



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

DESARROLLO DE UN SISTEMA PARA EL
CONTROL DEL EJERCICIO PRESUPUESTAL
EN DEPENDENCIAS DE LA UNAM

T E S I S
QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION
P R E S E N T A N
AÑA M. ALVARADO YEPEZ
NORMA A. VARGAS GONZALEZ

DIRECTOR DE TESIS
ING. SEBASTIAN POBLANO ORDOÑEZ

MEXICO, D.F.

TESIS CON
FALLA DE ORIGEN

1989



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis 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

	Página
INTRODUCCION	4
CAPITULO I. CONCEPTOS GENERALES Y ANTECEDENTES DEL SISTEMA	7
A) Conceptos y definiciones involucrados en el control presupuestal	8
B) Antecedentes del sistema de control presupuestal en la UNAM	17
CAPITULO II. METODOLOGIA DE DESARROLLO	19
A) Objetivo de la metodología	20
B) Técnicas Estructuradas	23
C) Otras metodologías	29
CAPITULO III. ESTUDIO GENERAL DEL SISTEMA DE PRESUPUESTO	32
A) Diagnóstico de la situación actual	33
B) Definición de necesidades	35
C) Definición de alternativas de Solución	37
D) Estudio de factibilidad	39

CAPITULO IV. PLANEACION DEL SISTEMA DE PRESUPUESTO	41
A) Alcances del sistema	42
B) Recursos	43
C) Estructura Organizacional	44
D) Mecanismos de supervisión y control	45
CAPITULO V. ANALISIS DEL SISTEMA DE PRESUPUESTO	46
A) Análisis del sistema en su contexto	47
B) Análisis del sistema actual	49
C) Análisis del sistema a desarrollar	51
D) Diccionario de Datos	58
E) Miniespecificaciones	62
CAPITULO VI. DISEÑO DEL SISTEMA DE PRESUPUESTO	70
A) Carta Estructurada	71
B) Pseudocódigo	85
C) Definición del Software para el desarrollo	100
D) Diseño de Entradas y Salidas	105
E) Diseño de Archivos	127

CAPITULO VII. DESARROLLO DEL SISTEMA DE PRESUPUESTO	132
A) Programación	
1. Codificación	133
2. Archivos auxiliares	136
B) Integración	139
C) Pruebas	143
CAPITULO VIII. DOCUMENTACION, INSTALACION Y MANTENIMIENTO DEL SISTEMA	146
A) Documentación	147
B) Plan de Instalación	148
C) Capacitación	151
CONCLUSIONES	153
BIBLIOGRAFIA	156

I N T R O D U C C I O N

La información es un recurso que cada vez toma mayor importancia para las organizaciones. En esto ha influido el hecho de que el avance de la tecnología ha conseguido incrementar notablemente la velocidad con que viaja la información. Concretamente nos referimos a las innovaciones en el campo de la comunicación de datos y en el procesamiento de la información por medio de la computadora.

El control de la aplicación del presupuesto económico genera un tipo de información que para muchas organizaciones como la Universidad Nacional Autónoma de México, tiene enorme importancia, debido a que en este se basan, para planificar las acciones que se realizarán en un determinado periodo de tiempo.

La Universidad Nacional Autónoma de México, con el objeto de cumplir de una manera satisfactoria su compromiso ante la sociedad, cuenta en la actualidad con más de ciento cincuenta dependencias, donde se incluyen escuelas, facultades, institutos y dependencias administrativas entre otras.

El manejo de la información para el control del presupuesto que requiere la UNAM, debido al tamaño y complejidad de su organización, se realiza a diferentes niveles: a nivel global, o sea, el control de la aplicación de la cantidad total de ingresos, representados por la aportación del gobierno, cuotas y otros conceptos; y a nivel dependencias.

El nivel dependencia, que es el caso en que nos enfocamos, y objeto del presente trabajo, se refiere a la distribución que cada dependencia realiza bajo ciertas políticas y lineamientos generales del presupuesto asignado y el ejercicio del mismo, es decir, cada dependencia universitaria lleva el control presupuestal de la misma, el cual lo puede hacer de una manera manual o automática. Debido a la gran cantidad de información que algunas dependencias manejan y a la forma en que lo hacen, el proceso automático es el más recomendable.

La Dirección General de Servicios de Cómputo para la Administración, es una dependencia universitaria que tiene a su cargo entre otras funciones, asesorar y auxiliar a otras dependencias en el manejo automatizado por computadora de los procesos de la información administrativa, entre las que figura el control presupuestal.

Es así que conjuntamente con la DGSCA se lleva a cabo la realización del presente trabajo que tiene como título DESARROLLO DE UN SISTEMA PARA EL CONTROL DEL EJERCICIO PRESUPUESTAL EN DEPENDENCIAS DE LA UNAM. Dicho sistema pretende facilitar el manejo del control presupuestal a través del uso de la computadora, además de contar con diferentes niveles de ayuda, lo cual permite que el sistema sea más amigable o accesible con el usuario. Por otro lado se tendrá un módulo de autopresentación, con el cual se puede introducir a los nuevos usuarios en el uso del sistema.

La presente tesis tiene como finalidad dar un panorama amplio en cuanto a la metodología de desarrollo del sistema, ésta involucra varias fases en las cuales se describen de manera concisa y detallada los diferentes pasos a seguir.

En el capítulo primero se presentan conceptos y definiciones que se utilizan en el control presupuestal, estos conceptos pretenden crear un campo de conocimiento de la terminología que se emplea a lo largo de este trabajo referido al sistema de control presupuestal.

En el capítulo segundo se hace una semblanza de las metodologías empleadas para el diseño de sistemas de programación, en especial nos referimos a las técnicas estructuradas en las cuales nos apoyamos en este caso.

El estudio general del sistema de control presupuestal, se encuentra contemplado dentro del capítulo tercero, se incluyen puntos tales como: definición del sistema, diagnóstico del sistema actual, definición de necesidades, así como un breve estudio de la factibilidad del mismo.

Lo referente a la planeación del sistema, lo encontramos en el capítulo cuarto, donde se plantean de manera formal los requisitos preliminares de recursos para el desarrollo del sistema y por otro lado un ciclo de vida de éste, además de otros puntos de suma importancia como mecanismos de supervisión y control del proyecto.

El capítulo quinto se enfocó al análisis del sistema, sobre el que se plantea un seguimiento de tipo estructurado. Para ello se emplean diferentes herramientas, que nos proporcionan la metodología utilizada; dicho análisis comprende a grandes rasgos, diagramas de flujos de datos que muestran la estructura que el sistema de presupuesto contempla.

Posteriormente en el capítulo sexto se inicia el diseño propio del sistema. Este consta de varias etapas tales como la presentación en primera instancia, de una carta estructurada donde se describe cada módulo de que consta. Este capítulo es la parte medular del sistema ya que en él se determina el diseño definitivo a desarrollarse.

La parte más práctica la encontramos en el capítulo séptimo, en donde se lleva a cabo la programación del sistema, así como la definición de la estructura de los archivos auxiliares que se utilizarán. Otra parte importante es la presentación de las pruebas una vez integrado el sistema completo, para éstas se emplean datos cuyas generalidades son determinadas.

Una vez concluido el desarrollo, en el capítulo octavo se presenta un manual del usuario el cual tiene como objeto proporcionarle a éste, un documento que le guíe a lo largo de la manipulación del sistema. También se incluye en este capítulo el manual de operación que tiene como finalidad proporcionar la información necesaria acerca de la manera en que opera el sistema de presupuesto en la práctica.

El plan de instalación del sistema, así como el mantenimiento que este debe tener a lo largo de su ciclo de vida se incluye en el capítulo noveno.

En el rubro de conclusiones se describen los resultados generales del desarrollo, tales como la forma en que funcionó la metodología, los problemas que se presentaron y como se solucionaron, entre otras cosas.

Finalmente, se presenta una lista de todas las referencias bibliográficas que se requirieron para la realización de esta tesis.

Cabe hacer notar que aunque el sistema está enfocado para ser operado en dependencias universitarias, debido a la estandarización que existe con respecto a la forma en que se controla el ejercicio presupuestal, dicho sistema puede ser utilizado en cualquier dependencia que lleve el control presupuestal por programas.

C A P I T U L O I

CONCEPTOS GENERALES Y ANTECEDENTES DEL SISTEMA

Antes de iniciar el desarrollo del sistema, es importante conocer la definición o concepto de presupuesto, entender la utilidad y necesidad del control presupuestal, identificar la técnica de presupuesto por programas como una herramienta para dicho control y conocer la terminología utilizada en la mencionada técnica, por ser la base del sistema a desarrollar.

Por otro lado, debido a que el presupuesto y su control, no es nada nuevo, es conveniente presentar un panorama general de como se ha llevado a cabo este proceso en la UNAM.

Los puntos anteriormente mencionados son los que se desarrollan en el presente capítulo dividido en dos incisos, el primero se refiere a los conceptos y el segundo a los antecedentes.

A) CONCEPTOS Y DEFINICIONES INVOLUCRADOS
EN EL CONTROL PRESUPUESTAL

PRESUPUESTO

El presupuesto, en un sentido amplio, se puede definir como el instrumento que utiliza el sector público para calcular sus gastos y el consecuente financiamiento de tales gastos con el objeto de cumplir con sus fines en su carácter de regulador de la vida social y de su función de impulsar el desarrollo económico y social de un país.

La definición anterior se puede aplicar a lo que es el presupuesto en la UNAM, ya que los objetivos de ésta, son muy amplios y de carácter nacional.

PRESUPUESTO TRADICIONAL

El presupuesto, no siempre tuvo la importancia que tiene en la actualidad, ésta se incrementó a medida que las funciones del Estado se fueron acrecentando, esto último como consecuencia de la quiebra en la práctica de la teoría económica liberal, la cual, en sus expresiones más extremas, asignaba al Estado únicamente la función de prestar servicios públicos esenciales, tales como la impartición de justicia, el resguardo de las fronteras a través del ejército y los servicios administrativos generales. Conforme a esta teoría la libre acción de productores y consumidores crearían un equilibrio armónico en la economía. Sin embargo, la historia demostró lo erróneo de esta teoría con la crisis del año de 1929 y las depresiones periódicas de la economía, esto obligó a revisar los conceptos tradicionales de la economía clásica y el papel del Estado. Los economistas vieron en el gasto público una variable instrumental de equilibrio entre la oferta y la demanda globales. En este contexto, la importancia del Estado creció extraordinariamente y de manera paralela a la de las finanzas públicas. El estudio y forma de obtención de los ingresos monetarios y de la utilización de estos ingresos por el poder público para un determinado periodo, pasó a ocupar un plano muy importante.

Dentro de la Teoría de Finanzas Públicas, el presupuesto es definido como un documento financiero que prevé los ingresos y egresos públicos durante un periodo determinado, que generalmente es de un año y representa por tanto el instrumento puesto en manos de los servidores públicos para alcanzar metas y objetivos.

En el presupuesto tradicional, lo fundamental reside en las diversas formas en que se clasifican las magnitudes monetarias. La clasificación básica que se incluyen en éste son: por objeto del gasto, administrativa, funcional y económica.

PRESUPUESTO POR PROGRAMAS

El empleo del presupuesto por programas se concibió esencialmente para servir de instrumento a fin de cubrir las deficiencias que presenta el presupuesto tradicional. La principal deficiencia es que se sabe cuánto se gastará, pero no se tiene en el documento presupuestario elementos para juzgar qué se va a obtener a cambio de los fondos asignados. La idea inicial del presupuesto por programas, tuvo su origen en los Estados Unidos en la década de los cuarentas, y ahí mismo tuvo sus primeras aplicaciones. El concepto de programación, se refiere a una metodología para tomar decisiones, y elegir alternativas entre objetivos y medios.

El presupuesto por programas es un proceso integrado de formulación ejecución y evaluación de decisiones, tendiente a lograr una mayor racionalización de la función administrativa, aparte de una clara definición de objetivos y metas, que conlleva a la determinación de las acciones alternativas que pueden ejecutarse; la selección de las más adecuadas; su agrupación en programas; su cuantificación en función del destino del gasto; así como una clara determinación de funciones y responsabilidades.

Los principios esenciales del presupuesto por programas son:

- + Universalidad (inclusión de todos los ingresos y egresos)
- + Unidad (un solo documento)
- + Previsión
- + Periodicidad (programas a corto plazo)
- + Programación (establecer objetivos y metas con los medios y plazos para lograrlos)
- + Requisitos Formales (claridad, publicidad, exactitud, especificación, exclusividad)

Existen dos tipos de programas de acuerdo a su naturaleza, los programas de funcionamiento y los programas de inversión, los primeros están destinados a producir un servicio como es el caso de la UNAM, y los segundos tienen como finalidad producir un bien.

Los niveles programáticos de asignación de recursos en programas de funcionamiento son:

Función.- Es la finalidad establecida por los órganos, políticas o propósitos generales de la Institución. En la UNAM esos propósitos son los referentes a la Docencia, Investigación, Extensión Universitaria y Apoyo.

Programa.- Comprende el conjunto de actividades relativamente homogéneas encaminadas a cumplir con los propósitos genéricos expresados en una función por medio del establecimiento de objetivos y metas a los cuales se asignan recursos humanos, materiales y financieros, los cuales son administrados por una unidad ejecutora.

Subprograma.- Es la desgregación de un programa complejo con el propósito de mostrar la naturaleza diversa de las metas y costos que se expresan en un determinado programa.

Actividad.- Esta es una división más reducida de cada una de las acciones que se deben llevar a cabo para el logro de los objetivos y metas expresados en los programas.

Tarea.- Es una operación que se ejecuta para alcanzar las metas y objetivos de un resultado determinado.

Concretamente para las actividades científicas y tecnológicas, se maneja el nivel de:

Unidad Responsable.- Se refiere a niveles de responsabilidad y decisión que dentro de una entidad ejecuten o coordinen: actividades de investigación y desarrollo experimental, educación superior, difusión de conocimientos científicos y tecnológicos y actividades asociadas.

Por otro lado, los niveles programáticos en programas de inversión son: Programa, subprograma, proyecto, obra y operación.

En general e independientemente del tipo de programa, las etapas del presupuesto por programas son las siguientes: formulación, aprobación, ejecución, control presupuestario y evaluación de resultados.

El programa es el instrumento central del Sistema de Presupuesto por Programas, a través de éste se plantean las necesidades y se analizan posibilidades, se asignan los recursos, se ejecutan las acciones, se contabiliza el gasto y se evalúa el cumplimiento de las metas. El programa se integra de los siguientes elementos:

Objetivo. - Es la expresión cualitativa de los fines que se pretenden alcanzar. Los objetivos deben ser claros, realistas, lógicos y susceptibles de ser cuantificados, a través de una unidad de medida.

Meta. - Es la representación numérica del resultado final que se pretende alcanzar para lograr el objetivo planteado.

Unidad de Medida. - Así se le denomina al bien o servicio que se considera como representativo del resultado final a alcanzar en un programa y de cuya cantidad a lograr resulta una meta.

Recursos. - Son los insumos, tanto financieros como materiales y humanos que harán posible la realización de las actividades que permitan alcanzar objetivos y metas.

Unidad Ejecutora. - Se refiere a la unidad o unidades organizacionales encargadas de llevar a cabo la obtención de los objetivos y las metas fijados en el programa, así como la administración de los recursos asignados.

Por todo lo dicho anteriormente, el presupuesto por programas es la opción que se ha tomado como la más ventajosa. Desde 1976 y como propósito del C. Presidente de la República se ha tratado de institucionalizar la práctica del presupuesto por programas en todo el país.

La Universidad Nacional Autónoma de México ha adoptado dicha técnica desde hace aproximadamente 15 años.

Desde hace aproximadamente 15 años en que se adoptó el Presupuesto por Programas como una técnica presupuestal propia de la Institución, se sigue perfeccionando su aplicación, debido más que nada, a que dicha técnica constituye un valioso elemento que permite contemplar el cumplimiento de objetivos y metas, para lo cual agrupa la totalidad de las actividades a desarrollar y los recursos destinados para ello. Así mismo y como consecuencia de lo anterior, coadyuva al mejor cumplimiento de las funciones de docencia, investigación y extensión Universitaria, mediante la óptima asignación y utilización de los recursos con que se cuenta.

La asignación del presupuesto en la UNAM está a cargo de la Dirección General del Presupuesto por Programas que depende de la Secretaría General Administrativa.

La aplicación de esta técnica requiere de una serie de normas y procedimientos para su funcionamiento y la necesidad de una revisión permanente a efecto de que su operación marche acorde con su propia evolución y con el desarrollo de toda la estructura universitaria.

Estructura Programática.

La estructura programática es la parte central del Sistema de Presupuesto por Programas de la Universidad, en ella se conjugan sistemáticamente los programas y subprogramas que comprenden las actividades que conducen al desarrollo de las funciones de docencia, investigación, extensión universitaria y apoyo.

Esta estructura es el marco conceptual en el que se presentan los objetivos y metas que se pretenden lograr, así como los recursos humanos, materiales y financieros que se requieran para llevarlos a cabo, lo cual permite una correlación entre resultados y recursos.

En la integración de la estructura Programática las dependencias universitarias tienen una participación relevante por la información cuantitativa y cualitativa que proporcionan en cada ejercicio presupuestal.

Integración de la Estructura Programática.

La Universidad ha fundamentado, para este ejercicio, su estructura programática-Presupuestaria cuyo contenido se sustenta en las funciones expresadas.

1. Docencia

- 1.1 Educación Media Superior
- 1.2 Educación Técnica
- 1.3 Educación de Licenciatura
- 1.4 Educación de Posgrado
- 1.5 Servicios a Estudiantes
- 1.6 Servicios de Administración Escolar
- 1.7 Servicios de Coordinación, Apoyo y Superación a la Docencia
- 1.8 Adaptación y Mantenimiento en Docencia

2. Investigación

- 2.1 Investigación en Ciencias y Desarrollo Tecnológico
- 2.2 Investigación en Humanidades y Ciencias Sociales
- 2.3 Servicios de Coordinación, Apoyo y Superación a la Investigación
- 2.4 Adaptación y Mantenimiento en Investigación

3. Extensión Universitaria

- 3.1 Difusión de Actividades Artísticas, Científicas y Culturales
- 3.2 Extensión Educativa
- 3.3 Servicios de Divulgación
- 3.4 Servicios a la Comunidad
- 3.5 Adaptación y Mantenimiento en Extensión Universitaria

4. Apoyo

- 4.1 Dirección
- 4.2 Planeación
- 4.3 Servicios Administrativos Generales
- 4.4 Adaptación y Mantenimiento en Apoyo

Los programas antes citados se aplican a los siguientes ramos:

Ramo	Descripción
100	Organos de Dirección
200	Organos de Investigación Humanística
300	Organos de Investigación Científica
400	Facultades, Escuelas y Centros de Enseñanza
500	Organos Complementarios de Enseñanza y Serv.
600	Organos de Extensión Universitaria
700	Organos de Serv. Admon. y para el Desarrollo
800	Organos de Adaptación y Mantenimiento de Obras e Instalaciones.
900	Organo Coordinador de Serv. y Prestaciones Sociales.

En general el ramo representa, en esencia la ubicación del organismo de acuerdo a sus funciones. Los ramos particularizados constituyen las claves de las dependencias universitarias.

Con los elementos anotados anteriormente se constituye una codificación denominada Código Programático. A continuación se define este concepto y los elementos que lo constituyen y que no hayan sido definidos en los párrafos anteriores.

Código Programático.- Es un conjunto de dígitos entre los que se pueden distinguir las claves de programas, subprogramas, dependencias, subdependencias, partida de gasto y dígito de control, que juntos constituyen la base para procesar la información que demanda el sistema de presupuesto por programas.

Partida de gasto.- Es el elemento presupuestario en que se dividen los gastos y se clasifican las erogaciones de acuerdo con el objetivo específico del gasto. En la UNAM dicha partida se agrupa de acuerdo al tipo de gasto, generando así otro rubro conocido como Grupo. Además las Partidas se clasifican en: Directas, Centralizadas y Complementarias.

Dígito de Control.- Está formado por dos números de los cuales el primero identifica a la dependencia centralizadora del gasto y el segundo constituye el dígito verificador de la correcta integración del código programático. Este último es el resultado de un serie de operaciones efectuadas en los demás dígitos que integran a dicho código y se utiliza para la captura de datos.

El control presupuestal incluye dos procesos generales: la asignación y el ejercicio. Existen varias formas en que se puede ejercer el presupuesto, estas se denominan transacciones. Las transacciones que se utilizan en la Universidad se describen a continuación:

- + Asignación. Es el importe autorizado para sufragar los gastos.
- + Transferencia. Es la disminución que se hace a una asignación original de un código programático ya existente para ampliar otro.
- + Ampliación. Es el aumento que se hace a una asignación a partir de la transferencia.
- + Compromiso. Es el importe de una obligación contraída.
- + Ejercido. Es el pago del importe, ya sea directo o de un compromiso.
- + Reasignación. Es la asignación que se determina después de haber especificado la asignación inicial, y que se define durante el desarrollo del ejercicio presupuestal.
- + Cancelación. Es la supresión de un compromiso o un ejercido.

Toda la información que se va generando a partir del ejercicio presupuestal se va registrando manualmente en un documento denominado póliza.

Una póliza es un documento de carácter interno en el que se registran las operaciones que incluyen anotaciones y cantidades, además generalmente se anexan los comprobantes que justifican la póliza. Las pólizas permiten la división del trabajo y permiten la comprobación de operaciones.

Las pólizas se pueden clasificar de acuerdo al sistema que se utiliza, así tenemos: sistemas con un registro o póliza única; sistema de dos registros o de diario y caja; y sistema de tres registros o de diario, de ingresos y egresos de caja.

De esta manera, es como la Universidad Nacional Autónoma de México lleva a cabo su Ejercicio Presupuestal basado en la Estructura Programática.

Todo lo anterior nos da un panorama amplio de lo que es el presupuesto y su terminología, con esto podemos ubicarnos mejor en el contexto de la etapa del presupuesto que se pretende desarrollar, ésta es, el control del ejercicio. Con respecto a ésta, en el siguiente apartado se presenta la forma en cómo se ha llevado a cabo desde que se implementó la técnica programática automatizada.

B) ANTECEDENTES DEL SISTEMA DE CONTROL PRESUPUESTAL EN LA UNAM

La automatización del primer sistema para el control del ejercicio presupuestal estuvo a cargo de la Dirección General de Servicios de Computo para la Administración, antes Centro de Servicios de Computo, desde hace aproximadamente 15 años, y surgió como una inquietud por parte del Centro de Investigación en Matemáticas Aplicadas, Sistemas y Servicios (CIMASS) actualmente Instituto (IIMAS).

Algunas características de este sistema son: operaba en computadores Burroughs 5500 y 6700; está escrito en lenguaje Algol, utiliza mucha memoria y se ejecuta en proceso batch.

Después del C.S.C, otras dependencias como el Instituto de Materiales, la Dirección General de Obras y la Facultad de Medicina, se interesaron en utilizar el sistema por lo que a éste se le hicieron algunas modificaciones para hacerlo conversacional con objeto de que sea trabajado por terminales a través de teleproceso. El sistema seguía operando en el computador Burroughs 6700, de donde más tarde sería trasladado a un computador A9 donde actualmente se encuentra.

Aproximadamente hace cinco años, fue cuando comenzó en la UNAM la descentralización de los procesos administrativos, es decir, se trataba de que cada dependencia pudiera efectuar sus procesos en equipos propios de la misma, ya que resultaba problemático el acceso al computador central Burroughs 6700. Para llevar a cabo tal objetivo, la Universidad adquirió una gran cantidad de microcomputadoras Cromemco y la DGSCA desarrolló un sistema para el control del presupuesto que operaba en los sistemas C-10 y C-ZERO, con 64 KB de memoria principal y dos drives para memoria secundaria.

Este nuevo sistema para microcomputadora se realizó basándose en el sistema del equipo A9, y contaba con la opción de consulta aunque muy restringida por la capacidad del equipo.

Se eligió como lenguaje para el desarrollo el BASIC ESTRUCTURADO de Cromemco, ya que ésta resultaba ser el más estándar en esos equipos además de ser más fácil el traslado de Algol a Basic.

La realización del nuevo sistema no resultó ser una tarea fácil. Tomó cerca de un año y estuvo en operación masiva cerca de tres años, después de los cuales el equipo Cromemco se ha ido sustituyendo por microcomputadoras personales PC compatibles.

Lo anterior determinó la necesidad de realizar otro sistema para utilizarse en las nuevas computadoras PC que se estaban adquiriendo, sin embargo, por la urgente necesidad de que se tuviera el sistema de inmediato, ya que muchas dependencias lo requerían, la acción que se tomó fue la de trasladar el sistema de Cromemco a PC. Esto se hizo a través de protocolo de comunicación y se adaptaron los programas de Basic Estructurado a GWBASIC.

Las Características del sistemas en PC son:

- + Es interactivo
- + Es conversacional
- + Posibilidad de consulta
- + Posibilidad de emisión de 5 reportes

Sin embargo, el sistema tiene la desventaja de que la implementación para diferentes dependencias resulta muy complicada debido a que se modifican directamente los programas fuentes. Además por causas que en otro capítulo se comentarán con más detalle, el sistema actual contiene rutinas de ordenamiento en DBASE III PLUS. Por todo esto, el mantenimiento de dicho sistema se ha vuelto muy complicado, a pesar de que su funcionamiento es bueno.

La versión del sistema para computadora PC es el que actualmente utilizan la mayoría de las dependencias universitarias y es el se tomará como base para la realización de un nuevo sistema, objetivo de este trabajo, para tratar de mejorar y optimizar el ya existente.

C A P I T U L O I I

METODOLOGIA DE DESARROLLO

Como consecuencia de la necesidad de planificar, operar y diseñar sistemas cada día más complejos que solucionen los grandes problemas sociotécnicos en los medios académicos e industriales, ha surgido una nueva filosofía o metodología llamada Ingeniería de Programación.

El objeto final de este trabajo es llevar a cabo el desarrollo de un sistema, sin embargo, no se puede llegar a este sin antes haber entendido el problema, sin haber hecho un estudio y una planeación y sin haber considerado otros aspectos que se incluyen en lo que se denomina metodología de desarrollo.

Este capítulo, trata precisamente del objetivo de la metodología y particulariza sobre una en especial, por ser a nuestro juicio la más completa y general, aunque también se mencionan otras metodologías utilizadas en la actualidad. Este apartado nos da la estructura de los subsecuentes capítulos. Se puede decir, que en los siguientes apartados encontraremos ejemplificados los conceptos que aquí se definen.

A) OBJETIVO DE LA METODOLOGIA

El manejo de la información por medio de computadoras, se hace día a día más frecuente, sin embargo el diseñar sistemas adecuados y eficientes para la manipulación de dicha información no resulta fácil.

En ocasiones el analista se enfrenta a problemas de diseño, el programador a problemas de codificación y el usuario se enfrenta con un sistema deficiente, eso se debe en gran medida a la falta de uso de una técnica bien definida para el diseño del sistema.

Las técnicas empleadas son diversas, pero todas pretenden llegar a lo mismo: desarrollar sistemas que sean eficientes, fácilmente implementables y bien documentados.

Dichas técnicas se agrupan en lo que se conoce como Ingeniería de Software o Ingeniería de Programación.

La Ingeniería de Programación es una disciplina que hasta la década de los sesentas aún no se había establecido. Surgió como una solución a lo que se conoce como la Crisis del Software.

La problemática principal que se detectó en dicha crisis se puede definir en los siguientes puntos:

- + Los costos se incrementaban en forma exponencial.
- + Los proyectos no se terminaban a tiempo ni con el presupuesto programado.
- + El mantenimiento del sistema absorbe la mayor parte de recursos de la gente de desarrollo.

Los problemas anteriores conllevan a la generación de problemas asociados de los que se destacan los siguientes:

- + Insatisfacción del usuario con el sistema terminado.
- + Calidad dudosa del software.
- + Dificultad de mantenimiento del software actual.

Uno de los factores que se tomaron como referencia para detectar la crisis, fue la comparación de costos entre el hardware y el software, mientras que en uno disminuían, en el otro aumentaban considerablemente.

Lo que pretende la ingeniería de software, es establecer métodos y técnicas que lleven paso a paso a la realización de sistemas de cómputo eficaces, eficientes, confiables, transportables y rentables. Por otro lado, define lo que es el ciclo de vida de un proyecto de software.

CICLO DE VIDA DE UN SISTEMA DE PROGRAMACION.

Existe un número considerable de metodologías de desarrollo, pero en todas se puede distinguir las siguientes fases:

+ Estudio del sistema.

Se refiere a una revisión general del sistema actual, de la que se deriva la detección y definición de necesidades, se plantean diferentes alternativas para satisfacer las necesidades. Se realiza un estudio de factibilidad que incluye una revisión de los recursos para determinar si es factible su utilización y un análisis beneficio/costo. El estudio de factibilidad se hace generalmente para todas las alternativas propuestas, ya que de éste se concluye la más viable. Cabe aclarar que de este estudio se puede concluir que la realización del sistema no es factible, y ya no se continuaría con las siguientes etapas.

+ Planeación.

Una vez que se definió que se necesita el nuevo sistema y que es factible llevarlo a cabo, se hace una planeación de cómo se desarrollará el mismo. Primero se definen los alcances del proyecto. Se determina también la disposición de los recursos cronológicamente y se establecen mecanismos de supervisión y control de avance. Con respecto a las herramientas de control del proyecto, se tienen las opciones de: Gráficas de Gantt, las cuales consisten de un reporte en forma de tabla donde se incluye la información del avance, responsable, actividades, etc.; la otra opción es la realización de la Ruta Crítica, la cual consiste en una red donde los nodos denotan las actividades a realizarse, la ruta crítica suele utilizarse en proyectos muy grandes donde se requiere gran coordinación. En la planeación también se define la estructura organizacional para la realización del sistema, es decir, se asignan responsables.

+ Analisis y Diseño.

El análisis y diseño son las etapas más determinantes dentro del ciclo, ya que de éstas depende la vida del sistema. Es aquí donde se define el QUE, el COMO y CON QUE del desarrollo. En esta etapa, se determinan los datos que va a manejar el sistema, la localización de los mismos (archivos que se van a utilizar), los procesos que se requieren y la forma en que se comunicarán. Con los resultados de esta fase, se puede definir el lenguaje más óptimo para que se haga el desarrollo del sistema.

+ Programación y Pruebas.

Se refiere a la codificación de los programas en un lenguaje de computación y a su depuración hasta dejarlos en buen funcionamiento.

+ Liberación, instalación y documentación.

Esta etapa es en la que se presenta el sistema terminado al usuario con la documentación necesaria para su operación y mantenimiento.

+ Mantenimiento.

Son los cambios, reducciones o ampliaciones que el sistema vaya requiriendo durante su vida operable.

+ Deceso del sistema.

Ocurre cuando se requiere de un nuevo sistema que mejore al anterior. Esto sucede cuando la organización donde operaba el sistema cambia de políticas, cuando el sistema se vuelve obsoleto por equipo de hardware o por que se requiere de otro tipo de procesos que no son compatibles con los que se plantearon al principio del sistema en deseso. El nuevo sistema pasará por todas las etapas citadas.

Las etapas anteriores son las correspondientes al ciclo de vida de un sistema computarizado.

A continuación se presenta una metodología que en la actualidad es la más utilizada por ser una de las más completas.

B) TÉCNICAS ESTRUCTURADAS

Anteriormente no se empleaban técnicas bien definidas para el desarrollo de sistemas o si se utilizaba alguna metodología no era general. La metodología estructurada incluye el uso de técnicas generales, ya que se enfocan a analizar el problema lógico y no físico. Además dichas técnicas están fundamentadas matemáticamente.

La Técnica Estructurada incluye básicamente los siguientes pasos:

- + Análisis estructurado
- + Diseño estructurado
- + Programación o Codificación estructurada.

ANÁLISIS ESTRUCTURADO

El análisis estructurado, es la primera fase en el proyecto del desarrollo de un sistema EDP (electronic data process). Donde se definen los requerimientos del usuario y se documentan.

Para entender el análisis estructurado, es importante examinar los pasos que usualmente se siguen en el análisis de sistemas clásico. En éste, el analista prepara un documento que describe el propósito del sistema y que el usuario revisa. El documento está formado por una gran cantidad de páginas escritas en términos técnicos, con las siguientes características:

- Es monolítico, es decir, debe ser leído de principio a fin.
- Es redundante, se da la misma información en diferentes lugares.
- Es difícil de modificar y mantener.
- Se preocupa por la parte física del sistema y no por la lógica.

De las características anteriores, se puede deducir que existen problemas que el análisis estructurado trata de resolver.

El análisis estructurado es básicamente el uso de herramientas gráficas en la documentación, para producir un nuevo tipo de especificaciones funcionales.

Las herramientas en las que se basa el análisis estructurado son:

DIAGRAMA DE FLUJO DE DATOS. Es el modelo de flujo de datos a través del sistema en forma gráfica. Los elementos del diagrama son: fuente y destino de los datos; flujo de datos; procesos y almacenamiento de datos. Dichos elementos se esquematizan en las siguientes figuras:

almacenamiento

por medio de dos líneas paralelas se representan los archivos.

| fuente |
| o |
destino
cuadros o rectangulos representan el origen y el destino de los datos.

: :
: :

en círculos se representan los procesos.

----->
la flecha indica el flujo de los datos. Sobre o bajo la flecha se ponen los nombres de los datos.

DICCIONARIO DE DATOS. Es la colección organizada de definiciones lógicas de los nombres de los datos que se muestran en el diagrama de flujo de datos. La estructura que se sigue para la definición de los datos es la siguiente:

<nombre del dato> = <definición>

donde para la definición se utiliza la siguiente notación:

x = a + b	x	consiste	del	dato	elemental	a	y	b						
x = {a b}	"	"	"	"	"	"	"	" o "						
x = (a)	"	"	"	"	"	opcional	a							
x = {a}	"	"	"	"	"	de	cero	o	más	ocurrencias	de	a		
x = w(a)	"	"	"	"	"	w	o	más	ocurrencias	de	a			
x = (a)z	"	"	"	"	"	z	"	menos	"	"				
x = w(a)z	"	"	"	"	"	número	de	ocurrencias	de	a	entre	w	y	z.

ESPECIFICACIONES DEL PROCESO O MINIESPECIFICACIONES. El propósito de éste es permitir la descripción rigurosa y precisa de las políticas (pero no la implementación de las tácticas) representadas en los procesos de los niveles más bajos del diagrama de flujo de datos.

La especificación del proceso puede realizarse de diversas formas: fórmulas gráficas; tablas de decisión o en pseudocódigo.

DIAGRAMA DE RELACION DE ENTIDADES (E-R). Este se refiere a la relación existente de los datos que se almacenan. Cada dato en DFD corresponde a un objeto en el E-R.

DIAGRAMA DE TRANSICION DE ESTADOS (STD). Estos son útiles en sistemas de tiempo real y es una modificación del diagrama de flujo de datos.

DISENO ESTRUCTURADO

En el diseño del sistema se trata de lograr una especificación clara y completa de los requerimientos de software de un sistema.

Módulo y Modularidad.

Un aspecto importante a considerar en el diseño estructurado es la modularidad. Un módulo se puede definir como un conjunto de instrucciones a las que se les puede asignar un nombre (módulo lógico).

Existen dos formas principales de clasificar a los módulos.

Según su función, hay módulos de control para establecer relaciones, tomar decisiones y llamar a otros módulos; de proceso, para realizar alguna tarea operativa; de rutinas auxiliares, para usos generalizados como validación, inicialización e impresión; de interfases E/S, los cuales involucran a algún dispositivo periférico; de errores y excepciones, los cuales se encargan de recuperar errores del sistema y reestablecerlo para su ejecución normal.

Según su ejecución, hay módulos secuenciales, es decir uno después de otro; incrementales cuando se ejecutan procesos intermedios; y paralelos si se ejecutan en forma simultánea.

Una característica de los módulos es que sean independientes, lo cual es la base de la modularidad. Un módulo debe ser lo mayor cohesivo posible y debe tener el menor acoplamiento.

COHESION. Es la interrelación que existe entre los elementos (instrucciones y datos) de un mismo módulo. Existen varios tipos de cohesión: coincidental, cuando se hace un módulo para no repetir un código o bien un bloque muy largo se fragmenta en módulos pequeños; lógica si existe relación de módulos generales como validación; temporal que es igual a la lógica pero relacionada con el tiempo; procedural; comunicacional; secuencial y funcional.

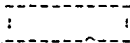
ACOPLAMIENTO. Es una medida cualitativa que se refiere a la interrelación entre módulos de un sistema. Los criterios de acoplamiento se refieren a los parámetros, el número y forma en que se llaman, a las áreas comunes de datos, a las variables de control (banderas), etc.

Estructura del Sistema.

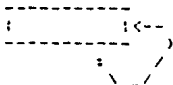
La estructura del sistema no nos da el flujo de datos sino la dependencia entre módulos.

El diseño estructurado incluye herramientas gráficas para su documentación como:

CARTA ESTRUCTURADA O CARTA HIPO (hierarchical input process output). Es la organización jerárquica de un sistema que nos muestra la organización y subordinación hacia los módulos. En la carta se pueden distinguir lo que se conoce como FAN-OUT, que se refiere a las llamadas de un módulo hacia los subordinados y FAN-IN que se refiere a las llamadas que recibe un módulo subordinado directo. En la elaboración de la carta estructurada se utiliza la simbología de Constantine, la cual se define a continuación.



condiciona el llamado hacia el módulo subordinado.



procesos iterativos

: : : :

módulos predefinidos

: :

procesos paralelos

parámetros

O----->

variables de control

@-----@

datos o estructuras de datos

En general la metodología del diseño estructurado (Constantine y Yourdon), sigue los siguientes pasos:

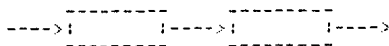
- + revisión y evaluación del DFD del análisis.
- + se subdivide en subsistemas.
- + evaluación para cada subsistema de los recursos y del tipo de sistema (en línea, batch, teleproceso, etc.).
- + para cada subsistema se realiza una revisión de los archivos definiéndolos físicamente. Se normaliza y se definen volúmenes (No. de registro, transacciones, etc.).
- + se evalúa el tipo de diseño. Transformación y Transacciones.

CODIFICACION O PROGRAMACION ESTRUCTURADA

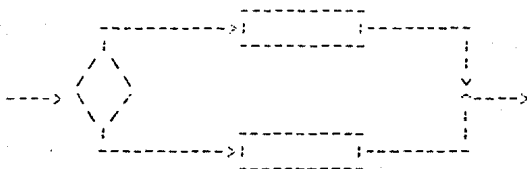
El término fue concebido por el profesor Edsger Dijkstra a mediados de la década de los sesentas. Fue la primera de las técnicas estructuradas. Esta técnica se basa en la teoría de que si el código del programa se escribe usando solamente las tres estructuras: secuencia, selección e iteración; se tendrían pocos errores y podría modificarse con mayor velocidad y seguridad.

Las bases teóricas de todos los procedimientos lógicos tradicionalmente se describen con diagramas de flujo. A mediados de los sesentas, Conrado Bohm y Guissepe Jacopini concluyeron matemáticamente que cualquier procedimiento lógico, esto es, cualquier diagrama de flujo, se puede derivar de las siguientes tres combinaciones:

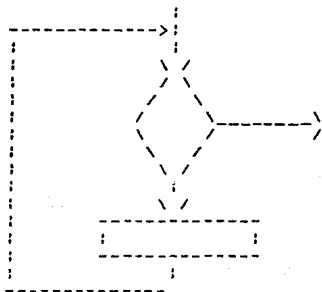
secuencia



selección



iteración



C) OTRAS METODOLOGIAS

Para el diseño de sistemas, en particular de software, existen diversas técnicas, las cuales, nos guían a través de diversos pasos. En esta ocasión, hablaremos precisamente sobre estas técnicas y sus generalidades. Entre las más empleadas y las más conocidas podemos citar las siguientes:

- 1: Diseño Top-Down
- 2: Diseño Bottom-Up
- 3: Diseño por Prototipos

Diseño Top-Down.

El diseño Top-Down es básicamente una descomposición de procesos los cuales están enfocados sobre la estructura de control del programa. El primer paso es estudiar los diferentes aspectos en que una tarea o labor puede ser descompuesta, esto es, fraccionar esa tarea en un número de funciones independientes o módulos. Este es el primer paso en la descomposición. El segundo paso es romper cada uno de esos módulos en submódulos independientes. Este proceso es repetido hasta obtener módulos que puedan ser indivisibles en submódulos. Claramente, un módulo de la estructura puede extenderse a un nivel más bajo que el siguiente.

Dentro del diseño Top-Down los niveles de detalle bajos esconden niveles mucho más bajos. Sin embargo solamente aquellos niveles que involucran a los datos como interfaz entre un nivel y otro, y los niveles de control, son definidos. Por otro lado si una estructura de datos está profundamente contenida dentro de un módulo de nivel bajo, ésta no necesita indispensablemente estar especificada dentro del módulo, esto se haría hasta llegar a la fase del diseño de cada proceso. Si los datos son compartidos por diferentes módulos en algún nivel, entonces la estructura de datos a ser utilizada debe ser escogida antes de procesar un nivel más bajo.

En el diseño Top-Down podemos comenzar a detallar una estructura de control en el primer nivel de descomposición y producir el código necesario. Similarmente, el código de control para cada módulo de bajo nivel será escrito, y el último código a ser escrito será el de aquel módulo de más bajo nivel.

Es difícil cuantificar, en este tipo de diseño, cuándo un módulo es suficientemente pequeño y cuándo es necesario descomponerlo en submódulos, a este respecto existen diferentes criterios para determinarlo.

Diseño Bottom-Up.

En este tipo de diseño, primeramente se visualiza un sistema típico de diseño, y se decide, por experiencia, intuición o quizás por un análisis rápido qué parte o partes del diseño son las más dificultosas o las más limitadas. Las partes cruciales son primero investigadas y es necesario realizar un diseño de decisiones.

En algunos casos el diseño Bottom-Up representa una síntesis. Las especificaciones de las partes claves son dadas, y el diseño es formulado para estas partes claves. Si el diseño no es factible sobre las especificaciones dadas y resulta ser contrastante, los requerimientos se cambian y se procede a otro nuevo diseño.

El diseño Bottom-Up procede primeramente a analizar las partes cruciales, no así, en el diseño Top-Down, que va de lo más general a lo más particular, esto tiene como desventaja el no poder determinar desde un principio si se marcha sobre un diseño equivocado o poco factible. En Bottom-Up se determina primeramente la factibilidad del diseño analizando, como ya se mencionó, las partes cruciales sobre las especificaciones ya dadas.

Si cada módulo es codificado tan rápido como la construcción de su diseño, entonces la entrada es llamada codificación Bottom-Up. Es posible tener un diseño Top-Down y una codificación Bottom-Up y viceversa.

Diseño por Prototipo.

El diseño por prototipos es una técnica en la que se realiza un conjunto de pantallas, menús, impresiones, etc. de como quedará el sistema de acuerdo a las especificaciones proporcionadas por el usuario. El usuario al revisar el prototipo puede esclarecer para sí mismo algunas dudas y de esta manera se va perfeccionando el sistema.

Los prototipos se realizan con la combinación de lenguajes de alto nivel, generadores de pantallas y reportes y las facilidades de los manejadores de bases de datos.

Las motivaciones que existen para la realización de prototipos son: la comunicación entre analista y usuario; son fáciles y rápidos mientras que el modelo escrito y el análisis se lleva meses; aclara el diálogo hombre-máquina.

Sin embargo, para que esta técnica sea factible se deben cumplir las siguientes condiciones: solo debe haber un usuario o un pequeño grupo; existe el modelo de datos o es fácil su creación; la aplicación es pequeña o mediana; todos los participantes deben estar de acuerdo con la técnica.

De entrada se puede observar que el desarrollo que nuestro trabajo requiera no puede utilizar la técnica por prototipo debido principalmente a que el número de usuarios de este sistema es muy grande.

Otras metodologías.

En la mezcla del diseño Top-Down y Bottom-Up, a menudo resulta que comenzamos en medio del problema y trabajamos por ambos lados tanto hacia arriba (Bottom-up) como hacia abajo (Top-down). Esta técnica es usualmente llamada diseño Middle-out. En problemas complejos, en los cuales generalmente es difícil decidir cómo modularizar el primer nivel (o algún otro nivel) se emplea esta técnica. En tales casos, se debe considerar una lista de sistemas de entrada y decidir qué funciones son necesarias para procesar estas entradas. La técnica de diseño Back-to-front es algunas veces aplicada igualmente a este tipo de problemas. Similarmente, podríamos comenzar con los requerimientos de salida y trabajar hacia atrás, esta técnica es llamada Front-to-back. En algunos casos las estructuras de datos son cruciales a la solución y el término Data-directed es apropiado.

Como se podrá observar, los siguientes capítulos siguen la estructura de lo que se conoce como el ciclo de vida de un sistema de software, pero se hace énfasis en la utilización de las técnicas estructuradas.

C A P I T U L O I I I

ESTUDIO GENERAL DEL SISTEMA DE PRESUPUESTO

La finalidad y objeto del estudio general del sistema es facilitar la información que la dirección necesita para tomar la decisión de comenzar o no el desarrollo del sistema.

En este capítulo se empezará por definir la situación existente y hacer una evaluación de la misma. De esto, se derivará la definición explícita de las necesidades que serán entre otros puntos, los objetivos del nuevo sistema. Con esto se pueden plantear diversas opciones de solución, cada una de las cuales se estudian para ver si son factibles.

De los resultados del estudio de factibilidad se tiene la información para decidir si se lleva a cabo el desarrollo del sistema.

A) DIAGNOSTICO DE LA SITUACION ACTUAL

Existe en la Universidad Nacional Autónoma de México varios sistemas que llevan a cabo el control del ejercicio presupuestal.

El sistema en el equipo mayor, que fué el primero en desarrollarse en la Dirección General de Servicios de Computo para la Administración, funciona actualmente en el equipo A9 de dicha dirección. Es un sistema escrito en lenguaje Algol y requiere 280 KB de memoria para los programas fuente y 140 KB para los programas objeto. Su operación se realiza en proceso batch, para esto las dependencias que lo utilizan requieren de capturistas que dejen los datos grabados en cintas, las cuales son traídas a la dirección para que los datos sean procesados por la A9. Para operar el sistema se requieren conocimientos de CANDE y WFL.

El primer proceso que realiza el sistema sobre los datos es un diagnóstico de los mismos para detectar errores. En caso de que existan errores se recapturan los datos con las correcciones.

La información de los movimientos que se realizan desde el inicio del control presupuestal hasta un determinado momento, se presentan en ocho reportes: presupuesto detallado por partidas, detallado por unidad responsable, concentrado por unidad responsable, concentrado por unidad responsable y grupo, global por unidad responsable, informe para redistribución presupuestal e informe global por partida.

El diagnóstico y los informes se solicitan directamente a la A9 a través de terminales localizadas en las diferentes dependencias que hacen uso del sistema, para esto se requiere de equipo adicional de comunicaciones tales como modems y líneas telefónicas.

Son cuatro las dependencias universitarias que utilizan actualmente el sistema.

El sistema de presupuesto realizado para funcionar en un microcomputador Cromemco funciona con los sistemas C-10 o C-ZERO para dos drives. Está escrito en Basic estructurado, requiere un diskette de 365 KB para los programas fuente y otro para los ejecutables. Su operación es iterativa con captura de datos por pantalla. Contiene un módulo de consulta por pantalla y otro para resumir los datos y así ocupar menos memoria (cierre de periodo). Incluye también un proceso para efectuar modificaciones sobre los datos archivados para los casos de error. Presenta la información en 5 reportes: global por partida, global por unidad responsable, detallado por partida, detallado por unidad responsable-partida, concentrado por unidad responsable-grupo.

Las dependencias universitarias que utilizan actualmente la versión Cromemco del sistema de presupuesto son 15 de las cuales las 13 pretenden cambiar a la versión PC.

La versión PC del sistema de control presupuestal, funciona en equipos microcomputadores PC compatibles con IBM. Se encuentra escrito en GW-BASIC. Se puede operar en equipos con un drive y un disco duro o con dos drives. La operación y los módulos que contiene el sistema son los mismos que en la versión Cromemco.

Para la generación de reportes se requieren rutinas de ordenamiento, las cuales se encuentran escritas en DBASE III PLUS y están compiladas con CLIPPER. Esto debido a que el ordenamiento de DBase requiere menos memoria principal que el ordenamiento de Basic, lo cual causaba problemas de capacidad.

Para algunas versiones de GW-Basic se requiere una definición previa de parámetros tales como el número de archivos.

El número de dependencias que utilizan esta versión es de cuarenta y cinco.

Existe una versión más del sistema de presupuesto, esta opera en equipos de la familia B20, requiere dos diskettes para programas fuentes y otros más para los objetos. Tiene los mismos módulos que la versión PC. Los informes incluyendo el sort se encuentran en COBOL. Hasta el momento ninguna dependencia utiliza esta versión.

El sistema en el computador A9 requiere de muchos recursos, pero el equipo los puede satisfacer en forma completa y eficaz.

La limitación del sistema en equipos Cromemco es básicamente la capacidad de memoria de estos, sin embargo, son pocos los usuarios que lo utilizan y se prevé que sean menos aún. El mantenimiento y adaptación del sistema son complejos al igual que en la versión PC.

El sistema de control presupuestal en microcomputadoras personales PC funciona bien para los usuarios. No existen tantas limitaciones de capacidad de memoria como en Cromemco (exceptuando las funciones de sort) sin embargo, la adaptación del sistema es compleja y peligrosa ya que se realiza directamente en los programas fuente, además el mantenimiento en general resulta difícil debido a que se requiere conocer bastante bien Basic y DBase III.

B) DEFINICION DE NECESIDADES

Del inciso anterior podemos concluir lo siguiente:

Por el momento no se requiere adaptaciones, cambios o adiciones al sistema de presupuesto de la A9.

El sistema de control presupuestal en Cromemco se puede mejorar, pero debido a que su uso es reducido con tendencias al desuso, no es recomendable mejorar un sistema que no se utilizará.

El sistema de control presupuestal para PC es el que más se utiliza y es en el que se requieren muchas mejoras.

De ahora en adelante, cuando nos refiramos al sistema de control presupuestal, se debe entender que es la versión para microcomputadora personal PC.

A continuación listaremos y explicaremos las necesidades que se requieren satisfacer con la realización del sistema, objeto de esta tesis.

Las características funcionales que debe contemplar el sistema son las siguientes:

- + Se necesita un sistema para el control del ejercicio presupuestal que funcione en equipos computadores PC compatibles.
- + El sistema debe ser accesible al usuario con varios niveles de ayuda, con los que se explique el modo de operación y las características de los datos que se solicitan.
- + El manejo del sistema debe ser lo suficientemente sencillo para que sea operado sin requerir mayores conocimientos de computación.
- + Se necesita que el sistema incluya una forma de autopresentación para introducir a los nuevos usuarios en su utilización.
- + La adaptación del sistema a las diferentes dependencias debe ser fácil y debe hacerse sin alterar los programas fuentes.
- + Es necesario que el mantenimiento del nuevo sistema sea una tarea fácil, para lo cual todos los programas que se requieran deben estar escritos en el mismo lenguaje.

Los procesos que se deben contemplar en el sistema son los siguientes:

+ Se requiere de un sistema con el que se puedan dar de alta las asignaciones y movimientos que se hagan sobre estas asignaciones (transacciones). Este proceso de alta se debe realizar como una captura de datos por pantalla. Al dar de alta las transacciones, se deben validar los datos, en el caso de que se detecte algún error, se debe presentar la opción de ayuda en la que se explicará la forma correcta de introducir los datos.

+ Siempre se registran primero todas las asignaciones, por lo que sería conveniente tener un proceso exclusivo para tal registro y otro independiente para los demás tipos de transacciones.

+ Los datos se deben ir almacenando en archivos, los cuales se asignarán a disco duro y/o diskette. Además debe haber al menos un respaldo de la información.

+ El sistema debe incluir un módulo de modificaciones para el caso de que exista algún error en los datos de los archivos.

+ Se deberá tener la opción de consultar la información por pantalla. La consulta deberá ser flexible, contemplando diferentes datos como llaves de acceso para la consulta, tales como:

- Consulta por grupo,
- Consulta por póliza,
- Consulta por partida,
- Consulta por totales.

+ Se requiere que por medio del sistema se obtengan informes, se deben obtener al menos los mismos tipos de reportes con que se cuenta en la actualidad, es decir:

- Informe global por Partida,
- Informe global por unidad responsable,
- Informe detallado por partida,
- Informe detallado por unidad responsable-partida.
- Informe concentrado por unidad responsable-grupo.

+ Debido a que la cantidad de transacciones puede ser muy grande y en un momento dado solo interese el estado actual del presupuesto y no su historia. Se debe contar con un proceso que resuma la información. Esto para optimizar el espacio en la memoria y disminuir el tamaño de los informes.

C) DEFINICION DE ALTERNATIVAS DE SOLUCION

Dentro del análisis del sistema, generalmente nos planteamos diversas estrategias para resolver el problema, estas deben estar vinculadas hacia una posible solución. Así mediante una evaluación previa de cada una de estas estrategias, se determina cual será la más viable.

En este caso, hacemos mención de tres estrategias de solución con sus respectivas evaluaciones y el criterio de elección de la alternativa a llevarse a cabo.

Alternativa 1.

Podemos comenzar por considerar, dado que el Sistema de Control de Presupuesto ya existe, que este siguiera funcionando tal cual, es decir, como primera alternativa consideramos la de no hacer nada.

Sin embargo al evaluar esta alternativa, encontramos que el sistema actual, aunque cumple con las más esenciales necesidades, es un sistema que no proporciona facilidad de implementación y adaptación a los requerimientos de cada dependencia.

Alternativa 2.

Esta alternativa nos muestra un panorama distinto al anterior, dado que el sistema actual presenta los problemas ya mencionados, podemos considerar éstos como partes aisladas y tratar de algún modo de sustituir aquellas que no funcionen de manera adecuada por otras que si lo hagan. Por otro lado, detallando en estas partes encontramos que en el módulo de emisión de reportes que emplea un archivo de tipo secuencial, presentó problemas en el ordenamiento del mismo, por lo que se tuvo que recurrir a rutinas externas en lenguaje DBase III Plus para realizar este proceso.

Así mismo, evaluando tal alternativa, encontramos que estas partes se encuentran vinculadas en ciertos campos con otras tantas, tal que, al intentar modificar una parte habría que modificar la parte que se encuentre vinculada con la anterior, y así sucesivamente, esto es, dado que el lenguaje no es el más adecuado para este tipo de sistemas, la estructura del sistema en GW-Basic posee un gran acoplamiento o dependencia entre módulos.

Alternativa 3.

Esta nos ofrece otra forma de atacar el problema, podemos desarrollar un sistema completamente diferente, en un lenguaje diferente que ofrezca al usuario un panorama distinto de operación y al programador un sistema que sea fácilmente adaptable a las necesidades del usuario.

El considerar como estrategia de solución el hecho de realizar un sistema completamente nuevo, implica como es obvio, hacer un análisis de las necesidades que se definieron previamente, determinar las medidas de solución para cada necesidad y evaluar en conjunto esta estrategia propuesta.

Esto trae como consecuencia el determinar si un sistema que está desarrollado en un lenguaje de tercera generación, que ocasiona problemas al modificar el campo de una variable, y que es muy complejo, debe ser sustituido por otro que sea el reverso de éste?. Consideramos, bajo la pregunta anterior que la respuesta es afirmativa, sobre todo que como en la actualidad se cuenta con herramientas más poderosas para resolver el problema, rechazaremos uno de ellas para obtener un sistema completamente actual.

Dado que la primera alternativa no nos ofrece ningún cambio y por lo cual no alcanza a cumplir las necesidades mencionadas en el apartado anterior, no la podemos considerar como una estrategia factible, por otro lado, la segunda alternativa nos propone únicamente modificar el sistema para alcanzar a cubrir las necesidades citadas, sin embargo, nuestro sistema seguiría siendo obsoleto y complejo. Es así como consideramos, a nuestro juicio, que la tercera alternativa es la factible a ser realizada.

D) ESTUDIO DE FACTIBILIDAD

Los sistemas computarizados ofrecen al usuario información actualizada y confiable. Tal es el caso de un sistema de Presupuesto que nos determina el estado económico y financiero de la organización que tenga como propósito ejercerlo, esto se refleja en una mejor asignación de los recursos de dicha organización.

Como se ha mencionado anteriormente, el sistema a desarrollar no solo tendría como objetivo el ser operado en Dependencias Universitarias, sino además podría ser adaptado a las necesidades de alguna empresa que lo solicitara. Es así como se procede a un análisis de factibilidad del sistema propuesto, con la alternativa factible determinada en el punto anterior.

Recursos.

Al hablar de recursos, nos estamos refiriendo concretamente, a todos aquellos elementos que emplearemos para el diseño y desarrollo del sistema; este tipo de recursos los podemos dividir en tres grupos: humanos, tecnológicos y financieros.

Primeramente, detallaremos en recursos humanos ya que sin ellos, obviamente, ningún proyecto podría llevarse a cabo, ésta es entonces la parte más importante que debemos considerar.

El diseño y desarrollo del sistema para control del ejercicio presupuestal siguiendo la técnica por programas será realizado por las personas autoras de este documento con asesoramiento a cargo del ingeniero Sebastián Poblano Ordóñez.

Los recursos tecnológicos son también importantes, ya que sin ellos no podríamos darle forma a las ideas ni llevarlas a cabo, es por ello que dentro de los recursos tecnológicos podemos mencionar los siguientes:

- + Microcomputadora tipo PC
- + Impresora

En cuanto a software:

- + Sistema Operativo MS-DOS
- + Lenguajes de programación de 4a. generación
- + Procesador de Palabra WordStar

Medios de almacenamiento masivo:

- + Disco duro tipo Winchester de 20 MB
- + Discos flexibles de 5 1/4, doble cara, doble densidad de 360KB
- + Papel para impresión

Los recursos financieros, son importantes para poder obtener los recursos tecnológicos, que finalmente, éstos empleados por los humanos transforman el ingenio en una realidad. Los recursos financieros serán provistos por la Dirección General de Servicios de Cómputo Administrativo de la UNAM.

Una vez determinados todos los recursos a ser empleados para el diseño, desarrollo e implantación del Sistema de Control Presupuestal, se procederá a hacer un análisis para determinar la factibilidad del mismo.

Beneficios.

Para determinar la factibilidad de un proyecto se emplean diversas técnicas, en algunos análisis la relación es beneficio-costo, sin embargo dada la situación económica actual del país es difícil cuantificar monetariamente los recursos a ser empleados dado que nuestras estimaciones de hoy serían obsoletas a las de mañana, es por esto que nuestro análisis se remite a una relación recursos-beneficio, es decir, solo determinamos qué recursos serán empleados y de qué manera serán adquiridos para llevar a cabo la estrategia de solución y los beneficios que nos proporciona dicha estrategia.

Podemos enumerar dichos beneficios:

- 1: Gran facilidad de operación.
- 2: Gran funcionalidad.
- 3: En cuestión de mantenimiento el sistema será versátil.
- 4: Fácilmente implementable y adaptable a las necesidades del usuario.
- 5: Ofrecerá diversos niveles de ayuda al usuario en caso de que éste ignore su operación.

De esta manera podemos notar que esta estrategia nos ofrece grandes beneficios y por tanto nuestra relación recursos-beneficios resulta ser satisfactoria, es por ello que nuestro análisis nos muestra la factibilidad del sistema a desarrollar.

C A P I T U L O I V

PLANEACION DEL SISTEMA DE PRESUPUESTO

En todo desarrollo de sistemas de cualquier tipo, es importante la planeación, con ésta de algún modo, estamos vinculando nuestras actividades con un programa establecido de ellas.

En este apartado se mencionarán los mecanismos de planeación y control del desarrollo y diseño del proyecto, siendo ésta una parte muy importante para la terminación oportuna de nuestras actividades a llevar a cabo.

En el primer punto se contemplarán los alcances del sistema, es decir, qué se pretende del mismo; para el segundo punto presentamos una planeación del empleo de los recursos que se requerirán, y finalmente, el tercer punto nos llevará a determinar de qué manera se supervisará y controlará el diseño y desarrollo del sistema.

A) ALCANCES DEL SISTEMA

En esta ocasión, nos detendremos a hablar un poco acerca de las metas que el sistema puede alcanzar, es decir, qué es lo que el sistema pretende proporcionarnos.

Se pretende que el sistema pueda ser adaptado de acuerdo a las necesidades de cada dependencia que así lo solicite, o sea, un sistema que sea fácilmente implementable, además de ser amigable para el usuario ofreciéndole diferentes niveles de ayuda y que permita tener versatilidad.

El hecho es que, se buscará alcanzar un alto grado de funcionalidad, aunado a una fácil comprensión de la programación para personas ajenas al desarrollo del mismo. Esto se pretende lograr dado que en algunos casos el dar mantenimiento a un sistema complejo que se desconoce, resulta ser tardado y naturalmente costoso, en este caso, el sistema deberá ser sencillo pero envuelto en muchas cualidades.

El sistema a desarrollar podrá ser puesto a disposición de todas las dependencias universitarias que requieran hacer uso de él a partir del próximo año. No puede ser antes, ya que el control se lleva desde el inicio del año, además de que el periodo faltante para su puesta en operación, servirá para probar exhaustivamente el sistema.

Aunque el diseño y desarrollo del sistema, se realizará para que funcione en la Universidad Nacional Autónoma de México, éste podrá funcionar en cualquier institución que lleve el control del ejercicio presupuestal con la técnica de presupuesto por programas. Esto se menciona, para hacer notar la generalidad del sistema.

Por otro lado no se pretende que el sistema sea negociable, es decir, no se pondrá a disposición como un paquete comercial.

El ciclo de vida del proyecto estará limitado, considerando los grandes avances tecnológicos que en materia de Software y Hardware se están presentando hoy en día. Pensando en el posible éxito de este sistema se podría trazar un plan de mejoras continuas para evitar que quede obsoleto en un mercado de cambios constantes.

B) RECURSOS

Comenzemos primeramente por recordar a nuestros lectores que los recursos a ser empleados para el diseño y desarrollo de nuestro sistema fueron especificados en el capítulo anterior.

Para hablar de planeación de recursos, es necesario saber de antemano lo que esto implica. El programar de alguna manera el cómo, cuándo y dónde serán utilizados nuestros recursos no es una parte sencilla, ya que ésta se encuentra siempre bajo modificaciones según se vaya avanzando en el proyecto. Sin embargo podemos generalizar un poco tomando en cuenta que primero se diseña y después se desarrolla, así pues nuestro plan de recursos se resumiría como sigue:

Para la documentación del proyecto, incluyendo la etapa de planeación que es en la que nos encontramos, se cuenta con una microcomputadora PC compatible IBM, discos flexibles de 5 1/4 de pulgada con 360 KBytes de memoria, impresora, papel para impresión, procesador de palabras, entre otros recursos de tipo secundario. Como se ha mencionado a lo largo de este documento en capítulos anteriores, la Dirección General de Servicios de Cómputo para la Administración es el organismo encargado de proveer de tales recursos desde su fase inicial hasta su fase final.

Las etapas de análisis y diseño, que también forman parte de la documentación del proyecto, no necesitarán de más recursos que los que se mencionaron anteriormente.

En la parte de desarrollo del sistema, se contará con elementos tales como la microcomputadora tipo PC, con disco duro, discos flexibles de 5 1/4 de pulgadas con 360 KBytes de memoria, lenguajes de tercera y cuarta generación, paquetes procesadores de texto, impresora y papel para impresión.

Cabe señalar que estos recursos se encontrarán a disposición sin horario fijo y el lugar de trabajo será la misma Dirección General de Servicios de Cómputo para la Administración.

C) ESTRUCTURA ORGANIZACIONAL

Como se había mencionado anteriormente, la estructura organizacional consiste en la definición de funciones y distribución de responsabilidades entre las personas que desarrollarán el presente trabajo.

Este punto de la planeación en nuestro caso será muy sencillo. Básicamente quienes se encargarán del trabajo de desarrollo, así como de la documentación del proyecto, serán las autoras de esta tesis.

El responsable del proyecto ante las autoridades, es el Ing. Sebastián Poblano Ordoñez, jefe del Departamento de Asesoría e Implantación de Sistemas de la Dirección General de Servicios de Computo para la Administración de la UNAM, además el mismo coordinará y asesorará en todo el desarrollo del proyecto.

Por otro lado, también se podrá contar con la asesoría del personal que labora en la DGSCA en el departamento donde se desarrolla este proyecto, para todo lo referente a los conceptos administrativos y técnicas que se requieran conocer.

Debido a que los principales encargados del desarrollo del sistema, requieren conocer todos los aspectos relacionados con éste, no habrá divisiones radicales en la distribución del trabajo, más bien se trabajará conjuntamente en cada una de las etapas que se incluyen en el trabajo.

D) MECANISMOS DE SUPERVISION Y CONTROL

Una parte importante dentro del diseño y desarrollo de sistemas es sin duda alguna, la supervisión y control de éste, es decir, de qué manera se va determinando el avance del sistema, tomando en cuenta todos los factores que influyen en su creación.

Para este caso, en el desarrollo del Sistema de Control Presupuestal Programas se han determinado ciertos mecanismos, estos consisten de las siguientes etapas:

- 1: Coordinación de la fase de diseño.
- 2: Una vez concluida la fase anterior contamos con la coordinación de la fase de desarrollo del sistema, esto involucra toda la etapa de programación así como las pruebas modulares.
- 3: Coordinación de la fase de pruebas del sistema una vez concluido éste y de manera global.
- 4: Coordinación de la fase de implantación del sistema.

La estimación de tiempos de cada una de las fases no serán determinados por ahora.

Cada coordinación y supervisión se encontrarán a cargo del Ing. Sebastián Poblano Ordoñez.

El control del proyecto se encontrará a cargo de las personas autoras de este documento, esto permitirá que el sistema sea dinámico en todas sus fases, tratando de lograr la excelencia en su diseño, programación y presentación.

Tanto la coordinación como la supervisión y control se realizarán periódicamente según se vaya avanzando en las diferentes etapas que comprenden el sistema.

C A P I T U L O V

ANALISIS DEL SISTEMA DE PRESUPUESTO

Uno de los aspectos más importantes que se puede obtener del análisis es el saber lo que se va a hacer. En nuestro caso para tener una mejor idea del "qué", es necesario conocer el contexto en el cual se encuentra el sistema que vamos a desarrollar, es decir, en forma general vamos a analizar el origen y destino de la información de presupuesto y vamos a ver en qué parte de todo el proceso presupuestal de las dependencias, interviene el control del ejercicio presupuestal.

Una vez conociendo el contexto, es necesario analizar el sistema de control del ejercicio presupuestal que actualmente está funcionando así como el nuevo desarrollo que se va a realizar. Este análisis se llevará a cabo haciendo uso de las técnicas estructuradas como el diagrama de flujo de datos (DFD), el diccionario de datos y las miniespecificaciones. Las dos últimas técnicas mencionadas se referirán solo al sistema a desarrollar, ya que éste englobará al sistema actual.

A) ANALISIS DEL SISTEMA EN SU CONTEXTO

Saber dónde y para qué va a servir el sistema que se va a desarrollar es algo que nos ayuda a entender mejor la problemática del sistema.

En nuestro caso, al analizar el sistema en su contexto también se podrá analizar al usuario del sistema. Con el conocimiento del tipo de usuario, se podrán entender mejor las características operacionales del sistema a desarrollar y que se definieron dentro de las necesidades que se deben cumplir en el nuevo sistema.

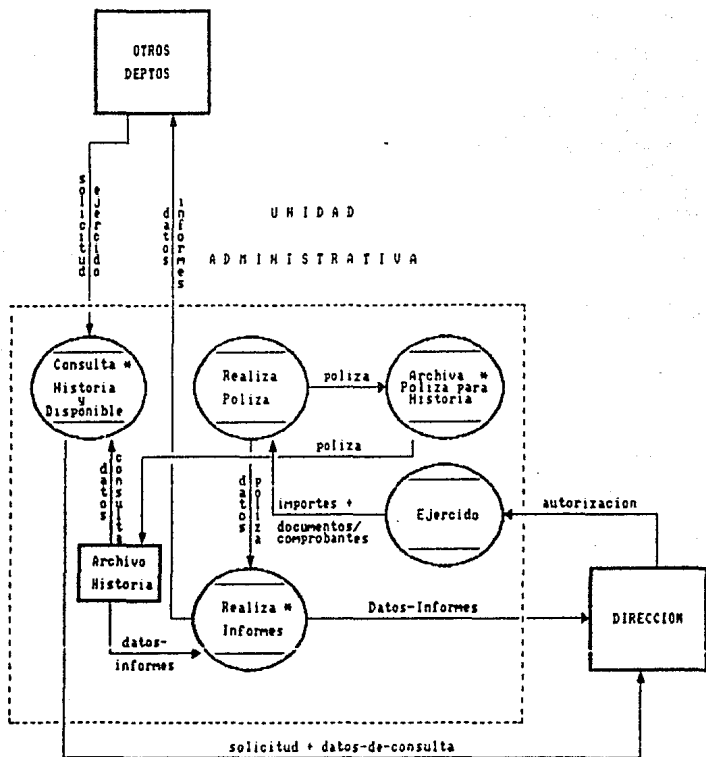
El control del ejercicio del presupuesto es un proceso que requiere que se efectúen ciertas actividades, las cuales se inician desde el momento en que se requiere ejercer, ya que para esto hay que saber si se puede ejercer, para obtener tal conocimiento se consultan totales de lo disponible y algunas veces es necesario conocer la historia de lo ya ejercido.

El sistema que se pretende desarrollar es una herramienta para la realización de las actividades que se necesitan hacer, por ejemplo mantener la historia del ejercicio presupuestal al día, hacer la consulta en forma rápida y directa y podrá ahorrar todo el trabajo referente a la realización de informes.

En el diagrama de flujo de datos que se presentará a continuación, se podrán apreciar las principales actividades, descritas en forma de procesos que ya se han mencionado. En el diagrama también se podrá observar que el usuario del sistema de control presupuestal se localiza generalmente en lo que en la UNAM se denomina Unidad Administrativa. Es decir es aquí donde se define al usuario de nuestro sistema.

El usuario, que generalmente pertenecerá a la unidad administrativa, será una persona con conocimientos de la técnica programática para el presupuesto y todos los conceptos que se manejan alrededor de esta área. Por otro lado, la mayoría de los usuarios tendrán poco o nada de conocimientos sobre tópicos de computación.

CONTEXTO DEL SISTEMA DE PRESUPUESTO



* Actividades referentes al control del ejercicio presupuestal

B) ANALISIS DEL SISTEMA ACTUAL

En el siguiente diagrama de flujo de datos se podrán observar los procesos y los archivos con los que cuenta actualmente el sistema de control del ejercicio presupuestal. En dicho diagrama se reflejan las relaciones existentes entre los procesos y los archivos.

Los procesos de asignación y actualización tiene una función muy similar, ya que ambos consultan el archivo que contiene los códigos programáticos (prespto.dat) y actualizan al archivo de transacciones (histpre.dat), aunque la validación que hacen es inversa con respecto a la existencia del código programático de la transacción. es decir, mientras en la asignación no se permite dar de alta registros cuyo código ya ha sido asignado, en la actualización no se permite dar altas con códigos que no hayan sido asignados. Por otro lado, la asignación actualiza el archivo prespto.dat, el proceso de actualización solo consulta a dicho archivo.

Quando se comienza a utilizar el sistema de control del ejercicio presupuestal, lo primero que se realiza es la asignación. Para todas las transacciones que se dan de alta por medio de este proceso no es necesario introducir la clave de la transacción por que de antemano se puede saber.

El proceso de modificación se encuentra fuera del menú principal del sistema por dos razones: la primera es que este módulo se hizo con posterioridad a todos los demás módulos del sistema, y segundo por que el usuario al no ver un proceso de modificación en el menú pone más atención al proporcionar los datos al sistema.

A todos los usuarios del sistema de presupuesto se les hace la recomendación de hacer respaldos de sus datos cada determinado tiempo y en forma especial antes de realizar el proceso de cierre de periodo, para que los usuarios puedan realizar dicho respaldo es necesario que se le entrene en el manejo básico de directorios y comandos para respaldar del sistema operativo MSDOS, ya que es con este con el que se creó el sistema de presupuesto y por lo tanto funciona con él, además de que el MSDOS es uno de los más generalizados aparte de que es el que proporcionan la mayoría de los fabricantes de equipo PC compatibles con IBM.

C) ANALISIS DEL SISTEMA A DESARROLLAR

Al empezar a analizar los objetivos del sistema, se distinguen tres subsistemas que se pueden realizar y manejar en forma independiente.

A continuación explicaremos brevemente cada subsistema, y en seguida se presentarán los diagramas de flujo de datos de cada uno de ellos.

I.- Por una parte se tendrá el subsistema procesador de la información o de operación, que es el objeto principal de este trabajo. Este sistema será el encargado de mantener los archivos de datos actualizados y correctos. Para llevar a cabo el cumplimiento de esta función se tendrá los procesos de altas en los que se incluyen la asignación y actualización del presupuesto, y un proceso de modificación.

Otra función que tiene que realizar el sistema es procesar los datos de la forma en que lo requiera el usuario y presentárselos, para esto se contará con los módulos de informes y consulta.

Con objeto de hacer más accesible al sistema al usuario, el sistema también incluirá dos módulos para protección de su información. Estos son los módulos de respaldo y recuperación de archivos.

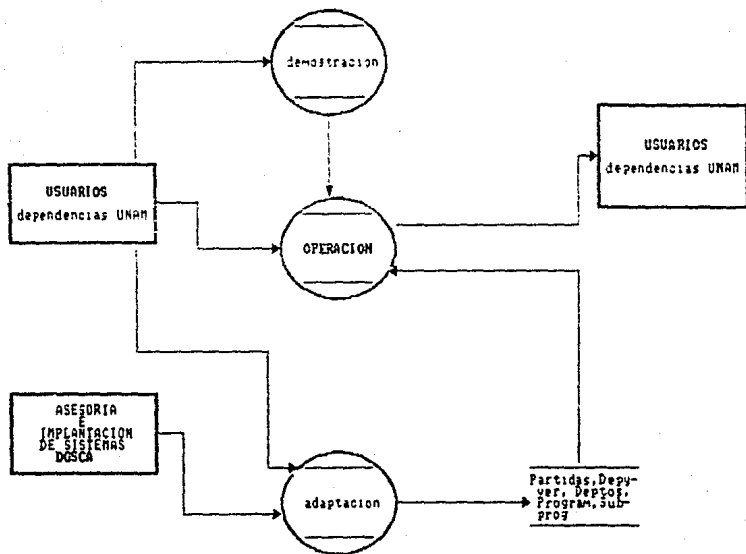
Además este sistema contará con diferentes niveles de ayuda, es decir, una explicación de cómo se opera el sistema en los diferentes procesos. La explicación de la operación se encuentra almacenada en archivos.

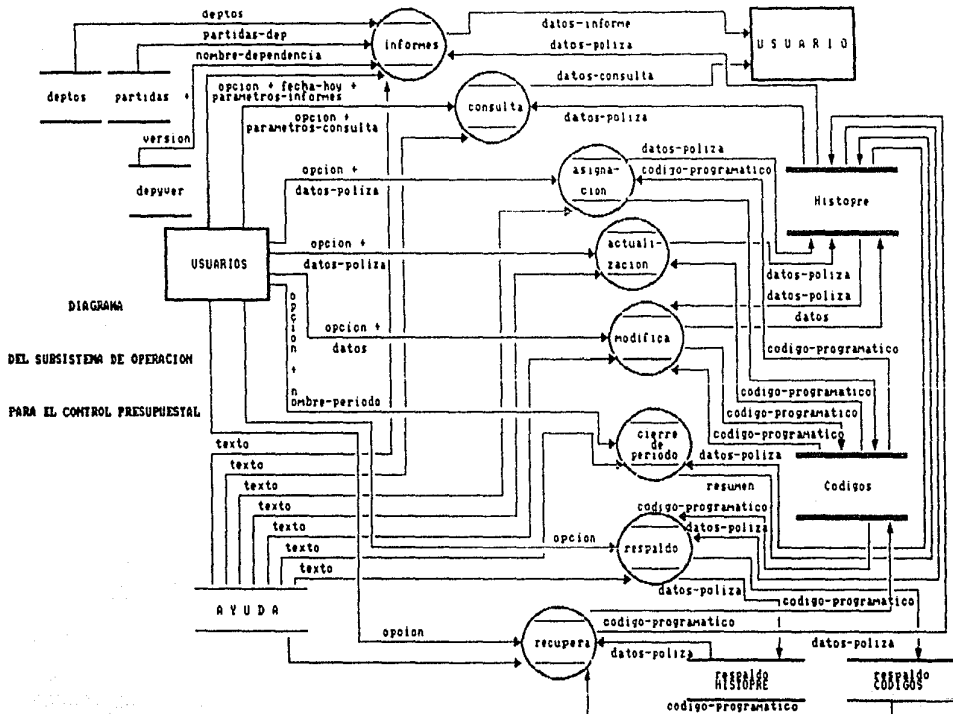
Todos los procesos anteriores, sus relaciones por medio de los datos y los archivos se muestran en Diagrama del Subsistema de Operación para el Control Presupuestal. También se incluyen diagramas más detallados de cada proceso.

II.- Por otro lado, se requiere que el sistema sea fácilmente adaptable a diferentes dependencias. Para esto se crearán diferentes archivos que contienen la información específica para cada dependencia. La representación gráfica de este subsistema lo podemos encontrar en el Diagrama del Subsistema para Adaptación del Sistema de Presupuesto.

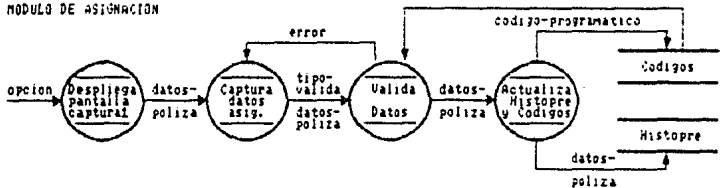
III.- Finalmente la última parte del sistema, es lo referente a la presentación o autodemstración del sistema. Esto lo podemos apreciar en el Diagrama del Subsistema para Demostración.

DIAGRAMA GENERAL DEL SISTEMA DE CONTROL PRESUPUESTAL

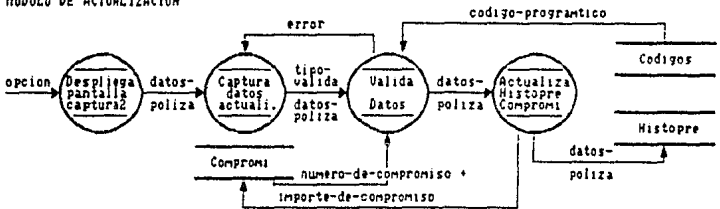




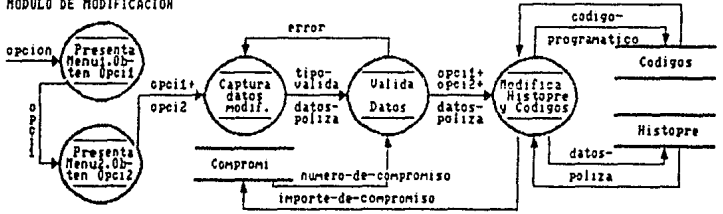
MODULO DE ASIGNACION



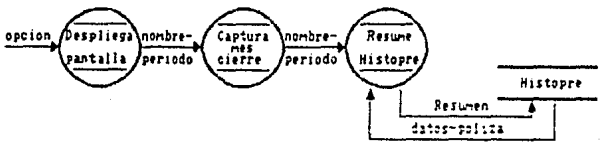
MODULO DE ACTUALIZACION



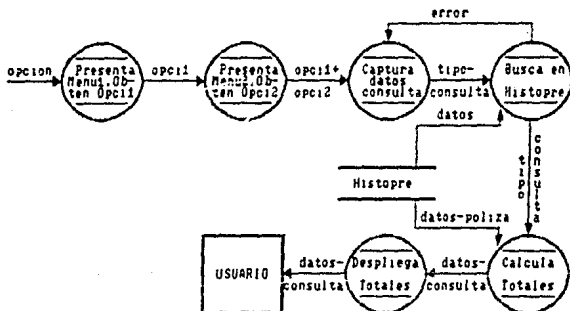
MODULO DE MODIFICACION



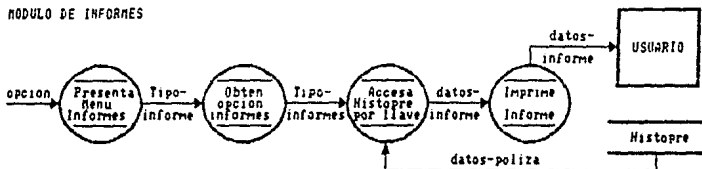
MODULO DE CIERRE DE PERIODO



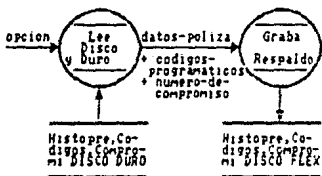
MODULO DE CONSULTA



MODULO DE INFORMES



MODULO DE RESPALDO



MODULO DE RECUPERACION

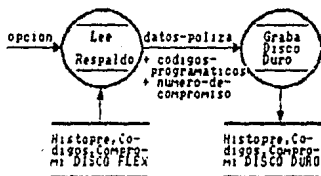


DIAGRAMA DEL SUBSISTEMA PARA ADAPTACION

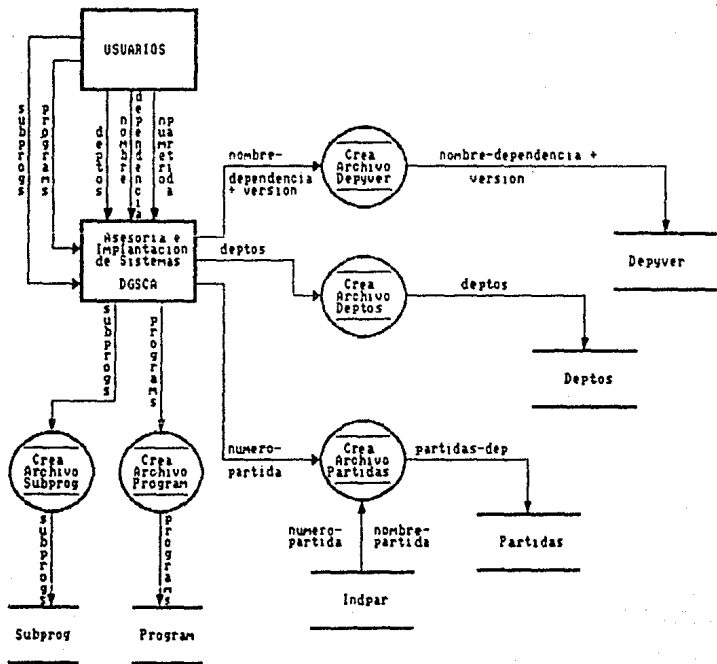
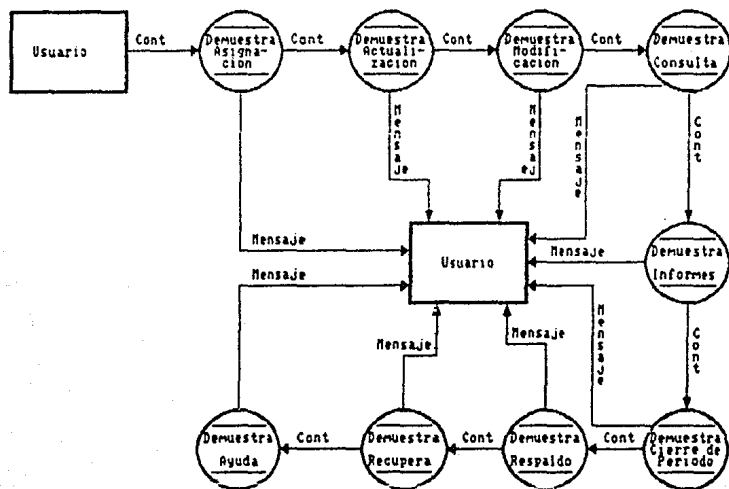


DIAGRAMA DEL SUBSISTEMA PARA DEMOSTRACION



D) DICCIONARIO DE DATOS

La descripción de los datos que se utilizan en el control presupuestal y que se especifican en el DFD. En la metodología de desarrollo se presenta como el diccionario de datos.

A continuación se definen dichos datos.

Datos-Póliza = número de póliza
+ fecha movimiento
+ código programático
+ clave transacción
/ número de compromiso
+ descripción
+ importe.

Datos = número de póliza
/ fecha movimiento
/ código programático
/ clave transacción
/ número de compromiso
/ descripción
/ importe.

Código-Programático = programa
+ subprograma
+ dependencia
+ subdependencia
+ partida.

Número-de-Compromiso = número de compromiso o
número de referencia.

Importe-de-Compromiso = importe asociado a un número
de compromiso.

Fecha-hoy = día hoy
+ mes hoy
+ año hoy.

Tipo-Informe = número asociado al tipo de informe =
1/2/3/4/5.

Datos-Informe = encabezado
+ nombre dependencia
/ nombre partidas
/ nombre departamentos
+ totales.

Tipo-Consulta = tipo de consulta
+ global o uresp
+ (número de póliza
/ número de grupo
/ número partida.

Datos-Consulta = importes
+ totales.

Nombre-Periodo = nombre del mes de cierre
/ (nombre del mes de cierre
+ número de cierre en el mes).

Resumen = número de póliza
+ fecha cierre
+ código programático
+ clave transacción
/ número de compromiso
+ ("Total al cierre " + nombre periodo)
+ (total asignado por código programático
/ total transferencia por código progrático
/ total ampliación por código programático
/ total compromiso por código programático
/ total ejercido por código programático
/ total reasignación por código programático).

Nombre-Dependencia = nombre de la dependencia.

Versión = año en el que se va a utilizar el sistema.

Deptos = clave número de departamento
+ nombre del departamentos.

Número-Partida = número de las partidas.

Nombre-Partida = nombra de la partida.

Partidas-Dep = número partidas
+ nombre partidas.

Programs = número de los programas.

Subprog = número de los subprogramas.

Mensaje = Explica y demuestra módulo informes
/ Explica y demuestra módulo consulta
/ Explica y demuestra módulo asignación
/ Explica y demuestra módulo actualización
/ Explica y demuestra módulo modificación
/ Explica y demuestra módulo cierre de periodo
/ Explica respaldo
/ Explica recuperación
/ Explica y demuestra ayuda.

Texto = (Explica Fecha
+ (Explica informe global por partida
/ Explica informe global por departamento
/ Explica informe detallado por partida
/ Explica informe detallado depto.partida
/ Explica informe concentrado depto.grupo))
/ (Explica consulta por grupo
/ Explica consulta por póliza
/ Explica consulta por partida
/ Explica consulta por totales)
/ ((Explica proceso asignación
/ Explica proceso actualización
/ Explica proceso modificación)
+ (Explica fecha
/ Explica póliza
/ Explica código programático
/ Explica clave transacción
/ Explica descripción
/ Explica importe))
/ (Explica cierre de periodo
+ Explica nombre de periodo)
/ (Explica proceso de respaldo
/ Explica proceso recupera).

Cont = Opción para continuar.

Opción = Asignación
/ Actualización
/ Modificación
/ Consulta
/ Informes
/ Cierre de Periodo
/ Respaldo
/ Recupera
/ Ayuda
/ Termina.

Opci1 = (tipo de consulta =
 consulta por póliza
 / consulta por partida
 / consulta por grupo
 / consulta por totales)
/ (tipo de modificación =
 modificación por poliza
 / modificación por código programático
 / modificación por campo).

Opci2 = (tipo de consulta =
 consulta global
 / consulta por unidad responsable)
/ (tipo de modificación =
 corrección
 / cancelación).

Error = dato o parámetro inválido.

Tipo-Valida = parámetro para validar un dato =
1/2/3/4/5/6/7/8

E) MINIESPECIFICACIONES

Las miniespecificaciones son las especificaciones de los procesos que se determinaron en DFD. Es decir, se explicará en forma general pero lo más claro posible, las acciones que realizarán los procesos.

! Proceso: Asignación !

Especificaciones:

Entrada. Se piden los datos-póliza al usuario por medio de la pantalla.

Proceso. Básicamente se trata de un proceso de validación de los datos que se capturan. Para esto se busca el código programático en el archivo de códigos ya asignados. Si ya existe el código no permite la salida del proceso. Si no existe realiza la salida del proceso. De los demás datos se verifican que sean del tipo correcto y que estén definidos en la versión que se utiliza.

Salida. La salida del proceso es la actualización o el registro de las nuevas transacciones tanto en el archivo histopre como en el de códigos.

! Proceso: Actualización !

Especificaciones:

Entrada. Se piden los datos-póliza al usuario por medio de la pantalla.

Proceso. También se trata de un proceso de validación y actualización. La validación en este proceso es contraria a la de la asignación con respecto al código programático. Los demás datos se validan como en la asignación, a excepción del número de compromiso o de referencia, para transacciones que lo requieran, en este caso se verifica la existencia o no de dicho número y que los importes se encuentren dentro del rango.

Salida. La salida del proceso es el registro de las nuevas transacciones en el archivo histopre.

! Proceso: Modifica !

Especificaciones:

Entrada. La entrada a este procesos es el dato del archivo que se va a modificar y el nuevo valor de dicho dato.

Proceso. Valida el nuevo valor que se pretende actualizar. Localiza el registro a ser localizado y realiza la modificación.

Salida. Dejar en el archivo de Histopre los datos ya modificados.

! Proceso: Cierre de Periodo !

Especificaciones:

Entrada. El nombre del periodo del cierre, generalmente es el mes en el que se realiza el cierre:

Proceso. Obtiene el total de todas las transacciones para cada código programático. Las descripciones son sustituidas por un mensaje que indica que los importes son los totales del cierre efectuado.

Salida. Los totales sustituyen a todos los registros del archivo histopre.

! Proceso: Consulta !

Especificaciones:

Entrada. El tipo de consulta, es decir, dato por el que se quiere buscar.

Proceso. Buscar en el archivo histopre los registros que cumplan con el tipo de consulta requerido. Se obtienen algunos totales.

Salida. Mostrar en pantalla los datos y totales que resulten del proceso.

! Proceso: Informes !

Especificaciones:

Entrada. El tipo de informe, es decir, el detalle y el nivel de totales que se quieren obtener.

Proceso. Ordenar la información de la forma requerida para el informe. Generalmente se trata de obtener totales a diferentes niveles.

Salida. Sacar a impresión la información procesada.

! Proceso: Respaldo !

Especificaciones:

Entrada. Los datos de los archivos histopre y códigos.

Proceso. Copiar, hacer duplicado de los datos. El respaldo se realiza de datos del disco duro a disco flexible.

Salida. Dejar en disco flexible una copia de los archivos histopre y códigos.

! Proceso: Recupera !

Especificaciones:

Entrada. Los datos de los archivos histopre y códigos del disco flexible.

Proceso. Copiar, hacer duplicado de los datos que se encuentran en el respaldo. La recuperación es de los datos del disco flexible a disco duro.

Salida. Dejar en disco duro una copia de los datos del disco flexible.

! Proceso: Crea-Archivo-Depyver !

Especificaciones:

Entrada. El nombre de la dependencia a la que se le va a dar una versión del sistema y el año de la versión del sistema en el que se va a utilizar.

Proceso. Crear un archivo con el nombre de la dependencia y el año en el que se utilizará la versión.

Salida. Dejar en memoria secundaria el nombre de la dependencia.

! Proceso: Crea-Archivo-Deptos !

Especificaciones:

Entrada. Nombre de los departamentos que determine la dependencia a la que se le adapta el sistema y una clave asignada a cada departamento.

Proceso. Crear un archivo que contenga la clave y nombre de los departamentos que maneja la dependencia.

Salida. Dejar en memoria secundaria las claves y nombres de los departamentos.

! Proceso: Crea-Archivo-Partidas !

Especificaciones:

Entrada. El número de las partidas que utilizará la dependencia y los registros del archivo de las todas las partidas.

Proceso. Crear un archivo de partidas a partir de otro archivo que contiene todas las partidas que maneja la universidad. El nuevo archivo solo contendrá las partidas que requiera la dependencia.

Salida. Dejar en memoria secundaria, el número y descripción de las partidas de la dependencia.

! Proceso: Crea-Archivo-Program !

Especificaciones:

Entrada. El número de los programas que utilizará la dependencia.

Proceso. Crear un archivo de que contenga el número de todos los programas que vaya a utilizar la dependencia.

Salida. Dejar en memoria secundaria, el número de los programas que utiliza la dependencia.

! Proceso: Crea-Archivo-Subprog !

Especificaciones:

Entrada. El número de los subprogramas que utilizará la dependencia.

Proceso. Crear un archivo de que contenga el número de todos los subprogramas que vaya a utilizar la dependencia.

Salida. Dejar en memoria secundaria, el número de los subprogramas que utiliza la dependencia.

! Proceso: Demuestra-Asignación !

Especificaciones:

Entrada. Opción para iniciar la demostración.

Proceso. Dar una explicación de lo que realiza el proceso de asignación y una demostración. Para esto se utilizarán ejemplos concretos y específicos.

Salida. Mostrar en pantalla las explicaciones y demostraciones.

! Proceso: Demuestra-Actualización !

Especificaciones:

Entrada. Opción para continuar la demostración.

Proceso. Dar una explicación de lo que realiza el proceso de actualización y una demostración. De nuevo para este proceso se utilizaran ejemplos con datos concretos y específicos.

Salida. Mostrar en pantalla las explicaciones y demostraciones.

! Proceso: Demuestra-Modificación !

Especificaciones:

Entrada. Opción para continuar la demostración.

Proceso. Dar una explicación de lo que realiza el proceso de modificación y una demostración. Como en la asignación y en la actualización se pretende que haya ejemplos para que la explicación se más fácil y la demostración más completa.

Salida. Mostrar en pantalla las explicaciones y demostraciones.

! Proceso: Demuestra-Consulta !

Especificaciones:

Entrada. Opción para continuar la demostración.

Proceso. Dar una explicación de lo que realiza el proceso de consulta y una demostración. Se tratarán se seguir paso a paso el proceso que realmente se efectúa en el subsistema de operación.

Salida. Mostrar en pantalla las explicaciones y demostraciones.

! Proceso: Demuestra-Informes !

Especificaciones:

Entrada. Opción para continuar la demostración.

Proceso. Dar una explicación de lo que realiza el proceso de informes y una demostración de los menús y la información que se requiere. Una demostración estrictamente en este caso no se hará por que se limitaría a tener una impresora al momento de la demostración.

Salida. Mostrar en pantalla las explicaciones y demostraciones.

! Proceso: Demuestra-Cierre-de-Periodo !

Especificaciones:

Entrada. Opción para continuar la demostración.

Proceso. Dar una explicación de lo que realiza el proceso de cierre de periodo. Se ejemplificarán los resultados de este proceso haciendo incapié en la recomendaciones que se hacen al utilizar este proceso.

Salida. Mostrar en pantalla las explicaciones.

! Proceso: Demuestra-Respaldo !

Especificaciones:

Entrada. Opción para continuar la demostración.

Proceso. Dar una explicación de lo que realiza el proceso de respaldo. Una demostración estricta no se hará por que se requiere tener un disco flexible limpio.

Salida. Mostrar en pantalla las explicaciones.

! Proceso: Demuestra-Recuperación !

Especificaciones:

Entrada. Opción para continuar la demostración.

Proceso. Dar una explicación de lo que realiza el proceso de recuperación. Como en el caso anterior la demostración no se hará debido a que se requiere de un disco flexible con datos.

Salida. Mostrar en pantalla las explicaciones.

! Proceso: Demuestra-Ayuda !

Especificaciones:

Entrada. Opción para continuar la demostración.

Proceso. Dar una explicación de cómo funciona la ayuda del subsistema de operación. Y dar un ejemplo de como se presenta esta opción.

Salida. Mostrar en pantalla las explicaciones.

Todo lo anterior nos da una clara idea de lo que se quiere hacer, que es el objetivo del análisis. Además también ya se definieron los datos que se necesitarán, la forma en que serán utilizados, es decir si los datos quedarán archivados, y también se tienen definidos los procesos.

C A P I T U L O V I

DISEÑO DEL SISTEMA DE PRESUPUESTO

El diseño es la etapa anterior al desarrollo, es aquí donde se debe definir "como" se hará el desarrollo. Siguiendo con las técnicas estructuradas, la primera parte del diseño consiste en realizar la carta estructurada. En dicha carta se especifican los procesos que se definieron en el DFD del análisis, subdividiéndolos en pequeños módulos independientes.

En el pseudocódigo o español estructurado se describen detalladamente los procesos que se involucran en el sistema.

Antes de iniciar el desarrollo es necesario conocer "con qué" se va a realizar, por lo que es en el diseño, por ser la última etapa antes del desarrollo, donde se define el software necesario.

En esta etapa se puede definir en forma más explícita las entradas, salidas y archivos que utiliza el sistema, e inclusive se puede pensar en el diseño de éstas tomando en cuenta las facilidades del software a utilizar.

A) CARTA ESTRUCTURADA

La estructura de software se define como la representación jerárquica que indica la relación entre módulos o elementos del software empleados para resolver el problema definido en el análisis de requerimientos. La evolución de la estructura de software comienza con la definición del problema; y la solución del mismo ocurre cuando cada parte del problema es resuelto por uno o más elementos de software.

La estructura de software representa una arquitectura que implica una jerarquización del control. Cabe señalar que la estructura de software no representa el flujo de los datos a través de los módulos, aspectos procedurales como secuencia de procesos, ni formas de decisiones dentro de dicha estructura.

Es importante simplificar al máximo una estructura de software ya que de ello depende que tan complejo será el desarrollo del sistema. Si se cuenta con un diseño complejo será difícil llegar a un desarrollo simple y viceversa.

Dentro del diseño de la estructura de software existen diversas metodologías y cada una de ellas es empleada de acuerdo al tipo de sistema que se esté desarrollando, para este caso la metodología de Diseño Estructurado está enfocada a sistemas científicos y administrativos.

Metodología de Diseño Estructurado.

Esta metodología fue desarrollada por Constantine y Yourdon, que desde los 60's se han dedicado a implementar diversas técnicas enfocadas al diseño estructurado. Para llevar a cabo esta metodología se tienen los siguientes pasos a seguir:

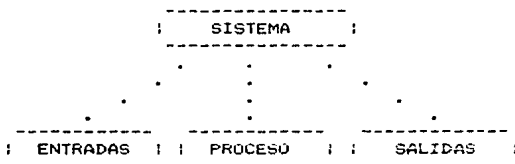
- 1: Se hace una revisión y evaluación del diagrama de flujo de datos (DFD) de acuerdo a los requerimientos.
- 2: Se subdivide dicho DFD en subsistemas.
- 3: Se evalúa para cada subsistema los recursos y el tipo de sistema.
- 4: Para cada subsistema se realiza una revisión de los archivos definiéndolos físicamente.

5: Se evalúa el tipo de diseño.

- a) Transformación.
- b) Transacciones.

Diseño por Transformación.

En términos generales, en el diseño por transformación se determina el punto de máxima abstracción del sistema que se define como el lugar en el diagrama de flujo de datos (DFD) en el cual las entradas se procesan para convertirse en salidas. El modelo de estructura de software que se aplica es el siguiente:

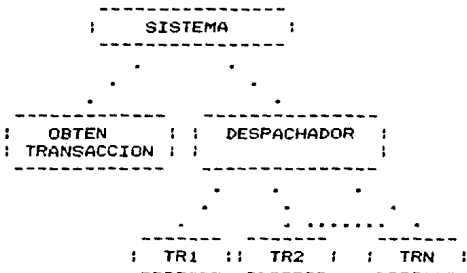


El punto de máxima abstracción se localiza en la parte del PROCESO que es realmente donde se realiza la transformación. Una vez determinada la carta estructurada se aplican sobre ella los criterios de cohesión y acoplamiento definidos en el capítulo segundo y se genera una nueva estructura.

Diseño por Transacciones.

El modelo por transacciones se caracteriza por el movimiento de los datos a través de un camino de recepción denominado 'Obtención de la Transacción' que convierte la información externa en una transacción. La información permanece estática hasta no obtener un valor externo que determine el flujo de los datos a través de una ruta denominada 'Despachador'. El despachador determina el flujo de la información a través de una o más transacciones contenidas en él, pero el flujo de los datos sólo podrá ser por una ruta a la vez, ello depende del valor externo. El punto en el cual la información puede tomar varios caminos (dentro del DFD) se le denomina punto de transacción.

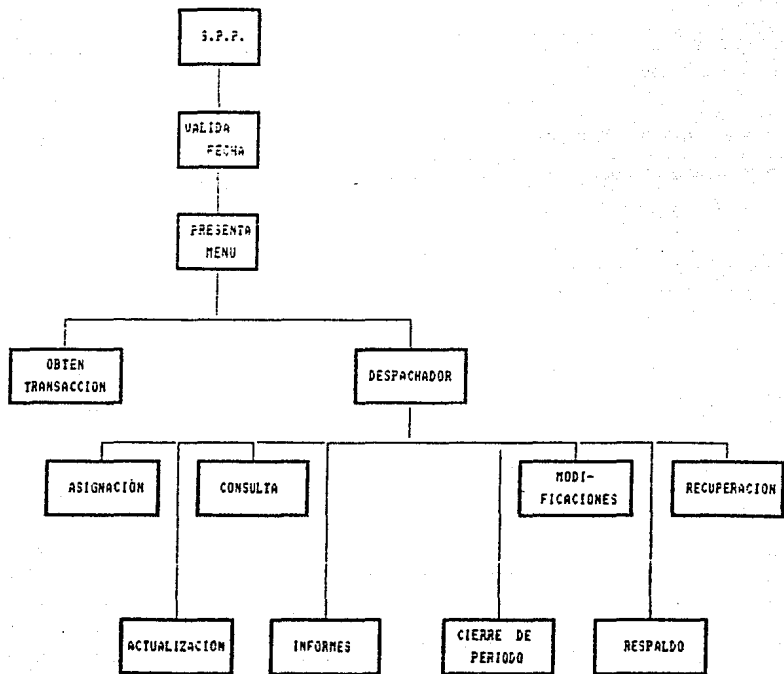
Una vez que se ha hecho una revisión del DFD, de los archivos y estructuras de datos a ser empleadas se procede a determinar que partes del DFD tienen el modelo por transformación y cuales por transacciones, una vez identificadas las partes se localiza el punto de transacción y las características de cada camino de flujo de los datos. Finalmente se emplea una estructura que tiene el siguiente modelo:



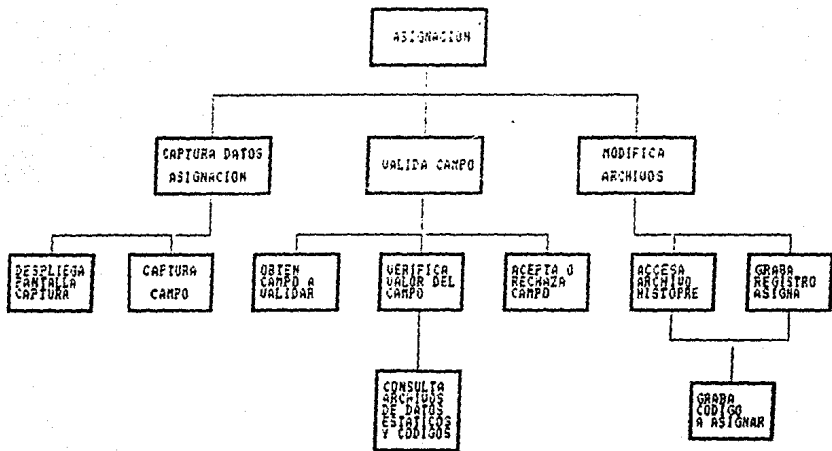
Para nuestro desarrollo en particular se empleará el diseño por transacciones dado que nuestro sistema contará con diferentes módulos de elección.

En las siguientes páginas se presentan los diagramas de las cartas estructuradas de cada uno de los módulos o procesos que se incluyen dentro del sistema de control del ejercicio presupuestal.

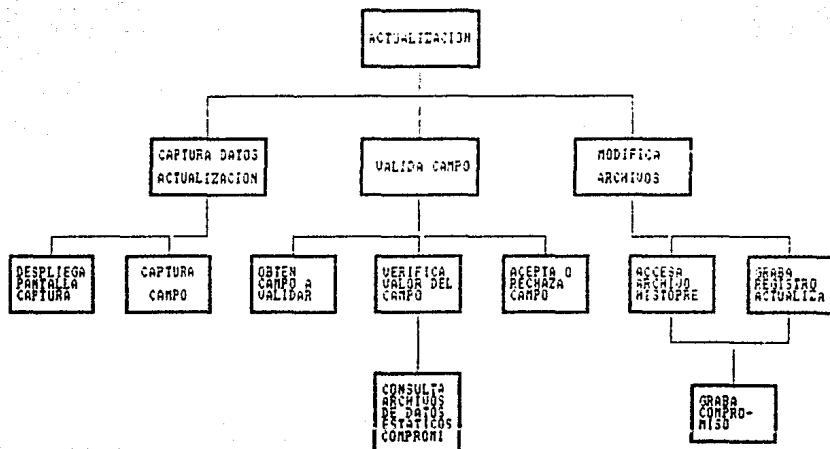
CARTA ESTRUCTURADA DEL SUBSISTEMA DE OPERACION



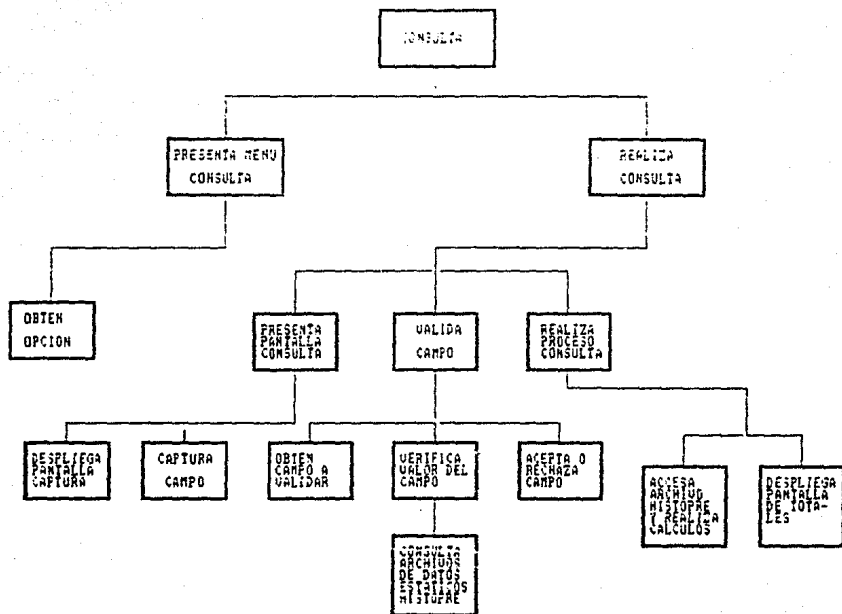
CARTA ESTRUCTURADA DEL MÓDULO DE ASIGNACION



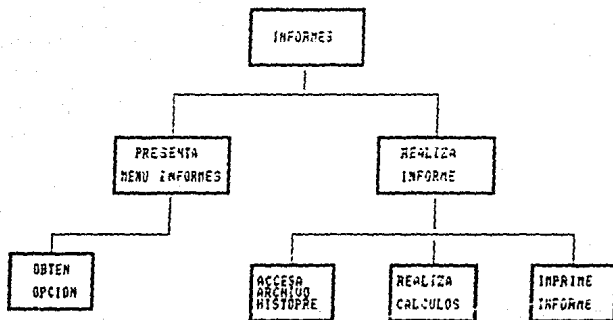
CARTA ESTRUCTURADA DEL MODULO DE ACTUALIZACION



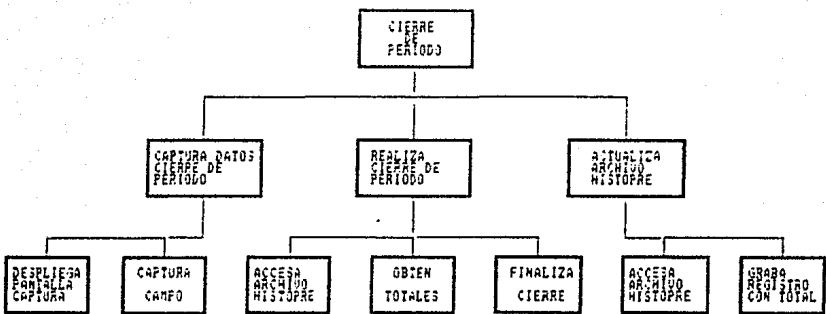
CARTA ESTRUCTURADA DEL MÓDULO DE CONSULTA



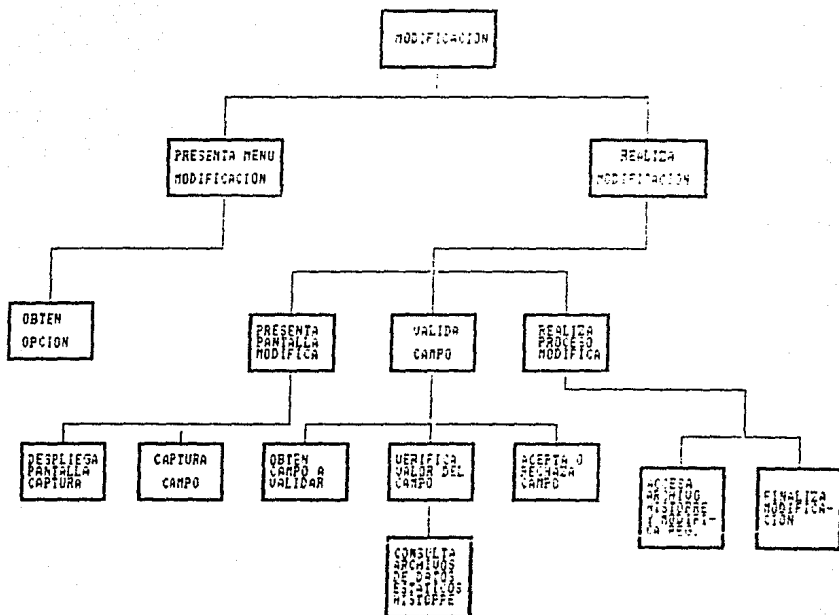
CARTA ESTRUCTURADA DEL MÓDULO DE INFORMES



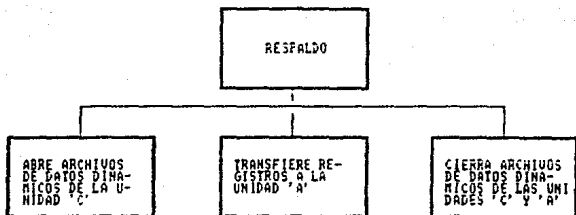
CARTA ESTRUCTURADA DEL MODULO DE CIERRE DE PERIODO



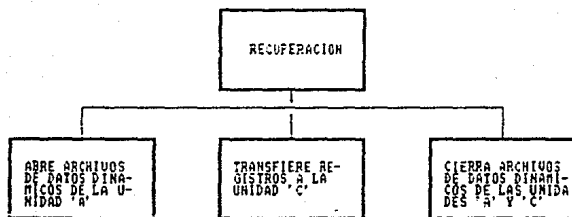
CARTA ESTRUCTURADA DEL MODULO DE MODIFICACION



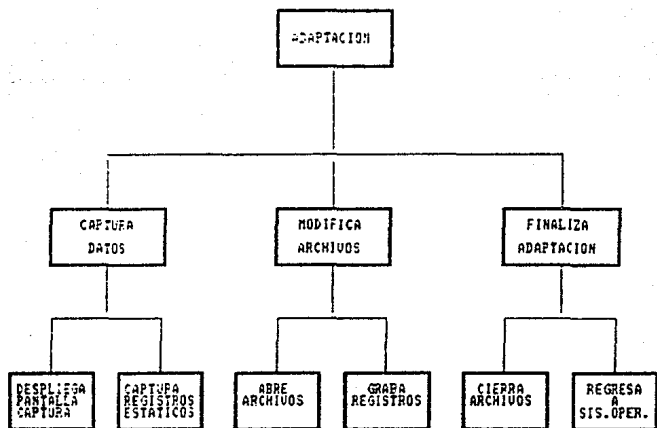
CARTA ESTRUCTURADA DEL MODULO DE RESPALDO



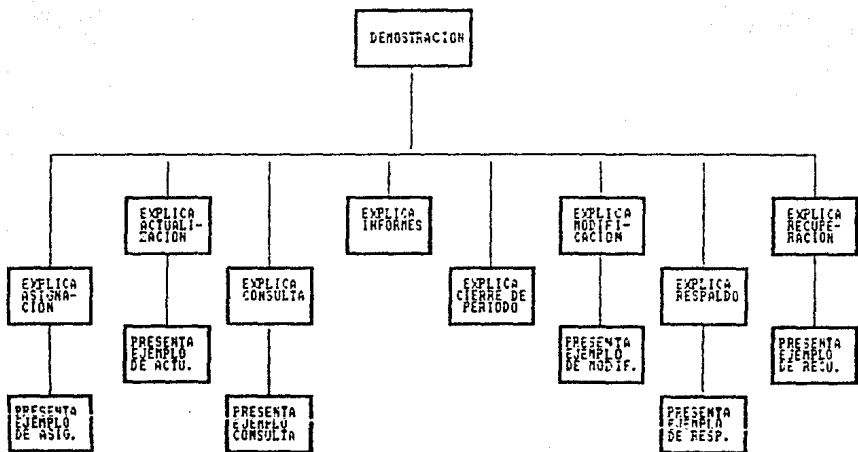
CARTA ESTRUCTURAL DEL MÓDULO DE RECUPERACION



CARTA ESTRUCTURADA DEL SUBSISTEMA DE ADAPTACION



CARTA ESTRUCTURADA DEL SUBSISTEMA DE DEMOSTRACION



B) PSEUDOCODIGO

El pseudocódigo llamado Program Design Language (PDL) es un 'lenguaje' auxiliar que utiliza el vocabulario de algún lenguaje y la sintaxis de otro. El PDL es usado comúnmente para describir un diseño específico, combinando una forma de narrativa sencilla con una manera formal de procedimientos que pueden ser transferidos posteriormente a un lenguaje de alto nivel u otro.

Para nuestro diseño emplearemos una forma de PDL en español utilizando al mismo tiempo estructuras de los lenguajes de programación estructurados ya conocidas, la finalidad de ello es detallar los procesos especificados anteriormente.

SUBSISTEMA DE OPERACION PARA EL CONTROL PRESUPUESTAL

: Módulo: Asignación :

Proceso 1.0 Despliega-Pantalla-Captura

Entradas : Opción

Pseudocódigo:

Limpia pantalla
presenta marco
presenta encabezados
despliega 'póliza'
despliega 'fecha'
despliega 'descripción'
despliega 'importe'
despliega 'código programático'

Salida: Datos-Póliza

Proceso 2.0 Captura-Datos-Asig

Entradas : Datos-Póliza

Pseudocódigo:

Mientras no FIN
captura póliza
ve a valida
captura fecha
ve a valida
captura descripción
captura importe
ve a valida
captura código programático
ve a valida
ve a actualiza-histopre-códigos
ve a despliega-pantalla-captura-asig
fin mientras

Salidas: Datos-Póliza. Tipo-Valida

Proceso 3.0 Actualiza-Histopre-Códigos

Entradas : Datos-Póliza

Pseudocódigo:

```
Usa archivo HISTOPRE
graba hclavmov
graba hdescrip
graba hnumref
graba himporte
graba hfecha
graba hpóliza
graba hprogram
graba hsubprog
graba hpartida
graba hdeparto
graba hdictrol
Usa archivo CODIGOS
graba cprogram
graba csubprog
graba cpartida
graba cdeparto
graba cdictrol
```

Salida: Datos-Póliza, Código-Programático

! Módulo: Actualización !

Proceso 4.0 Despliega-Pantalla-Captura2

Entrada: Opción

Pseudocódigo:

```
Limpiar pantalla
presentar marco
presentar encabezados
despliega póliza
despliega fecha
despliega tipo de movimiento
despliega número de referencia
despliega descripción
despliega importe
despliega código programático
```

Salida: Datos-Póliza

Proceso 5.0 Captura-Datos-Actualiz

Entrada: Datos-Póliza

Pseudocódigo:

```
Mientras no "FIN"  
  captura póliza  
  valida  
  captura fecha  
  valida  
  captura tipo de movimiento  
  valida  
  captura número de referencia  
  valida  
  captura descripción  
  captura importe  
  valida  
  captura código programático  
  valida  
  actualiza Histopre  
  despliega-pantalla-captura  
Fin mientras
```

Salida: Tipo-Valida, Datos-Póliza

Proceso 6.0 Actualiza-Histopre-y-Compromi

Entrada: Datos-Póliza

Pseudocódigo:

```
Activa archivo Histopre  
graba clave de movimiento  
graba descripción  
graba número de referencia  
graba importe  
graba fecha  
graba póliza  
graba programa  
graba subprograma  
graba partida  
graba departamentamento  
graba digito de control
```

Salida: Datos-Póliza, Número-de-Compromiso, Importe-de-Compromiso.

! Módulo: Modificación !

Proceso 7.0 Presenta-Menú-y-Obten-Opcii

Entrada: Opción

Pseudocódigo:

```
Limpia pantalla  
presenta marco  
presenta encabezados  
presenta tipos de modificación1  
captura opcii
```

Salida: Opcii

Proceso 8.0 Presenta-Menú2-y-Obten-Opci2

Entrada: Opci1

Pseudocódigo:

Limpia pantalla
presenta marco
presenta encabezados
presenta tipos de modificación2
captura opc12

Salida: Opci1, Opci2

Proceso 9.0 Captura-Datos-Modificar

Entrada: Opci1,Opci2

Pseudocódigo:

Si opc11='1' y opc12='1'
captura póliza actual
valida
captura póliza nueva
valida
modifica archivo Histopre
FIN SI
Si opc11='1' y opc12='2'
captura póliza a cancelar
valida
borra registro de Histopre
FIN SI
Si opc11='2' y opc12='1'
captura código programático actual
valida
captura código programático nuevo
valida
modifica archivo Histopre
FIN SI
Si opc11='2' y opc12='2'
captura código programático a cancelar
valida
borra registro de Histopre
FIN SI
Si opc11='3' y opc12='1'
captura código programático
valida
captura póliza
valida
captura fecha
valida
captura clave de movimiento
valida
presenta registros a modificar
captura de nuevo campos
modifica archivo Histopre
FIN SI
Si opc11='3' y opc12='2'
captura código programático
valida
captura póliza

valida
captura fecha
valida
captura clave de movimiento
valida
presenta registros a modificar
captura de nuevo campos
borra registro de Histopre

FIN SI

Salida: Tipo-Valida, Datos-Póliza

Proceso 10.0 Modifica-Histopre-y-Códigos

Entrada: Opc11, Opc12, Datos-Póliza

Pseudocódigo:

Si opc11='1' y opc12='1'
 usa archivo Histopre
 Mientras no fin de archivo
 busca póliza actual
 reemplaza póliza actual por
 póliza nueva

FIN Mientras

FIN SI

Si opc11='1' y Opc12='2'
 usa archivo Histopre
 Mientras no fin de archivo
 busca póliza actual
 borra registro

FIN Mientras

FIN SI

Si opc11='2' y opc12='1'
 usa archivo Histopre
 Mientras no fin archivo
 busca código actual
 reemplaza código actual por
 código nuevo

FIN Mientras

usa archivo Códigos
localiza registro con código=código
actual
borra registro

FIN SI

Si opc11='2' y opc12='2'
 usa archivo Histopre
 Mientras no fin de archivo
 busca código actual
 borra registro

FIN Mientras

usa archivo Códigos
localiza registro con código=código
actual
borra registro

FIN SI

```
Si opc11='3' y opc12='1'
  usa archivo Histopre
  reemplaza registro con campo capturado
FIN SI
Si opc11='3' y opc12='2'
  usa archivo Histopre
  borra registro con código, póliza,
  fecha y clave movimiento capturados
FIN SI
```

Salida: Código-Programático, Datos-Póliza

! Módulo: Cierre de Periodo !

Proceso 11.0 Despliega-Pantalla

Entrada: Opción

Pseudocódigo:

```
Limpia pantalla
presenta marco
presenta encabezados
despliega 'Nombre del mes de cierre'
captura nombre del mes de cierre
resume Histopre
```

Salida: Nombre-Periodo

Proceso 12.0 Resume Histopre

Entrada: Nombre-Periodo

Pseudocódigo:

```
Copia estructura de Histopre a Auxhist
activa archivo Histopre
Mientras no fin de archivo
  lee registro
  Si clave de movimiento='1'
    acumula importe ASIGNACION
  FIN SI
  Si clave de movimiento='2'
    acumula importe TRANSFERENCIA
  FIN SI
  Si clave de movimiento='3'
    acumula importe AMPLIACION
  FIN SI
  Si clave de movimiento='4'
    graba registro en Auxhist
  FIN SI
  Si clave de movimiento='5'
    graba registro en Auxhist
  FIN SI
  Si clave de movimiento='6'
    acumula importe EJER.-DIRECTO
  FIN SI
FIN Mientras
```

```

Si importe asignación > 0
  reemplaza registro en Auxhist con
  número de referencia='0000' y
  póliza='000'
FIN SI
Si importe transferencia > 0
  reemplaza registro en Auxhist con
  número de referencia='0000' y
  póliza='000'
FIN SI
Si importe ampliación > 0
  reemplaza registro en Auxhist con
  número de referencia='0000' y
  póliza='000'
FIN SI
Si importe ejercido-directo > 0
  reemplaza registro en Auxhist con
  número de referencia='0000' y
  póliza='000'
FIN SI
borra todos los registros de Histopre.DBF
copia todos los registros de Auxhist a
Histopre.DBF

```

Salida: Resumen

Módulo: Consulta

Proceso 13.0 Presenta-Menú1-y-Obten-Opci1

Entrada: Opci1

Pseudocódigo:

```

Limpia pantalla
Presenta marco
presenta encabezado
presenta tipos de consulta1
captura opción1

```

Salida: Opci1

Proceso 14.0 Presenta-Menú2-y-Obten-Opci2

Entrada: Opci1

Pseudocódigo:

```

Limpia pantalla
presenta marco
presenta encabezado
presenta tipos de consulta2
captura opción2

```

Salida: Opci1, Opci2

Proceso 15.0 Captura-Datos-Consulta

Entrada: Opci1, Opci2

Pseudocódigo:

```
Si opc11='1' y opc12='1'  
  captura dato=grupo  
  busca dato  
  captura dato=unidad responsable  
  busca dato  
FIN SI  
Si opc11='1' y opc12='2'  
  captura dato=grupo  
  busca dato  
FIN SI  
Si opc11='2' y opc12='1'  
  captura dato=póliza  
  busca dato  
  captura dato=unidad responsable  
  busca dato  
FIN SI  
Si opc11='2' y opc12='2'  
  captura dato=póliza  
  busca dato  
FIN SI  
Si opc11='3' y opc12='1'  
  captura dato=partida  
  busca dato  
  captura dato=unidad responsable  
  busca dato  
FIN SI  
Si opc11='3' y opc12='2'  
  captura dato=partida  
  busca dato  
FIN SI  
Si opc11='4' y opc12='1'  
  captura dato=unidad responsable  
  busca dato  
FIN SI  
Si opc11='4' y opc12='2'  
  calcula totales  
FIN SI
```

Salida: Tipo-Consulta

Proceso 16.0 Busca-en-Histopre

Entrada: Dato

Pseudocódigo:

```
Mientras dato=grupo  
  activa Histopre indexado por  
  grupo-u.responsable  
  localiza dato-grupo  
  Si fin de archivo  
    captura dato  
  FIN SI  
  calcula totales  
FIN Mientras
```

Mientras dato=póliza
 activa Histopre indexado por
 póliza-u.responsable
 localiza dato-grupo
 Si fin de archivo
 captura dato
 FIN SI
 calcula totales

FIN Mientras

Mientras dato=partida
 activa Histopre indexado por
 partida-u.responsable
 localiza dato-grupo
 Si fin de archivo
 captura dato
 FIN SI
 calcula totales

FIN Mientras

Mientras dato=unidad responsable
 activa Deptos secuencialmente
 localiza dato=unidad responsable
 Si fin de archivo
 captura dato
 FIN SI
 calcula totales

FIN Mientras

Salida: Tipo-Consulta

Proceso 17.0 Calcula-Totales

Entrada: Tipo-Consulta

Pseudocódigo:

Si clave de movimiento='1'
 incrementa asignado
 FIN SI

Si clave de movimiento='2'
 incrementa asignado
 FIN SI

Si clave de movimiento='3'
 incrementa asignado
 FIN SI

Si clave de movimiento='4'
 incrementa compromiso
 FIN SI

Si clave de movimiento='5'
 incrementa ejercido
 FIN SI

Si clave de movimiento='6'
 incrementa ejercido
 FIN SI

Salida: Datos-Consulta

Proceso 18.0 Despliega-Totales

Entrada: Datos-Consulta

Pseudocódigo:

```
Limpia pantalla
presenta marco
presenta encabezado
despliega total asignado
despliega compromiso
despliega ejercido
despliega disponible
```

Salida: Datos-Consulta

! Módulo: Informes !

Proceso 19.0 Presenta-Manu y Obten-Tipo-de-Informe

Entrada: Opción

Pseudocódigo:

```
Limpia pantalla
presenta marco
presenta encabezado
presenta opciones de informes
captura tipo de informes
```

Salida: Tipo-Informe

Proceso 20.0 Accesa-Histopre-por-Llave

Entrada: Tipo-Informe

Pseudocódigo:

```
Si tipo informe='1'
  activa archivo Histopre indexado por
  partida
  Mientras no fin de archivo
    Mientras partida sea igual
      acumula totales
      imprime informe
    FIN Mientras
  limpia totales
  FIN Mientras
FIN SI
Si tipo informe='2'
  activa archivo Histopre indexado por
  unidad responsable
  Mientras no fin de archivo
    Mientras unidad responsable igual
      acumula totales
      imprime informe
    FIN Mientras
  limpia totales
  FIN Mientras
FIN SI
```

```

Si tipo informe='3'
activa archivo Histopre indexado por
partida
Mientras no fin de archivo
    Mientras partida igual
        acumula totales
        imprime informe
    FIN Mientras
    limpia totales
FIN Mientras
FIN SI
Si tipo informe='4'
activa archivo Histopre indexado por
unidad responsable-partida
Mientras no fin de archivo
    Mientras unidad responsable igual
        Mientras partida igual
            acumula totales1
            imprime informe
        FIN Mientras
        acumula totales2
        limpia totales1
    FIN Mientras
    limpia totales2
FIN Mientras
FIN SI
Si tipo informe='5'
activa archivo Histopre indexado por
unidad responsable-grupo
Mientras no fin de archivo
    Mientras grupo igual
        Mientras unidad responsable igual
            acumula totales1
            imprime informe
        FIN Mientras
        acumula totales2
        limpia totales1
    FIN Mientras
    limpia totales2
FIN Mientras
FIN SI

```

Salida: Datos-Informe

Proceso 21.0 Imprime-Informe

Entrada: Datos-Informe

Pseudocódigo:

```

    Imprime encabezados
    imprime registro
    imprime totales

```

Salida: Datos-Informe

! Módulo: Respaldo !

Proceso 22.0 Lee-Disco-Duro

Entrada: Opción

Pseudocódigo:

Lee archivos con extensión .DBF

graba respaldo

Lee archivo con extensión .NTX

graba respaldo

Salida: Datos-Póliza, Códigos-Programáticos, Número-de-Compromiso.

Proceso 23.0 Graba Resplado

Entrada: Datos-Póliza, Códigos-Programáticos, Número-de-Compromiso.

Pseudocódigo:

Mientras no fin de archivo

 copia registro a la unidad A:

 FIN Mientras

Salida: Datos-Póliza, Códigos-Programáticos, Número-de-Compromiso.

! Módulo: Recuperación !

Proceso 24.0 Lee-Respaldo

Entrada: Opción

Pseudocódigo:

Lee de la unida A: archivos con extensión .DBF

graba disco duro

lee de al unidad a: archivos con extensión .NTX

graba disco duro

Salida: Datos-Póliza, Códigos-Programáticos, Número-de-Compromiso.

Proceso 25.0 Graba Disco Duro

Entrada: Datos-Póliza, Códigos-Programáticos, Número-de-Compromiso.

Pseudocódigo:

Mientras no fin de archivo

 COPIA registro a la unidad C:

 FIN Mientras

Salida: Datos-Póliza, Códigos-Programáticos, Número-de-Compromiso.

: Modulo Auxiliar: Valida :

Proceso 26.0 Valida
Entrada: Tipo-Valida, Dato
Pseudocódigo:

```
Si tipo-valida='1'  
  verifica si clave de movimiento se encuentra  
  entre 1-6  
FIN SI  
Si tipo-valida='2'  
  verifica que número de referencia se  
  numérico  
  Si clave de movimiento='4'  
    verifica que número de referencia no este  
    en archivo Compromi  
  FIN SI  
  Si clave de movimiento='5'  
    verifica que número de referencia exista  
    en Compromi  
  FIN SI  
FIN SI  
Si tipo-valida='3'  
  verifica que 0 < mes < 12  
  verifica que 0 < dia < 31  
FIN SI  
Si tipo-valida='4'  
  verifica que póliza sea numérica  
FIN SI  
Si tipo-valida='5'  
  verifica que programa este en archivo  
  Program  
  verifica que subprograma este en  
  archivo Subprog  
  verifica que partida este en archivo  
  Partidas  
  verifica que departamento este en  
  archivo Deptos  
  verifica que código programático no  
  este en archivo Códigos  
FIN SI  
Si tipo-valida='6'  
  verifica que programa este en archivo  
  Program  
  verifica que subprograma este en  
  archivo Subprog  
  verifica que partida este en archivo  
  Partidas  
  verifica que departamento este en  
  archivo Deptos  
  verifica que código programático este  
  en archivo Códigos  
FIN SI
```

```
Si tipo-valida='7'  
  verifica que importe sea numerico  
  Si clave de movimiento='5'  
    verifica que importe no exeda al  
    importe del compromiso, revisando el  
    archivo Compromi  
  FIN SI  
FIN SI  
Si tipo-valida='8'  
  verifica el código programático  
FIN SI
```

Salida: Datos-Póliza, Error

SUBSISTEMA PARA ADAPTACION

```
-----  
; Módulo: Crea-Archivo-Depyver ;  
-----
```

Proceso 27.0

Entrada: Nombre-Dependencia, Versión

Pseudocódigo:

```
Limpia pantalla  
presenta marco  
presenta mensaje 'Versión'  
pide versión  
presenta mensaje 'Dependencia'  
pide nombre de la dependencia  
guarda los valores dados en un archivo  
denominado Depyver
```

Salida: Nombre-Dependencia, Versión

```
-----  
; Módulo: Crea-Archivo-Deptos ;  
-----
```

Proceso 28.0

Entrada: Deptos

Pseudocódigo:

```
Limpia pantalla  
presenta marco  
presenta mensaje 'Departamentos'  
Mientras haya departamento por dar de alta  
  pide clave de departamento  
  pide nombre de departamento  
FIN Mientras
```

Salida: Deptos

```
-----  
; Módulo: Crea-Archivo-Program ;  
-----
```

Proceso 30.0
Entrada: Programs
Pseudocódigo:

```
Limpia pantalla
presenta marco
presenta mensaje 'Programas'
Mientras haya programas por dar de alta
  pide numero de programa
  usa archivo Program
  graba en Program el numero de
  programa
FIN Mientras
```

Salida: Programs

; Módulo: Crea-Archivo-Subprog ;

Proceso 31.0
Entrada: Subprogs
Pseudocódigo:

```
Limpia pantalla
presenta marco
presenta mensaje 'Subprograma'
Mientras haya subprogramas por dar de alta
  pide número de subprograma
  usa archivo Subprog
  graba en Subprog el número de
  subprograma
FIN Mientras
```

Salida: Subprogs

; Módulo: Crea-Archivo-Partidas ;

Proceso 29.0
Entrada: Número-Partida, Nombre-Partida
Pseudocódigo:

```
Limpia pantalla
presenta marco
presenta mensaje 'Partida'
Mientras haya partidas por dar de alta
  pide número de partida
  usa archivo Indpar
  busca en Indpar el numero de
  partida
  guarda el nombre de la partida en
  variable auxiliar
  usa archivo Partidas
  graba en Partidas el numero y
  nombre de la partida
FIN Mientras
```

Salida: Número-Partida, Nombre-Partida

C) DEFINICION DEL SOFTWARE PARA EL DESARROLLO

Una vez que ya se sabe que es lo **QUE** se va a hacer y **COMO** se va a hacer, viene la fase de **CON QUE** se realizará el proyecto. En nuestro caso, es aquí donde se definen las herramientas de software que se utilizarán para el desarrollo del proyecto.

La selección está delimitada por los recursos de hardware, principalmente lo referente al equipo de cómputo. Es decir, el software deberá funcionar en equipos PC compatibles con IBM.

Para la parte del proyecto correspondiente al sistema de control del ejercicio presupuestal se requiere de software con orientación administrativa. Las opciones que tenemos para realizar esta parte son:

1. Lenguajes de alto nivel como Cobol. Este es un buen lenguaje para realizar un sistema administrativo en PC, ya que se pueden realizar manejos de pantalla para la captura y presentación de datos, además es un lenguaje muy estándar, lo cual facilita su mantenimiento y adaptación. Sin embargo, se tiene la desventaja de que los programas fuente tienen una cantidad muy grande de código, lo que influye de manera determinante en el tiempo de desarrollo.

2. Manejadores de Bases de Datos. Concretamente nos referimos al DBase III Plus, ya que es uno de los más utilizados hoy en día, lo cual implicaría modernizar el desarrollo de los nuevos sistemas. Por otro lado es muy sencilla su utilización. El Dbase III Plus es muy buen software para realizar sistemas administrativos, además cuenta con generadores de pantallas y reportes. Por las características que se mencionaron, realizar sistemas en Dbase III Plus es sumamente fácil y rápido. Sin embargo, existe una desventaja y es que Dbase es un intérprete lo cual implica que para ejecutar los programas, se requiere que esté presente todo el sistema de Dbase, esto conlleva a la utilización de gran cantidad de memoria para cada versión.

Otra opción es utilizar CLIPPER, este es todo un sistema completo, el cual incluye un manejador de base de datos compatible con DBase III Plus, es un compilador y un ligador los cuales generan programas ejecutables a partir de código Dbase III Plus, esto elimina la desventaja que se menciona respecto a la realización del sistema en DBase.

Por lo dicho anteriormente, el sistema de control del ejercicio presupuestal se realizará en DBase III Plus y será compilado con Clipper.

Para la realización del subsistema de adaptación, se utilizará el mismo software que para el subsistema de operación.

Para la parte de demostración se requiere de software con posibilidades gráficas. Para este subsistema se utilizará un paquete denominado Storyboard, ya que éste está orientado precisamente hacia las demostraciones.

A continuación se describirá en forma general y breve las características del software que se va a utilizar.

DBase III Plus

Es un sistema manejador de bases de datos relacional que opera en modo de intérprete.

Una base de datos es un conjunto de archivos relacionados.

El sistema manejador nos permite aprovechar de una forma eficiente la información de los archivos.

Lo de sistema relacional se refiere a la teoría en la que se sustenta la operación del sistema. A este respecto cabe mencionar que existen básicamente tres modelos para el diseño de bases de datos. El modelo relacional, el modelo jerárquico y el modelo de red, y otros más como el modelo entidad-relación y el maestro-evento.

Modelo Relacional.

Existe un fundamento matemático para el modelo relacional basándose en el álgebra y cálculo relacional. Sobre este aspecto no entraremos en detalles pero se puede consultar de la bibliografía las referencias (18), (19), (20) o (21).

El enfoque relacional a bancos de datos fue concebido por E.F. Codd y continúa revolucionando desde 1970. Este enfoque visualiza al banco de datos lógico como una tabla bidimensional llamada relación.

Sin involucrarnos con las definiciones formales, en el modelo relacional se tienen los siguientes términos:

Tuple. En un archivo sería uno de los registros o en la tabla sería cada uno de los renglones.

Llave Primaria. Uno de los campos del registro.

Atributos. Los campos del registro que no son llave primaria.

Dominio. Es el conjunto de registros referidos a un solo campo. En la tabla sería una columna.

La manipulación de datos se realiza mediante el SEQUEL, que es un sublenguaje de datos no por procedimientos tal que utiliza las facultades funcionales tanto del álgebra como del cálculo relacional. Es un lenguaje autocontenido, estructurado y con palabras claves para evitar el uso de notación y conceptos matemáticos.

Para que una base de datos pueda tener sus propias características de exhaustiva, sin redundancia y estructurada, se lleva a cabo lo que es la normalización. Dicha normalización consiste en aplicar las siguientes formas:

1a Forma Normal.

Una relación está en primera forma normal si todos sus dominios son simples (no hay campos vectores).

2a Forma Normal.

Una relación está en segunda forma normal si los dominios que no son llaves son funcionalmente dependientes de la llave primaria.

3a Forma Normal.

Una relación está en tercera forma normal si los dominios no llaves son independientes entre sí, pero funcionalmente dependientes de la llave primaria.

Características de DBase III Plus.

Es un sucesor del DBase II, el primer administrador de bases de datos popular para microcomputadora.

Wayen Rattif diseñó el sistema Vulcano basándose en un administrador de grandes computadoras de la Jet Propulsion Laboratory (JPL). Después George Tate comercializó Vulcano como Dbase II y creó la compañía Ashton-Tate para distribuirlo.

Dbase III fue diseñado para obtener un pleno rendimiento de las microcomputadoras de 16 bits, y está escrito en lenguaje de programación C. Fue introducido en 1984 y el Dbase III Plus en 1985, el cual ofrece más facilidades para programación.

El sistema Dbase III Plus, corre en computadoras personales IBM y compatibles. El sistema tiene las siguientes limitaciones:

- 1 billón de registros
- 2 billones de bytes (equivalentes a 100 20-megabytes en disco duro)
- 4000 bytes por registro
- 128 campos por registro
- 254 bytes por campo
- 512 Kbytes por campo memo
- 15 archivos abiertos
- 10 archivos de base de datos abiertos al mismo tiempo
- 7 archivos indexados abiertos
- 1 archivo de formato por archivo de base de datos

Los requerimientos del sistema son:

- corre en PC,XT,AT compatibles con IBM
- funciona con sistemas operativos MS-DOS o PC-DOS versión 2.0 o mayores.
- 256k de memoria para versiones menores a 2.0 o 2.1
- para utilizar otro editor o el comando RUN se requiere al menos 384 KB y de preferencia 512 KB de memoria.

Clipper

La sintaxis y la lógica de Clipper emula al lenguaje de programación del intérprete DBase III.

Los programas de Clipper pueden ser compilados y ligados ofreciendo las siguientes ventajas: mayor velocidad en la ejecución y seguridad del código fuente.

Clipper funciona en PC, XT o AT IBM o compatible, con sistema operativo PC-DOS o MS-DOS versiones 2.0 o mayores. Requiere de 256 KB de memoria RAM.

Para el funcionamiento de Clipper se requiere del siguiente software:

- Clipper de Nantucket, Inc.
- DBase III de Ashton-Tate, Inc.
- MS-DOS de Microsoft, Inc.
- Plink86 de Phoenix Software Associates, Ltd.

Storyboard

Es un sistema que funciona en computadoras personales IBM o compatibles y ayuda a crear figuras y diagramas para preparar presentaciones.

El sistema está formado de 4 partes:

+ Picture Maker. Esta parte nos permite crear y modificar ilustraciones. Se puede escribir texto, dibujar figuras, generar cartas y encimar imágenes entre ilustraciones.

+ Picture Taker. Captura imágenes en pantalla de otros programas de aplicación.

+ Story Editor. Permite organizar ilustraciones de la presentación.

+ Story Teller. Permite presentar historias construidas con story editor.

D) DISEÑO DE ENTRADAS Y SALIDAS

Las entradas y las salidas son dos aspectos indispensables para el funcionamiento de todo sistema, ya que éstos son los que están en contacto con el usuario y éste a su vez es quien da vida al sistema a través del uso o aplicación del mismo.

Se puede decir que la parte medular de los sistemas, es decir, los datos que se procesan y la lógica de dicho proceso, está en función de lo que se quiere obtener del sistema, esto es, el proceso está en función de las salidas.

Por otro lado, los datos que se requieren para las salidas, de alguna manera deben entrar al sistema, por lo que las entradas dependen también de las salidas.

Ya se había mencionado que las entradas y las salidas son los medios por los cuales el usuario hace funcionar o tiene contacto con el sistema, es por esto que éstas deben ser diseñadas de tal modo que faciliten el uso del sistema al usuario.

Para el caso concreto de nuestro sistema, las entradas corresponderán a las pantallas de entrada de datos, y las salidas corresponderán a los informes o reportes y a las pantallas del módulo de consulta.

DISEÑO DE PANTALLAS DE ENTRADA

Las pantallas o entradas de nuestro sistema, se refieren a la forma en que el usuario proporcionará los datos al sistema.

Como una forma de facilitar el uso y comprensión de dicho sistema se ha optado por el manejo de menús.

Las pantallas que se incluyen dentro de este rubro son las pantallas de menús y las pantallas de captura en asignación, actualización y modificación.

Hay en nuestro sistema otros tipos de pantallas que no son entradas, como son las pantallas de consulta.

A continuación se presentarán los formatos de las pantallas de entrada.

Los nombres entre "< >" implica que son variables.

Para la entrada de la opción inicial, en la que se elegirá el proceso que se desea realizar, se tendrán dos pantallas de menú principal:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	OPCIONES DEL SISTEMA
<ol style="list-style-type: none">1 ASIGNACION2 ACTUALIZACION3 CONSULTA4 INFORMES5 CIERRE DE PERIODO	
OPCION:	
+ MAS OPCIONES ? AYUDA 0 FIN SISTEMA	

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	OPCIONES DEL SISTEMA
<ol style="list-style-type: none">1 MODIFICACION2 RESPALDO3 RECUPERACION	
OPCION:	
+ MAS OPCIONES ? AYUDA 0 FIN SISTEMA	

Para la entrada de los datos que se capturarán para el sistema se tendrán las siguientes pantallas:

Pantalla de Asignación

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)		ASIGNACION
{ Poliza }	{ Fecha }	
{Codigo Programatico }		
{ Descripcion }		
{Importe }		
OPCION:		
" FIN " para terminar ? AYUDA		

Pantalla de Actualización

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)		ACTUALIZACION
{ Poliza }	{ Fecha }	
{Codigo Programatico }	Clave de Nov. Num. Referencia	
{ Descripcion }		
{Importe }		
OPCION:		
" FIN " para terminar ? AYUDA		

La entrada para elegir el tipo de consulta será la siguiente:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	CONSULTA
1 POR GRUPO 2 POR POLIZA 3 POR PARTIDA 4 POR TOTALES	
OPCION:	
? AYUDA	0 MENU ANTERIOR

Se tendrá otra entrada para el tipo de consulta, que se presentará dentro de cada opción elegida en la entrada anterior, ésta será:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	CONSULTA (TIPO)
1 POR U. RESPONSABLE 2 GLOBAL	
OPCION:	
? AYUDA	0 MENU ANTERIOR

Para la consulta por grupo y global se tendrá la siguiente pantalla de entrada:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	CONSULTA GRUPO-GLOBAL
GRUPO { }	
? AYUDA	FIN PARA TERMINAR

Para la consulta por grupo y unidad responsable se tendrá la siguiente pantalla de entrada:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	CONSULTA GRUPO-U.RESP.
GRUPO { } U. RESPONSABLE { }	
? AYUDA	FIN PARA TERMINAR

Para la consulta por partida y global se tendrá la siguiente pantalla de entrada:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	CONSULTA PARTIDA-GLOBAL
PARTIDA	
? AYUDA	FIN PARA TERMINAR

Para la consulta por partida y unidad responsable se tendrá la siguiente pantalla de entrada:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	CONSULTA PARTIDA-U.RESP.
PARTIDA U. RESPONSABLE	
? AYUDA	FIN PARA TERMINAR

Para la consulta por póliza y global se tendrá la siguiente pantalla de entrada:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	CONSULTA POLIZA-GLOBAL
POLIZA { }	
? AYUDA	FIN PARA TERMINAR

Para la consulta por póliza y unidad responsable se tendrá la siguiente pantalla de entrada:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	CONSULTA PLZA.-U.RESP.
POLIZA { } U. RESPONSABLE { }	
? AYUDA	FIN PARA TERMINAR

Para la consulta por totales se tendrá la siguiente pantalla de entrada:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	CONSULTA TOTS-U.RESP.
U. RESPONSABLE	
? AYUDA FIN PARA TERMINAR	

La entrada para elegir el tipo de informe será la siguiente:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	INFORMES
1 INFORME GLOBAL POR PARTIDA 2 INFORME GLOBAL POR UNIDAD RESPONSABLE 3 INFORME DETALLADO POR PARTIDA 4 INFORME DETALLADO POR U.RESPONSABLE-PARTIDA 5 INFORME CONCENTRADO POR U.RESPONSABLE-GRUPO	
OPCION:	
? AYUDA 0 REGRESA A MENU PRINCIPAL	

La entrada para elegir el tipo de modificación que se quiera realizar, será la siguiente:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	MODIFICACION
 1 MODIFICA TODA UNA POLIZA 2 MODIFICA TODO UN CODIGO PROGRAMATICO 3 MODIFICA ALGUN CAMPO 	
OPCION:	
? AYUDA	0 REGRESA A MENU PRINCIPAL

Se tendrá otra entrada para el tipo de modificación dentro de cada opción elegida en la entrada anterior, ésta será:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	MODIFICACION (TIPO)
 1 CORRECCION 2 CANCELACION 	
OPCION:	
? AYUDA	0 REGRESA A MENU ANTERIOR

Para el proceso de corrección del módulo de modificación por póliza se tendrá la siguiente pantalla de entrada:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	MOD. POLIZA CORRECCION
Teclee poliza actual []	
Teclee poliza nueva []	
? AYUDA	FIN PARA TERMINAR

Para el proceso de cancelación del módulo de modificación por póliza se tendrá la siguiente pantalla de entrada:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	MOD. POLIZA CANCELACION
Teclee poliza a cancelar []	
? AYUDA	FIN PARA TERMINAR

Para el proceso de corrección del módulo de modificación por código programático se tendrá la siguiente pantalla de entrada:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	MOD. COD. PROG. CORRECCION
Teclee código programático actual []
Teclee código programático nuevo []
? AYUDA	FIN PARA TERMINAR

Para el proceso de cancelación del módulo de modificación por código programático se tendrá la siguiente pantalla de entrada:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	MOD. COD. PROG. CANCELACION
Teclee código programático a cancelar []
? AYUDA	FIN PARA TERMINAR

Para el proceso de corrección del módulo de modificación por campo se tendrá la siguiente pantalla de entrada:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	MOD. POR CAMPO CORRECCION
Codigo Programatico --> []	
Poliza --> []	
Fecha (DiasMes) --> []	
Tipo de movimiento --> []	
? AYUDA	FIN PARA TERMINAR

Para el proceso de cancelación del módulo de modificación por campo se tendrá la siguiente pantalla de entrada:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	MOD. POR CAMPO CANCELACION
Codigo Programatico --> []	
Poliza --> []	
Fecha (DiasMes) --> []	
Tipo de movimiento --> []	
? AYUDA	FIN PARA TERMINAR

En el caso de corrección por campo se tendrá otra pantalla para la entrada de los datos nuevos y en la cual se presentarán también los datos actuales de los registros, dicha pantalla será:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	MOD. POR CAMPO CORRECCION
Poliza { }	Fecha { }
Código Programático { }	Clave de Mov. Num. Referencia { }
D e s c r i p c i o n { }	
I m p o r t e { }	
Es el registro a modificar ? (Si o No) []	
? AYUDA	FIN PARA TERMINAR

La entrada para el módulo de cierre de periodo será:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	CIERRE DE PERIODO
Proporciona el nombre del periodo: { }	
? AYUDA	'FIN' REGRESA A MENU PRINCIPAL

Los módulos de respaldo y recuperación solicitarán una entrada para que se lleve a efecto el proceso correspondiente, las pantallas de entrada para estos casos son las siguientes:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	RESPALDO
<p>COLOCAR DISCO FLEXIBLE EN LA UNIDAD DE FLOPPIES</p> <p>Teclar "RETURN" cuando este listo el disco</p>	

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	RECUPERACION
<p>COLOCAR DISCO FLEXIBLE EN LA UNIDAD DE FLOPPIES</p> <p>Teclar "RETURN" cuando este listo el disco</p>	

DISEÑO DE INFORMES Y PANTALLAS DE SALIDA

Los informes o salidas de nuestro sistema serán la forma en que éste presentará los resultados al usuario y son el objeto principal de dicho sistema.

Los tipos de informes y la distribución de la información dentro de los mismos ya se encuentra determinada por el sistema actual.

Una de las cosas que se pretenden mejorar en nuestro sistema, es lo referente al tiempo en que se obtienen los informes, para esto, hemos determinado que dichos informes se obtendrán a partir de archivos indexados que se actualizan constantemente, en vez de que cada ocasión que se obtenga algún reporte se gaste tiempo en el proceso de la información para ordenarla y dejar una copia del archivo de historia con los datos ordenados.

Otra de las salidas de nuestro sistema, son las pantallas donde se muestran los resultados del módulo de consulta.

En las siguientes páginas, encontraremos los formatos de cada uno de los cinco informes que se obtienen por el sistema.

También se tienen las pantallas de salida correspondientes al proceso de consulta, éstas se podrán observar después de los formatos para los informes.

En el caso de las salidas, se tiene que hacer más incapié en el hecho de que se presentan formatos, ya que la información en éstos casos es más variable que para las entradas, debido a que los datos de salida depende de la versión y del usuario que utilice el sistema. Por esos solo se presentarán los formatos y se harán las siguientes consideraciones:

- * la información entre " < > " es variable
- * el formato de los informes se calculará para formas de 132 columnas por 51 renglones.

1. INFORME GLOBAL POR PARTIDA:

Fecha: <dia>/<mes>/<version>	Página:<#>
UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO <DEPENDENCIA> UNIDAD ADMINISTRATIVA INFORME GLOBAL POR PARTIDA ASIG. TRANS. AMPLI. TOT. COMPROM. EJER. DISP.	
PARTIDAS:	
<Nombre Partida>	
<# Partida>	\$<importe>.....\$<importe>
.	
.	
<Nombre Partida>	
<# Partida>	\$<importe>.....\$<importe>
Total del Grupo	
<# grupo>	\$<importe>.....\$<importe>

2. INFORME GLOBAL POR UNIDAD RESPONSABLE:

Fecha: <dia>/<mes>/<version>	Página:<#>
UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO <DEPENDENCIA> UNIDAD ADMINISTRATIVA INFORME GLOBAL POR UNIDAD RESPONSABLE ASIG. TRANS. AMPLI. TOT. COMPROM. EJER. DISP.	
U.RESPONSABLE	
<Departamento>	\$<importe>.....\$<importe>
.	
.	
<Departamento>	\$<importe>.....\$<importe>
TOTALES	\$<importe>.....\$<importe>
TOTALES DIRECTAS	\$<importe>.....\$<importe>
TOTALES CENTRAL.	\$<importe>.....\$<importe>
TOTALES COMPLEM.	\$<importe>.....\$<importe>

3. INFORME DETALLADO POR PARTIDA:

```
-----  
: Fecha: <dia>/<mes>/<version>                               Pagina:<#> :  
:                                                             :  
: UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO                   :  
: <DEPENDENCIA>                                             :  
: UNIDAD ADMINISTRATIVA                                     :  
:                                                             :  
: INFORME DETALLADO POR PARTIDA                             :  
: PROGRAMAS: <#> PARTIDA: <#> <Nombre de la partida>      :  
: SUBPROGRAMA: <#> DIGITO DE CONTROL: <#>                 :  
: FECHA POL. CONCEP. ASIG. TRANS. AMPLI. TOT. COMPR. EJER. DISP. :  
: <fecha><#><descri> $<importe>.....$<importe>          :  
: .                                                         :  
: .                                                         :  
: <fecha><#><descri> $<importe>.....$<importe>          :  
: -----  
: SUMA POR PARTIDA $<importe>.....$<importe>          :  
-----
```

4. INFORME DETALLADO POR UNIDAD-RESPONSABLE-PARTIDA:

```
-----  
: Fecha: <dia>/<mes>/<version>                               Pagina:<#> :  
:                                                             :  
: UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO                   :  
: <DEPENDENCIA>                                             :  
: UNIDAD ADMINISTRATIVA                                     :  
:                                                             :  
: INFORME DETALLADO POR UNIDAD-RESPONSABLE-PARTIDA         :  
: UNIDAD RESPONSABLE: <# departamento> <nombre departamento> :  
: PROGRAMAS: <#> PARTIDA: <#> <Nombre de la partida>      :  
: SUBPROGRAMA: <#> DIGITO DE CONTROL: <#>                 :  
: FECHA POL. CONCEP. ASIG. TRANS. AMPLI. TOT. COMPR. EJER. DISP. :  
: <fecha><#><descri> $<importe>.....$<importe>          :  
: .                                                         :  
: .                                                         :  
: <fecha><#><descri> $<importe>.....$<importe>          :  
: -----  
: SUMA POR PARTIDA $<importe>.....$<importe>          :  
-----
```

5. INFORME CONCENTRADO POR UNIDAD-RESPONSABLE-GRUPO:

Fecha: <dia>/<mes>/<versión>		Página:<#>				
UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO <DEPENDENCIA> UNIDAD ADMINISTRATIVA						
INFORME CONCENTRADO POR UNIDAD-RESPONSABLE-GRUPO						
UNIDAD RESPONSABLE: <# departamento> <nombre departamento>						
GRUPO: <# grupo>						
PROG.	SUBPROG.	PART.	NOM.DE LA PART.	ASIG.	EJER.	DISP.
<#>	<#>	<#>	<nom.partida>	\$<importe>	\$<importe>
	.					
	.					
<#>	<#>	<#>	<nom.partida>	\$<importe>	\$<importe>
TOTALES				-----		
				\$<importe>	\$<importe>
TOTALES DIRECTAS				\$<importe>		
TOTALES CENTRALIZADAS				\$<importe>		
TOTALES COMPLEMENTARIAS				\$<importe>		

Los formatos para las pantallas de salida del sistema de presupuesto se muestran a continuación:

Para la consulta por grupo y global se tendrá la siguiente pantalla de salida:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	CONSULTA GRUPO-GLOBAL
 <u>TOTALES POR GRUPO (N)</u> ASIGNADO: \$ (IMPORTE) COMPROMETIDO: \$ (IMPORTE) EJERCIDO: \$ (IMPORTE) DISPONIBLE: \$ (IMPORTE) OPCION: CONTINUAR OPCION (S) MENU ANTERIOR (N)	

Para la consulta por grupo y unidad responsable se tendrá la siguiente pantalla de salida:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	CONSULTA GRUPO-U.RESP.
 GRUPO (N) UNIDAD RESPONSABLE (N) (NOMBRE DEL DEPARTAMENTO) ASIGNADO: \$ (IMPORTE) COMPROMETIDO: \$ (IMPORTE) EJERCIDO: \$ (IMPORTE) DISPONIBLE: \$ (IMPORTE) OPCION: CONTINUAR OPCION (S) MENU ANTERIOR (N)	

Para la consulta por póliza y global se tendrá la siguiente pantalla de salida:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	CONSULTA POLIZA-GLOBAL
<u>TOTALES POR POLIZA (N)</u>	
ASIGNADO: \$ (IMPORTE)	
COMPROMETIDO: \$ (IMPORTE)	
EJERCIDO: \$ (IMPORTE)	
DISPONIBLE: \$ (IMPORTE)	
OPCION:	
CONTINUAR OPCION (S) MENU ANTERIOR (N)	

Para la consulta por póliza y unidad responsable se tendrá la siguiente pantalla de salida:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	CONSULTA PLZA-U.RESP.
POLIZA (N)	
UNIDAD RESPONSABLE (N) (NOMBRE DEL DEPARTAMENTO)	
ASIGNADO: \$ (IMPORTE)	
COMPROMETIDO: \$ (IMPORTE)	
EJERCIDO: \$ (IMPORTE)	
DISPONIBLE: \$ (IMPORTE)	
OPCION:	
CONTINUAR OPCION (S) MENU ANTERIOR (N)	

Para la consulta por totales y global se tendrá la siguiente pantalla de salida:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	CONSULTA TOTAL-GLOBAL
<u>TOTAL GLOBAL</u>	
ASIGNADO: \$ <IMPORTE>	
COMPROMETIDO: \$ <IMPORTE>	
EJERCIDO: \$ <IMPORTE>	
DISPONIBLE: \$ <IMPORTE>	
OPCION:	
CONTINUAR OPCION (S) MENU ANTERIOR (N)	

Para la consulta por totales y unidad responsable se tendrá la siguiente pantalla de salida:

SISTEMA DE PRESUPUESTO (NOMBRE DE LA DEPENDENCIA)	CONSULTA TOTAL-U.RESP.
UNIDAD RESPONSABLE (N) <NOMBRE DEL DEPARTAMENTO>	
ASIGNADO: \$ <IMPORTE>	
COMPROMETIDO: \$ <IMPORTE>	
EJERCIDO: \$ <IMPORTE>	
DISPONIBLE: \$ <IMPORTE>	
OPCION:	
CONTINUAR OPCION (S) MENU ANTERIOR (N)	

E) DISEÑO DE ARCHIVOS

Desde que se realizó el análisis se definieron los datos que se utilizan en el sistema y se determinó también cuales deben ser almacenados. En esta parte del diseño se describirá en forma explícita y agrupando en archivos los datos del sistema.

----- Archivo Depyver -----

Este archivo contiene el nombre de la dependencia que va a utilizar el sistema, este parámetro se envía a los encabezados de las pantallas y de los informes. Además el archivo contendrá el número que indica el año de la versión que se utiliza.

Nombre campo	Llave	Tipo	Longitud	Observaciones
versión		numérico	2	año en que se usa el sist.
nomdep		alfanumérico	30	nombre de la dependencia

Organización: Secuencial

----- Archivo Leptos -----

Este archivo contiene el nombre de los departamentos o unidades responsables que se utilizan en la dependencia. El número de la dependencia se captura en forma de clave dentro del código programático.

Nombre campo	Llave	Tipo	Longitud	Observaciones
clavedep	*	numérico	3	clave del departamento
nomdep		alfanumérico	30	nombre del departamento

Organización: Secuencial

Archivo Program

En este archivo se definirán las claves de los programas que utiliza la dependencia y que forman parte del código programático.

Nombre campo	Llave	Tipo	Longitud	Observaciones
programa	*	numé- rico	2	clave del programa

Organización: Secuencial

Archivo Subprog

En este archivo se definirán las claves de los subprogramas que define la dependencia y que forman parte del código programático.

Nombre campo	Llave	Tipo	Longitud	Observaciones
subprog	*	numé- rico	2	clave del subprograma

Organización: Secuencial

 Archivo Partidas

Este archivo contiene el número y nombre de las partidas que utiliza la dependencia.

Nombre campo	Llave	Tipo	Longitud	Observaciones
numpar	*	numérico	3	número de la partida
nompar		alfanumérico	80	nombre de la partida

Organización: Indexado

 Archivo Indpar

Este archivo contiene el número y nombre de todas las partidas que se utilizan en la Universidad Nacional Autónoma de México.

Nombre campo	Llave	Tipo	Longitud	Observaciones
ptd1	*	numérico	3	número de la partida
nombrel		alfanumérico	80	nombre de la partida

Organización: Indexado

 Archivo Códigos

Este archivo contiene los códigos programáticos asignados, es decir, los códigos con los cuales se pueden realizar transacciones.

Nombre campo	Llave	Tipo	Longitud	Observaciones
cprogram	*	numérico	2	
csubprog	*	numérico	2	
cpartida	*	numérico	3	
cdeparto	*	numérico	3	departamento o unidad resp.
cdictrol	*	numérico	2	digito de control

Organización: Indexado

 Archivo Compromi

Compromi, contiene el número de compromiso o de referencia a un compromiso dado de alta. Se encuentra el importe actualizado referente al compromiso.

Nombre campo	Llave	Tipo	Longitud	Observaciones
cnumref		numérico	4	número de referencia o compromiso.
cimporte		numérico	15	
ccodigo		numérico	12	código programático

Organización: Secuencial

 Archivo Histopre

Este archivo contiene la información que se va generando durante la operación del sistema.

Nombre campo	Llave	Tipo	Longitud	Observaciones
hclavmov		alfanumérico	1	clave del movimiento o transacción
hdescrip		alfanumérico	30	texto referente al tipo de transacción
hnumref		numérico	4	número de referencia o compromiso
himporte		numérico	15	
hfecha		numérico	4	fecha de movimiento
hpoliza		numérico	3	
hprogram		numérico	2	
hsubprog		numérico	2	
hpartida		numérico	3	
hdeparto		numérico	3	departamento o unidad resp.
hdictrol		numérico	2	dígito de control

Organización: Secuencial

C A P I T U L O V I I

DESARROLLO DEL SISTEMA DE PRESUPUESTO

En este capítulo encontraremos la parte medular de este trabajo, aquí es donde se llevará a la práctica todo lo que se ha visto desde el estudio hasta el diseño del sistema.

Se comenzará este capítulo con la programación, la cual abarca todo lo que se refiere a introducir en la computadora el código del lenguaje en el que se desarrollará el sistema y los textos que se requieren para el funcionamiento del mismo, concretamente nos referimos a los textos para las ayudas.

También durante la programación se realizan las pruebas individuales de cada proceso que se va terminando.

La integración del sistema se irá realizando conforme se vayan teniendo los módulos ya probados individualmente. Una vez que se tenga el sistema completo se procederá a realizar pruebas más completas cuyas particularidades se definen dentro del inciso de pruebas.

A) PROGRAMACION

Al llegar a esta parte del desarrollo se comienza a tener contacto con el software que se va a utilizar, si es que no se ha tenido antes.

Con los paquetes a utilizar (Dbase III Plus y Story-Board básicamente) instalados y con la documentación respectiva se hacen las pruebas necesarias para conocer el software y sus facilidades.

Una vez que se conocen lo suficiente los paquetes se procede a la codificación de cada uno de los módulos que conforman el sistema y que fueron definidos en las miniespecificaciones. La codificación se hace basándose en el pseudocódigo.

Al momento de programar es cuando se realiza todo lo visto en el análisis y el diseño, solo que al hacerlo prácticamente en el desarrollo se tienen que detallar más las especificaciones que se definieron antes. En algunos casos se tienen que readaptar algunas definiciones para aprovechar cabalmente o hacer un mejor uso del software.

Las consideraciones a hacer en el paso preliminar y necesario para la programación, es decir la codificación, se explican con más detalle en las siguientes líneas.

Más adelante agrupados dentro de lo que denominamos archivos auxiliares se explica en forma general, la manera y las razones de por que se requieren ciertos archivos.

CODIFICACION.

Es conveniente mencionar, que la técnica a ser empleada en cuanto a la programación será estructurada/mnemotécnica/modular, esto es, la programación estructurada resulta ser más versátil y práctica, sea cual fuere el lenguaje de programación que se emplee. El mnemotecnismo se refiere a utilizar nombres de variables que estén relacionadas con su función. La modulación del sistema nos permitirá una detección más rápida de errores, así como también el poder realizar pruebas individuales que nos permitan avanzar con mayor rapidez y con mayor probabilidad de éxito.

A la hora de comenzar la codificación se debe tener un orden para ir realizando los diferentes módulos. el orden que nosotros seguimos y las razones por las cuales seguimos tal secuencia son las que a continuación se mencionan:

Preparación del archivo general de partidas. Esto se refiere a tener un archivo con el número y nombre de todas las partidas universitarias. El sistema actual de control del ejercicio presupuestal requiere de un archivo con organización relativa, el cual se creó a través de uno con organización secuencial que se procesa a partir de un programa en cobol para transformarlo. Por lo anterior para tener el archivo que el nuevo sistema requiere, se codifica un programa en lenguaje Cobol para que cree un archivo secuencial, incertándole el número de la partida, haciendo uso del archivo relativo existente. El archivo secuencial se transforma en una archivo DBase con instrucciones del mismo intérprete.

Codificación del subsistema de adaptación. El objeto principal del subsistema de adaptación es el de crear el ambiente necesario para que el sistema de control del ejercicio presupuestal funcione para una dependencia en particular, es decir el sistema de adaptación debe crear los archivos constantes y que contienen los datos que el sistema de presupuesto requiere los cuales son definidos por las dependencias universitarias. La razón por la que se ha decidido iniciar la codificación de los módulos pertenecientes al subsistema de adaptación es que los archivos que crea dicho subsistema son necesarios para para poder probar cualquiera de los módulos de cualquier otro subsistema, mientras que para probar éste, no requiere de ningún otro. Dentro de la codificación de este subsistema se tienen la presentación del mismo, las pantallas de captura de:

- + versión (año del periodo en que se utilizará el sistema) y nombre de la dependencia.
- + clave y nombre de los departamentos.
- + clave de las partidas, el nombre de dichas partidas se tomará del archivo de partidas que previamente se preparó.
- + clave de los programas.
- + clave de los subprogramas, y despedida del sistema de adaptación.

Codificación de la presentación y estructura general. Esto se refiere a la presentación y estructura del sistema de control del ejercicio presupuestal, para guiarnos en la secuencia a seguir en la realización de los módulos que integran dicho subsistema. La estructura se define en los menús principales.

Codificación de módulos de respaldo y recuperación. Estos se codifican al principio por que pueden ayudar a realizar algunas tareas de respaldo durante el desarrollo de los demás módulos. Por otro lado son sencillos y no requieren de otros procedimientos para su prueba, más que la creación de los archivos a respaldar, lo cual se necesita para los demás módulos.

Codificación de módulos de asignación y actualización. Estos se requieren para introducir datos de prueba, lo cual es importante para la realización de algunas pruebas en algunos otros módulos, además de ser los primeros en el menú principal. Debido a que la validación de los datos que se introducen es una de las principales tareas de la asignación y de la actualización y que además es la misma para estos y otros casos en que se requiere validar, se codifica aparte, como un procedimiento general, un proceso de validación.

Codificación de módulos de consulta, informes, cierre de periodo y modificación. Se menciona estos juntos y en el orden en el que aparecen en el menú principal, por que se pueden realizar simultáneamente, son independientes entre si lo cual facilita las pruebas de los mismos.

Codificación del programa de ayuda. Este es un programa que determina a través del parámetro que recibe el nivel de ayuda que solicita el usuario, y controla el despliegue del texto correspondiente a la explicación requerida. Este programa se hace al final, una vez que se completan todos los módulos operativos del subsistema, ya que no es indispensable para el funcionamiento de dicho subsistema.

Creación del subsistema de demostración. Después de que se conoce lo suficiente el paquete con el que se realiza la demostración, se comienzan a realizar las pantallas donde se expliquen y donde se ejemplifican cada uno de los módulos del subsistema de operación. Se realizan primero las pantallas utilizando el programa "Picture Maker" del paquete mencionado. Después se enlazan todas las pantallas utilizando el programa "Story Editor" de la misma paquetera. Una vez que se tiene la historia que demuestra el sistema de presupuesto ésta puede ser invocada mediante el programa "Story Teller" que forma parte también del paquete de STORYBOARD. El último paso a realizar para la terminación del subsistema de demostración, es la de hacer ejecutable para ser llamado desde el sistema operativo dicho subsistema.

Hablando de los nombres de los módulos podemos decir los siguiente:

+ Los nombres de los módulos que son llamados desde el menú principal del subsistema de control del ejercicio presupuestal, están formados por 6 letras, comienzan con "SP" y las siguientes 4 letras son el inicio del nombre de la tarea que realizan: spalig, spactu, spcons, spinfo, spcipa, spmodi, spresp, sprecu.

+ Debido a las características del lenguaje que se utiliza (dbase III plus) se generan múltiples programas pequeños los cuales agrupamos en un solo archivo para cada módulo a excepción de algunos muy grandes o muy generales. El nombre de dicho archivo comienza con "PROCS" y las tres primeras letras del módulo al que corresponden los programas que agrupan: procsasi, procsact, procscon, procsinf, procsmod.

El código fuente de los programas que conforman el sistema no se incluye en el presente documento, debido a la magnitud de éste, pero se tiene aparte un listado con todos los programas fuente, el cual puede ser consultado como referencia o para el mantenimiento del sistema.

ARCHIVOS AUXILIARES.

Estos son archivos que no fueron descritos explícitamente en el diseño, pero que durante el desarrollo del sistema se hace indispensable tener definidas ciertas características de éstos.

Los archivos auxiliares de este sistema son:

Archivo de ayuda.

La ayuda en el sistema de presupuesto consiste en explicar por la pantalla lo que el usuario solicite, de acuerdo al lugar (menús o pantallas de captura de datos) donde pida la ayuda.

El archivo de ayuda se considera como un archivo auxiliar, ya que para la operación del control del ejercicio presupuestal no es indispensable.

La mayor parte del archivo de ayuda está formado por los textos correspondientes a las explicaciones de los procesos que efectúa el sistema y a las explicaciones de los datos que se capturan.

El archivo de ayuda también se encarga del control de despliegue de los textos, dicho control se lleva a cabo con instrucciones de DBase III Plus.

Archivos Índice.

Durante la programación del sistema de control del ejercicio presupuestal, nos encontramos con diferentes opciones para resolver algún problema de programación, así, tenemos que al programar los módulos de consulta, informes y modificación, los cuales requieren que se tenga acceso a la información en un cierto orden, se nos presentaron dos opciones para tener ordenada la información, una opción era la de ordenar el archivo HISTOPRE por los campos requeridos cada vez que se ejecutaran algunos de los módulos ya mencionados. Otra opción es la de tener varios archivos índice, uno por cada tipo de consulta, informes o modificación, según se requiera. Dichos archivos índice se actualizan al mismo tiempo que se modifica el archivo HISTOPRE.

La opción que elegimos fué la de tener archivos indexados, dado que el acceso a la información es muy rápido, esto trae como consecuencia el evitar hacer ordenamientos de datos cada vez que se requiera acceder la información.

Como podemos observar, en este caso la variable tiempo es favorecida con este tipo de acceso, no así la memoria, dado que para el otro caso no sería necesario que estuvieran todo el tiempo en memoria los archivos con los datos ordenados.

De acuerdo a como se requiere el orden de acceso a los datos se tienen los siguientes archivos indices:

HPARUR1
 HGPOUR1
 HPOLUR1
 HPPSDC1
 HURPPSD1
 HCOPOFT1
 HURPSP1

El archivo de códigos está indexado por todos los campos del mismo, en el mismo orden en que se captura el código programático, es decir, programa, subprograma, partida, dígito de control y departamento. El archivo que contiene los indices se denomina CODIGOS1.

A continuación se presenta la estructura de los diferentes archivos indices para Histopre:

Nombre archivo	Llave de acceso	Módulo o proceso que lo utiliza
HPARUR1	hpartida+hdeparto	Informe global por partida (1) Consulta por partida (1)
HGPOUR1	hgrupo+hdeparto	Consulta por grupo (1)
HPOLUR1	hpóliza+hdeparto	Consulta por póliza (2)
HPPSDC1	hpartida+hprogram+ hsubprog+hdictrol	Informe (3)
HURPPSD1	hdeparto+hpartida+ hprogram+hsubprog+ hdictrol	Informe (4,5,2) Modificación por código (2)
HCOPOFT1	hdeparto+hpartida+ hprogram+hsubprog+ hdictrol+hpóliza+ hfecha+hclavmov	Modificación por campo (3)
HURPSP1	hdeparto+hprogram+ hsubprog+hpartida	Informe detallado por unidad resp.- partida (4)

B) INTEGRACION

La integración de cualquier sistema de cómputo se puede definir como la combinación adecuada de cada uno de los módulos o programas que se han realizado y probado por separado.

Para integrar un sistema de cómputo se tienen varias formas de hacerlo, estas son las siguientes:

Integración Tradicional.

Quando se realiza un tipo de integración tradicional se sigue la siguiente secuencia:

- + Escribir y probar cada uno de los módulos por separado (unit-test).
- + Agrupar módulos en subsistemas y probarlos (subsystem-test).
- + Agrupar los subsistemas para integrar todo el sistema y se prueba.

Integración Incremental.

En este tipo de integración se consideran las técnicas estructuradas, por lo que resulta más acorde con la metodología que se ha seguido para el desarrollo del sistema.

En general, para integraciones incrementales se sigue la siguiente secuencia:

- + Probar un módulo.
- + Adicionar ese módulo a la combinación existente.
- + Probar la nueva combinación.

Dentro del tipo de integración incremental se tienen ciertas variantes las cuales se explican a continuación:

1.- Integración Top-Down Incremental.

En este caso se prueban los módulos superiores del DFD donde los módulos subordinados son simulados por un módulo "stub". Un módulo "stub" no realiza la función del módulo que simula, solo despliega letreros.

Cada módulo "stub" se va reemplazando por módulos reales hasta llegar a la parte inferior del sistema.

2.- Integración Bottom-Up Incremental.

Este tipo de integración es contraria a la de Top-Down Incremental.

3.- Integración Sandwich Incremental.

Este tipo de integración es una mezcla de los dos tipos anteriores.

La forma en que el sistema de control presupuestal se fué integrando fué por el método de Integración Top-Down Incremental.

La integración de nuestro sistema se va dando poco a poco y según se vaya requiriendo. Por el orden en que se fué programando se tiene que un nuevo módulo o proceso que se haya terminado se integra a lo que ya se encuentra hecho, de tal modo que el nuevo proceso integrado puede o no requerir de algunas tareas que estén ya implementadas en el sistema, pero nunca requerirán de algo que todavía no se tenga.

En nuestro caso tenemos diferentes niveles de integración, la integración de cada uno de los módulos de un subsistema y la integración de cada subsistema al sistema completo.

Recordando del análisis, los subsistemas con que se cuenta son: el de adaptación, el de operación (sistema de control del ejercicio presupuestal) y el subsistema de demostración.

Tanto el sistema de adaptación como el de demostración giran en torno al de operación, ya que el primero crea el ambiente necesario para que el de operación valide y muestre los datos correspondientes a una dependencia específica. La demostración es un complemento para la operación y sirve sobre todo a los nuevos usuarios del sistema.

La integración entre la adaptación y la operación se da a través de los archivos: partidas, programas, subprogs, depts y depyver. Adapta crea los archivos y control presupuestal hace uso de ellos.

La demostración es completamente independiente de los otros dos subsistemas ya que lo podemos considerar como complemento de éstos.

Por otro lado, la integración del subsistema de adaptación, es decir, de los módulos que lo conforman, es secuencial, o sea, cada módulo es llamado uno después de otro, sin posibilidad de cambiar el orden. Se pasa de una pantalla de captura a otra y no tiene menús.

En el caso de la integración del subsistema de operación, se puede decir que es paralela, esto es, que cada módulo se va adicionando al sistema general, al que tiene el control, y la integración de un módulo es independiente a la de cualquier otro, ya que se tiene una estructura definida por medio de menús que hace posible este tipo de integración, por lo que se puede elegir cualquier módulo para ser ejecutado sin que se tenga que pasar primero por otro.

Pero por supuesto que no se debe perder la lógica del proceso, ya que aunque se pueden llamar y ejecutar por separado cada proceso, no sería lógico, por ejemplo, modificar datos que no existan en la base de datos, primero se ejecuta un proceso de asignación y si es necesario uno de actualización, para después hacer una consulta o una modificación.

COMPILACION

Una vez que ya se tiene completo el sistema de control del ejercicio presupuestal, se compila. La compilación se considera dentro de la integración por que al final de ésta, se tiene un solo archivo que contiene el código ejecutable y que integra todos los programas del sistema de control del ejercicio presupuestal e inclusive integra también el archivo de ayuda.

La compilación se lleva a cabo con el sistema Clipper. Se compila el programa SISPRE.PRG el cual directa o indirectamente hace referencia a todos los demás programas que forman el sistema. Clipper genera un archivo de código objeto denominado SISPRE.OBJ.

Para obtener el código ejecutable del sistema se hace uso del ligador que utiliza Clipper, éste es Plink86. El ligador toma el archivo generado por el compilador y genera un nuevo archivo que puede ser ejecutado, dicho archivo generado será en nuestro caso SISPRE.EXE.

Antes de que el sistema compilado pueda funcionar, es necesario transformar los archivos índices. Para esto se hace un programa en Clipper que genere dichos archivos índices. Dicho programa se denomina INDI.PRG.

Una vez que se ha compilado y ligado el sistema de presupuesto, éste está listo para ser implementado.

C) PRUEBAS

Podemos entender por prueba al proceso de ejecutar un programa con el fin de encontrar los errores cometidos a lo largo de todos los procesos de desarrollo del sistema. Esto involucra desde el planeamiento de requerimientos, el análisis y finalmente el diseño, esto es a fin de poder corregir conflictos entre lo diseñado y lo que se quiere producir. Las pruebas entonces, tienen como finalidad depurar el sistema.

Dentro de la etapa de pruebas del software se consideran básicamente cuatro pasos:

- 1: Inspecciones y recorridas de los productos generados.
- 2: Diseño de casos de pruebas.
- 3: Ejecución de las pruebas.
- 4: Evaluación de las pruebas.

El objetivo de las inspecciones y recorridas es el detectar errores de omisión, lógica o inconsistencia.

Existen diferentes técnicas que permiten detectar, de algún modo, el mayor número de errores, estas técnicas son las llamadas de caja blanca y de caja negra.

Las pruebas de caja blanca o lógicas, permiten diseñar los casos de prueba de acuerdo a la estructura interna del programa. Los criterios para el diseño de casos de prueba se describen a continuación:

Cobertura de instrucciones:

Este criterio advierte que todas las instrucciones del programa se ejecuten por lo menos una vez, por lo cual resulta ser un criterio muy deficiente e inútil.

Cobertura de decisiones:

Establece que cada secuencia lógica de instrucciones debe ejecutar se una vez si y otra no, esto es, que cada decisión debe tener un falso y un verdadero.

Cobertura de condiciones:

Establece que cada condición en cada decisión tenga todos los resultados posibles, esto es, que si una decisión contiene varias condiciones, esta debe contemplar todos los resultados posibles.

Cobertura de condición múltiple:

Establezca que todas las combinaciones posibles de resultados de condición en cada decisión y todos los puntos de entrada se invoquen por lo menos una vez.

Las pruebas de caja negra, permiten diseñar los casos de prueba en función de las especificaciones del programa, sin tener que consultar la lógica interna.

Existen diferentes tipos y niveles de pruebas, durante la programación se efectúan pruebas individuales a cada módulo. Dichas pruebas individuales se realizan con el objeto de detectar desde errores de sintaxis hasta errores lógicos.

Para esta técnica se emplean diferentes criterios de diseño de casos de pruebas, que a continuación se presentan.

Particiones de equivalencia: Aquí se realiza una partición del conjunto de casos de pruebas que pueden ser abiertos con el mínimo número de casos de pruebas.

Ejecución de las pruebas: Una vez que se han elegido los casos de pruebas a emplear, se procede a ejecutarlas, con la finalidad de detectar errores.

Para la ejecución de los casos de pruebas existen cinco maneras de llevarlas a cabo, a diferentes niveles:

Pruebas modulares: Son aquellas que se ejecutan una vez concluido cada módulo que conforma al sistema, esto permite disminuir los errores en la ejecución de pruebas a un nivel más alto.

Pruebas de integración: Estas pruebas se realizan una vez ejecutadas las pruebas modulares, y también tienen como finalidad localizar errores de codificación.

Pruebas funcionales: Estas pruebas tienen como objetivo encontrar errores en las especificaciones de los requerimientos, esto se hace una vez depurado el sistema de errores de sintaxis.

Pruebas al sistema: Este tipo de pruebas nos permiten identificar las discrepancias entre el sistema final y el que se quería realizar.

Pruebas de aceptación: Tienen como finalidad comparar al sistema final con el planteado en los objetivos.

Para el caso del Sistema de Presupuesto se realizaron varios tipos de pruebas, siguiendo la técnica de caja negra, primeramente se llevaron a cabo las pruebas modulares, para depurar a cada módulo por separado de errores de sintaxis, una vez concluida esta tarea se procedió a realizar las pruebas de integración, en esta etapa se obtuvieron buenos resultados.

Las pruebas funcionales fueron parte muy importante dentro del proceso de pruebas al sistema ya que en esta fase se empleó más tiempo que en las etapas anteriores, en ésta se realizó un ciclo de pruebas ya que por modificar la función de algunos módulos nos obligamos a comenzar con la primera etapa de las pruebas.

Finalmente se realizaron las pruebas al sistema de manera exitosa, al igual que en las pruebas de aceptación, ya que en la mayoría de los casos siempre se marchó sobre los objetivos dispuestos en el diseño que en su oportunidad se detalló, aunque cabe señalar que no se estuvo exento de pequeñas discrepancias entre lo que el usuario quería obtener y lo desarrollado.

Una vez concluidas las diferentes etapas de pruebas, se realizaron un tipo muy especial de pruebas, las de volumen, estas se refieren a la cantidad de información que soporta el sistema como máximo, en particular, el Sistema de Presupuesto soporta gran cantidad de información, que de acuerdo a la estimación del volumen de datos que las dependencias usuarias manejan como máximo, muy difícilmente se podrá llegar al límite.

Cabe señalar que para el efecto de la mayoría de los tipos de pruebas, se emplearon datos reales, esto es, datos empleados por dependencias usuarias, aunque para las pruebas modulares se emplearon datos ficticios.

C A P I T U L O V I I I

DOCUMENTACION, INSTALACION Y MANTENIMIENTO DEL SISTEMA

La documentación del sistema, es parte importante dentro de la terminación del mismo, en este caso la documentación es somera dados los objetivos planteados al respecto. Sin embargo en los primeros párrafos de este capítulo se dan algunas referencias para la documentación a detalle.

Con la instalación del sistema termina lo que propiamente se denomina el desarrollo de un sistema de computación y objetivo de la presente tesis, las etapas posteriores corresponden al mantenimiento de dicho sistema, sin embargo ni la instalación ni el mantenimiento se llevarán a la práctica por el momento, por cuestiones de tiempo, lo que se hace es una planeación de la instalación y capacitación del usuario. Lo referente al mantenimiento no se puede preveer ya que ésto se da durante la producción del sistema, lo cual no se considera dentro del presente trabajo.

Dentro del plan de instalación se mencionan algunas actividades que se tienen que realizar para cumplir con uno de los objetivos planteados al inicio de esta tesis, en el que se pretende facilitar al usuario las tareas que éste tenga que realizar.

En lo referente a la capacitación se planean las sesiones de demostración del sistema y otras más, explicativas si es necesario.

A) DOCUMENTACION

El proceso de documentación, aunque resulta una actividad en ocasiones tediosa, es muy necesaria, ya que en ella se especifican los procesos de operación del sistema. Esto es, existen dos clases de documentos que emplean dicha información aunque en diferentes términos para cada caso.

A continuación presentamos brevemente una explicación de lo que sería cada uno de estos documentos en términos de información.

El manual de usuario, es uno de los documentos que incluyen, desde la especificación del tipo de información que se requiere para hacer operar al sistema (formas de captura), hasta los procesos de instalación y operación de cada módulo que conforma al mismo, este documento es una guía de uso del sistema para el usuario final ya que se refiere básicamente a la utilización de éste como herramienta.

El manual de operación es un documento más complejo, ya que en él se incluye información de tipo operativa, es decir, se menciona y detalla la manera en que está estructurado el Sistema de Presupuesto, los procesos de instalación del mismo y cada una de las actividades de los módulos del sistema que los conlleva a realizar su función específica. Este manual va dirigido a usuarios expertos en computación y conocedores de las herramientas empleadas en el desarrollo del sistema.

El manual de usuario es más sencillo que el manual de operación, ya que aunque en ambos se proporciona información sobre un mismo producto, dicha información está enfocada a diferentes personas.

Ambos documentos no son incluidos en este capítulo ya que por su contenido, haría al presente trabajo demasiado redundante, es por ello que se cuenta con un 'Manual de Usuario' y un 'Manual de Operación' como documentos separados e independientes de la presente tesis.

Dichos manuales son proporcionados gratuitamente a los usuarios que los soliciten en el departamento de Asesoría e Implementación de Sistemas de la Dirección General de Servicios de Cómputo para la Administración.

B) PLAN DE INSTALACION

La planeación de la instalación del sistema de control del ejercicio presupuestal en dependencias de la UNAM, se hace tomando en cuenta que en la mayoría de los casos será el propio usuario el que se encargue de dicha instalación, es por esto que la misma debe ser lo más sencilla posible.

Para instalar el sistema se requiere de lo siguiente:

- + Tener lista una versión preparada con los datos que proporcione el usuario. Para esto se hace uso del subsistema de Adapta con que cuenta el sistema.
- + Crear en el disco duro del equipo del usuario una estructura de directorios y subdirectorios, que sea la adecuada para la mejor distribución y manejo de la información que mantiene el usuario en su equipo. Además la estructura debe ser acorde a la forma en como el sistema hace uso de los archivos y programas.
- + Hacer una copia del código ejecutable del sistema, en el disco duro, en el directorio y subdirectorio correspondiente.
- + Copiar los archivos donde se encuentra la información constante de la dependencia, información tal como el nombre de dicha dependencia, el año o la versión del periodo en que se utilizará el sistema de control presupuestal, las partidas que usa la dependencia, así como los programas y subprogramas que define.
- + También como parte de la instalación se debe copiar en disco duro, las estructuras de los archivos que contendrán los datos del sistema durante la producción del mismo, éstos deberán localizarse en el directorio y subdirectorio correspondiente para su utilización.
- + Otros archivos que se deben tener en disco duro, son los referentes al subsistema de demostración, para esto se debe proporcionar al usuario un disco flexible que contenga el subsistema y copiarlo.

Para que el usuario pueda instalar el Sistema de Presupuesto, se le hará entrega de dos discos flexibles, los cuales contendrán el subsistema de operación, solo el archivo ejecutable, todos los archivos de datos que se utilizan, el subsistema de demostración y los archivos auxiliares para la instalación.

Como se puede observar, el proceso de instalación consta básicamente de copiar de discos flexibles donde se encuentra la versión preparada del usuario, a disco duro donde se operará el sistema. Por ello, para que el usuario pueda instalar su sistema requerirá del conocimiento básico de algunos comandos del sistema operativo MSDOS, ya que bajo este opera el sistema. Los comandos a los que nos referimos se enfocan a los siguiente:

- uso, creación y manejo de directorios y subdirectorios, y
- comandos para copiar.

El conocimiento anterior se le puede dar al usuario dentro de la capacitación del mismo, sin embargo si dicho usuario es una persona que nunca ha tenido contacto con un equipo de computo, lo cuál se ha observado que es muy frecuente, hará complicada la capacitación. Por lo anterior se puede facilitar la instalación y con ello dicha capacitación si se hace un proceso de instalación que realice la misma con la mínima intervención del usuario.

El proceso de instalación se realiza aprovechando algunas de las ventajas que da el sistema operativo para crear archivos que MSDOS los vea como ejecutables y que contengan comandos del mismo sistema operativo. Dichos archivos se denominan "archivos con extensión .BAT".

Cada una de las instrucciones del archivo que en nuestro caso se denomina INSTALA.BAT, hace los procesos necesarios ya mencionados para la instalación del sistema, es decir: crear directorios y subdirectorios, copiar archivos de datos constantes, copiar las estructuras de los archivos para los datos de producción, copiar el código ejecutable, y copiar lo referente al subsistema de demostración.

Teniendo un archivo de instalación con las características ya mencionadas, éste debe encontrarse y proporcionarse al usuario en el mismo disco flexible donde se encuentran los datos y estructuras a copiar. La idea es que el usuario con solo encender el equipo y dar las siguientes instrucciones tenga instalado el sistema de control del ejercicio presupuestal:

- C) a:
- a) instala

A continuación se presenta el contenido del archivo INSTALA.BAT.

```
rem instalacion del sp
md sp
cd sp
md archivos
md datos
md progs
md demos
copy \archivos\*. * c:\sp\archivos\*. *
copy \datos\*. * c:\sp\datos\*. *
copy \progs\*. * c:\sp\progs\*. *
copy \demos\*. * c:\sp\demos\*. *
cd\
copy sis-pre.bat c:
rem "SISTEMA INSTALADO"
```

Una facilidad que se le puede brindar al usuario es que pueda operar el sistema de presupuesto desde el directorio raíz, para esto se puede tener el código ejecutable en la raíz, pero no es lo mas conveniente, ya que esta situación se prestaría a cometer posibles errores.

Se tiene otra opción, que es la de crear otro archivo con extensión .BAT y que se ejecute desde la raíz. Así el usuario no tendría que hacer cambios de directorios ni tendría necesidad de conocer los comandos del sistema operativo que para tal fin se tienen.

El contenido de dicho archivo que denominaremos SIS-PRE.BAT, es el siguiente:

```
cd sp
sispre
cd\
cls
```

C) CAPACITACION

La Capacitación consistirá en preparar al usuario en la utilización del sistema.

Una vez terminado el desarrollo, el sistema de control del ejercicio presupuestal, quedará a cargo del Departamento de Asesoría e Implanteación de Sistemas de la Dirección General de Servicios de Computo para la Administración, esto quiere decir que cualquier dependencia universitaria que desee hacer uso del sistema, debe acudir a dicho departamento, el cual se encargará de la preparación de una versión del sistema adecuada a la dependencia que la solicita.

Respecto a la capacitación del usuario, ésta estará a cargo del Departamento ya mencionado, ya que éste ha realizado dicha función en innumerables ocasiones con los sistemas que integran el paquete administrativo, es por esto que el plan de capacitación se base en la experiencia que ya se tiene.

Los usuarios del sistema de presupuesto son los mismos encargados del control manual del ejercicio presupuestal o alguno de sus colaboradores.

La primera etapa de la capacitación se presenta desde el momento en que se les da una demostración a los usuarios en una primera entrevista, la cual se programa cuando el usuario solicita el sistema por primera vez.

La demostración en el caso del nuevo sistema de control del ejercicio presupuestal, varía en que desde ahora se cuenta con un sistema de demostración. La demostración de un sistema cuando no se cuenta con un subsistema de demostración se realiza directamente sobre una versión adaptada para tal función. Una persona que conoce el sistema se encarga de mostrar al nuevo usuario las características del sistema que requiere utilizar. La ventaja de tener un subsistema de demostración es que facilita el trabajo de las explicaciones, ya que el subsistema las proporciona, evitando así que al demostrador se le pase algún detalle irrelevante. Además el subsistema de demostración contiene ejemplos por lo que ahorra el trabajo de inventarlos a la hora de la demostración.

Por otro lado se le puede proporcionar al usuario el subsistema de demostración, para que el mismo pueda usarlo cuando lo requiera o para que capacite a alguna otra persona que vaya a utilizar el sistema.

En concreto, la capacitación en la primera etapa, consiste en una demostración del sistema al usuario por parte del personal del Departamento de Asesoría quien le resolverá sus dudas con ayuda del subsistema de demostración.

Después de la primera demostración, se pasa a la etapa de adaptación e instalación para enseguida continuar con la capacitación.

Para fines de capacitación, la instalación puede sufrir alguna variante la cual consistirá básicamente en el equipo a utilizar, es decir, si el usuario solicita manejar su sistema en equipo del Departamento de Asesoría para capacitarse mejor, y si hay disponibilidad de hacerlo, la instalación se realiza en un equipo del Departamento.

El usuario puede asistir al Departamento a utilizar la versión adaptada de su sistema con datos verídicos o datos de prueba si lo prefiere, mientras contará con la asesoría del personal que labora en el Departamento hasta que se haya completado la capacitación del usuario.

Cuando el usuario utiliza el sistema en su propio equipo, es decir, cuando ya se encuentra capacitado para usarlo por su cuenta, aún así cuenta con el apoyo del Departamento de Asesoría e Implantación de Sistemas para cualquier duda o problema que se le presente al usuario.

CONCLUSIONES

Con el desarrollo de la presente tesis hemos podido llevar a la práctica, la teoría que se había visto a lo largo de los estudios realizados.

Por otro lado se cumplió con el principal objetivo de la tesis, que fué el de desarrollar un sistema en microcomputadora para el Control del Ejercicio Presupuestal.

Ahora se cuenta con un sistema de presupuesto muy accesible para el usuario ya que posee una serie de ventajas, que en contraposición al sistema con el que se contaba anteriormente, resulta ser muy eficiente, dichas ventajas son, por ejemplo, el poseer ayuda inmediata e interactiva, lo cual lo hace aún más amigable al usuario final, así mismo, la operación de los diferentes módulos garantiza la consistencia de los datos, parte esencial dentro de cualquier organización. Por otro lado, el subsistema de demostración facilita la presentación del subsistema de operación a los nuevos usuarios, actividad que anteriormente resultaba ser un tanto complicada.

En términos de operación del sistema, asegura un acceso y modificación de la información más rápidos dentro de los módulos de consulta, cierre de periodo, modificación, informes, respaldo y recuperación. Las altas de los datos también son instantáneas en el caso de asignación y actualización.

Adicionalmente a todas las ventajas mencionadas, el usuario cuenta también con un documento en el que se le proporciona la información necesaria para la operación del sistema. Dicha información se incluye en el "Manual del Usuario".

El nuevo sistema de presupuesto presenta la ventaja de que se puede adaptar fácilmente a cualquier dependencia universitaria que quiera hacer uso de él. Esto gracias a que se cuenta con un subsistema de adaptación con el que resulta muy sencillo preparar diferentes versiones del sistema.

Otro de los objetivos que se cumplió, fué el que se haya desarrollado un sistema actualizado respecto a las herramientas de las que se hizo uso. Nos referimos a los nuevos lenguajes de computadora (lenguajes de 4a. generación), en equipos que hoy en día son de gran utilidad (computadoras PC compatibles con IBM).

Al cumplir con este objetivo se logró también facilitar el trabajo del mantenimiento del sistema, ya que ahora se encuentra todo el sistema en un mismo lenguaje.

Otra de las cosas que facilita el mantenimiento del sistema, es que se cuenta con un documento en el que se tiene la información de lo que hace el sistema y cómo lo hace. Dicha información se encuentra en el "Manual de Operación".

Al realizar la presente tesis, conocimos en forma general alguna de las técnicas administrativas modernas para el control presupuestal, nos referimos a la técnica de Presupuesto por Programas.

El sistema que se desarrollo se basa en la técnica mencionada y por la generalidad de la misma, el sistema también es muy general, por lo que se puede utilizar en cualquier dependencia que emplee dicha técnica.

La metodología que se utilizó para llevar a cabo el desarrollo del sistema, se basó principalmente en las técnicas estructuradas, las cuales fueron descritas en su oportunidad. Al usarlas, pudimos apreciar las ventajas y desventajas que ofrece el realizar un sistema siguiendo una determinada metodología.

Entre las ventajas que pudimos observar, es que la metodología sirve como una guía, la cual nos ayuda a tener un mejor control de las actividades que se tienen que realizar, y así, al seguir un plan ya establecido en el que previamente se estudiaron los pasos más adecuados a seguir, también se observan ventajas respecto al tiempo de desarrollo de un sistema.

La desventaja que pudimos apreciar, no es una desventaja de la metodología en sí, es solo que para nuestro caso en particular el aplicar estrictamente las técnicas estructuradas no nos resultó tan eficaz como hubiera sido en otros casos, en los que se hubiera aprovechado al máximo las características de las técnicas mencionadas. básicamente nos referimos a las opciones gráficas que nos brinda la técnica.

En la mayoría de los casos, el principal problema respecto al desarrollo de sistemas de cómputo es la comunicación con el usuario. Dicho problema se ve notablemente disminuido al hacer uso de las herramientas del análisis estructurado, como son el Diagrama de Flujo de Datos (DFD). Sin embargo en nuestro caso no había comunicación directa con el usuario final, por lo que el DFD no cumplió su principal cometido, ya que las personas encargadas de definirnos las características del sistema a desarrollar, fueron personas conocedoras de la computación, es por ello que se facilitó mucho la comunicación.

Así en nuestro caso concreto, el DFD fué como un complemento que más sirvió para la documentación del trabajo que para el análisis del sistema, y apesar de que se cuentan con herramientas gráficas bastante buenas, el elaborar el DFD no dejó de ser una tarea tediosa. Lo mismo podemos decir acerca del diseño estructurado, concretamente con la realización de la Carta Estructurada.

Al terminar el desarrollo del sistema pudimos comprobar las ventajas que nos ofrecen los lenguajes de 4a. generación, tales como DBase y Clipper, ya que por un lado se desarrollan sistemas en un periodo de tiempo relativamente corto y los lenguajes son muy fáciles de aprender. Por otra parte los sistemas obtenidos son muy rápidos y amigables.

En general con la realización de la presente tesis se pudo hacer uso de diversos paquetes para microcomputadora, como son los procesadores de textos, paquetes de graficación, sistemas administradores de bases de datos, sistemas para realizar presentaciones, etc. Conocimos sus principales características y apreciamos sus ventajas y desventajas. Aprendimos a utilizar las más modernas herramientas para desarrollar sistemas acordes a la era tecnológica en la que vivimos.

Finalmente, podemos concluir, que el Sistema de Presupuesto desarrollado e implementado, se encuentra sujetos a mejoras ya que el avance en materia de software es apresurado, es por ello que las herramientas para desarrollar sistemas nos permiten siempre mejorar los sistemas actuales, sea cual fuere su aplicación.

B I B L I O G R A F I A

- 1) INTRODUCCION A LA TECNICA DEL PRESUPUESTO POR PROGRAMAS.
Ferreiro, Luis Alberto Santana
Universidad Nacional Autónoma de México
1980.
- 2) PRESUPUESTO CON ORIENTACION PROGRAMATICA LINEAMIENTOS GENERALES
1976.
- 3) CATALOGO PRESUPUESTAL E INSTRUCTIVO DEL EJERCICIO 1988
UNAM 1988.
- 4) ANTEPROYECTO DE PRESUPUESTO 1985 INSTRUCTIVO
UNAM 1985.
- 5) CATALOGO PRESUPUESTAL E INSTRUCTIVO DEL EJERCICIO 1985
UNAM 1985.
- 6) CATALOGO PRESUPUESTAL E INSTRUCTIVO DEL EJERCICIO 1981
UNAM 1981.
- 7) SISTEMA DE PRESUPUESTO.
MANUAL DE IMPLEMENTACION Y OPERACION EN UN COMPUTADOR CENTRAL BURROUGHS.
Dirección General de Servicios de Cómputo para la Administración.
UNAM 1985.
- 8) SISTEMA DE PRESUPUESTO.
MANUAL DE IMPLEMENTACION Y OPERACION EN UN MICROCOMPUTADOR.
Dirección General de Servicios de Cómputo para la Administración.
UNAM 1985.

- 9) APUNTES DEL CURSO:
USO DE LA COMPUTADORA EN LA ADMINISTRACION UNIVERSITARIA.
Dirección General de Servicios de Cómputo para la
Administración.
UNAM 1986.
- 10) CONTABILIDAD. TERCER CURSO. METODO AUTODIDACTICO.
C.P. Antonio Méndez V. y Javier Méndez V.
- 11) TESIS: DISEÑO E IMPLEMENTACION DE UN SISTEMA DE
BANCO DE HORAS VIA MICROCOMPUTADORA.
Autor: Juan Carlos Reza García.
Director: Ing. Sebastian Poblano Ordoñez.
1986.
- 12) TESIS: DESARROLLO TEORICO Y PRACTICO DE UN
SISTEMA DE INFORMACION PARA LA
ADMINISTRACION CENTRAL DE LA U.N.A.M.
Autor: José Luis Olmedo Talavera.
Director: Ing. Sebastian Poblano Ordoñez.
1985.
- 13) TESIS: DESARROLLO DE UN SISTEMA DE CONTABILIDAD
PARA PC USANDO DBASE III.
Autor: Ingrid Morgan Sotomayor.
Director: Ing. Sebastian Poblano Ordoñez.
1987.
- 14) MANAGING THE STRUCTURES TECHNIQUES STRATEGIES FOR
SOFTWARE DEVELOPMENTS IN THE 1990'S.
Edward Yourdon.
Yourdon Press. 1988.
- 15) SOFTWARE ENGINEERING
Design, Reliability, Management
Martin L. Shooman
McGraw-Hill Book Company
1983.
- 16) MANUAL DE LOS SISTEMAS DE INFORMACION
W. Hartman, H. Matthes, A. Proeme
Parainfo S.A.
Madrid 1985.

- 17) SOFTWARE ENGINEERING: A PARCTITIONER'S APPROACH
Roger S. Pressman
McGraw Hill
1982.
- 18) ADVENCED DATABASE TECHNIQUES
Daniel Martin.
Editorial: The MIT Press
London, England 1986.
- 19) DATABASE DESIGN
Gio Wiederhold
McGraw-Hill
1983.
- 20) PRINCIPLES OR DATA-BASE MANAGEMENT
James Martin
Printice-Hall
- 21) SISTEMA DE ADMINISTRADOR DE BANCOS DE DATOS
Alfonso F. Cardenas
Limusa
1983.
- 22) SISTEMAS DE ADMINISTRACION DE BASES DE DATOS
David Kruglinski
AOsborne/McGraw Hill
1984.
- 23) APLIQUE EL DBASE III
Edward Jones
McGraw-Hill
1986.
- 24) DBASE III PLUS APPLICATIONS LIBRARY
Thomas W. Carlton y Charles O. Stewart III.
QUE orporation
1987.
- 25) CLIPPER COMPILER USER MANUAL
Nantucket USA