaderos por medio del SOMIRE.
Reintegro.	Moneda.	Devoluciones hechas al SOMIRE.
Estado.	Númérico (Simple).	0:Cancelado; 1:Diferido; 2:Aplicado.
Observaciones.	Texto (200).	Observaciones realizadas para poder realizar el pago.



Atributos	Tipo de Datos	Descripción
COMPROMETIDO.		
No Compromiso.	Número (Entero).	Número del Compromiso.
Tipo.	Texto (1).	Tipo del Compromiso: Contrato o Pedido.
Programa.	Texto (2).	Programa de la Clave Presupuestal: AA, 5K, 3W, ZZ.
Subprograma.	Texto (2).	Subprograma de la Clave Presupuestal: 01,02,03,04,05,11.
Proyecto.	Texto (2).	Proyecto de la Clave Presupuestal: 00, 11.
Unidad Responsable.	Texto (3).	Unidad Responsable de la Clave Presupuestal: 101, 103, 111, 112, 113, 114, 115, 116, 200, 300.
Partida.	Texto (4).	Partida de la Clave Presupuestal.
Digito Identificador.	Texto (2).	Digito Identificador Clave Presupuestal: 10,11,20,60,61.
Tipo de Pago.	Texto (2).	Tipo de Pago de la Clave Presupuestal: 02, 03, 04, 06.
Fecha.	Fecha/Hora (Fecha Corta).	Fecha en que se finca el Compromiso.
Importe Total.	Moneda.	Importe Total del Compromiso.
Importe Re.tante.	Moneda.	Importe Restante del Compromiso Fincado.
Concepto.	Texto (50).	Concepto del Compromiso.
Beneficiario.	Texto (50).	Nombre del Beneficiario del Compromiso.
Estado.	Numérico (Simple).	0:Cancelado; 1:Diferido; 2:Aplicado.
CONCILIACION.		
Conciliación.	Contador.	Contador de Conciliaciones.
Número de Cuenta.	Texto (50).	Número de Cuenta.
Fecha.	Fecha/Hora (Fecha Corta).	Fecha de Elaboración.
Su Saldo.	Moneda.	Monto.
Nuestro Saldo.	Moneda.	Monto.
Estado.	Numérico (Simple).	0:Cancelado; 1:Diferido; 2:Aplicado.
CORRECCIONES.		
Referencia.	Numérico (Entero).	Número de la Corrección.
Fecha.	Fecha/Hora (Fecha Corta).	Fecha en que se Elabore la Corrección.
Número de Cuenta.	Texto (25).	Número de la Cuenta a Afectar.
Importe.	Moneda.	Importe de la Corrección.
I/E.	Texto (1).	El Importe es un Ingreso ó un Egreso.
Concepto.	Memo.	Motivo por el cual se hace la Corrección.
Tipo de Transaccion.	Texto (2).	Tipo de la Transacción P. Corrección.
Tipo de Cuenta.	Texto (1).	C: Cheques; I: Inversión.
Estado.	Numérico (Simple).	0: Cancelado; 1: Diferido; 2: Aplicado.
CUENTA BANCARIA.		
Número de Banco.	Numérico (Entero Largo).	Número del Banco.
Número de Cuenta.	Texto (50).	Número de Cuenta.
Tipo de Cuenta.	Texto (1).	Tipode Cuenta: (C: Cheque; I: Inversión).
Número de Sucursal.	Numérico (Byte).	Número de Sucursal.
Plaza.	Texto (50).	Plaza de la Sucursal.
Saldo.	Moneda.	Monto Actual de la Cuenta.
OC/JL.	Texto (2).	Manejador de la Cuenta.
Aplicado.	Texto (1).	Indicador si ya fue aplicada la Cuenta Bancaria.



Atributos	Tipo de Datos	Descripción
CHEQUES.		
Número de Banco.	Numérico (Entero Largo).	Número de Banco.
Número de Cuenta.	Texto (50).	Número de Cuenta.
Número de Cheque.	Numérico (Doble).	Número de Cheque.
Fecha..	Fecha/Hora (Fecha Corta).	Fecha de Expedición del Cheque.
Tipo de Cuenta.	Texto (1).	Tipo de Cuenta (C: Cheques ; I: Inversiones).
Beneficiario.	Texto (50).	Proveedor o Beneficiario del Cheque.
Importe.	Moneda.	Importe del Cheque expedido.
Justificación.	Texto (150).	Motivo del Cheque.
Justificaciol.	Texto (50).	Clave Presupuestal a la que afecta el Cheque.
Numero de SOMIRE.	Numérico (Entero Largo).	Folio asignado al SOMIRE.
SOREVI.	Numérico (Entero).	Folio asignado al SOREVI.
rfc_benef.	Texto (13).	RFC Beneficiario.
Folio de Traspaso	Numérico (Entero).	Número del Traspaso.
Cancelacion.	Fecha/Hora (Fecha Corta).	Fecha de Cancelación del Cheque.
Estado.	Numérico (Simple).	0:Cancelado; 1:Diferido; 2:Aplicado.
Poliza.	Si/No.	0:NO ; 1:SI
DETALLE ADECUACION.		
No_ autoriz.	Numérico (Simple)	Número del Documento con que se realiza la Adecuación.
Programa.	Texto (2).	Programa de la Clave Presupuestal: AA, 5K, 3W, ZZ.
Subprograma.	Texto (2).	Subprograma de la Clave Presupuestal: 01,02,03,04,05,11.
Proyecto.	Texto (2).	Proyecto de la Clave Presupuestal: 00, 11.
Unidad Responsable.	Texto (3).	Unidad Responsable de la Clave Presupuestal: 101, 103, 111, 112, 113, 114, 115, 116, 200, 300.
Partida.	Texto (4).	Partida de la Clave Presupuestal.
Digito Identificador.	Texto (2).	Digito Identificador Clave Presupuestal: 10,11,20,60,61.
Tipo de Pago.	Texto (2).	Tipo de Pago de la Clave Presupuestal: 02, 03, 04, 06.
Contador.	Numérico (Simple).	Cantidad de Adecuación.
Fuente.	Numérico (Byte).	Fuente Financiamiento de la Adecuación Presupuestal.
Total.	Moneda.	Total de la Cantidad Asignada en la Modificación.
Modif_enero.	Moneda.	Cantidad de Modificación al Mes de Enero.
Modif_febrero.	Moneda.	Cantidad de Modificación al Mes de Febrero.
Modif_marzo.	Moneda.	Cantidad de Modificación al Mes de Marzo.
Modif_abril.	Moneda.	Cantidad de Modificación al Mes de Abril.
Modif_mayo.	Moneda.	Cantidad de Modificación al Mes de Mayo.
Modif_junio.	Moneda.	Cantidad de Modificación al Mes de Junio.
Modif_julio.	Moneda.	Cantidad de Modificación al Mes de Julio.
Modif_agosto.	Moneda.	Cantidad de Modificación al Mes de Agosto.
Modif_Septiembre.	Moneda.	Cantidad de Modificación al Mes de Septiembre.
Modif_octubre.	Moneda.	Cantidad de Modificación al Mes de Octubre.
Modif_noviembre.	Moneda.	Cantidad de Modificación al Mes de Noviembre.
Modif_diciembre.	Moneda.	Cantidad de Modificación al Mes de Diciembre.



Atributos	Tipo de Datos	Descripción
DETALLE CONCILIACION.		
Contador.	Contador.	Contador Automático
Conciliación.	Númérico (Entero largo).	Número de Conciliación.
Fecha.	Fecha/Hora (Fecha Corta).	Fecha del Movimiento.
Concepto.	Texto (50).	Motivo del Movimiento.
Sus C Argos.	Moneda.	Monto.
Nuestros Cargos.	Moneda.	Monto.
Sus Abonos.	Moneda.	Monto.
Nuestros Abonos.	Moneda.	Monto.
DETALLE DEL SOMIRE.		
Número de Somire.	Númérico (Entero Largo).	Número del SOMIRE.
Programa.	Texto (2).	Programa de la Clave Presupuestal: AA, 5K, 3W, ZZ.
Subprograma	Texto (2).	Subprograma de la Clave Presupuestal: 01,02,03,04,05,11.
Proyecto.	Texto (2).	Proyecto de la Clave Presupuestal: 00, 11.
Unidad Responsable.	Texto (3).	Unidad Responsable de la Clave Presupuestal: 101, 103, 111, 112, 113, 114, 115, 116, 200, 300.
Partida.	Texto (4).	Partida de Gasto de la Clave Presupuestal.
Digito Identificador.	Texto (2).	Digito Identificador Clave Presupuestal: 10,11,20,60,61.
Tipo de Pago.	Texto (2).	Tipo de Pago de la Clave Presupuestal: 02, 03, 04, 06.
Factura.	Texto (10)	Número de la Factura.
Descuentos.	Moneda.	Descuentos del Somire.
Importe Total.	Moneda.	Importe de la Factura.
DETALLE DOCORECO.		
Movimiento.	Contador.	Movimiento del Docoreco.
Número.	Númérico (Byte).	Número del Docoreco.
Entidad.	Texto (2).	JE:Junta Local Estatal; JD:Junta Local Distrital; VO: Vocalías; CC:Centros Regionales de Computo.
Númeroent.	Númérico (Doble).	0:RFE; 1:Aguascalientes... 32:Zacatecas.
Programa.	Texto (2).	Programa de la Clave Presupuestal: AA, 5K, 3W, ZZ.
Subprograma	Texto (2).	Subprograma de la Clave Presupuestal: 01,02,03,04,05,11.
Proyecto.	Texto (2).	Proyecto de la Clave Presupuestal: 00, 11.
Unidad Responsable.	Texto (3).	Unidad Responsable de la Clave Presupuestal: 101, 103, 111, 112, 113, 114, 115, 116, 200, 300.
Partida.	Texto (4).	Partida de Gasto de la Clave Presupuestal.
Digito Identificador.	Texto (2).	Digito Identificador Clave Presupuestal: 10,11,20,60,61.
Tipo de Pago.	Texto (2).	Tipo de Pago de la Clave Presupuestal: 02, 03, 04, 06.
Denominación.	Texto (150).	Concepto de la Partida Presupuestal.
Importe.	Moneda.	Monto de la Comprobación.



Atributos	Tipo de Datos	Descripción
DETALLE INVERSIONES.		
Contador.	Contador.	Número de Movimientos en el día.
Número de Cuenta.	Texto (25).	Número de la Cuenta de Inversión.
Fecha.	Fecha/Hora (Fecha Corta).	Fecha del Movimiento.
Concepto.	Texto (50).	Concepto del Movimiento de la Cuenta de Inversión.
Total.	Moneda.	Total de la Inversión.
Tasa Bancaria.	N Numérico (Simple).	Tasa de Interes Depósito Bancario.
Interes Bancario.	Moneda.	Intereses Generados Depósito Bancario.
Importe Bancario.	Moneda.	Importe Depósito Bancario.
Tasa Gubernamental.	N Numérico (Simple).	Tasa de Interes Valores Gubernamentales.
Interes Gubernamental.	Moneda.	Intereses Generados Valores Gubernamentales.
Importe Gubernamental.	Moneda.	Importe Valores Gubernamentales.
DETALLE SOREVI.		
SOREVI.	N Numérico (Entero).	Folio asignado al SOREVI.
rfc_benef.	Texto (13).	RFC del Beneficiario.
Contador.	N Numérico (Simple).	Cantidad de Personas Comisionadas
Beneficiario.	Texto (50).	Nombre del Comisionado
Destino.	Texto (50).	Destino de la Comisión.
Actividad.	Memo.	Actividades a realizar el Comisionado.
Importe.	Moneda.	Total asignado al Comisionado.
DOCORECO.		
Numero.	N Numérico (Byte).	Fuente de Financiamiento de Adecuación Presupuestal.
Entidad.	Texto (2).	Descripción de la Fuente de Financiamiento.
Númeroent.	N Numérico (Doble).	0:RFE; 1:Aguascalientes... 32:Zacatecas.
Fecha.	Fecha/Hora (Fecha Corta).	Fecha de Recepción del Docoreco.
Estado.	N Numérico (Byte).	0:Cancelado; 1:Diferido; 2:Aplicado.
FUENTE.		
Fuente.	N Numérico (Byte).	Fuente de Financiamiento de Adecuación Presupuestal.
Descripción.	Texto (25).	Descripción de la Fuente de Financiamiento.
Tipo.	Texto (1).	Tipo de Adecuación a aplicar: Interna, Externa, Ambas.
INVERSIONES.		
Número de Cuenta.	Texto (25).	Número de la Cuenta de Inversiones.
Fecha.	Fecha/Hora (Fecha Corta).	Fecha del Movimiento de la Cuenta de Inversiones.
Saldo Inicial.	Moneda.	Saldo al Inicio del Día.
Saldo Final.	Moneda.	Saldo al Final del Día.
Saldo Bancario Inicial.	Moneda.	Saldo Bancario Inicial.
Saldo Bancario Final.	Moneda.	Saldo Bancario Final.
Saldo Gubernamental Inicial.	Moneda.	Saldo Valores Gubernamentales Inicial.
Saldo Gubernamental Final.	Moneda.	Saldo Valores Gubernamentales Final.
Estado.	N Numérico.	0:Cancelado; 1:Diferido; 2:Aplicado.
MES.		
Mes.	N Numérico (Byte).	Número del Mes correspondiente en el Año en Curso.
Descripción.	Texto (10).	Nombre del Mes correspondiente en el Año en Curso.
Cierre.	N Numérico (Simple).	Verificador del cierre en la Tabla Modificado.



Atributos	Tipo de Datos	Descripción
MODIFICADO.		
Programa.	Texto (2).	Programa de la Clave Presupuestal: AA, 5K, 3W, ZZ.
Subprograma.	Texto (2).	Subprograma de la Clave Presupuestal: 01.02.03.04.05.11.
Proyecto.	Texto (2).	Proyecto de la Clave Presupuestal: 00, 11.
Unidad Responsable.	Texto (3).	Unidad Responsable de la Clave Presupuestal: 101, 103, 111, 112, 113, 114, 115, 116, 200, 300.
Partida.	Texto (4).	Partida de la Clave Presupuestal.
Digito Identificador.	Texto (2).	Digito Identificador Clave Presupuestal: 10,11,20,60,61.
Tipo de Pago.	Texto (1).	Tipo de Pago de la Clave Presupuestal: 02, 03, 04, 06.
Status.	Texto (1).	Estado de la Clave Presupuestal: Original o Subregirada.
Fecha_mod.	Fecha/Hora (Fecha Corta).	Fecha de la Modificación.
Pto_ori01.	Moneda.	Presupuesto Original Enero.
Pto_ori02.	Moneda.	Presupuesto Original Febrero.
Pto_ori03.	Moneda.	Presupuesto Original Marzo.
Pto_ori04.	Moneda.	Presupuesto Original Abril.
Pto_ori05.	Moneda.	Presupuesto Original Mayo.
Pto_ori06.	Moneda.	Presupuesto Original Junio.
Pto_ori07.	Moneda.	Presupuesto Original Julio.
Pto_ori08.	Moneda.	Presupuesto Original Agosto.
Pto_ori09.	Moneda.	Presupuesto Original Septiembre.
Pto_ori10.	Moneda.	Presupuesto Original Octubre.
Pto_ori11.	Moneda.	Presupuesto Original Noviembre.
Pto_ori12.	Moneda.	Presupuesto Original Diciembre.
Ampliacio01.	Moneda.	Ampliación Autorizada al Presupuesto Original Enero.
Ampliacio02.	Moneda.	Ampliación Autorizada al Presupuesto Original Febrero.
Ampliacio03.	Moneda.	Ampliación Autorizada al Presupuesto Original Marzo.
Ampliacio04.	Moneda.	Ampliación Autorizada al Presupuesto Original Abril.
Ampliacio05.	Moneda.	Ampliación Autorizada al Presupuesto Original Mayo.
Ampliacio06.	Moneda.	Ampliación Autorizada al Presupuesto Original Junio.
Ampliacio07.	Moneda.	Ampliación Autorizada al Presupuesto Original Julio.
Ampliacio08.	Moneda.	Ampliación Autorizada al Presupuesto Original Agosto.
Ampliacio09.	Moneda.	Ampliación Autorizada al Presupuesto Original Septiembre.
Ampliacio10.	Moneda.	Ampliación Autorizada al Presupuesto Original Octubre.
Ampliacio11.	Moneda.	Ampliación Autorizada al Presupuesto Original Npviembre.
Ampliacio12.	Moneda.	Ampliación Autorizada al Presupuesto Original Diciembre.
Reduccion01.	Moneda.	Reducción Autorizada al Presupuesto Original Enero.
Reduccion02.	Moneda.	Reducción Autorizada al Presupuesto Original Febrero.
Reduccion03.	Moneda.	Reducción Autorizada al Presupuesto Original Marzo.
Reduccion04.	Moneda.	Reducción Autorizada al Presupuesto Original Abril.
Reduccion05.	Moneda.	Reducción Autorizada al Presupuesto Original Mayo.
Reduccion06.	Moneda.	Reducción Autorizada al Presupuesto Original Junio.
Reduccion07.	Moneda.	Reducción Autorizada al Presupuesto Original Julio.
Reduccion08.	Moneda.	Reducción Autorizada al Presupuesto Original Agosto.
Reduccion09.	Moneda.	Reducción Autorizada al Presupuesto Original Septiembre.
Reduccion10.	Moneda.	Reducción Autorizada al Presupuesto Original Octubre.
Reduccion11.	Moneda.	Reducción Autorizada al Presupuesto Original Noviembre.
Reduccion12.	Moneda.	Reducción Autorizada al Presupuesto Original Diciembre.
Pto_act01.	Moneda.	Presupuesto Actual Enero.
Pto_act02.	Moneda.	Presupuesto Actual Febrero.
Pto_act03.	Moneda.	Presupuesto Actual Marzo.
Pto_act04.	Moneda.	Presupuesto Actual Abril.



Pto_act05.	Moneda.	Presupuesto Actual Mayo.
Pto_act06.	Moneda.	Presupuesto Actual Junio.
Pto_act07.	Moneda.	Presupuesto Actual Julio.
Pto_act08.	Moneda.	Presupuesto Actual Agosto.
Pto_act09.	Moneda.	Presupuesto Actual Septiembre.
Pto_act10.	Moneda.	Presupuesto Actual Octubre.
Pto_act11.	Moneda.	Presupuesto Actual Noviembre.
Pto_act12.	Moneda.	Presupuesto Actual Diciembre.
Comprometido01.	Moneda.	Recursos Comprometidos del Presupuesto Enero.
Comprometido02.	Moneda.	Recursos Comprometidos del Presupuesto Febrero.
Comprometido03.	Moneda.	Recursos Comprometidos del Presupuesto Marzo.
Comprometido04.	Moneda.	Recursos Comprometidos del Presupuesto Abril.
Comprometido05.	Moneda.	Recursos Comprometidos del Presupuesto Mayo.
Comprometido06.	Moneda.	Recursos Comprometidos del Presupuesto Junio.
Comprometido07.	Moneda.	Recursos Comprometidos del Presupuesto Julio.
Comprometido08.	Moneda.	Recursos Comprometidos del Presupuesto Agosto.
Comprometido09.	Moneda.	Recursos Comprometidos del Presupuesto Septiembre.
Comprometido10.	Moneda.	Recursos Comprometidos del Presupuesto Octubre.
Comprometido11.	Moneda.	Recursos Comprometidos del Presupuesto Noviembre.
Comprometido12.	Moneda.	Recursos Comprometidos del Presupuesto Diciembre.
Ministrado01.	Moneda.	Recursos Ministrados del Presupuesto Enero.
Ministrado02.	Moneda.	Recursos Ministrados del Presupuesto Febrero.
Ministrado03.	Moneda.	Recursos Ministrados del Presupuesto Marzo.
Ministrado04.	Moneda.	Recursos Ministrados del Presupuesto Abril.
Ministrado05.	Moneda.	Recursos Ministrados del Presupuesto Mayo.
Ministrado06.	Moneda.	Recursos Ministrados del Presupuesto Junio.
Ministrado07.	Moneda.	Recursos Ministrados del Presupuesto Julio.
Ministrado08.	Moneda.	Recursos Ministrados del Presupuesto Agosto.
Ministrado09.	Moneda.	Recursos Ministrados del Presupuesto Septiembre.
Ministrado10.	Moneda.	Recursos Ministrados del Presupuesto Octubre.
Ministrado11.	Moneda.	Recursos Ministrados del Presupuesto Noviembre.
Ministrado12.	Moneda.	Recursos Ministrados del Presupuesto Diciembre.
Ejercido01.	Moneda.	Recursos Ejercidos del Presupuesto Enero.
Ejercido02.	Moneda.	Recursos Ejercidos del Presupuesto Febrero.
Ejercido03.	Moneda.	Recursos Ejercidos del Presupuesto Marzo.
Ejercido04.	Moneda.	Recursos Ejercidos del Presupuesto Abril.
Ejercido05.	Moneda.	Recursos Ejercidos del Presupuesto Mayo.
Ejercido06.	Moneda.	Recursos Ejercidos del Presupuesto Junio.
Ejercido07.	Moneda.	Recursos Ejercidos del Presupuesto Julio.
Ejercido08.	Moneda.	Recursos Ejercidos del Presupuesto Agosto.
Ejercido09.	Moneda.	Recursos Ejercidos del Presupuesto Septiembre.
Ejercido10.	Moneda.	Recursos Ejercidos del Presupuesto Octubre.
Ejercido11.	Moneda.	Recursos Ejercidos del Presupuesto Noviembre.
Ejercido12.	Moneda.	Recursos Ejercidos del Presupuesto Diciembre.
Disponible01.	Moneda.	Recursos Disponibles del Presupuesto Enero.
Disponible02.	Moneda.	Recursos Disponibles del Presupuesto Febrero.
Disponible03.	Moneda.	Recursos Disponibles del Presupuesto Marzo.
Disponible04.	Moneda.	Recursos Disponibles del Presupuesto Abril.
Disponible05.	Moneda.	Recursos Disponibles del Presupuesto Mayo.
Disponible06.	Moneda.	Recursos Disponibles del Presupuesto Junio.
Disponible07.	Moneda.	Recursos Disponibles del Presupuesto Julio.
Disponible08.	Moneda.	Recursos Disponibles del Presupuesto Agosto.
Disponible09.	Moneda.	Recursos Disponibles del Presupuesto Septiembre.
Disponible10.	Moneda.	Recursos Disponibles del Presupuesto Octubre.
Disponible11.	Moneda.	Recursos Disponibles del Presupuesto Noviembre.
Disponible12.	Moneda.	Recursos Disponibles del Presupuesto Diciembre.



--	--	--

Atributos	Tipo de Datos	Descripción
MOVIMIENTOS.		
Número de Cuenta.	Texto (25).	Número de Cuenta del Movimiento.
Número Transacción.	Númerico (Doble).	Número de: Cheque, Radiación, Traspaso del Movimiento.
Tipo Transacción.	Texto (2).	Tipo de Transacción Efectuada: Cheque, Radiación, Traspaso, Recibo de Ingresos.
I/E.	Texto (1).	Tipo del Movimiento: Ingreso o Egreso.
Tipo de Cuenta.	Texto (1).	C:Cheques; I:Inversión.
Fecha.	Fecha/Hora (Fecha Corta).	Fecha en que se realizó el Movimiento.
Justificación.	Memo.	Motivo del Movimiento.
Importe Ing. eso.	Moneda.	Monto de la Cuenta de Ingresos.
Importe Egreso.	Moneda.	Monto de la Cuenta de Egresos.
Conciliado.	Si/No.	Se realizó la Conciliación Si o No.
RECIBO DE INGRESO.		
Número de Cuenta.	Texto (50).	Número de Cuenta.
Número de Recibo.	Númerico (Doble).	Número del Recibo de Ingreso.
Número de Banco.	Númerico (Entero Largo).	Número de Banco.
Tipo de Cuenta.	Texto (1).	Tipo de Cuenta (C:Cheques; I:Inversión).
Fecha.	Fecha/Hora (Fecha Corta).	Fecha en que se generó el Recibo de Ingreso.
Concepto.	Texto (70).	Concepto del Recibo de Ingreso.
Importe.	Moneda.	Importe del Recibo de Ingreso.
Procedencia.	Texto (35).	Lugar de quien manda el Recibo de Ingreso.
Estado.	Númerico (Simple).	0: Cancelado; 1:Diferido; 2:Aplicado.
Cancelacion.	Fecha/Hora (Fecha Corta).	Fecha de Cancelación del Recibo de Ingreso.
REINTEGRO.		
No. Recibo.	Texto (7).	Número de Recibo de Ingresos.
Fecha.	Fecha/Hora (Fecha Corta).	Fecha de Recibo de Ingresos.
Monto.	Moneda.	Monto del Recibo.
Programa.	Texto (2).	Programa de la Clave Presupuestal: AA, 5K, 3W, ZZ.
Subprograma.	Texto (2).	Subprograma de la Clave Presupuestal: 01.02.03,04.05.11.
Proyecto.	Texto (2).	Proyecto de la Clave Presupuestal: 00, 11.
Unidad Responsable.	Texto (3).	Unidad Responsable de la Clave Presupuestal: 101, 103, 111, 112, 113, 114, 115, 116, 200, 300.
Partida.	Texto (4).	Partida de la Clave Presupuestal.
Digito Identificador.	Texto (2).	Digito Identificador Clave Presupuestal: 10,11,20,60,61.
Tipo de Pago.	Texto (1).	Tipo de Pago de la Clave Presupuestal: 02, 03, 04, 06.
Banco.	Texto (50).	Nombre del Banco.
Cuenta.	Texto (50).	Nombre de la Cuenta del Banco.
Concepto.	Texto (50).	Justificación del Reintegro.
Estado.	Númerico (Simple).	0:Cancelado; 1:Diferido; 2:Aplicado.
SALDOS.		
Número de Cuenta.	Texto (25).	Número de Cuenta.
Fecha.	Fecha/Hora (Fecha Corta).	Fecha en la que se Recibió el Saldo.
Tipo de Cuenta.	Texto(1).	Tipo de Cuenta (C:Cheques; I:Inversiones).
Saldo.	Moneda.	Monto Actual de la Cuenta.



Atributos	Tipo de Datos	Descripción
SEGUIMIENTO PRESUPUESTAL.		
Programa.	Texto (2).	Programa de la Clave Presupuestal: AA, 5K, 3W, ZZ.
Subprograma.	Texto (2).	Subprograma de la Clave Presupuestal: 01,02,03,04,05,11.
Proyecto.	Texto (2).	Proyecto de la Clave Presupuestal: 00, 11.
Unidad Responsable.	Texto (3).	Unidad Responsable de la Clave Presupuestal: 101, 103, 111, 112, 113, 114, 115, 116, 200, 300.
Partida.	Texto (4).	Partida de la Clave Presupuestal.
Digito Identificador.	Texto (2).	Digito Identificador Clave Presupuestal: 10,11,20,60,61.
Tipo de Pago.	Texto (1).	Tipo de Pago de la Clave Presupuestal: 02, 03, 04, 06.
Fecha l.	Fecha/Hora (Fecha Corta).	Fecha en que se realizó el Pedido o Contrato.
Pedido.	Numérico (Entero).	Número del Pedido.
Contrato.	Numérico (Entero).	Número del Contrato.
SOMIRE.	Numérico (Entero Largo).	Folio Asignado al SOMIRE.
Sorevi.	Numérico (Entero).	Folio Asignado al SOREVI.
Factura.	Texto (15).	Número de la Factura.
Cheque.	Numérico (Doble).	Número del Cheque.
Fecha de Pago.	Fecha/Hora (Fecha Corta).	Fecha de Expedición del Cheque.
Concepto.	Texto (150).	Justificación del SOMIRE.
Proveedor o Beneficiario.	Texto (150).	Beneficiario del SOMIRE.
Comprometido.	Moneda.	Monto Comprometido del Movimiento.
Ejercido.	Moneda.	Monto Ejercido del Movimiento.
Observaciones.	Texto (50).	Estado en que se realizó el Movimiento.
Mes.	Numérico (Doble).	Mes en que se realizó el Movimiento.
SOMIRE.		
Número de SOMIRE.	Numérico (Entero Largo).	Folio Asignado al SOMIRE.
Número de Compromiso.	Numérico (Entero).	Número del Compromiso que paga el SOMIRE.
Tipo de Compromiso.	Texto (1).	Tipo de Compromiso: Contrato o Pedido.
Fecha de Recepción	Fecha/Hora (Fecha Corta).	Fecha de recepción del SOMIRE.
Proveedor o Beneficiario.	Texto (150).	Beneficiario del SOMIRE.
Fecha de Pago.	Fecha/Hora (Fecha Corta).	Fecha en la cual se pagará el SOMIRE.
Monto.	Moneda.	Importe del SOMIRE.
Concepto.	Memo.	Justificación del SOMIRE.
Estado.	Numérico (Simple).	0:Cancelado; 1:Diferido; 2:Aplicado.
M/E.	Numérico (Byte).	0:Ejercido; 1:Ministrado.
TiPol.	Numérico (Doble).	Tipo de Póliza a Utilizar.
SOREVI.		
SOREVI.	Numérico (Entero).	Folio Asignado al SOREVI.
Recepcion.	Fecha/Hora (Fecha Corta).	Fecha de Recepción del SOREVI.
Pago.	Fecha/Hora (Fecha corta).	Fecha en la cual se pagará el SOREVI.
Programa.	Texto (2).	Programa de la Clave Presupuestal: AA, 5K, 3W, ZZ.
Subprograma.	Texto (2).	Subprograma de la Clave Presupuestal: 01,02,03,04,05,11.
Proyecto.	Texto (2).	Proyecto de la Clave Presupuestal: 00, 11.
Unidad Responsable.	Texto (3).	Unidad Responsable de la Clave Presupuestal: 101, 103, 111, 112, 113, 114, 115, 116, 200, 300.
Partida.	Texto (4).	Partida de la Clave Presupuestal.
Digito Identificador.	Texto (2).	Digito Identificador Clave Presupuestal: 10,11,20,60,61.
Tipo de Pago.	Texto (1).	Tipo de Pago de la Clave Presupuestal: 02, 03, 04, 06.
Total.	Moneda.	Importe del SOREVI.



Estado.	Númerico (Simple).	0:Cancelado; 1:Diferido; 2:Aplicado

Atributos	Tipo de Datos	Descripción
TABLA ANUAL.		
Programa.	Texto (2).	Programa de la Clave Presupuestal: AA, 5K, 3W, ZZ
Subprograma.	Texto (2).	Subprograma de la Clave Presupuestal: 01,02,03,04,05,11.
Proyecto.	Texto (2).	Proyecto de la Clave Presupuestal: 00, 11.
Unidad Responsable.	Texto (3).	Unidad Responsable de la Clave Presupuestal: 101, 103, 111, 112, 113, 114, 115, 116, 200, 300.
Partida.	Texto (4).	Partida de la Clave Presupuestal.
Digito Identificador.	Texto (2).	Digito Identificador Clave Presupuestal: 10,11,20,60,61.
Tipo de Pago.	Texto (1).	Tipo de Pago de la Clave Presupuestal: 02, 03, 04, 06.
Asignacion_enero.	Moneda.	Cantidad de Asignación al Mes de Enero.
Asignacion_febrero.	Moneda.	Cantidad de Asignación al Mes de Febrero.
Asignacion_marzo.	Moneda.	Cantidad de Asignación al Mes de Marzo.
Asignacion_abril.	Moneda.	Cantidad de Asignación al Mes de Abril.
Asignacion_mayo.	Moneda.	Cantidad de Asignación al Mes de Mayo.
Asignacion_junio.	Moneda.	Cantidad de Asignación al Mes de Junio.
Asignacion_julio.	Moneda.	Cantidad de Asignación al Mes de Julio.
Asignacion_agosto.	Moneda.	Cantidad de Asignación al Mes de Agosto.
Asignacion_septiembre.	Moneda.	Cantidad de Asignación al Mes de Septiembre.
Asignacion_octubre.	Moneda.	Cantidad de Asignación al Mes de Octubre.
Asignacion_noviembre.	Moneda.	Cantidad de Asignación al Mes de Noviembre.
Asignacion_diciembre.	Moneda.	Cantidad de Asignación al Mes de Diciembre.
Total.	Moneda.	Total de las Cantidades asignadas en el Año.
TABLA CLAVE.		
Programa.	Texto (2).	Programa de la Clave Presupuestal: AA, 5K, 3W, ZZ
Subprograma.	Texto (2).	Subprograma de la Clave Presupuestal: 01,02,03,04,05,11.
Proyecto.	Texto (2).	Proyecto de la Clave Presupuestal: 00, 11.
Unidad Responsable.	Texto (3).	Unidad Responsable de la Clave Presupuestal: 101, 103, 111, 112, 113, 114, 115, 116, 200, 300.
Partida.	Texto (4).	Partida de la Clave Presupuestal.
Digito Identificador.	Texto (2).	Digito Identificador Clave Presupuestal: 10,11,20,60,61.
Tipo de Pago.	Texto (1).	Tipo de Pago de la Clave Presupuestal: 02, 03, 04, 06.
TABLA DE ERRORES.		
Codigo de Error.	Númerico (Entero Largo).	Número del Error.
Cadena de Error.	Memo.	Descripción del tipo de Error.
TABLA DETALLES POLIZAS.		
Movimiento.	Contador.	Movimiento de la Póliza.
Mes.	Númerico (Byte).	Mes de la Póliza.
Número de Poliza.	Númerico (Entero).	Número de la Póliza.
Tipo de Poliza.	Texto (1).	Tipo de la Poliza Expedida.
Cuenta.	Texto (16).	Cuenta, Subcuenta, SSubcuenta, SSSubcuenta del Cat. de C
Descripcion.	Texto (150).	Descripción de la Combinación de las Cuenta, SubCuenta.
Parcial.	Moneda.	Parcial de la Póliza.
Cargo.	Moneda.	Debe de la Póliza.
Abono.	Moneda.	Haber de la Póliza.



Atributos	Tipo de Datos	Descripción
TABLA DI.		
Digito Identificador.	Texto (2).	Digito Identificador Clave Presupuestal: 10.11.20.60.61.
Tipo de Pago.	Texto (1).	Tipo de Pago de la Clave Presupuestal: 02. 03. 04. 06.
TABLA POLIZAS.		
Mes.	Númérico (Byte).	Mes de la Poliza.
Número de Poliza.	Númérico (Entero).	Numero de la Poliza Expedida.
Tipo de Poliza.	Texto (1).	Tipo de la Poliza Expedida.
Fecha.	Fecha/Hora (Fecha Corta).	Fecha en que se Expidió la Poliza.
Concepto.	Texto (50).	Concepto de la Poliza.
Referencia.	Texto (40).	Referencia de la Poliza.
Estado.	Númérico (Simple).	0:Cancelado; 1:Diferido; 2:Aplicado.
TABLA PR.		
Programa.	Texto (2).	Programa de la Clave Presupuestal: AA, 5K, 3W, ZZ.
Subprograma.	Texto (2).	Subprograma de la Clave Presupuestal: 01.02.03.04.05, 11.
Proyecto.	Texto (2).	Proyecto de la Clave Presupuestal: 00, 11.
Descripcion.	Texto (45).	Descripcion de la tema formada por el Programa Subprograma y Proyecto.
TABLA PTDA.		
Partida.	Texto (4).	Número de la Partida de la Clave Presupuestal Corres.
Descripcion.	Texto (120).	Descripción de la Partida Presupuestal Correspondiente.
TABLA UR.		
Unidad Responsable.	Texto (3).	Unidad Responsable de la Clave Presupuestal: 101, 103, 111, 112, 113, 114, 115, 116, 200, 300.
Descripcion.	Texto (70).	Descripción de las Unidades del I.F.E.
TRASPASO.		
Número Traspaso.	Númérico (Entero Largo).	Número Asignado al Traspaso.
Tipo.	Texto (1).	Tipo: Traspaso o Radicación.
Cuenta Origen.	Texto (50).	Cuenta de la cual se Retirarán Fondos.
Fecha.	Fecha/Hora (Fecha Corta).	Fecha en que se Elaboró el Traspaso.
Referencia.	Texto (15).	Número del Documento que da Instrucción.
Cuenta Destino.	Texto (50).	Cuenta a la cual se Depositarán Fondos.
Importe.	Moneda.	Cantidad Motivo del Movimiento.
Justificacion.	Memo.	Motivo del Traspaso.
Reg/Int.	Númérico (Entero).	Registro Interno del Movimiento.
Estado.	Númérico (Simple).	0.Cancelado; 1:Diferido; 2:Aplicado.
Cancelacion.	Fecha/Hora (Fecha Corta).	Fecha de Cancelación del Traspaso.

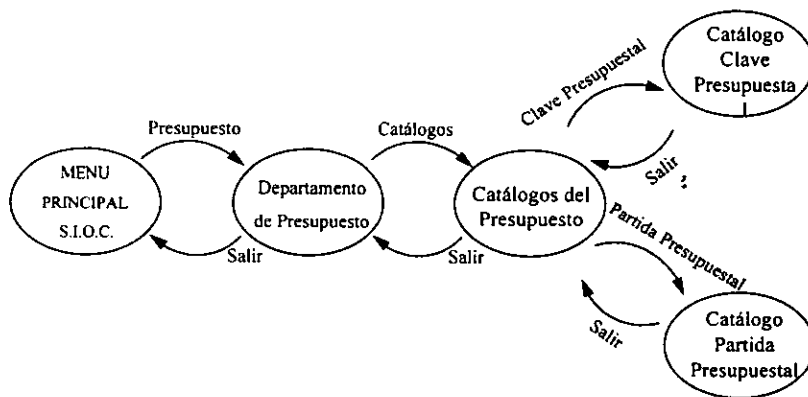


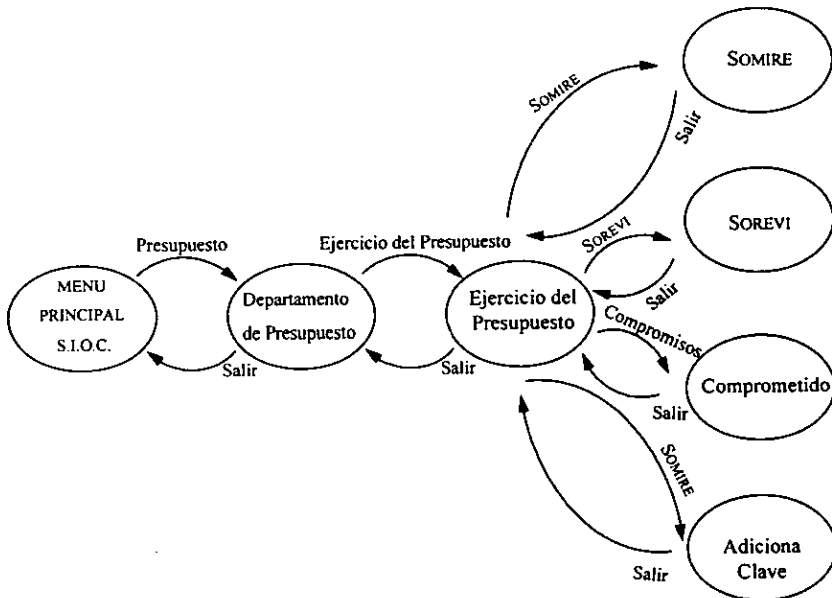
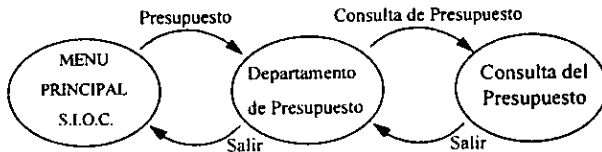
Atributos	Tipo de Datos	Descripción
USO REPORTE AVANCE.		
Programa.	Texto (2).	Programa de la Clave Presupuestal: AA, 5K, 3W, ZZ
Subprograma.	Texto (2).	Subprograma de la Clave Presupuestal: 01.02.03.04.05.11.
Proyecto.	Texto (2).	Proyecto de la Clave Presupuestal: 00, 11.
Unidad Responsable.	Texto (3).	Unidad Responsable de la Clave Presupuestal: 101, 103, 111, 112, 113, 114, 115, 116, 200, 300.
Partida.	Texto (4).	Partida de la Clave Presupuestal.
Digito Identificador.	Texto (2).	Digito Identificador Clave Presupuestal: 10,11,20,60,61.
Tipo de Pago.	Texto (1).	Tipo de Pago de la Clave Presupuestal: 02, 03, 04, 06.
Status.	Texto (1).	Estado de la Clave Presupuestal: Original o Sobregirada.
Fecha_Mod.	Fecha/Hora Fecha Corta).	Fecha de la Modificación.
ORIGINAL.	Moneda.	Presupuesto Original.
AMPLIACIONES.	Moneda.	Ampliación Autorizada al Presupuesto Original.
REDUCCIONES.	Moneda.	Reducción Autorizada al Presupuesto Original.
MODIFICADO.	Moneda.	Presupuesto Actual.
MESCALEN.	Moneda.	Presupuesto Candelarizado Mes.
ACUMUCALEN.	Moneda.	Presupuesto Calendarizado Acumulado.
MESCOMPROME.	Moneda.	Recursos Comprometidos del Presupuesto.
ACUCOMPROME.	Moneda.	Comprometido Acumulado.
MESMINIS.	Moneda.	Recurso Ministrado del Presupuesto.
ACUMINIS.	Moneda.	Ministrado Acumulado.
MESEJER.	Moneda.	Recurso Ejercido del Presupuesto.
ACUEJER.	Moneda.	Ejercido Acumulado.
SUMA.	Moneda.	Suma de Comprometido+Ministrado+Ejercido.
MESXEJER.	Moneda.	Recurso Disponible del Presupuesto.
ACUXEJER.	Moneda.	Ejercido Acumulado.
ANUALXEJER.	Moneda.	Por Ejercer Anual.

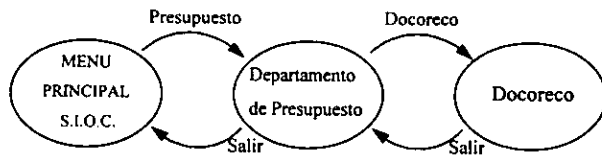
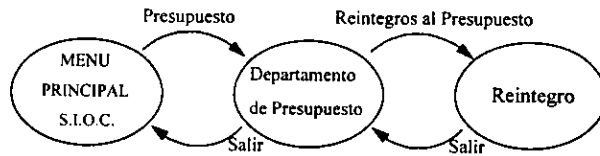
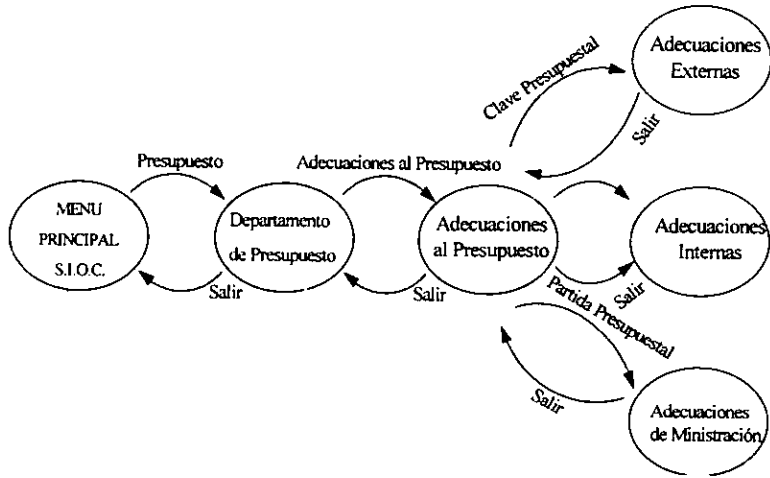


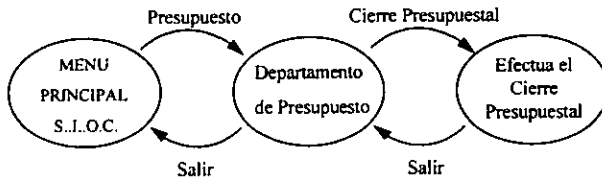
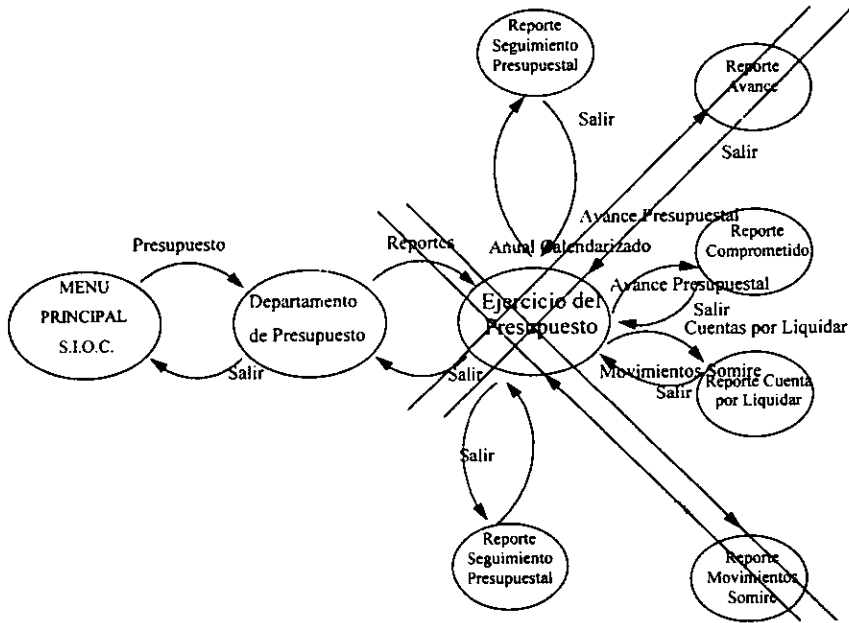
Apendice **C** | Diagramas de Transición de Estados

C.1.- Diagramas de Transición de Estados para el Departamento de Presupuesto.



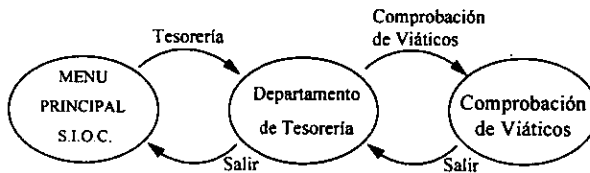
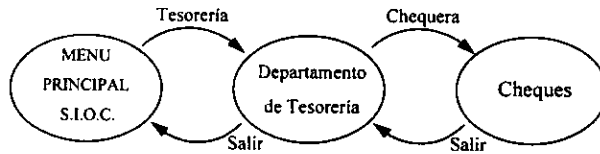
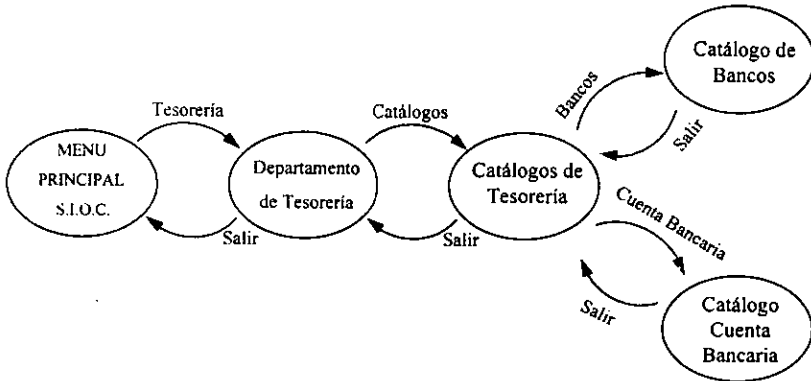


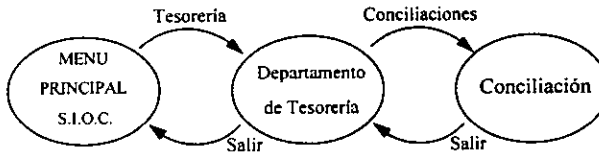
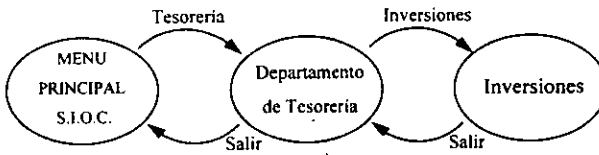
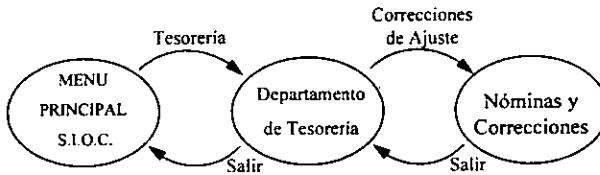
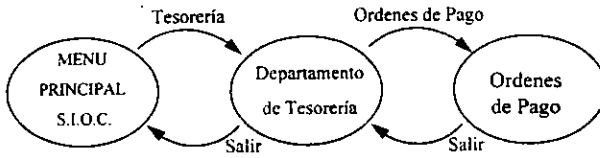


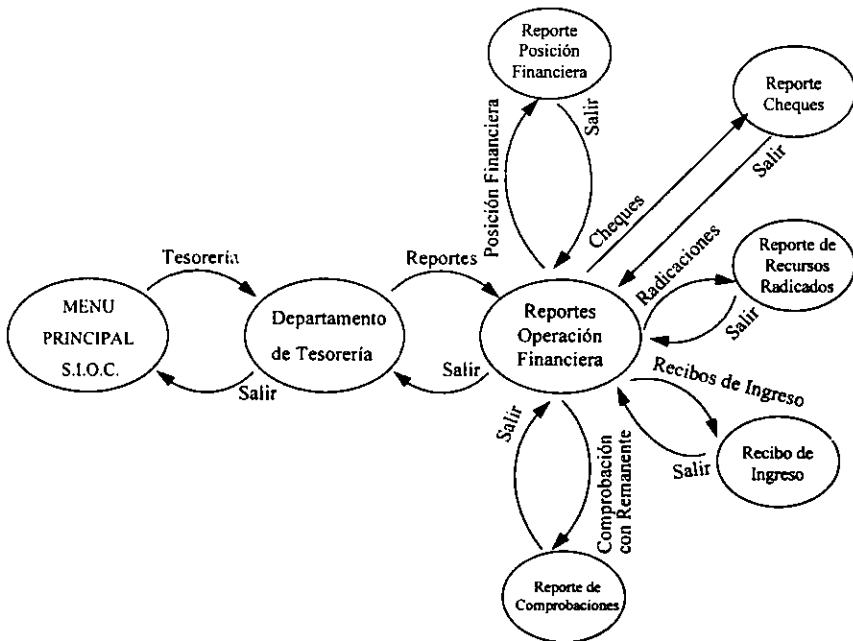
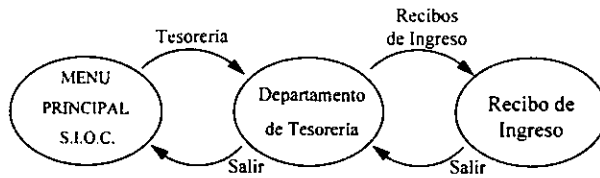




C.2.- Diagramas de Transición de Estados para el Departamento de Tesorería









C.3.- Diagramas de Transición de Estados para el Departamento de Contabilidad

