

REPOSITORIO ACADÉMICO DIGITAL INSTITUCIONAL

**Control de gasto presupuestario para la dirección de presupuestos del
Gobierno del Estado**

Autor: Jorge Tavera Ochoa

**Tesis presentada para obtener el título de:
Ing. en Sistemas Computarizados [sic]**

**Nombre del asesor:
Héctor Guerrero Guadarrama**

Este documento está disponible para su consulta en el Repositorio Académico Digital Institucional de la Universidad Vasco de Quiroga, cuyo objetivo es integrar, organizar, almacenar, preservar y difundir en formato digital la producción intelectual resultante de la actividad académica, científica e investigadora de los diferentes campus de la universidad, para beneficio de la comunidad universitaria.

Esta iniciativa está a cargo del Centro de Información y Documentación "Dr. Silvio Zavala" que lleva adelante las tareas de gestión y coordinación para la concreción de los objetivos planteados.

Esta Tesis se publica bajo licencia Creative Commons de tipo "Reconocimiento-NoComercial-SinObraDerivada", se permite su consulta siempre y cuando se mantenga el reconocimiento de sus autores, no se haga uso comercial de las obras derivadas.





UNIVERSIDAD VASCO DE QUIROGA

ESCUELAS DE SISTEMAS COMPUTARIZADOS

CONTROL DE GASTO PRESUPUESTARIO
PARA LA DIRECCION DE PRESUPUESTOS
DEL GOBIERNO DEL ESTADO.

TESIS

PARA OBTENER EL TITULO DE:

LICENCIADO EN SISTEMAS COMPUTARIZADOS

PRESENTA

JORGE TAVERA OCHOA

ASESOR DE TESIS .
ING. HECTOR GUERRERO GUADARRAMA

MORELIA MICH.

SEPTIEMBRE 1997



7
AVALA

T15



AGRADECIMIENTOS

UNIVERSIDAD
VASCO DE QUIROGA

ESCUELAS DE SISTEMAS COMPUTARIZADOS

CONTROL DE GASTO PRESUPUESTARIO
PARA LA DIRECCION DE PRESUPUESTOS
DEL GOBIERNO DEL ESTADO.

TESIS

PARA OBTENER EL TITULO DE:

LICENCIADO EN SISTEMAS COMPUTARIZADOS

PRESENTA

JORGE TAVERA OCHOA



MORELIA MICH.

SEPTIEMBRE 1997



AGRADECIMIENTOS

A Dios.	1
Que siempre ha iluminado mi camino y mi vida con amor, felicidad y sabiduría, que siempre en los momentos difíciles me ha reconfortado y apoyado.	2
Objetivo General	3
A mis Padres.	5
Jorge y Alicia, que han sabido inculcarme el hábito del estudio y la superación, y que, con su amor y confianza hicieron posible el presente trabajo.	6
A mis hermanos.	7
Varinka, Mauricio y Rodrigo que han estado siempre apoyándome en mis objetivos y metas.	7
A mis tíos.	11
Bricio, René, Marco, José Manuel, y sus respectivas esposas y mi tía Doris. Gracias por el apoyo y amor brindado en toda mi vida.	11
A mis amigos.	35
Alejandro, Agustín, Isidro, Esteban, Gerardo, Juan, Fernando, Sandra y Gauri que siempre me han brindado su apoyo, tiempo, dedicación y su amistad. A Elsa que ha sabido impulsarme para alcanzar mis metas y objetivos, y que con su cariño me ha llenado de felicidad.	35
A mi asesor.	41
Ing. Héctor Guerrero Guadarrama, que como tal es excelente y que además encontré en él a un amigo de corazón que supo guiarme en la realización del presente trabajo.	41
A mis compañeros de generación.	43
José, José de Jesús, Richard, José Manuel, René, Juan Carlos, César(†), Carlos Antonio, Miguel, Hugo y Sandra, gracias por su apoyo y amistad.	43
Oficio de Modificación	47
Documento Múltiple	48
Salidas de Almacén	60
Requisición de Bienes	62
Diseño Lógico	54
Elementos del Diseño Lógico	56
Diseño de Salidas	58

INDICE

Diseño Físico.....	89
Herramientas para el Diseño.....	89
Herramientas de la Estrategia de Datos.....	90
Introducción.....	1
Antecedentes.....	2
Objetivo General.....	3
Objetivo Específicos.....	5
Metas.....	6
Hipótesis.....	7
Planeación del Sistema.....	8
Análisis del Problema.....	11
Entrevistas.....	11
Análisis de los Documentos para Realizar Erogaciones.....	32
Diseño de la Solución.....	35
Objetivos del Diseño.....	35
Estructura del Presupuesto de Egresos.....	35
Diseño de la Base de Datos.....	40
Diccionario de Datos.....	112
Catálogo General.....	40
Catálogo de Partidas.....	41
Distribución.....	43
Movimientos.....	44
Documento de Ejecución.....	45
Oficio de Modificación.....	47
Documento Múltiple.....	48
Salidas de Almacén.....	50
Requisición de Bienes.....	52
Diseño Lógico.....	54
Herramientas del Diseño Lógico.....	56
Diseño de Salidas.....	58

Diseño Físico	89
Herramientas para el Desarrollo de Sistemas	89
Herramientas de la Estrategia de Datos	90
Sugerencias en la Forma de Codificar	91
Diseño Físico del Sistema	92
Conectividad	100
Prueba y Depuración	104
El Aseguramiento de la Calidad	104
Estrategias de Prueba	105
Prácticas de Prueba	105
Diseño de Datos de Prueba	106
Implantación y Evaluación	107
Capacitación del Personal	107
Método de Conversión	108
Evaluación	109
Conclusiones y Recomendaciones	110
Consideraciones	111
Bibliografía	112
Manual del Usuario	113
Menú Principal	113
Presupuesto	114
Erogaciones	117
Catálogos	118
Reportes	119
Utilerías	120
Ayuda	122
Código Fuente	

Es por lo tanto necesario que se lleve una sistematización del presupuesto para no quedar rezagados en el rápido avance tecnológico que se vive actualmente.

INTRODUCCION

La necesidad de prever situaciones futuras ha nacido siglos atrás, así tenemos que lo El objetivo de este trabajo es sistematizar el proceso de distribución y ejecución del gasto presupuestal para la Dirección de Presupuesto del Gobierno del Estado.

Actualmente las computadoras han adquirido gran auge en todos los terrenos de la vida humana, podemos encontrar computadoras en el hogar y en de grandes Instituciones Gubernamentales o Privadas. La computación ha facilitado las actividades cotidianas y de trabajo del ser humano, los procesos se han tornado más sencillos y la información se ha vuelto una pieza fundamental para el desarrollo de nuevas aplicaciones.

En la actualidad cada organismo o unidad gubernamental en México, cuenta con formas El hecho de presupuestar para organismo públicos implica la selección de fines y también la de los medios para alcanzar dichos fines. Los sistemas presupuestarios públicos cuentan con canales complejos de intercambio de información, la cual se procesa sobre lo que se otorga y con valoraciones de lo que se está consiguiendo.

En la actualidad el sistema presupuestario del Gobierno del Estado, que está controlado por la Dirección de Presupuesto, lleva a cabo todas las erogaciones que se realizarán durante todo el año, esto es desde el pago de la nómina a sus empleados hasta la realización de obras públicas en beneficio de la sociedad.

La Dirección de Presupuesto se encarga de llevar un control estricto y apegado de cada gasto que realiza, con el objeto de saber en qué, cuánto, cómo, cuándo, dónde y por qué se está gastando.

Es por lo tanto necesario que se lleve una sistematización del presupuesto para no quedar rezagados en el rápido avance tecnológico que se vive actualmente.

ANTECEDENTES

La necesidad de prever situaciones futuras ha nacido siglos atrás, así tenemos que los propietarios de negocios sólo buscaban en la contabilidad, información acerca de cuánto les debían sus deudores, cuánto debían a sus acreedores, etc.

En 1912 el Gobierno de los Estados Unidos viendo la importancia de los presupuestos en la esfera gubernamental, nombró una comisión encargada de estudiar todo lo relativo a estas cuestiones y de esta forma tenemos que en 1921 se aprobó la Ley del Presupuesto Nacional o que es lo mismo el Gasto Público. Esta Ley pasó de esta forma a ser un instrumento de la administración oficial, de ese país.

Dicho sistema está formado por los siguientes elementos de entrada: Partidas de Gasto. En la actualidad cada organismo o unidad gubernamental en México, cuenta con formas presupuestales, como lo es la Dirección de Presupuesto del Gobierno del Estado, que se encarga de planear, preparar, controlar, informar y utilizar el presupuesto que tiene asignado durante todo el año, esto es con la finalidad de verificar que cada peso que se gaste, esté realmente siendo utilizado en lo que se planeó gastar.

Después de siete años se ha sentido la necesidad de sistematizar el presupuesto que tiene preparado la Dirección con el objeto de tener la información presupuestal verificada mes por mes y un resumen anual.

OBJETIVO GENERAL

Diseñar, construir e implantar un Sistema para el Control del Gasto Presupuestal que permita controlar, verificar y mantener actualizada la información sobre el gasto presupuestado que se piensa realizar.

El sistema de gasto Presupuestal sirve para calendarizar, en forma mensual, el gasto presupuestado durante todo un año. Esto tiene como finalidad distribuir y gastar adecuadamente los recursos monetarios que han sido canalizados, por parte del Gobierno del Estado, a la Dirección de Presupuesto.

Dicho sistema está formado por los siguientes elementos de entrada: Partidas de Gasto, Documento de Ejecución Presupuestaria, Oficio de Modificación Presupuestaria y Documento Múltiple. La función del primero es almacenar, en forma numérica y mensual, el monto monetario inicial de las cuentas en que se piensa gastar, como por ejemplo sueldos, prima vacacional, mobiliario, equipo médico etc., el segundo se encarga de proporcionar el dinero y de dar el visto bueno para que se realice el gasto, el tercero modifica el monto monetario inicial, ya sea, para aumentar o disminuir dicho monto y el cuarto realiza ajustes monetarios a las cuentas.

SISTEMA PARA EL CONTROL DEL GASTO PRESUPUESTAL

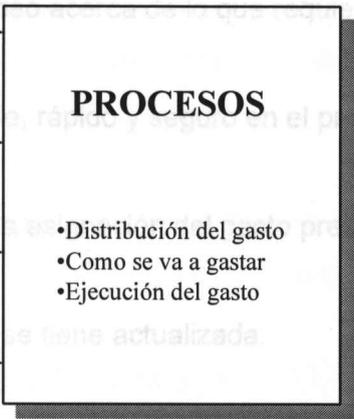
Entradas

Partidas deGasto

Documento de
Ejecución Presupuestaria

Oficio de Modificación
Presupuestaria

Documento Múltiple



Salidas

•Visualización por Pantalla
•Reportes

OBJETIVOS ESPECIFICOS

- ✕ Elaborar un análisis minucioso acerca de lo que requiere el sistema.
- ✕ Diseñar un sistema confiable, rápido y seguro en el proceso del gasto presupuestal.
- ✕ Agilizar la presentación en la asignación del gasto presupuestal.
- ✕ Facilitar la información que se tiene actualizada.
 - ✕ Proporcionar el análisis y diseño de la sistematización del gasto presupuestal en un lapso no mayor a 14 semanas.
 - ✕ Establecer la implantación y prueba del sistema en un periodo no mayor a 3 semanas.
 - ✕ Liberar y documentar totalmente el sistema 4 semanas después de su implantación.
 - ✕ Se harán recomendaciones para llegar a un nivel más allá del sistema.

METAS

El sistema Presupuestario de la Dirección de Presupuesto comprende varios subsistemas tales como : el sistema de ingreso, el sistema de egreso, el sistema de contabilidad y el sistema de gasto presupuestal.

El propósito de este proyecto se encuentra enmarcado dentro del objetivo general y por lo tanto se enfocará a la realización de las siguientes metas :

- ✘ Proporcionar el análisis y diseño de la sistematización del gasto presupuestal en un lapso no mayor a 14 semanas.
- ✘ Establecer la implantación y prueba del sistema en un período no mayor a 3 semanas.
- ✘ Liberar y documentar totalmente el sistema 4 semanas después de su implantación.
- ✘ Se harán recomendaciones para llegar a un nivel más allá del sistema.

HIPOTESIS

Aplicando la sistematización para el control del gasto presupuestal la confiabilidad de la información, el ahorro de tiempo en procesar la información serán mayores. La verificación del gasto será más efectiva.

Se aplicará el método de sistematización presupuestaria y el encargado del departamento de Sistemas en la Dirección de Presupuesto. También se les aplicará cuestionarios a los usuarios de otras dependencias que laboran en forma conjunta con la Dirección. Además incluirá las siguientes etapas:

A - Análisis del Problema.

B - Diseño del Sistema.

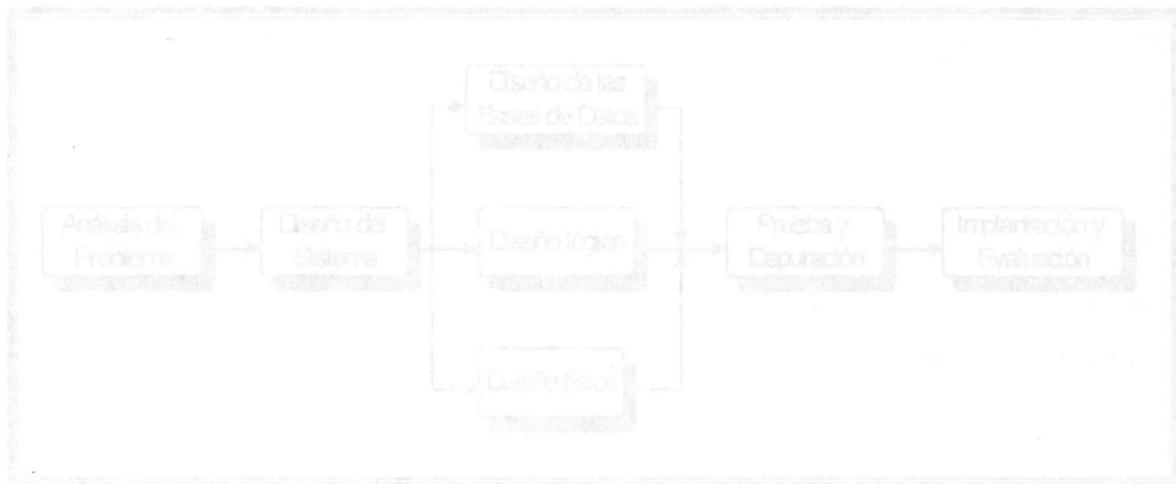
↳ Diseño de las Bases de Datos.

↳ Diseño lógico.

↳ Diseño Físico.

C - Prueba y depuración.

D - implantación y evaluación.



PLANEACION DEL SISTEMA

La metodología que se llevará a cabo para cubrir el objetivo trazado será mediante entrevistas personales al jefe de departamento de Ejecución y Control del Presupuesto, al jefe del departamento de Ejecución Presupuestaria y al encargado del departamento de Sistemas en la Dirección de Presupuesto. También se les aplicará cuestionarios a los usuarios de otras dependencias que laboran en forma conjunta con la Dirección. Además incluirá las siguientes etapas:

A.- Análisis del Problema.

B.- Diseño del Sistema.

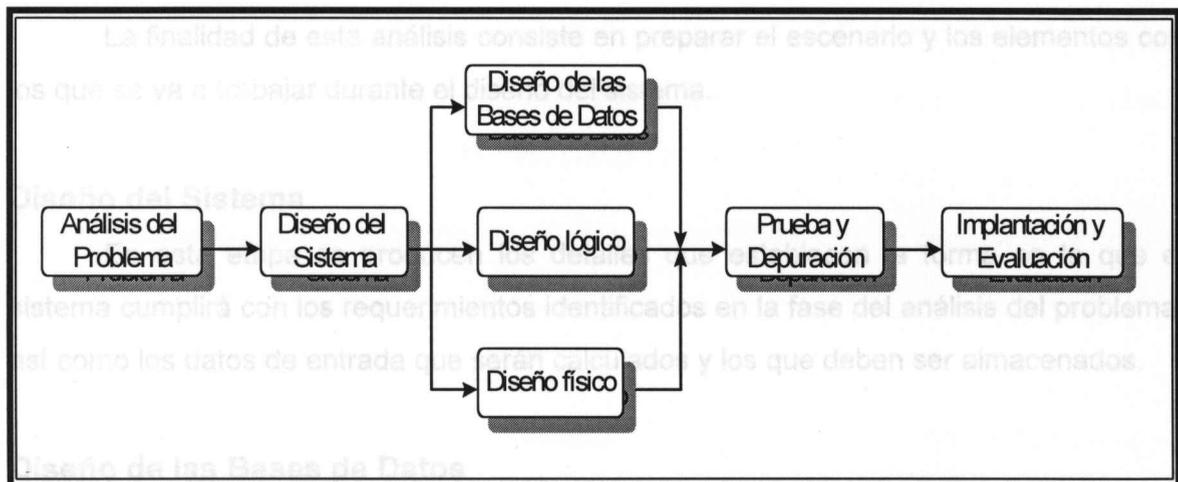
➤ Diseño de las Bases de Datos.

➤ Diseño lógico.

➤ Diseño Físico.

C.- Prueba y depuración.

D.- Implantación y evaluación.



Las etapas anteriormente señaladas corresponden al Ciclo de vida del desarrollo de sistemas. Dentro de cada fase se emplearán las herramientas del método de análisis estructurado, ya que éste nos va a permitir conocer el sistema en forma lógica y como opera.

Los elementos del análisis estructurado que se usarán son:

- Símbolos gráficos
- Diccionario de datos
- Descripción y procedimientos
- Estándares para describir y documentar el sistema

Análisis del problema o Investigación Preliminar

Uno de los aspectos fundamentales en el desarrollo de sistemas es comprender el problema en su totalidad; esto es, estudiarlo de manera profunda para poder definirlo y describirlo de tal manera, que de este conocimiento se puedan seleccionar los elementos que son relevantes para el desarrollo del sistema.

La finalidad de este análisis consiste en preparar el escenario y los elementos con los que se va a trabajar durante el diseño del sistema.

Diseño del Sistema

En esta etapa se producen los detalles que establecen la forma en la que el sistema cumplirá con los requerimientos identificados en la fase del análisis del problema, así como los datos de entrada que serán calculados y los que deben ser almacenados.

Diseño de las Bases de Datos

Habiendo identificado los elementos del sistema se puede proceder a diseñar las tablas o bases de datos donde se guardará la información.

Diseño Lógico

En esta etapa se realizará la apariencia que presentan las pantallas y los informes que el usuario contempla ya que esto representará para el usuario la forma en que desarrollará su trabajo

Diseño Físico

Es importante diseñar el programa de forma precisa de tal forma que cumpla con el objetivo planteado. Sin una planificación, el mantenimiento del software se convierte pronto en una parte importante del costo total.

Prueba del sistema

Antes de que los usuarios trabajen con la aplicación, ésta debe superar un programa definido de pruebas para evitar posteriores sucesos indeseables.

Durante esta fase el sistema se emplea de manera experimental para asegurarse de que el software no tenga fallas, es decir, que funcione de acuerdo con las especificaciones.

Implantación y evaluación

La implantación es el proceso de instalar el sistema, capacitar a los usuarios y construir todos los archivos de datos necesarios para utilizarlo.

Una vez determinado el monto la Tesorería General del Estado solicita la autorización del presupuesto de egresos al H. Congreso del Estado quien a su vez analiza y autoriza en forma global, es decir por U.P.P. y por dependencia.

La Tesorería General del Estado comunica a las dependencias por medio de oficios el importa autorizado para ejercerse en el año y le solicita a la dependencia la forma en que lo va a ejercer y le fija un techo financiero que es lo presupuestado a gastar a la cual tiene derecho la dependencia y esta a su vez lo distribuye de acuerdo a sus gastos, y de acuerdo al tipo de gasto lo divide en mensualidades; a esto se le denomina calendarización del gasto. La calendarización del gasto puede ser de dos tipos: el tipo 1 corresponde a una distribución del gasto en forma equitativa durante los 12 meses del año fiscal y el tipo 2 corresponde a una distribución parcial en algunos

ANALISIS DEL PROBLEMA

El objetivo de esta etapa es comprender situaciones, no resolver problemas. Solo después de comprender en su totalidad el sistema, se está en posición de hacer la recomendaciones pertinentes para su diseño.

Se elaborará un cuestionario encaminado a conocer como es y como funciona el presupuesto de egresos

ENTREVISTA AL JEFE DEL DEPARTAMENTO DE EJECUCION Y CONTROL DEL PRESUPUESTO

25 de Octubre de 1996

A) ¿Cómo se estructura el Presupuesto de Egresos?

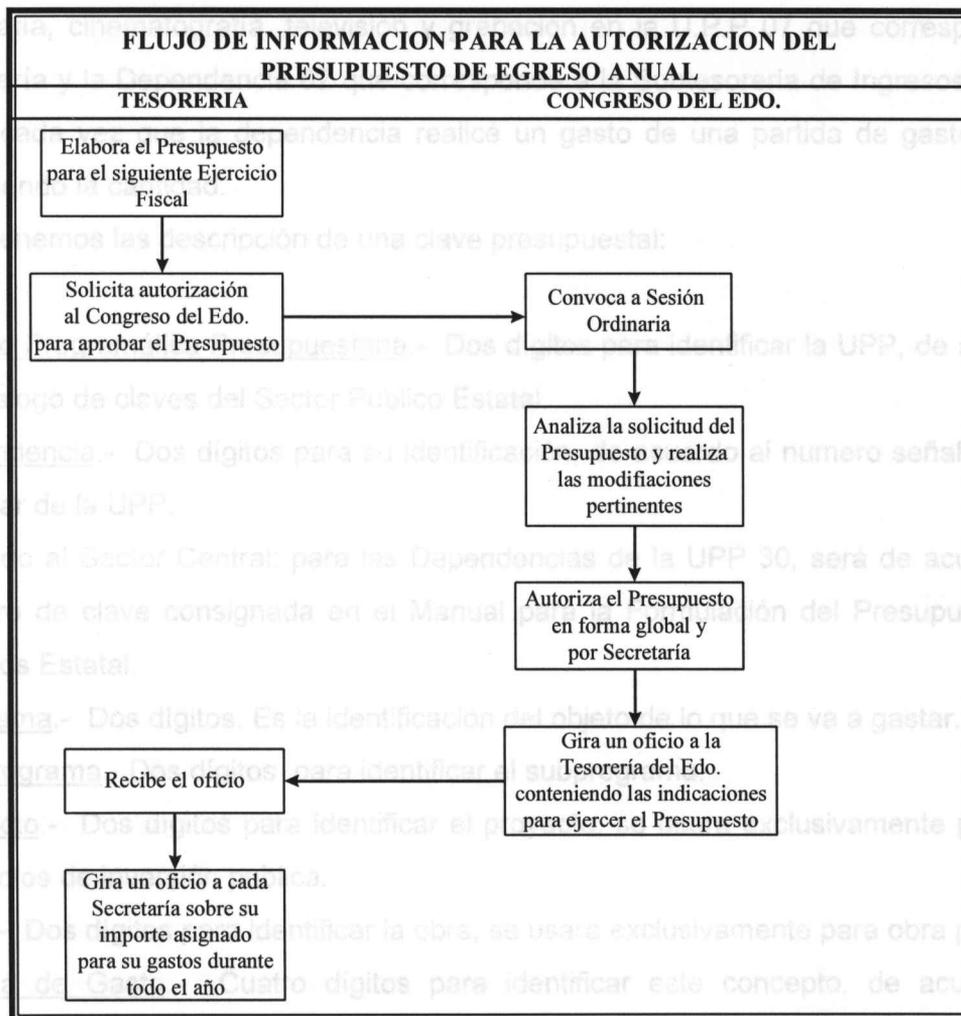
Cuando se aproxima el comienzo de un año fiscal (es el período comprendido de Enero a Diciembre) se hace un análisis del anteproyecto del presupuesto (por parte de la Tesorería General del Estado por medio de la Dirección de Programación y Presupuesto) que tiene como fin presupuestar el gasto o las erogaciones que realizarán las Unidades Programáticas Presupuestales (U.P.P.) en base a lo que se gasto en el año fiscal y el costo de la vida.

Una vez determinado el monto la Tesorería General del Estado solicita la autorización del presupuesto de egresos al H. Congreso del Estado quien a su vez analiza y autoriza en forma global, es decir por U.P.P. y por dependencia.

La Tesorería General del Estado comunica a las dependencias por medio de oficios el importe autorizado para ejercerse en el año y le solicita ala dependencia la forma en que lo va a ejercer y le fija un techo financiero que es lo presupuestado a gastar a la cual tiene derecho la dependencia y esta a su vez lo distribuye de acuerdo a sus gastos, y de acuerdo al tipo de gasto lo divide en mensualidades; a esto se le denomina calendarización del gasto. La calendarización del gasto puede ser de dos tipos: el tipo 1 corresponde a una distribución del gasto en forma equitativa durante los 12 meses del año fiscal y el tipo 2 corresponde a una distribución parcial en algunos

meses del año, es decir puede ser que para una partida de gasto no se efectuó una erogación (de la cantidad asignada para gastar) en algunos meses del año fiscal y sin embargo si en otros meses y a diferencia del tipo 1 que en todo los meses del año se esta gastando de una partida de gasto.

Por otra parte existe dos tipos de presupuesto de egresos: el presupuesto estatal y el presupuesto federal. El primero es el que asigna el gobierno del estado como la mencioné y el segundo lo asigna el gobierno federal de acuerdo a los programas que ellos estén manejando.



B) ¿Cómo se registran o se manejan las erogaciones?

Los gastos o erogaciones que realiza cada dependencia van siendo registradas por medio de una llave denominada Clave Presupuestal la cual esta compuesta de la

siguiente manera: U.P.P., Dependencia, Programa, Subprograma, Proyecto, Obra, Partida, Subregión, Dígito Identificador y Dígito Verificador. La Partida de gasto es el campo que nos indica en lo que se va a gastar el dinero. He aquí un ejemplo de una clave presupuestal:

1.-	U.P.P. DEP	PROG	SUBP	PROY	OBRA	PARTIDA	SUBR	DI	DV
2.-	07	02	01	01	00	00	2605	3	17

La partida 2605 nos esta indicando que se va a realizar un gasto en material de fotografía, cinematografía, televisión y grabación en la U.P.P 07 que corresponde a Tesorería y la Dependencia 02 que corresponde a la Subtesoreria de Ingresos. Por lo tanto cada vez que la dependencia realicé un gasto de una partida de gasto se irá reduciendo la cantidad.

Aquí tenemos las descripción de una clave presupuestal:

Unidad Programática Presupuestaria.- Dos dígitos para identificar la UPP, de acuerdo al catalogo de claves del Sector Publico Estatal.

Dependencia.- Dos dígitos para su identificación, de acuerdo al numero señalado por el titular de la UPP.

Referido al Sector Central; para las Dependencias de la UPP 30, será de acuerdo al numero de clave consignada en el Manual para la Formulación del Presupuesto de Egresos Estatal.

Programa.- Dos dígitos. Es la identificación del objeto de lo que se va a gastar.

Subprograma.- Dos dígitos para identificar el subprograma.

Proyecto.- Dos dígitos para identificar el proyecto; se usara exclusivamente para los proyectos de inversión publica.

Obra.- Dos dígitos para identificar la obra; se usara exclusivamente para obra publica.

Partida de Gasto.- Cuatro dígitos para identificar este concepto, de acuerdo al Clasificador General por Objeto del Gasto.

Subregión.- Un dígito para identificar este concepto en la región, de acuerdo al manual para la formulación del Presupuesto de Egresos Estatal. (El numero tres se usara invariablemente para ubicar el gasto corriente) para la ejecución de obra se hará la

regionalización, de acuerdo a los criterios del Programa Nacional de Solidaridad que divide al Estado en las siguientes regiones:

- | | |
|-------------------------------|--------------------------------|
| 1.- Región Patzcuaro-Zirahuen | 6.- Región Tierra Caliente |
| 2.- Región Zacapu | 7.- Región Costa |
| 3.- Región Lago de Cuitzeo | 8.- Región Meseta Purepecha |
| 4.- Región Bajío | 9.- Región Valle de Apatzingan |
| 5.- Región Oriente | 0.- Región Ciénaga de Chápala |

Dígito Identificador (D.I.).- Un dígito que permite la identificación de recursos.

Dígito Verificador (D.V.).- Un dígito que permite el registro confiable de la información.

En cuanto al catálogo de U.P.P. existen veintitrés, quince organismos, dos empresas de participación estatal, tres juntas, dos comisiones, dos comités y un fideicomiso, sujeto a control presupuestal. A continuación se detallan las U.P.P. con sus números, así como los Organismo Descentralizados, Empresas, Juntas, Comisiones, Comités y Fideicomisos.

NÚM.NOMBRE

- | | |
|----|---|
| 01 | Poder Legislativo. |
| 02 | Poder Judicial. |
| 03 | Ejecutivo del Estado. |
| 04 | Secretaría Particular. |
| 05 | Coordinación General de Comunicación Social. |
| 06 | Secretaría de Gobierno. |
| 07 | Tesorería General del Estado. |
| 08 | Secretaría de Comunicaciones y Obras Públicas. |
| 09 | Secretaría de Desarrollo Agropecuario y Forestal. |
| 10 | Secretaría de Desarrollo Industrial y Comercial. |
| 11 | Secretaría de Turismo. |
| 12 | Secretaría de Educación en el Estado. |

- 13 Procuraduría General del Estado.
- 14 Oficialía Mayor.
- 15 Coordinación de Programas Estatales de Solidaridad.
- 16 Coordinación de Apoyo Municipal.
- 17 Secretaría de Salud.
- 18 Secretaría de Desarrollo Urbano y Ecología.
- 19 Contraloría General del Estado.
- 20 Coordinación de Gestión Social.
- 23 Participaciones Municipales.
- 24 Erogaciones Adicionales.
- 25 Deuda Pública.

C) DEPENDENCIAS DE LA U.P.P 30

Esos formatos o solicitudes que usted menciona nosotros lo denominamos Organismos, Empresas, Juntas, Comités, Comisiones y Fideicomisos

NÚM.NOMBRE

- 31 Casa de las Artesanías.
- 33 Instituto Michoacano de la Juventud y el Deporte.
- 34 Instituto Michoacano de Cultura.
- 35 Sistema Michoacano de Radio y Televisión.
- 36 Centro de Convenciones de Morelia.
- 37 Parque Zoológico Benito Juárez.
- 38 Universidad Michoacana de San Nicolás de Hidalgo.
- 39 Patronato de Promotores Voluntarios.
- 40 Sistema para el Desarrollo Integral de la Familia Michoacana.
- 42 Instituto Catastral y Registral.
- 43 Transporte Público de Michoacán.
- 44 Promotora Michoacana de la Vivienda.
- 46 Consejo Tutelar de Morelia.
- 47 Promotora de Parques Agropecuarios de Michoacán.
- 51 Colegio de Bachilleres.

- 62 A.S.T.E.C.A. del Estado de Michoacán S.A. de C.V.
- 63 Empresas de Michoacán para el Desarrollo
- 71 Junta Local de Conciliación y Arbitraje.
- 72 Junta Local de Caminos
- 73 Comisión Federal de Electricidad Coordinación Divisional de Electrificación Rural en Michoacán.
- 74 Junta de Asistencia privada del Gobierno del Estado.
- 75 Comisión Estatal de Derecho Humanos.
- 80 Comité de Planeación para el Desarrollo del Estado de Michoacán.
- 81 Comité de Agua potable y Alcantarillado.
- 92 Fideicomiso de Parques Industriales de Michoacán.

C) ¿Cuales son los formatos o solicitudes para realizar un gasto?

Esos formatos o solicitudes que usted menciona nosotros lo denominamos documentos. Existen varios documentos: Documento de Ejecución Presupuestaria, Oficio de Modificación, Documento Múltiple, Requisiciones y Salidas de Almacén. Puede usted tener más información de estos documentos con el jefe del departamento de Ejecución Presupuestaria.

La Dirección de Programación y Presupuesto recibe los documentos

ENTREVISTA AL JEFE DE DEPARTAMENTO DE EJECUCION PRESUPUESTARIA

que cumplan con las normas y disposiciones establecidas para su llenado.

Primera Parte

A) ¿Cómo funciona el sistema de ejecución?

Aprobado el Presupuesto de Egresos por el H. Congreso del Estado, el mecanismo de ejecución realiza las siguientes actividades:

Central de Ejecución del Presupuesto.- Es el registro que, mediante la apertura de auxiliares, se realiza en la Dirección de Programación y Presupuesto para registrar las operaciones a nivel de clave presupuestaria.

Central de Documentos Presupuestarios.- Este se encuentra integrado por los documentos aprobados mediante los cuales se afecta legalmente el Presupuesto de Egresos del Estado. Los documentos de referencia son los siguientes: Documento de

Ejecución Presupuestaria y Pago, Requisición, Reporte de Salida de Almacén, Oficio de Modificación Presupuestaria, Documento Múltiple.

La Dirección de Programación y Presupuesto por su parte entrega a cada una de las Unidades Programáticas Presupuestarias una copia de los listados que contengan la información del Archivo Central de Ejecución del Presupuesto y Calendario de Pagos, para efectos de que estas abran sus auxiliares de registro.

Segunda Parte

A) El propósito es llevar a cabo el registro directo de cada una de las operaciones que afectan al presupuesto de las Unidades Programáticas Presupuestarias, organismos descentralizados, empresas de participación estatal, juntas, comisiones, comités y fideicomisos, de la Administración Pública Estatal y uniformar los criterios en cuanto al procedimiento del registro, acorde con los archivos centrales de presupuesto, para estar en condiciones de efectuar conciliaciones periódicas y detectar posibles omisiones de cargos y abonos presupuestales.

RECEPCIÓN, REVISIÓN Y AUTORIZACIÓN DE DOCUMENTOS PRESUPUESTARIOS.

La Dirección de Programación y Presupuesto recibe los documentos presupuestarios expedidos por las Unidades Programáticas Presupuestarias y verifica que cumplan con las normas y disposiciones establecidas para su llenado.

Así mismo, consulta el archivo central de ejecución del presupuesto para verificar las disponibilidades. Si existe saldo insuficiente lo registra en el archivo central de documentos presupuestarios, dando a conocer a las unidades Programáticas presupuestarias de su autorización; si no existe saldo insuficiente, no se realiza ningún registro y rechaza el documento, notificándole a la unidad Programática presupuestaria lo anterior, mediante la devolución del documento.

Si los documentos no se apegan a los lineamientos establecidos para su autorización, son devueltos a las Unidades Programáticas para que se proceda a la corrección de dichos documentos y se actualicen en forma inmediata sus auxiliares de registro.

Cuando los documentos presupuestarios (documento de ejecución presupuestaria y pago, oficio de modificación presupuestaria y documento múltiple) sean cancelados una vez que ya hayan sido autorizados por la Dirección de Programación y Presupuesto, se deberá remitir a esta una copia del documento, con la anotación o sello de cancelado.

Segunda Parte documento

A) ¿Cuál es la función o actividad del Documento de Ejecución Presupuestaria?

Este documento se expide para liquidar cualquier tipo de erogaciones con cargo al Presupuesto de Egresos del Estado.

Los documentos comprobatorios permanecerán en el archivo de cada Unidad Programática Presupuestaria para auditorías o evaluaciones posteriores, excepto en el caso de las compras globales por la Dirección de Recursos Materiales, así como los servicios generales, realizados por la Dirección de Egresos para su comprobación y registro.

En un Documento de Ejecución Presupuestaria y Pago se podrán afectar hasta quince claves presupuestarias con diferente beneficiario, siempre y cuando sean de la misma Dependencia o Programa.

La afectación presupuestaria y contable será a nivel de importes brutos.

El documento de Ejecución Presupuestaria y Pago será un documento no negociable, por lo que la Delegación Administrativa, elaborara los contrarecibos mediante los cuales se harán efectivos los cobros, estos comprobantes deberán ser entregados a los beneficiarios, hasta que las Delegaciones tengan en su poder la copia del documento autorizado por la Dirección de Programación y Presupuesto. Será facultad de la Tesorería General del Estado exigir la identificación respectiva del o los cobradores y certificar su acreditación legal para realizar los cobros.

especificar los siguientes campos:

Cuando se trate de retiro de fondos a favor de los Organismos, será el gestor autorizado de los mismos, quien se presentara en la oficina pagadora para hacer efectivo el cobro.

Encabezado del documento

El Documento de Ejecución Presupuestaria y Pago no deberá llevar tachaduras, borrones o enmendaduras, pues de incurrir en errores será motivo suficiente para rechazar el documento.

1.-U.P.P.- Dos dígitos. Es la Unidad Programática Presupuestal del Gobierno del

2.-Tipo de Documento - Un dígito. Es la forma de la hoja. Para

3.-Número de Documento - Dos dígitos. Es el número progresivo que identifica al documento;

4.-Día - Dos dígitos. Es el día en que se formula el documento de

5.-Mes - Dos dígitos. Es el mes en que se formula el documento de

6.-Año - Dos dígitos. Es el año en que se formula el documento de

7.-Importe - El monto por el cual se va a comprar.

8.-Código de movimiento - Un dígito. Es el código que se utiliza para

9.-Código de dependencia - Dos dígitos. Es el código de la dependencia que

10.-Código de ejecución - Dos dígitos. Es el código de la ejecución que

11.-Código de pago - Dos dígitos. Es el código de la forma de pago que

12.-Código de movimiento - Un dígito. Es el código que se utiliza para

13.-Código de dependencia - Dos dígitos. Es el código de la dependencia que

14.-Código de ejecución - Dos dígitos. Es el código de la ejecución que

15.-Código de pago - Dos dígitos. Es el código de la forma de pago que

16.-Código de movimiento - Un dígito. Es el código que se utiliza para

17.-Código de dependencia - Dos dígitos. Es el código de la dependencia que

18.-Código de ejecución - Dos dígitos. Es el código de la ejecución que

19.-Código de pago - Dos dígitos. Es el código de la forma de pago que

20.-Código de movimiento - Un dígito. Es el código que se utiliza para

21.-Código de dependencia - Dos dígitos. Es el código de la dependencia que

22.-Código de ejecución - Dos dígitos. Es el código de la ejecución que

23.-Código de pago - Dos dígitos. Es el código de la forma de pago que

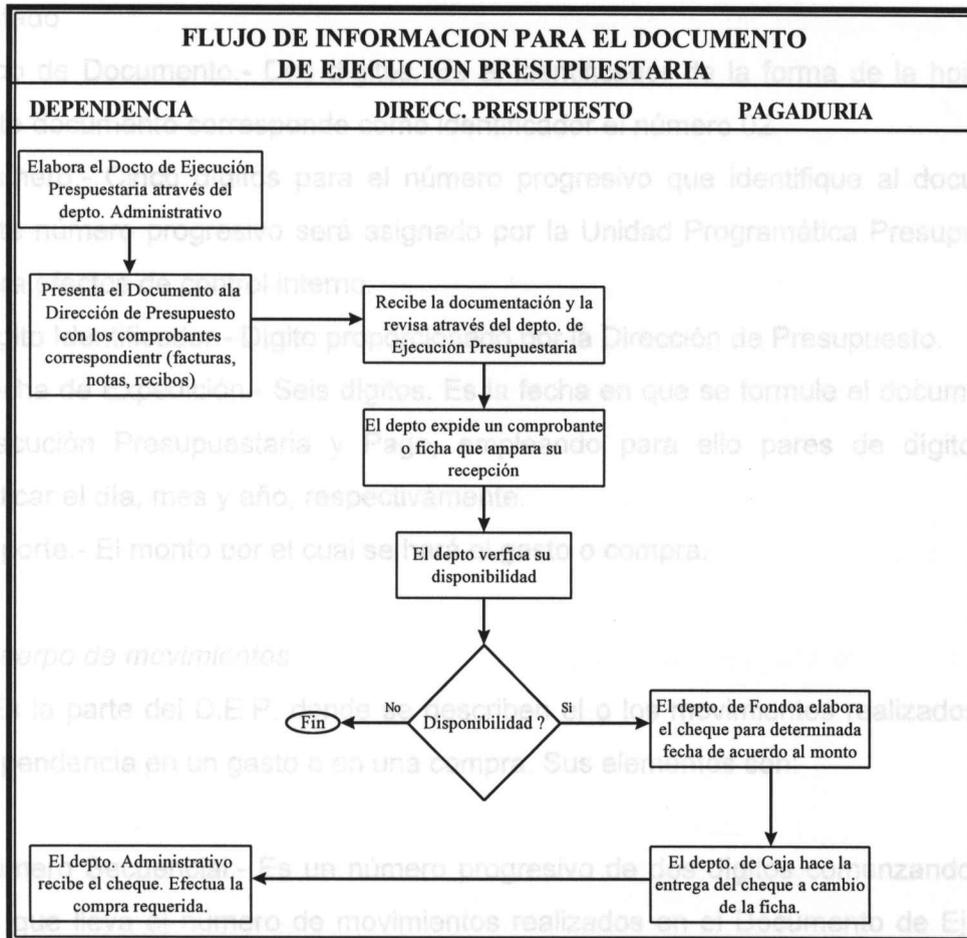
24.-Código de movimiento - Un dígito. Es el código que se utiliza para

25.-Código de dependencia - Dos dígitos. Es el código de la dependencia que

26.-Código de ejecución - Dos dígitos. Es el código de la ejecución que

27.-Código de pago - Dos dígitos. Es el código de la forma de pago que

28.-Código de movimiento - Un dígito. Es el código que se utiliza para



El Documento de Ejecución Presupuestaria es el único gestor ante la Tesorería General del Estado para solicitar dinero ya sea por un gasto o una compra mediante de una nota, factura etc., la cual funcionará como comprobante. Es elaborado por la dependencia solicitante mediante su delegación administrativa en el cual se especifican los siguientes campos:

La forma del Documento de Ejecución Presupuestaria esta dividida en dos partes: Identificación del documento y Desglose de movimientos a gastar

- Partida.- Indica en que se va a gastar el dinero.

Encabezado del documento es el lugar donde va realizar el gasto. Michoacán está

Es la parte del D.E.P. donde se especifican los datos principales de la dependencia. Sus elementos son:

de Egresos Estatal, y será exclusivo para obra, el numero tres ubica el

1.-U.P.P.- Dos dígitos. Es la Unidad Programática Presupuestal del Gobierno del Estado - Dígito Identificador - Verifica, mediante de un número que la Clave

2.-Tipo de Documento.- Dos dígitos. Es el identificador de la forma de la hoja. Para éste documento corresponde como identificador el número 02.

3.-Número.- Cinco dígitos para el número progresivo que identifique al documento; este número progresivo será asignado por la Unidad Programática Presupuestaria para efectos de control interno. la Clave Presupuestal Dependencia.

4.-Dígito Identificador.- Dígito proporcionado por la Dirección de Presupuesto.

5.-Fecha de Expedición.- Seis dígitos. Es la fecha en que se formule el documento de Ejecución Presupuestaria y Pago, empleando para ello pares de dígitos para indicar el día, mes y año, respectivamente. puestos a la partida.

6.-Importe.- El monto por el cual se hará el gasto o compra. de las deducciones al cargo presupuestal.

15 *Cuerpo de movimientos* Es tipo de comprobante que ampara al Documento de

Es la parte del D.E.P. donde se describen el o los movimientos realizados por la dependencia en un gasto o en una compra. Sus elementos son:

7.-Número Secuencial.- Es un número progresivo de dos dígitos comenzando con el 01 que lleva el número de movimientos realizados en el Documento de Ejecución Presupuestaria. el presupuesto de egresos, de acuerdo a las necesidades

8.-Clave Presupuestal.- Diez y ocho dígitos. Está contenida por varios subcampos:

- U.P.P. .- Unidad Programática Presupuestal el mismo tiempo que permita
- Dependencia .- Es la Dirección o Subsecretaria que pertenece a la U.P.P. mostrar
- Programa .- Es un objeto de lo que se va a gastar y porque se va a gastar
- Subprograma .- Es parte del objeto en que se va a gastar. Cuad. ejecución inicial, a la

se le denomina reducción presupuestaria; y por el contrario cuando se

- Proyecto .- Es el conjunto de obras a realizarse
- Obra.- Es el nombre de la obra en que se va a gastar
- Partida.- Indica en que se va a gastar el dinero.
- Subregión.- Es el lugar donde va realizar el gasto. Michoacán está dividido en 9 subregiones. Este dígito se tomara del catalogo de Regiones que forma parte del Manual para la Formalicen del Presupuesto de Egresos Estatal, y será exclusivo para obra, el numero tres ubica el gasto corriente.
- Dígito Identificador .- Verifica, mediante de un número que la Clave Presupuestal sea la correcta.

9.-Dígito Verificador.- Dígito que verifica que la Clave Presupuestal este bien escrita

10.- Beneficiario.- Es el sujeto que se va hacer acreedor a ese gasto el cual surge de la clave de la que se anote en la Clave Presupuestal Dependencia.

11.- Concepto.- Es la descripción de la partida en que se va a gastar.

12.- Cargo Presupuestal.- Es el monto por el cual se va a gastar con cargo a la partida.

13.- Deducciones.- Son la reducciones por impuestos a la partida.

14.- Liquido.- Es el total a gastar después de haber restado las deducciones al cargo presupuestal.

15.- Tipo de Justificación.- Es tipo de comprobante que ampara al Documento de Ejecución Presupuestaria.

16.- Número.- Es el folio del comprobante.

1.-U.P.P.- Dos digito. Es la Unidad Programática Presupuestal del Gobierno del

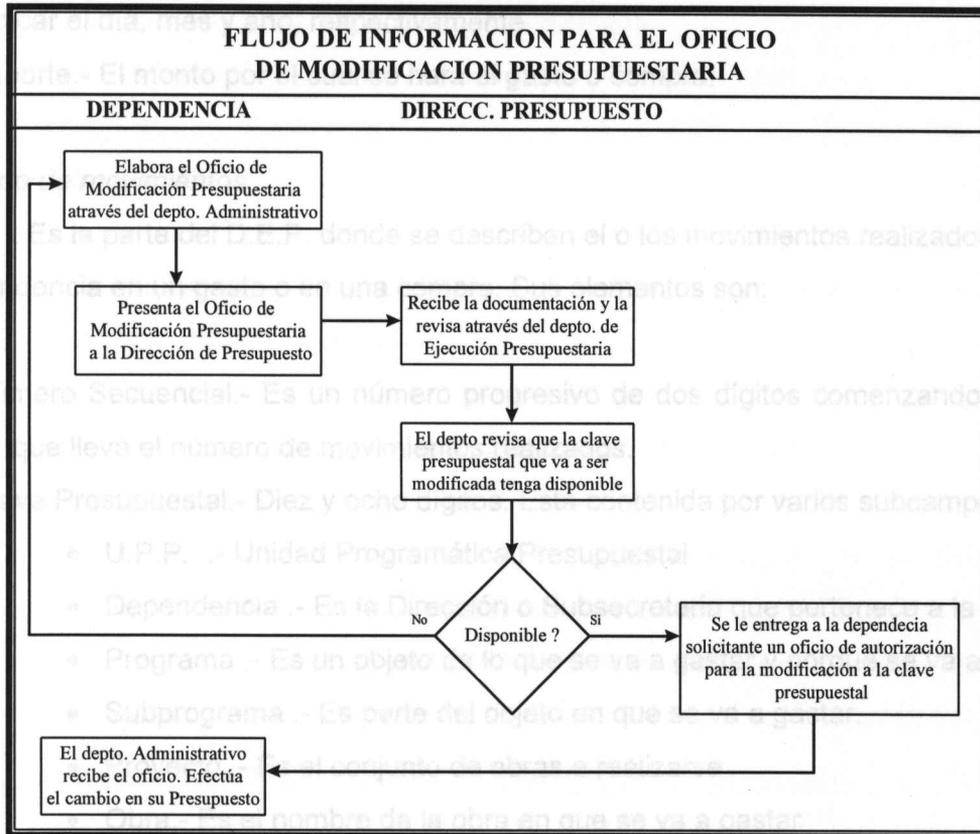
B) ¿Cuál es la función o actividad del Oficio de Modificación?

Este documento nos permite contar con un mecanismo ágil que permita efectuar las aclaraciones al presupuesto de egresos, de acuerdo a las necesidades debidamente justificadas que se presentan en las Unidades Programáticas Presupuestaria o Dependencias Ejecutoras del Gasto, al mismo tiempo que permita mostrar reflejo real de la situación que guarda en un momento dado el presupuesto.

4.- Dígito Identificador - Dígito proporcionado por la Dirección de Presupuesto.

Quando a una partida de gasto se le quita parte de su asignación inicial, a la acción realizada se le denomina reducción presupuestaria; y por el contrario cuando se

le agrega se le llama ampliación presupuestaria. La cantidad que se le reduce a una partida es la misma con que se le amplia a la otra.



Por la forma del Oficio de Modificación Presupuestaria podemos apreciar que está dividido en dos partes: Encabezado y Desglose de movimientos.

Encabezado del Documento

- 1.-U.P.P.- Dos dígitos. Es la Unidad Programática Presupuestal del Gobierno del Estado
- 2.-Tipo de Documento.- Dos dígitos. Es el identificador de la forma de la hoja. Para este documento corresponde como identificador el número 02.
- 3.-Número.- Cinco dígitos para el número progresivo que identifique al documento; este número progresivo será asignado por la Unidad Programática Presupuestaria para efectos de control interno.
- 4.-Dígito Identificador.- Dígito proporcionado por la Dirección de Presupuesto.

5.- Fecha de Expedición.- Seis dígitos. Es la fecha en que se formule el documento de Ejecución Presupuestaria y Pago, empleando para ello pares de dígitos para indicar el día, mes y año, respectivamente.

6.- Importe.- El monto por el cual se hará el gasto o compra.

13.- Importe.- Es el monto por el cual se va a ampliar y/o reducir las Claves

Cuerpo de movimientos

14.- Es la parte del D.E.P. donde se describen el o los movimientos realizados por la dependencia en un gasto o en una compra. Sus elementos son:

que contiene el Oficio de Modificación.

7.- Número Secuencial.- Es un número progresivo de dos dígitos comenzando con el

01 que lleva el número de movimientos realizados.

8.- Clave Presupuestal.- Diez y ocho dígitos. Está contenida por varios subcampos:

- U.P.P. .- Unidad Programática Presupuestal
- Dependencia .- Es la Dirección o Subsecretaría que pertenece a la U.P.P.
- Programa .- Es un objeto de lo que se va a gastar y porque se va a gastar.
- Subprograma .- Es parte del objeto en que se va a gastar.
- Proyecto .- Es el conjunto de obras a realizarse
- Obra.- Es el nombre de la obra en que se va a gastar
- Partida.- Indica en que se va a gastar el dinero.
- Subregión.- Es el lugar donde va a realizar el gasto. Michoacán está dividido en 9 subregiones. Este dígito se tomara del catalogo de Regiones que forma parte del Manual para la Formalicen del Presupuesto de Egresos Estatal, y será exclusivo para obra, el numero tres ubica el gasto corriente.
- Dígito Identificador .- Verifica, mediante de un número que la Clave Presupuestal sea la correcta.

9.- Dígito Verificador.- Dígito que verifica que la Clave Presupuestal este bien escrita

10.- Tipo de Movimiento.- Es el modo de como se va a afectar la clave presupuestal.

Existen 2 tipos: Compensada que indica que viene de una partida que tiene fondos y la Líquida que viene de una partida la cual no tiene fondos suficientes.

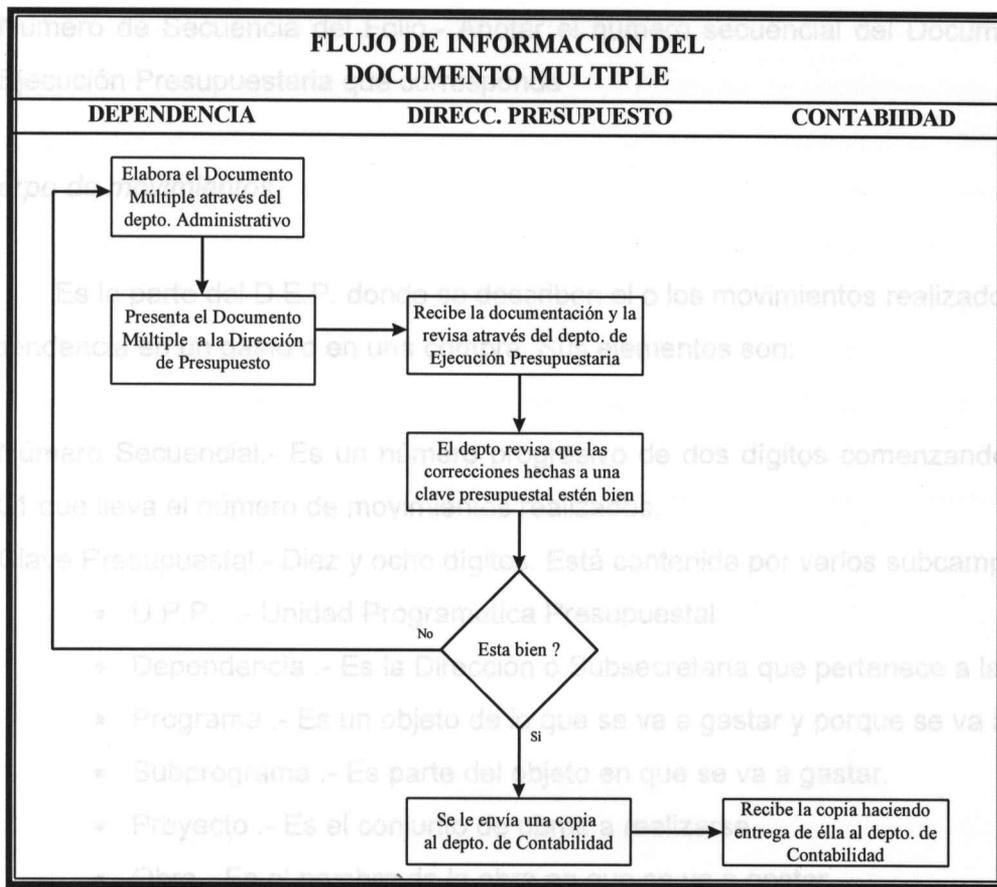
- 11.- Tipo de Operación.- Indica en cada Clave Presupuestal la operación que se está realizando. Si el tipo de operación es "02" es una reducción del gasto de un partida y si el tipo de operación es "03" es una ampliación.
- 12.- Concepto.- Es el nombre de la partida en que se va a gastar.
- 13.- Importe.- Es el monto por el cual se va ampliar y/o reducir las Claves Presupuestales.
- 14.- Calendario de Pagos.- En estas columnas se anotara el calendario o período de amortización de pagos que se afecta en los totales de las claves presupuestarias que contiene el Oficio de Modificación.

C) ¿Cuál es la función o actividad del Documento Múltiple?

Este documento tiene como finalidad corregir todas aquellas aplicaciones que por omisión de cargo equivocado o abono se dejaron de hacer, o se hicieron al presupuesto, respectivamente. Así mismo, sirve para enterar cantidades en efectivo o cheques reintegrados a la Tesorería, siendo justificados estos por recibo oficial; para ello, deberá anexarse una copia de dicho documento, como comprobante del ingreso con la anotación siguiente: Reintegro a Presupuesto.

Encabezado del Documento

- 1.- U.P.P.- Dos dígitos. Es la Unidad Programática Presupuestal del Gobierno del Estado.
- 2.- Tipo de Documento.- Dos dígitos. Es el identificador de la forma de la hoja. Para éste documento corresponde como identificador el número 02.
- 3.- Número.- Cinco dígitos para el número progresivo que identifique al documento; este número progresivo será asignado por la Unidad Programática Presupuestaria para efectos de control interno.
- 4.- Dígito Identificador.- Dígito proporcionado por la Dirección de Presupuesto.
- 5.- Fecha de Expedición.- Seis dígitos. Es la fecha en que se formule el documento de Ejecución Presupuestaria y Pago, empleando para ello pares de dígitos para indicar el día, mes y año, respectivamente.
- 6.- Folio.- Es el número con el cual se va a identificar el Documento de Ejecución Presupuestaria que se va a corregir.



Encabezado del Documento

- 1.-U.P.P.- Dos dígito. Es la Unidad Programática Presupuestal del Gobierno del Estado
- 2.-Tipo de Documento.- Dos dígitos. Es el identificador de la forma de la hoja. Para éste documento corresponde como identificador el número 02.
- 3.-Número.- Cinco dígitos para el número progresivo que identifique al documento; este número progresivo será asignado por la Unidad Programática Presupuestaria para efectos de control interno.
- 4.-Dígito Identificador.- Dígito proporcionado por la Dirección de Presupuesto.
- 5.-Fecha de Expedición.- Seis dígitos. Es la fecha en que se formule el documento de Ejecución Presupuestaria y Pago, empleando para ello pares de dígitos para indicar el día, mes y año, respectivamente.
- 6.-Folio.- Es el número con el cual se va a identificar al Documento de Ejecución Presupuestaria que se va a corregir

7.-Número de Secuencia del Folio.- Anotar el número secuencial del Documento de Ejecución Presupuestaria que corresponda

Cuerpo de movimientos

Es la parte del D.E.P. donde se describen el o los movimientos realizados por la dependencia en un gasto o en una compra. Sus elementos son:

8.-Número Secuencial.- Es un número progresivo de dos dígitos comenzando con el 01 que lleva el número de movimientos realizados.

9.-Clave Presupuestal.- Diez y ocho dígitos. Está contenida por varios subcampos:

- U.P.P. .- Unidad Programática Presupuestal
- Dependencia .- Es la Dirección o Subsecretaría que pertenece a la U.P.P.
- Programa .- Es un objeto de lo que se va a gastar y porque se va a gastar
- Subprograma .- Es parte del objeto en que se va a gastar.
- Proyecto .- Es el conjunto de obras a realizarse
- Obra.- Es el nombre de la obra en que se va a gastar
- Partida.- Indica en que se va a gastar el dinero.
- Subregión.- Es el lugar donde va realizar el gasto. Michoacán está dividido en 9 subregiones. Este dígito se tomara del catalogo de Regiones que forma parte del Manual para la Formalicen del Presupuesto de Egresos Estatal, y será exclusivo para obra, el numero tres ubica el gasto corriente.
- Dígito Identificador.- Verifica, mediante de un número que la Clave Presupuestal sea la correcta.

10.-Dígito Verificador.- Dígito que verifica que la Clave Presupuestal este bien escrita

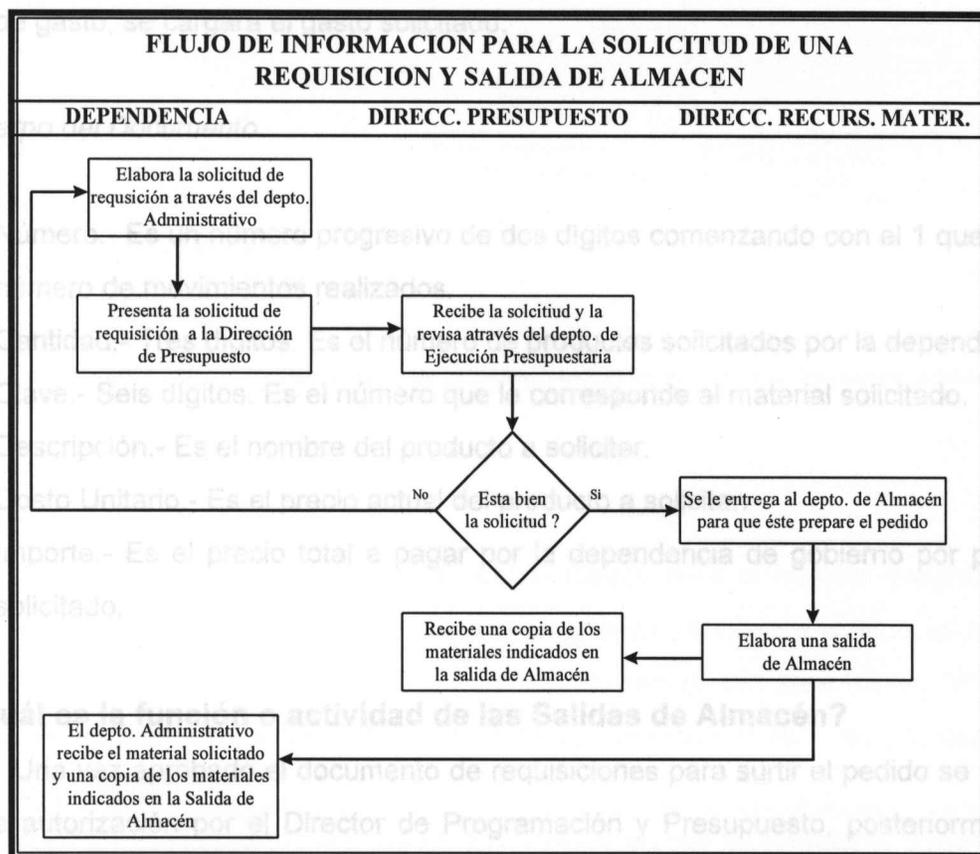
11.-Tipo de Operación.- Indica en cada Clave Presupuestal la operación que se está realizando. Si el tipo de operación es "22" es un abono a lo ejercido de un partida y si el tipo de operación es "21" es un cargo.

12.-Concepto.- Es el nombre de la partida en que se va a gastar.

13.-Importe.- Es el monto por el cual se va ampliar y/o reducir las Claves Presupuestales.

D) ¿Cuál es la función o actividad de las Requisiciones?

Es documento por el cual las U.P.P.'s y Dependencias se comprometen a utilizar ciertos recursos materiales por medio de una Partida de Gasto. En lo relativo a la tramitación de requisiciones, las U.P.P.'s y Dependencias de la Administración Pública Estatal, presentaran estas en la Dirección de Programación y Presupuesto, donde se revisará y analizará todo lo concerniente a la elaboración y justificación de los materiales, consultara en el archivo Central de Ejecución de Presupuesto si existe disponibilidad para la erogación y si esta es positiva se captura para dejar comprometido o respaldado su importe.



Encabezado del Documento

- 1.- Número de Requisición.- Cinco dígitos para el número progresivo que identifique al documento; este número progresivo ya viene asignado en la forma, además un dígito que identifique que tipo de requisición será. Si es "A" es una salida de

almacén, si es de "D" es una compra directa y si es "I" es una compra por equipo inventariado.

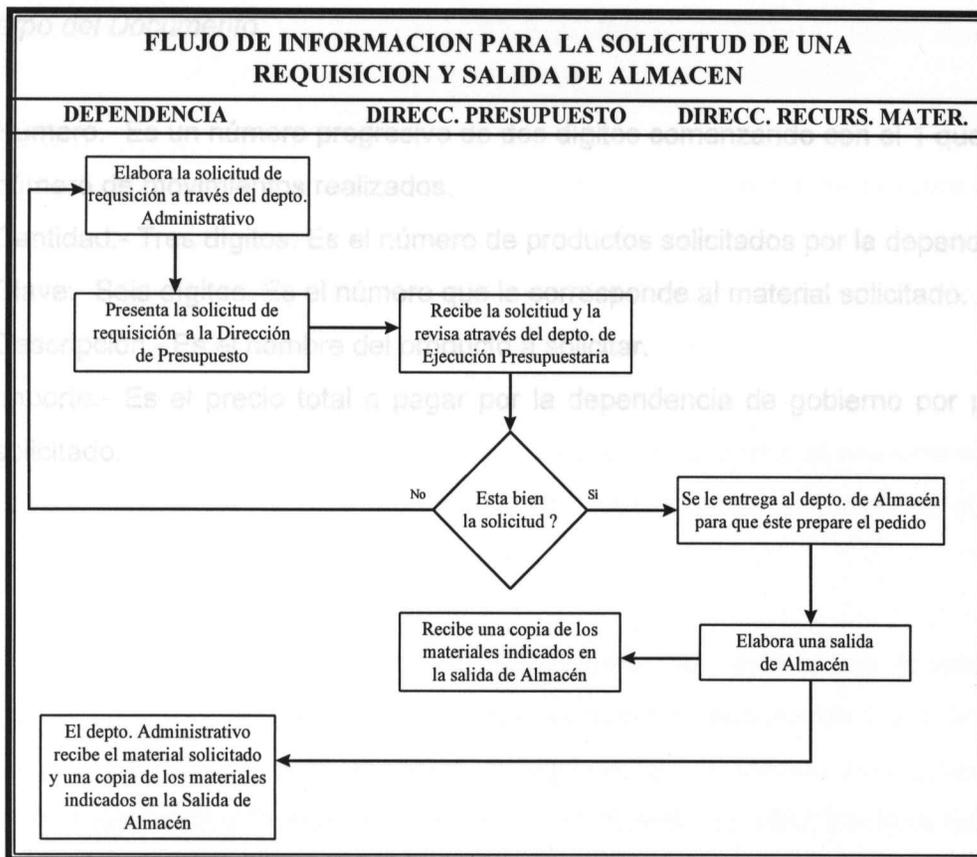
- 2.-Fecha de Expedición.- Seis dígitos. Es la fecha en que se formule el documento de Requisiciones, empleando para ello pares de dígitos para indicar el día, mes y año, respectivamente.
- 3.-U.P.P.- Dos dígitos. Indicará cual es la Unidad Programática Presupuestal solicitante.
- 4.-Dependencia.- Dos dígitos. Indicará cual es la entidad de gobierno solicitante.
- 5.-Clave Presupuestal.- Diez y ocho dígitos. Es la clave, que por medio de la partida de gasto, se cargará el gasto solicitado.

Cuerpo del Documento

- 1.-Número.- Es un número progresivo de dos dígitos comenzando con el 1 que lleva el número de movimientos realizados.
- 2.-Cantidad.- Tres dígitos. Es el número de productos solicitados por la dependencia.
- 3.-Clave.- Seis dígitos. Es el número que le corresponde al material solicitado.
- 4.-Descripción.- Es el nombre del producto a solicitar.
- 5.-Costo Unitario.- Es el precio actual del producto a solicitar.
- 6.-Importe.- Es el precio total a pagar por la dependencia de gobierno por producto solicitado.

E) ¿Cuál es la función o actividad de las Salidas de Almacén?

Una vez aprobado el documento de requisiciones para surtir el pedido se procede a la autorización por el Director de Programación y Presupuesto, posteriormente se entrega a la Entidad o Dependencia, quien a su vez la entrega a la Dirección de Recursos Materiales, para que se hagan los tramites administrativos necesarios. De considerarla correcta dará su visto bueno, para la autorización del Oficial Mayor de Gobierno. Posteriormente procederá a surtir los artículos solicitados, asimismo formulará la Salida de Almacén, la que será entregada a la U.P.P solicitante, así como a la Dirección de Programación y Presupuesto, a fin de actualizar los saldos de las claves presupuestarias.



Encabezado del Documento

- 1.- Número de Salida del Almacén.- Cinco dígitos para el número progresivo que identifique al documento; este número progresivo ya viene asignado en la forma, además un dígito que identifique que tipo de requisición será. Si es "A" es una salida de almacén, si es de "D" es una compra directa y si es "I" es una compra por equipo inventariado.
- 2.- Fecha de Expedición.- Seis dígitos. Es la fecha en que se formule el documento de Requisiciones, empleando para ello pares de dígitos para indicar el día, mes y año, respectivamente.
- 3.- U.P.P.- Dos dígitos. Indicará cual es la Unidad Programática Presupuestal solicitante.
- 4.- Dependencia.- Dos dígitos. Indicará cual es la entidad de gobierno solicitante.
- 5.- Clave Presupuestal.- Diez y ocho dígitos. Es la clave, que por medio de la partida de gasto, se cargará el gasto solicitado.

Cuerpo del Documento VISO A LA RECEPCIÓN DE DOCUMENTOS DE AFECTACIÓN AL PRESUPUESTO.

- 1.-Número.- Es un número progresivo de dos dígitos comenzando con el 1 que lleva el número de movimientos realizados.
- 2.-Cantidad.- Tres dígitos. Es el número de productos solicitados por la dependencia.
- 3.-Clave.- Seis dígitos. Es el número que le corresponde al material solicitado.
- 4.-Descripción.- Es el nombre del producto a solicitar.
- 5.-Importe.- Es el precio total a pagar por la dependencia de gobierno por producto solicitado.

dependencias mencionadas tienen facultad para afectar el presupuesto de las Unidades Programáticas Presupuestarias en base a los requerimientos que estas presenten, el procedimiento es el siguiente:

Aprobado el presupuesto por el H. Congreso del Estado, se procede a su ejecución conforme al calendario de pagos previamente establecido hasta el nivel de partida. La afectación al presupuesto de egresos podrá hacerla directamente cada Unidad Programática Presupuestaria, siempre y cuando no sean partidas que deben llevar el "Visto Bueno" de las direcciones globalizadoras de la Oficialía Mayor antes mencionada.

Cuando la afectación a las partidas presupuestarias no se refiera a las excepciones arriba mencionadas, la Unidad Programática Presupuestaria formulará un documento de ejecución presupuestaria y pago, siguiendo las indicaciones descritas en este Instructivo y lo enviara a la Dirección de Presupuesto para su autorización y radicación posterior a la Dirección de Egresos para su liquidación.

Cuando la afectación a las partidas presupuestarias sea por los conceptos cuya responsabilidad sea competencia de cualesquiera de las direcciones globalizadoras de la Oficialía Mayor, el procedimiento será el siguiente: organigramas.

En lo referente a la Dirección de Administración y Desarrollo de Personal, las U.P.P.s y Dependencias de la Administración Pública Estatal, Organismos y Fideicomisos, presentaran a dicha dirección las plantillas de personal, nominas, solicitudes de altas o bajas y demás movimientos del personal adscrito a las mismas, a

PROCEDIMIENTO PREVIO A LA RECEPCIÓN DE DOCUMENTOS DE AFECTACIÓN AL PRESUPUESTO.

Con base en las funciones establecidas en la Ley Orgánica de la Administración Pública Estatal, existen tres dependencias en la Oficialía Mayor que tienen carácter globalizador; la Dirección de Administración y Desarrollo de Personal, la Dirección de Recursos Materiales y la Dirección de Servicios Generales.

Las Dependencias mencionadas tienen facultad para afectar el presupuesto de las Unidades Programáticas Presupuestarias, en base a los requerimientos que estas presenten; el procedimiento es el siguiente:

Apr probado el presupuesto por el H. Congreso del Estado, se procede a su ejecución conforme al calendario de pagos previamente establecido hasta el nivel de partida. La afectación al presupuesto de egresos podrá hacerla directamente cada Unidad Programática Presupuestaria, siempre y cuando no sean partidas que deben llevar el "Visto Bueno" de las direcciones globalizadoras de la Oficialía Mayor antes mencionada.

Cuando la afectación a las partidas presupuestarias no se refiera a las excepciones arriba mencionadas, la Unidad Programática Presupuestaria formulará un documento de ejecución presupuestaria y pago, siguiendo las indicaciones descritas en este instructivo y lo enviara a la Dirección de Presupuesto para su autorización y radicación posterior a la Dirección de Egresos para su liquidación.

ANÁLISIS DE LOS DOCUMENTOS PARA REALIZAR EROGACIONES

Cuando la afectación a las partidas presupuestarias sea por los conceptos cuya responsabilidad sea competencia de cualesquiera de las direcciones globalizadoras de la Oficialía Mayor, el procedimiento será el siguiente: organigramas.

La U.P.P. realiza un gasto ordenado del presupuesto asignado.

En lo referente a la Dirección de Administración y Desarrollo de Personal, las U.P.P.'s y Dependencias de la Administración Pública Estatal, Organismos y Fideicomisos, presentaran a dicha dirección las plantillas de personal, nominas, solicitudes de altas o bajas y demás movimientos del personal adscrito a las mismas, a

efecto de que una vez realizadas las gestiones administrativas, se hagan las afectaciones presupuestarias mediante de la Nomina Mecanizada.

Las correcciones o adecuaciones en su caso, serán hechas por medio del Documento Múltiple; estas pueden estar motivadas por los cheques cancelados, bajas, omisiones o altas extemporáneas, para el sistema de nomina de procesamiento electrónico.

En lo relativo a la tramitación de requisiciones, las U.P.P.'s y Dependencias de la Administración Publica Estatal, presentaran estas en la Dirección de Programación y Presupuesto, donde revisara y analizara todo lo concerniente a la elaboración y justificación de los materiales, consultara en el archivo Central de Ejecución de Presupuesto si existe disponibilidad para la erogación y si esta es positiva se captura para dejar comprometido o respaldado su importe. A continuación se procede a la autorización por el Director de Programación y Presupuesto, posteriormente se entrega a la Entidad o Dependencia, quien a su vez la entrega a la Dirección de Recursos Materiales, para que se hagan los tramites administrativos necesarios. De considerarla correcta dará su Vo. Bo., para la autorización del Oficial Mayor de Gobierno. Posteriormente procederá a surtir los artículos solicitados, asimismo formulará la Salida de Almacén, la que será entregada a la U.P.P solicitante, así como a la Dirección de Programación y Presupuesto, a fin de actualizar los saldos de las claves presupuestarias.

c) Documento Múltiple

Ventajas

ANALISIS DE LOS DOCUMENTOS PARA REALIZAR EROGACIONES

a) Documento de Ejecución Presupuestaria

Ventajas

- ☛ La U.P.P. realiza un gasto ordenado del presupuesto asignado.
- ☛ Existe una seguridad de lo que se va a gastar no se derroche.
- ☛ Los requisitos que debe llenar la U.P.P. ya están preestablecidos

☛ El control de los documentos es manual por lo que ocasiona mucho papeleo.

d) El control del gasto realizado por una U.P.P. lo controla la Dirección de Presupuesto.

La U.P.P. puede realizar un pedido a la Dirección de Recursos Materiales acuerdo a

Desventajas

- ☛ El control de los documentos es manual por lo que ocasiona mucho papeleo.
- ☛ En los momentos de consultar la disponibilidad de alguna U.P.P. su búsqueda resulta lenta.

b) Oficio de Modificación Presupuestaria

Ventajas

☛ La U.P.P. puede realizar una corrección sobre aquellas aplicaciones por omisión de cargo equivocado o de abono. También puede reintegrar cantidades en efectivo o cheques.

- ☛ Los requisitos que debe llenar la U.P.P. ya están preestablecidos
- ☛ El control de las modificaciones realizadas por una U.P.P. lo controla la Dirección de Presupuesto.

Desventajas

- ☛ El control de los documentos es manual por lo que ocasiona mucho papeleo.
- ☛ En los momentos de consultar la disponibilidad de alguna U.P.P. su búsqueda resulta lenta.

c) Documento Múltiple

Ventajas

- ☛ La U.P.P. puede realizar una modificación de la cantidad a gastar asignado de acuerdo a sus necesidades
- ☛ Los requisitos que debe llenar la U.P.P. ya están preestablecidos
- ☛ El control de las modificaciones realizadas por una U.P.P. lo controla la Dirección de Presupuesto.

Desventajas

- ☛ El control de los documentos es manual por lo que ocasiona mucho papeleo.

d) RequisicionesVentajas

- ☛ La U.P.P. puede realizar un pedido a la Dirección de Recursos Materiales acuerdo a sus necesidades
- ☛ Los requisitos que debe llenar la U.P.P. ya están preestablecidos
- ☛ El control de las requisiciones solicitadas por una U.P.P. lo controla la Dirección de Presupuesto.

Desventajas

- ☛ El control de los documentos es manual por lo que ocasiona mucho papeleo.

e) Salidas de AlmacénVentajas

- ☛ La U.P.P., una vez hecho la requisición, obtiene lo solicitado.
- ☛ Los requisitos que debe llenar la U.P.P. ya están preestablecidos
- ☛ El control de pedidos realizados por una U.P.P. lo controla la Dirección de Presupuesto y la Dirección de Recursos Materiales.

Desventajas

- ☛ El control de los documentos es manual por lo que ocasiona mucho papeleo.

ESTRUCTURA DEL PRESUPUESTO DE EGRESOSEstructura General

La estructura de los elementos presupuesto de egresos se divide en dos tipos: los comunes y los exclusivos o particulares

Los primeros tienen como propósito facilitar la clasificación y ordenación del presupuesto de egresos las cuales son actividades que pertenecen a todos los elementos. La determinación de los campos se hizo en base a las entrevistas realizadas.

Los elementos exclusivos o particulares son aquellos características propias del reactivo que fueron analizados en la etapa del análisis.

DISEÑO DE LA SOLUCION

El diseño es la conversión de los requerimientos en métodos que los satisfagan. La esencia del diseño de sistemas es la selección y especificación de las características del sistema de información.

OBJETIVOS DEL DISEÑO

- ✎ Especificación de los elementos del diseño lógico: salida, entradas, archivos, bases de datos y procedimientos, todo en una forma que satisfagan los requerimientos del sistema.
- ✎ Asegurar que las características del sistema cumplan con los requerimientos del usuario, es decir, que satisfagan las necesidades de los usuarios si:
 - ✓ Realiza en forma apropiada los procedimientos correctos
 - ✓ Presenta información e instrucciones en una forma aceptable y efectiva
 - ✓ Produce resultados exactos
 - ✓ Proporciona una interfase y métodos de interacción aceptables
 - ✓ Es percibido por los usuarios como un sistema confiable.
- ✎ Proporcionar un sistema que sea fácil de utilizar.
- ✎ Proporcionar especificaciones detalladas para el desarrollo de software.

ESTRUCTURA DEL PRESUPUESTO DE EGRESOS

Estructura General

La estructura de los elementos presupuesto de egresos se divide en dos tipos: los comunes y los exclusivos o particulares

Los primeros tienen como propósito facilitar la clasificación y ordenación del presupuesto de egresos las cuales son actividades que pertenecen a todos los elementos. La determinación de los campos se hizo en base a las entrevistas realizadas.

Los elementos exclusivos o particulares son aquellas características propias del reactivo que fueron analizados en la etapa del análisis.

1) De A continuación se define los campos comunes a todos los elementos.

Elemento	Descripción
Partida de Gasto	En lo que se va a gastar
Clave del Material	Tipo de Material
U.P.P.	Unidad Programática Presupuestal
Dependencia	Subunidad perteneciente a la U.P.P.
Tipo de Documento 7.- Número	La forma que se utiliza para realizar una erogación
Dígito Verificador 2) Oficina de Modificación Presupuestaria 1.- Beneficiario 2.- Importe 3.- Tipo de Operación	Un dígito que permite el registro confiable de la información, el cual se calcula en la Dirección de Informática y Estadística.
Fecha de Expedición	Fecha en la cual se realiza el documento
Número Secuencial 6.- Mes Final	Es el número de movimientos que se realizan
Concepto	Descripción del gasto que se va a realizar
Número de Solicitud	Folio del documento
Dígito Identificador 2.- Importe 3.- Tipo de Operación	Un dígito que permite la identificación de recursos.
Clave Presupuestal 4.- Mes 4) Regulación	UPP, Dependencia, Programa, Subprograma, Proyecto, Obra, Partida, Dígito Identificador, Dígito Verificador.

Elementos Particulares

Se determinarán los elementos necesarios para la definición y aplicación de los elementos (estructuras de datos).

Por lo que se refiere a la definición de los elementos se hará una sola vez para cada reactivo diferentes. Se capturarán las claves de los elementos comunes a los elementos (U.P.P., Dependencia, Clave del Material, Clave Presupuestal), si ya existen y por el contrario, no se podrán dar de alta.

1) Documento de Ejecución Presupuestaria

1.-Importe

2.-Beneficiario

3.-Cargo Presupuestal

4.-Deducciones

5.-Liquido

6.-Tipo de Justificación

7.-Número

2) Oficio de Modificación Presupuestaria

1.-Beneficiario

2.-Importe

3.-Tipo de Operación

4.-Tipo de Movimiento

5.-Mes Inicial

6.-Mes Final

CAMPO: CLAVE PRESUPUESTARIA**3) Documento Múltiple**

1.-Beneficiario

2.-Importe

3.-Tipo de Operación

4.-Mes

4) Requisición

1.-Costo

2.-Importe

3.-Cantidad

4.-Descripción

5.-Tipo

5) Salidas de Almacén

1.-Costo

- 2.-Importe
- 3.-Cantidad
- 4.-Descripción
- 5.-Tipo

Dígitos de Verificación

Dos de los errores más comunes en el manejo de datos se presentan cuando los datos son capturados de forma incorrecta, estos errores se conocen como errores de transcripción.

La posibilidad de que estos errores ocurran al momento de capturar un clave presupuestal o un número de solicitud para realizar una erogación es muy alta. Actualmente existe un procedimiento manual para realizar dicho calculo por lo que se tomará dicho procedimiento para realizarlo en forma automatizada,

Procedimiento para el calculo del dígito verificador

CAMPO: CLAVE PRESUPUESTARIA

u.p.p.	dep	prog	subp	proy	obra	partida	subr
07	02	01	01	00	00	2605	3

CALCULO:

DISEÑO DE LAS BASES DE DATOS

$$3*2=6$$

$$5*3=15$$

$$\text{SUMA}=148$$

$$0*4=0$$

$$148/11=13$$

$$6*5=30$$

RESIDUO 5

$$2*6=12$$

DÍGITO 11-5

$$0*7=0$$

RESIDUO

$$0*8=0$$

DÍGITO 11-5

$$0*9=0$$

DÍGITO 6

$$0*2=0$$

$$1*3=3$$

$$0*4=0$$

DICCIONARIO DE DATOS

$$1*5=5$$

$$0*6=0$$

$$2*7=14$$

$$0*8=0$$

$$7*9=63$$

Base de Datos del Catálogo General (Grn) + Año.Dbf)

Descripción

La información que contiene esta base de datos es referente a la conformación de un catálogo de U.P.F. y Dependencia, Programa y Subprograma y finalmente Proyecto y Obra.

La estructura para formar un catálogo general resultó a partir de los resultados del análisis de requerimientos, en el cual se determinó que sería útil crear una sola base de datos, en lugar de tres, siendo que los campos mencionados en líneas anteriores tiene la

DISEÑO DE LAS BASES DE DATOS

Se puede decir que el diseño de datos es la actividad más importante del diseño ya que tiene una gran influencia en la calidad del software.

El diseño de las bases de datos consiste en determinar qué información requiere ser almacenada de forma permanente, los tipos de datos que contiene esta información y la longitud o tamaño de esos datos. Por lo tanto es importante forma estructuras o tablas que contengan los grupos de datos pertenecientes al mismo contexto de esta manera la duplicidad de información se evita y la introducción de datos se simplifica.

A continuación se presenta una descripción de cada una de las bases de datos que se diseñaron, sus estructuras, sus índices y las relaciones que mantienen con otros archivos.

DICCIONARIO DE DATOS

Para la creación de las bases de datos fue necesario crear un archivo único por año, dado que la asignación del Presupuesto de Egresos por año varía, de tal forma ante esta situación se construye una base de datos para cada año.

Base de Datos del Catálogo General (Gral + Año.Dbf)

Descripción

La información que contiene esta base de datos es referente a la conformación de un catálogo de U.P.P. y Dependencia, Programa y Subprograma y finalmente Proyecto y Obra.

La estructura para formar un catálogo general resultó a partir de los resultados del análisis de requerimientos, en el cual se determinó que sería útil crear una sola base de datos, en lugar de tres, siendo que los campos mencionados en líneas anteriores tiene la

misma longitud en clave y descripción por lo que para identificar el tipo de catálogo se requirió de un campo identificador.

Estructura de Datos

Campo	Tipo	Longitud	Decimal	Descripción
Tipo	Carácter	1	0	Identificador del catálogo
Clave	Carácter	2	0	Clave General
Subclave	Carácter	2	0	Subclave General
Descripción	Carácter	50	0	Descripción General

Índices

Se creó un índice con la siguiente expresión: "Tipo+Clave+Subclave", con el objeto de identificar las descripciones de las claves y subclaves. No podrán existir identificadores repetidos más sin embargo si claves y subclaves.

Relación con otras Bases de Datos

Base de Datos	Llave	Relación
Ejec + Año	Tipo + Clave + Subclave	U.P.P. y Dependencia
Ofic + Año	Tipo + Clave + Subclave	U.P.P. y Dependencia
Mulc + Año	Tipo + Clave + Subclave	U.P.P. y Dependencia
Reqc + Año	Tipo + Clave + Subclave	U.P.P. y Dependencia
Almc + Año	Tipo + Clave + Subclave	U.P.P. y Dependencia

Base de Datos del Catálogo de Partidas de Gasto (Part + Año.Dbf)

Estructura de Datos

Descripción

Esta base de datos contiene la clave de las partidas de gasto presupuestal, así como su descripción corta y descripción larga.

Campo	Tipo	Longitud	Decimal	Descripción
Material	Carácter	40	0	Descripción del Material
Forma	Carácter	5	0	Forma del Material

Estructura de Datos

Campo	Tipo	Longitud	Decimal	Descripción
Partida	Carácter	4	0	Partida de Gasto Presupuestal
Desc_Corta	Carácter	40	0	Descripción Corta de la Partida
Desc_Larga	Carácter	80	0	Descripción Larga de la Partida

Índices

Se construyó un índice con la siguiente expresión: "Partida", con el objeto de identificar su descripción. No podrán existir claves de partidas repetidas.

Relación con otras Bases de Datos

Base de Datos	Llave	Relación
Ejec + Año	Partida	Beneficiario
Ofic + Año	Partida	Beneficiario
Mulc + Año	Partida	Beneficiario
Reqc + Año	Partida	Partida de Gasto
Almc + Año	Partida	Partida de Gasto

Base de Datos del Catálogo de Material (Mate+ Año.Dbf)**Descripción**

Esta base de datos contiene la clave de los materiales existentes en Recursos Materiales, así como su costo, descripción y tipo.

Estructura de Datos

Campo	Tipo	Longitud	Decimal	Descripción
Material	Carácter	7	0	Clave del Material
Costo	Numérico	7	2	Precio del Material
Descrip	Carácter	40	0	Descripción del Material
Tipo	Carácter	5	0	Forma del Material

Índices con otras Bases de Datos

Se construyó un índice con la siguiente expresión: "Material", con el objeto de identificar su costo, descripción y tipo. No podrán existir claves de materiales repetidos.

Relación con otras Bases de Datos

Base de Datos	Llave	Relación
Reqc + Año	Material	Material
Almc + Año	Material	Material

Base de Datos de Distribución (Dist + Año.Dbf)

Descripción

La información que contiene esta base de datos es la concerniente a la distribución mensual asignada del presupuesto de egresos para cada Unidad Programática Presupuestal. La estructura fue formada a partir de los resultados del análisis de requerimientos.

Campo	Tipo	Longitud	Decimal	Descripción
Temp_cv	Carácter	18	0	Clave Presupuestal
Asignacion	Numérico	14	2	Cantidad monetaria inicial global
Tipo_Cal	Carácter	1	0	Tipo de Calendario
Fecha	Date	8	0	Fecha del movimiento
Tipo_Oper	Carácter	2	0	Tipo de operación
Mes	Numérico	2	0	Mes del movimiento
Total	Numérico	14	2	Cantidad monetaria por mes
Porcentaje	Numérico	6	2	Porcentaje de una cantidad monetaria

Índices

El índice para esta base de datos es mediante de la siguientes expresión: "Temp_cv, con el objeto de identificar que clave presupuestal se creo y cual va a ser su movimiento a lo largo del ejercicio fiscal por medio de la fecha.

Relación con otras Bases de Datos

Base de Datos	Llave	Relación
Asig + Año	Temp_cv + Dtos(Fecha)	Clave Presupuestal
Ofim + Año	Temp_cv	Clave Presupuestal
Ejem + Año	Temp_cv	Clave Presupuestal
Mulm + Año	Temp_cv	Clave Presupuestal
Reqm + Año	Temp_cv	Clave Presupuestal

*Base de Datos de Movimientos (Asig + Año.Dbf)***Descripción**

La información que contiene esta base de datos es el registro de los movimientos que van realizado cada U.P.P. por medio de su clave presupuestal.

Campo	Tipo	Longitud	Decimal	Descripción
No_Solic	Numérico	5	0	Número de Solicitud
Temp_cv	Carácter	18	0	Clave Presupuestal
Tipo_Oper	Carácter	2	0	Tipo de Operación
Asignacion	Numérico	14	2	Cantidad monetaria inicial global
Asig_Parci	Numérico	14	2	Cantidad monetaria mensual
Fecha_Mov	Date	8	0	Fecha del movimiento
Cargo	Numérico	14	2	Cantidad monetaria ejercida
Abono	Numérico	14	2	Cantidad monetaria agregada
Amplia	Numérico	14	2	Cantidad monetaria agregada a la asignación inicial
Reduce	Numérico	14	2	Cantidad monetaria reducida
Asig_Net	Numérico	14	2	Cantidad monetaria final global
Comprom	Numérico	14	2	Cantidad monetaria a gastar
Total	Numérico	14	2	Cantidad monetaria disponible

Índices con otras Bases de Datos

El índice para esta base de datos es por medio de la siguientes expresión: "Temp_cv+Dtos(Fecha), con el objeto de identificar fácilmente los movimientos que va registrándose en cada clave presupuestal.

Cuerpo (Ejec + Año. Dbf)

Relación con otras Bases de Datos

Ninguna

Base de Datos del Documento de Ejecución Presupuestaria

Para el almacenamiento de la información del Documento de Ejecución Presupuestaria se dividió en dos bases de datos: Encabezado y Cuerpo. La primera contiene la identificación de los movimientos realizados por el Documento de Ejecución Presupuestaria y el segundo contiene los movimientos realizados en dicho documento.

Encabezado (Ejec + Año. Dbf)

Campo	Tipo	Longitud	Decimal	Descripción
Depend	Carácter	2	0	Dependencia
UPP	Carácter	2	0	Unidad Programática Presupuestal
Tipo_Docto	Carácter	2	0	Tipo de Documento
No_Solic	Numérico	5	0	Número de Solicitud
Digito	Carácter	1	0	Dígito Verificador del Documento
Fecha_Exp	Date	8	0	Fecha de Expedición
Importe	Numérico	14	2	Cantidad monetaria a gastar.

Relación con otras Bases de Datos

Índices

El índice para esta base de datos es por medio de la siguientes expresión: "No_Solic", con el objeto de identificar, mediante le número de solicitud, fácilmente los movimientos que van registrándose.

Relación con otras Bases de Datos

Base de Datos	Llave	Relación
Ejem + Año	No_Solic	Número de Solicitud

Cuerpo (Ejem + Año.Dbf) movimientos realizados por el Oficio de Modificación Presupuestaria y al segundo contiene los movimientos realizados en dicho documento.

Campo	Tipo	Longitud	Decimal	Descripción
No_Solic	Numérico	5	0	Número de Solicitud
No_Sec	Carácter	2	0	Número Secuencial del movimiento
Clave_Pres	Carácter	18	0	Clave Presupuestal
Beneficiario	Carácter	30	0	Beneficiario
Concepto	Carácter	30	0	Concepto
Cargo	Numérico	14	2	Cargo Presupuestal
Deducción	Numérico	14	2	Deducción Presupuestal
Total	Numérico	14	2	Total del Gasto Presupuestal
Tipo	Carácter	2	0	Tipo de Justificación
Número	Numérico	7	0	Número del comprobante
Dig_Verif	Carácter	1	0	Dígito Verificador de la Clave Presupuestal

Índices

El índice para esta base de datos es mediante de la siguientes expresión: "No_Solic", con el objeto de identificar cual es el encabezado que corresponde con los movimientos que van registrándose.

Relación con otras Bases de Datos

Base de Datos	Llave	Relación
Asig + Año	Temp_cv	Clave Presupuestal
Ejec + Año	No_Solic	Número de Solicitud

Base de Datos del Oficio de Modificación Presupuestaria

Al igual que el Documento de Ejecución Presupuestaria el almacenamiento de la información se dividió en dos bases de datos: Encabezado y Cuerpo. La primera contiene la identificación de los movimientos realizados por el Oficio de Modificación Presupuestaria y el segundo contiene los movimientos realizados en dicho documento.

Encabezado (Ofic + Año. Dbf)

Campo	Tipo	Longitud	Decimal	Descripción
Depend	Carácter	2	0	Dependencia
UPP	Carácter	2	0	Unidad Programática Presupuestal
Tipo_Docto	Carácter	2	0	Tipo de Documento
No_Solic	Numérico	5	0	Número de Solicitud
Digito	Carácter	1	0	Dígito Verificador del Documento
Fecha_Exp	Date	8	0	Fecha de Expedición

Índices

El índice para esta base de datos es por medio de la siguientes expresión: "No_Solic", con el objeto de identificar, mediante le número de solicitud, fácilmente los movimientos que van registrándose.

Relación con otras Bases de Datos

Base de Datos	Llave	Relación
Ofim + Año	No_Solic	Número de Solicitud

Cuerpo (Ofim + Año.Dbf)

Campo	Tipo	Longitud	Decimal	Descripción
No_Solic	Numérico	5	0	Número de Solicitud
No_Sec	Carácter	2	0	Número Secuencial del movimiento
Clave_Pres	Carácter	18	0	Clave Presupuestal

Tipo_Doclo	Carácter	1	0	Tipo de movimiento
Concepto	Carácter	30	0	Concepto de Solicitud
Tipo_Oper	Carácter	2	0	Tipo de Operación del movimiento de
Importe	Numérico	14	2	Importe
Mes_Ini	Carácter	2	0	Mes Inicial
Mes_Fin	Carácter	2	0	Mes Final
Dig_Verif	Carácter	1	0	Dígito Verificador de la Clave Presupuestal

Índices

El índice para esta base de datos es mediante de la siguientes expresión: "No_Solic", con el objeto de identificar cual es el encabezado que corresponde con los movimientos que van registrándose.

Relación con otras Bases de Datos

Base de Datos	Llave	Relación
Asig + Año	Temp_cv	Clave Presupuestal
Ofic + Año	No_Solic	Número de Solicitud

Base de Datos del Documento Múltiple

Al igual que los documentos anteriores, el almacenamiento de la información se dividió en dos bases de datos: Encabezado y Cuerpo. La primera contiene la identificación de los movimientos realizados por el Documento Múltiple y el segundo contiene los movimientos realizados en dicho documento.

Encabezado (Mulc + Año. Dbf)

Campo	Tipo	Longitud	Decimal	Descripción
Solic	Numérico	5	0	Número de Solicitud del Documento de Ejecución
UPP	Carácter	2	0	Unidad Programática Presupuestal

Tipo_Docto	Carácter	2	0	Tipo de Documento
No_Solic	Numérico	5	0	Número de Solicitud
Dig_Ej	Carácter	1	0	Dígito Verificador del Documento de Ejecución.
Digito	Carácter	1	0	Dígito Verificador del Documento
Fecha_Exp	Date	8	0	Fecha de Expedición

Índices

El índice para esta base de datos es por medio de la siguientes expresión: "No_Solic", con el objeto de identificar, mediante le número de solicitud, fácilmente los movimientos que van registrándose.

Relación con otras Bases de Datos

Base de Datos	Llave	Relación
Mulm + Año	No_Solic	Número de Solicitud

Cuerpo (Mulm + Año.Dbf)

Campo	Tipo	Longitud	Decimal	Descripción
No_Solic	Numérico	5	0	Número de Solicitud
No_Sec	Carácter	2	0	Número Secuencial del movimiento
Clave_Pres	Carácter	18	0	Clave Presupuestal
Sec_Cabez	Carácter	2	0	Número Secuencial del Documento de Ejecución
Concepto	Carácter	30	0	Concepto
Tipo_Oper	Carácter	2	0	Tipo de Operación del movimiento
Importe	Numérico	14	2	Importe
Mesi	Carácter	2	0	Mes Modificado
Dig_Verif	Carácter	1	0	Dígito Verificador de la Clave Presupuestal

Índices

El índice para esta base de datos es mediante de la siguientes expresión: "No_Solic", con el objeto de identificar cual es el encabezado que corresponde con los movimientos que van registrándose.

Relación con otras Bases de Datos

Base de Datos	Llave	Relación
Asig + Año	Temp_cv	Clave Presupuestal
Mulc + Año	No_Solic	Número de Solicitud

Base de Datos de Salidas de Almacén.

Al igual que los documentos anteriores, el almacenamiento de la información se dividió en dos bases de datos: Encabezado y Cuerpo. La primera contiene la identificación de los movimientos realizados y el segundo contiene los movimientos realizados en dicho documento. Asimismo, si el tipo de salida fue "A" indica que es una salida de almacén, si fue "D" se realizó una compra directa y si fue "I" se realizó una compra de inventario.

Encabezado (Almc + Año. Dbf)

Campo	Tipo	Longitud	Decimal	Descripción
Depend	Carácter	20	0	Dependencia
UPP	Carácter	2	0	Unidad Programática Presupuestal
Tipo_Sal	Carácter	1	0	Tipo de Salida
No_Solic	Numérico	5	0	Número de Solicitud
Persona	Carácter	35	0	Nombre de la persona
Digito	Carácter	1	0	Dígito Verificador del Documento
Clave_Pres	Carácter	18	0	Clave Presupuestal
Requis	Numérico	5	0	Número de Requisición
Req_Tipo	Carácter	1	0	Tipo de Requisición
Fecha_Exp	Date	8	0	Fecha de Expedición

Índices Datos de Requisiciones

El índice para esta base de datos es mediante de la siguientes expresión: "Ident+Str(No_Solic)+Tipo_Sal", con el objeto de identificar el tipo del documento, su número de solicitud y finalmente el tipo de salida del documento.

Relación con otras Bases de Datos

Base de Datos	Llave	Relación
Almm + Año	Str(No_Solic) + Tipo Sal	Identificación, Número de Solicitud y Tipo de Salida

Cuerpo (Almm + Año.Dbf)

Campo	Tipo	Longitud	Decimal	Descripción
No_Solic	Numérico	5	0	Número de Solicitud
Tipo_Sal	Carácter	1	0	Tipo de Salida
Cantidad	Numérico	3	0	Cantidad
Clave	Carácter	7	0	Clave del artículo
Articulo	Carácter	40	0	Descripción del artículo
Importe	Numérico	9	2	Importe
Tipo	Carácter	5	0	Descripción de la forma
Costo_Exp	Numérico	7	2	Precio del artículo

Índices

El índice para esta base de datos es por medio de la siguientes expresión: "Str(No_Solic)+Tipo_Sal", con el objeto de identificar cual es el encabezado que corresponde con los movimientos que van registrándose.

Relación con otras Bases de Datos

Base de Datos	Llave	Relación
Asig + Año	Temp_cv	Clave Presupuestal
Almc + Año	Str(No_Solic) + Tipo Sal	Identificación, Número de Solicitud y Tipo de Salida

Base de Datos de Requisiciones

Al igual que los documentos anteriores, el almacenamiento de la información se dividió en dos bases de datos: Encabezado y Cuerpo. La primera contiene la identificación de los movimientos realizados y el segundo contiene los movimientos realizados en dicho documento. Asimismo, si el tipo de salida fue "A" indica que es una salida de almacén, si fue "D" se realizó una compra directa y si fue "I" fue una compra en bienes inmuebles (ojo)

Encabezado (Reqc + Año. Dbf)

Campo	Tipo	Longitud	Decimal	Descripción
Band	Carácter	1	0	Afectación de la Requisición
Depend	Carácter	20	0	Dependencia
UPP	Carácter	2	0	Unidad Programática Presupuestal
Req_Tipo	Carácter	1	0	Tipo de Requisición
No_Solic	Numérico	5	0	Número de Solicitud
Digito	Carácter	1	0	Dígito Verificador del Documento
Clave_Pres	Carácter	18	0	Clave Presupuestal
Total	Numérico	9	2	Importe
Fecha_Exp	Date	8	0	Fecha de Expedición

Índices

El índice para esta base de datos es mediante de la siguientes expresión: "Str(No_Solic)+Req_Tipo", con el objeto de identificar, su número de solicitud y el tipo de requisición.

Relación con otras Bases de Datos

Base de Datos	Llave	Relación
Reqm + Año	Str(No_Solic)+ Req_Tipo	Número de Solicitud y Tipo de Requisición

Cuerpo (Reqm + Año.Dbf)

DISEÑO LOGICO

Campo	Tipo	Longitud	Decimal	Descripción
No_Solic	Numérico	5	0	Número de Solicitud
Req_Tipo	Carácter	1	0	Tipo de Requisición
Cantidad	Numérico	3	0	Cantidad
Clave	Carácter	7	0	Clave del artículo
Descrip	Carácter	40	0	Descripción del artículo
Importe	Numérico	9	2	Importe
Tipo	Carácter	5	0	Descripción de la forma
Band	Carácter	1	0	Afectación de la Requisición
No	Carácter	2	0	Número Secuencial
Costo	Numérico	7	2	Precio del artículo

Índices

El índice para esta base de datos es mediante de la siguientes expresión: "Str(No_Solic)+Req_Tipo", con el objeto de identificar cual es el encabezado que corresponde con los movimientos que van registrándose.

Relación con otras Bases de Datos

Base de Datos	Llave	Relación
Asig + Año	Temp_cv	Clave Presupuestal
Reqc + Año	Str(No_Solic) +Req_Tipo	Número de Solicitud y Tipo de Requisición

que se realicen las tareas esenciales más eficientemente son:

1. Definir con mayor precisión los requerimientos del usuario en la fase del desarrollo del sistema
2. Preparar lo mejor posible la documentación del sistema.
3. Usar métodos más efectivos para el diseño de la lógica del procedimiento
4. Hacer un mejor uso de las herramientas y técnicas existentes.

DISEÑO LOGICO

El diseño lógico o diseño del sistema producen los detalles (entradas, salidas, archivos, bases de datos y procedimientos) que establecen la forma en que el sistema cumplirá con los requerimientos identificados durante la etapa del análisis.

Los dos objetivos operacionales que siempre se buscan en el diseño son: la confiabilidad y el mantenimiento del sistema.

Confiabilidad

Se dice que un sistema tiene confiabilidad si no produce fallas costosas o peligrosas al usarse de manera razonable. Existen dos niveles de confiabilidad:

- a) El sistema debe cumplir con los requerimientos especificados por el usuarios.
- b) El sistema debe producir los resultados que requiere el usuario.

Mantenimiento del sistema

Cuando se instalan los sistemas, generalmente se usan por periodos largos. La vida promedio de un sistema es de 4 a 6 años. Sin embargo este tiempo de uso trae consigo la necesidad de mantener continuamente el sistema.

Las claves para reducir la necesidad de mantenimiento, al igual que hacer posible que se realicen las tareas esenciales más eficientemente son:

1. Definir con mayor precisión los requerimientos del usuario en la fase del desarrollo del sistema
2. Preparar lo mejor posible la documentación del sistema.
3. Usar métodos más efectivos para el diseño de la lógica del procedimiento.
4. Hacer un mejor uso de las herramientas y técnicas existentes.

5. Realizar un diseño modular.

Las prácticas de diseño que se siguen para el software afectan en forma determinante la facilidad de mantenimiento de un sistema. Un software modular bien diseñado tiene grandes probabilidades de cumplir con los requerimientos de facilidad y confiabilidad.

Los principios que deben guiar el diseño de software son:

Principio	Descripción	Objetivos
Modularidad y Fragmentación	Diseño de un sistema como jerarquía de módulos	Diseñar la estructura en forma descendente con módulos que realicen funciones específicas.
Acoplamiento	La fuerza de las relaciones entre módulos	Maximizar la independencia entre los módulos minimizando el acoplamiento
Cohesión (integración)	La fuerza de las relaciones dentro de un módulo	Maximizar la cohesión; los elementos altamente relacionados deben estar en el mismo módulo
Extensión de control	Número de módulos subordinados al módulo que hace la llamada	Limitar la extensión de control de 5 a 7 módulos
Tamaño	Número de instrucciones que componen a un módulo	Limitar el tamaño de forma que la función de todo el módulo se centre en un sólo propósito
Uso compartido	Uso de un módulo por otros módulos	Evitar la duplicación permitiendo que los módulos sean llamadas por otros que necesitan la función de cada uno

Diseño Estructurado

Son tres elementos básicos usados en el desarrollo de los diagramas de flujo.

Proceso: El diseño estructurado es una aproximación modular y descendente a la tarea de desarrollo del software. El sistema entero puede así diseñarse y codificarse por módulos.

Proceso

Como puede notarse las ventajas de la modularidad y el diseño estructurado son claras:

- Un sistema modular se puede comprender mejor y su mantenimiento resulta por tanto más simple; esto es cierto no solo para el analista inicial que tiene que hacer cambios más tarde, sino para otras personas que puedan heredar el programa.
- Es posible dividir programas complejos en elementos más sencillos y manejables
- Se puede crear una biblioteca de módulos y utilizar los módulos en otro programa
- Es más fácil determinar la localización de errores de un programa en un módulo específico.

Iteración

HERRAMIENTAS DEL DISEÑO LÓGICO

mientras exista una condición dada o hasta que se de una condición.

Existen algunas herramientas de diseño que a la vez de ayudar en la tarea del diseño constituyen vías preferentes de comunicación al usuario de los conceptos propios del sistema. A continuación se describen algunas herramientas específicas que se utilizarán para documentar el diseño del software.

de flujo

Diagramas de flujo

Es la forma de representar una sucesión lógica o ilustrar qué proceso lleva a cabo un sistema. Son herramientas gráficas que forzan al diseñador a estructurar software que sea modular y descendente. Proporcionan una estructura a la que se pueden ajustar el desarrollo de software de aplicación.

Elementos Básicos DAS

Son tres elementos básicos usados en el desarrollo de los diagramas de flujo: procesos, decisión e iteración.

Proceso



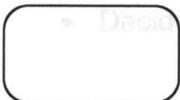
Los procesos o pasos de un programa se representan mediante un rectángulo. Este símbolo representa la iniciación de variables, actividades de entrada y salida y las llamadas para ejecutar otros procedimientos. Un nombre descriptivo breve, escrito dentro del rectángulo, establece el propósito del proceso.

Decisión



El símbolo de decisión representa condiciones alternativas que pueden ocurrir y que el programa debe de poder manejar. Muestran el equivalente de las estructuras IF-THEN-ELSE.

Iteración



El símbolo de iteración representa los ciclos y repetición de operaciones mientras exista una condición dada o hasta que se de una condición.

Conector



El símbolo de conector representa el enlace entre dos o más rutinas también se utiliza para facilitar la lectura y comprensión de los diagramas de flujo

DISEÑO DE SALIDAS

El término de salida se utiliza para denotar cualquier información producida por un sistema de información, ya sea impresa o en una pantalla.

Una de las características más importantes para los usuarios de un sistema de información son las salidas, ya que estas reflejan lo que produce el sistema.

Es importante seguir ciertos principios para su diseño, ya que se pueden presentar salidas confusas o muy complejas lo cual generaría efectos negativos en el sistema. Lo que se pretende es que sea un medio efectivo de comunicación entre el usuario y el sistema.

Para realizar el diseño de una salida se deberá tener en consideración lo siguiente:

- Asignación por Partida Presupuestal. - Permite imprimir el monto asignado a cada dependencia específica.
- Determinar qué información presentar.
- Disponer la presentación de la información en un formato aceptable.
- Decidir si la información será presentada en forma visual, verbal o impresa.

Diseño de la Salida Impresa

Un reporte o informe es el resumen de datos o información que se encuentran almacenados en un base de datos que se muestran según convenga en cada caso en un formato determinado.

Objetivos

- Expresar información relacionada con actividades pasadas, estado actual o proyecciones para el futuro.
- Señalar eventos importantes, oportunidades, problemas o advertencias.
- Iniciar una acción.
- Confirmar una acción.

La determinación de los reportes que estarán disponibles se hizo en base a las necesidades o requerimientos de los usuarios, de tal manera que los reportes provistos son:

- **Servicios Generales por U.P.P. y Dependencia**.- Permite imprimir todos aquellos
- **Catálogo de Partidas Presupuestales**.- Permite imprimir todas aquellas partidas de gasto que tiene el Gobierno del Estado.
- **Catálogo de U.P.P. y Dependencias**.- Permite imprimir todas aquellas entidades Gubernamentales que conforman a la Administración Pública.
- **Catálogo de Programas y Subprogramas**.- Permite imprimir todos aquellos programas y subprogramas de trabajo del Gobierno del Estado.
- **Catálogo de Proyectos y Obras**.- Permite imprimir todos aquellos proyectos y obras de inversión pública.
- **Asignación por U.P.P. y Dependencia**.- Permite imprimir todas las claves presupuestales con su monto por U.P.P. y dependencia en específica.
- **Asignación por Partida Presupuestal**.- Permite imprimir el monto asignado a cada partida de gasto por U.P.P. y Dependencia específica.
- **Movimientos por U.P.P. y Dependencia**.- Permite imprimir todos aquellos movimientos que haya realizado la U.P.P. por medio de su dependencia.
- **Movimientos por Clave Presupuestal**.- Permite imprimir todos aquellos movimientos que ha realizado una clave presupuestal determinada.
- **Ejecución Presupuestaria por U.P.P. y Dependencia**.- Permite imprimir todos aquellos Documentos de Ejecución Presupuestaria que ha utilizado una U.P.P. por medio de su dependencia.
- **Oficio de Modificación por U.P.P. y Dependencia**.- Permite imprimir todos aquellos documentos correspondientes al Oficio de Modificación que ha utilizado una U.P.P. por medio de su dependencia.
- **Documento Múltiple por U.P.P. y Dependencia**.- Permite imprimir todos aquellos documentos correspondientes al Documento Múltiple que ha utilizado una U.P.P. por medio de su dependencia.
- **Comprometido por U.P.P. y Dependencia**.- Permite imprimir todos aquellos documentos correspondientes a las Requisiciones que ha utilizado una U.P.P. por medio de su dependencia.

- ▣ Recursos Materiales por U.P.P. y Dependencia .- Permite imprimir todos aquellos documentos correspondientes a las Salidas de Almacén que ha utilizado una U.P.P. por medio de su dependencia.
- ▣ Servicios Generales por U.P.P. y Dependencia .- Permite imprimir todos aquellos documentos correspondientes Servicios Generales que ha utilizado una U.P.P. por medio de su dependencia.

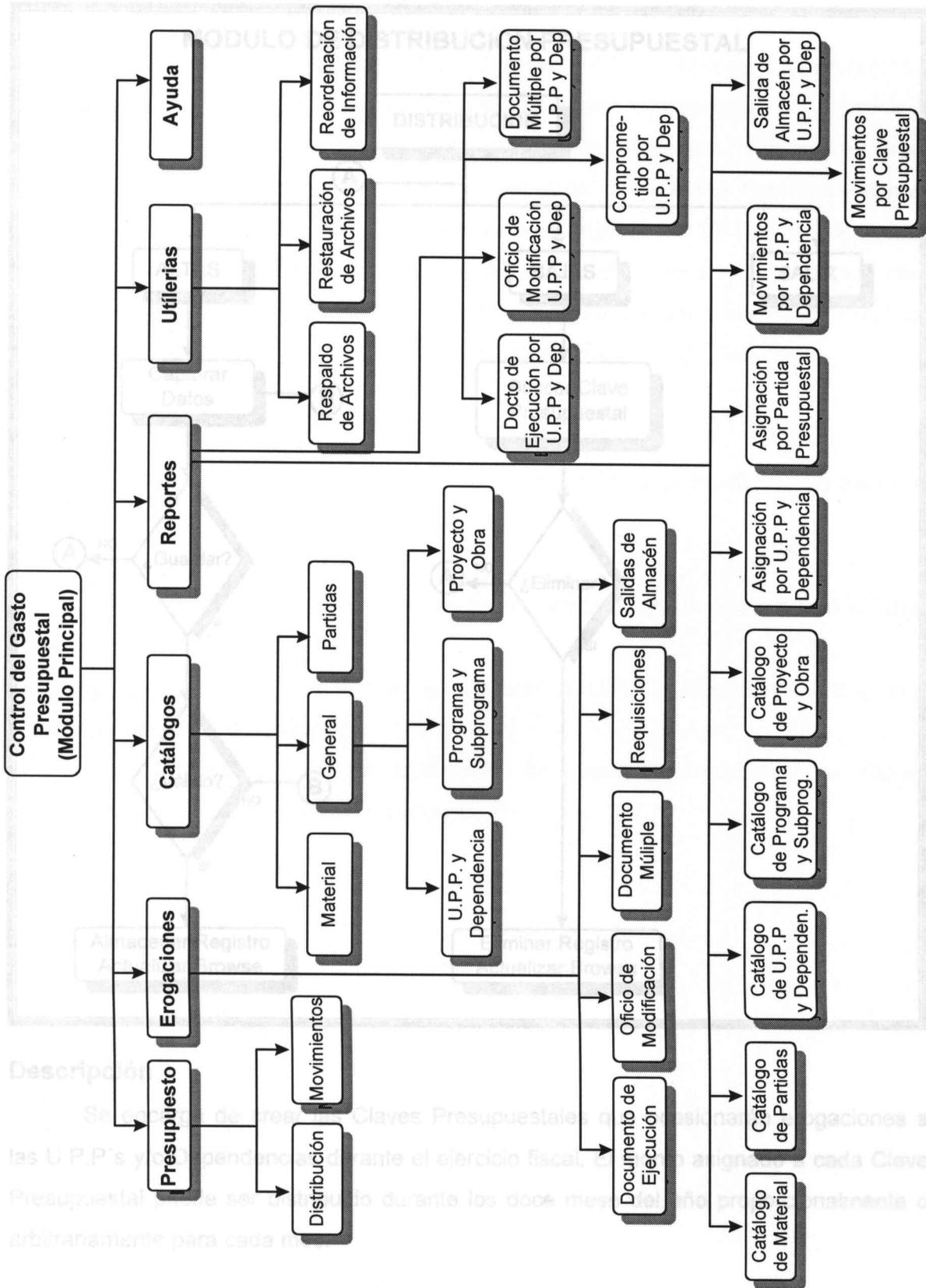
Diseño de la Salida en Pantalla

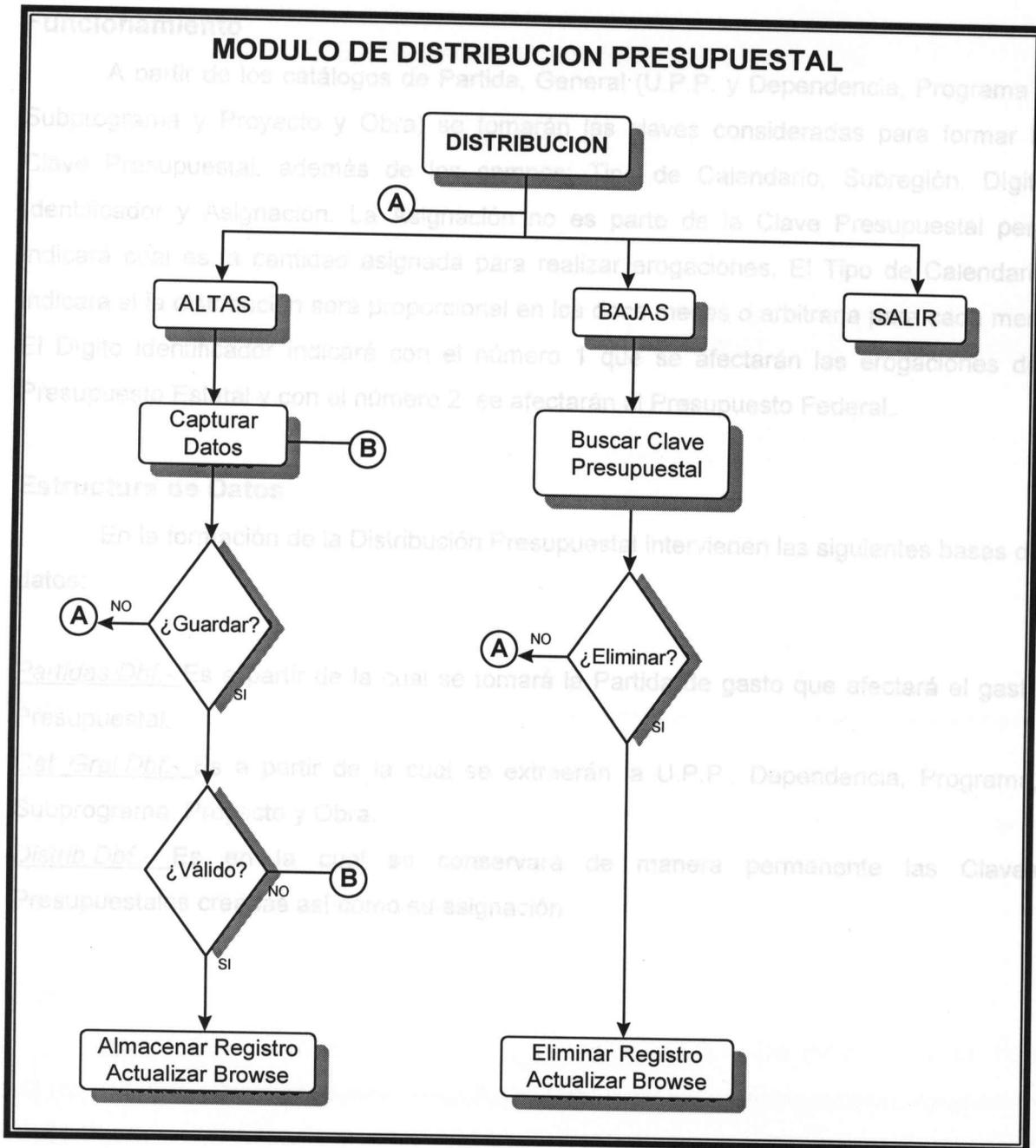
Existen varios principios que se deben de seguir para la elaboración de la salida en pantalla que a continuación se describen:

- ☞ **El sistema debe ser intuitivo.-** Esto significa que debe presentar toda la información que el usuario necesita saber de una manera que la entienda inmediatamente, sin tener que aprender algo nuevo.
- ☞ **Consistencia.-** Todas las ventanas, campos y controles deben comportarse de la misma manera, tanto dentro del sistema como con los estándares externos aceptados, lo cual reducirá la cantidad de información que el usuario nuevo tiene que aprender y por lo tanto la aplicación será más fácil de usar.
- ☞ **Retroalimentación.-** Es importante mantener al usuario bien informado de lo que está pasando en todo momento.
- ☞ **Indulgencia.-** El usuario es susceptible de cometer errores, la interfaz diseñada debe considerarlo y ser tan indulgente como sea posible cuando los errores ocurran.

Módulos del Sistema

Dentro de la etapa del diseño lógico es necesario especificar las operaciones principales que realizará el sistema, por esta razón se presenta a continuación cada módulo, su diagrama de flujo y la descripción del mismo así como su funcionamiento.





Descripción

Se encarga de crear las Claves Presupuestales que ocasionarán erogaciones a las U.P.P's y/o Dependencias durante el ejercicio fiscal. El monto asignado a cada Clave Presupuestal puede ser distribuido durante los doce mese del año proporcionalmente o arbitrariamente para cada mes.

Funcionamiento

A partir de los catálogos de Partida, General (U.P.P. y Dependencia, Programa y Subprograma y Proyecto y Obra) se tomarán las claves consideradas para formar la Clave Presupuestal, además de los campos: Tipo de Calendario, Subregión, Dígito Identificador y Asignación. La Asignación no es parte de la Clave Presupuestal pero indicará cual es la cantidad asignada para realizar erogaciones. El Tipo de Calendario indicará si la distribución será proporcional en los doce meses o arbitraria para cada mes. El Dígito identificador indicará con el número 1 que se afectarán las erogaciones del Presupuesto Estatal y con el número 2 se afectarán al Presupuesto Federal.

Estructura de Datos

En la formación de la Distribución Presupuestal intervienen las siguientes bases de datos:

Partidas.Dbfc. Es a partir de la cual se tomará la Partida de gasto que afectará el gasto Presupuestal.

Cat. Gral.Dbfc. Es a partir de la cual se extraerán la U.P.P., Dependencia, Programa, Subprograma, Proyecto y Obra.

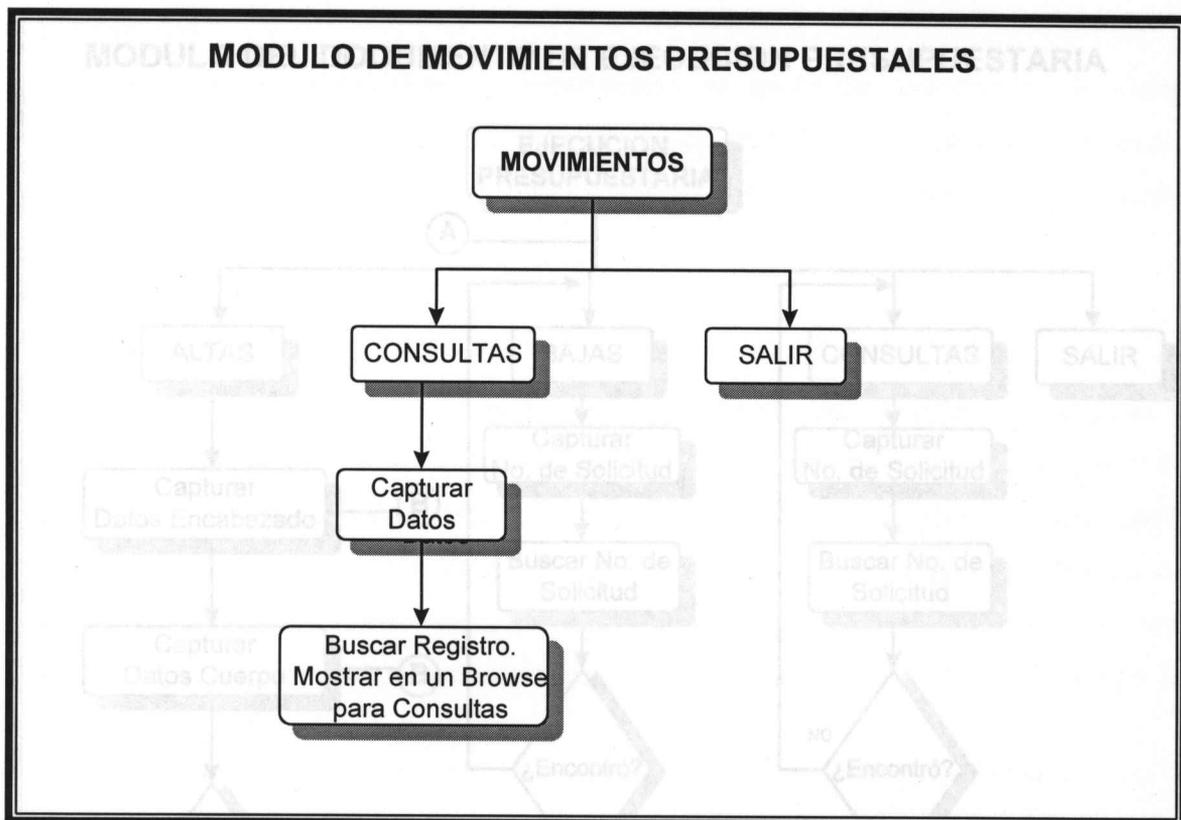
Distrib.Dbfc. Es en la cual se conservará de manera permanente las Claves Presupuestales creadas así como su asignación.

Funcionamiento

A partir de las Claves Presupuestales se registrarán la Fecha del movimiento, Tipo de movimiento, Asignación Inicial, Asignación Parcial, Ampliación, Reducción, Asignación Neta, Abonos, Ejercido y Disponible o Total. En consecuencia el usuario estará informado de manera adecuada que clase de movimiento se realizó.

Estructura de Datos

La única base de datos para conformar los movimientos es Asigna.Dbfc y para las consultas de movimientos en un determinado periodo se crea una base de datos aleatoria con los campos de Fecha de movimiento, Tipo de movimiento, Ejercido, Abono, Disponible.



Descripción

Se encarga de ir registrando los movimientos realizados de cada Clave Presupuestal, derivados por los gastos realizados por la U.P.P. y/o Dependencia. Asimismo permite ver los movimientos efectuados en determinado periodo.

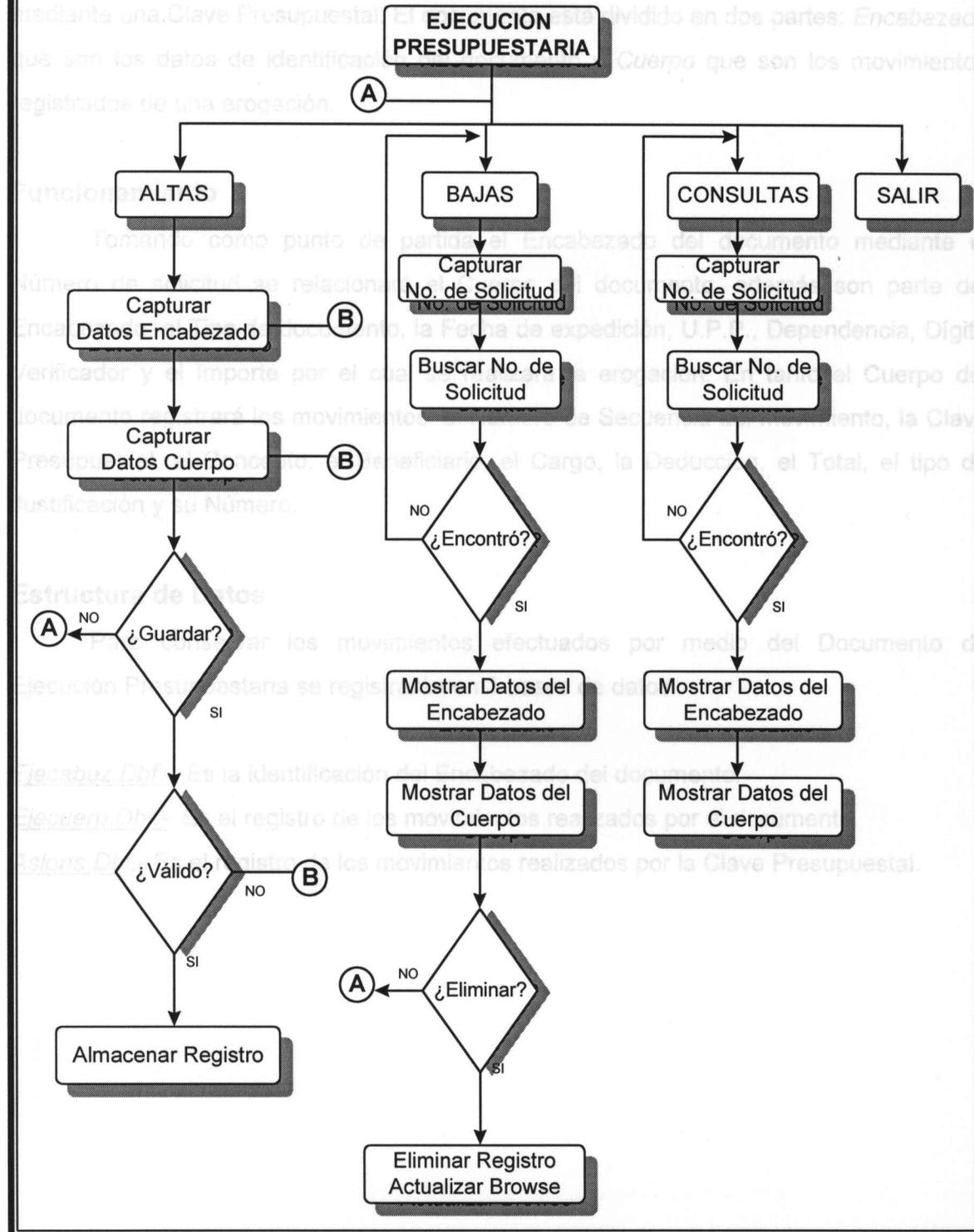
Funcionamiento

A partir de las Claves Presupuestales se registrarán la Fecha del movimiento, Tipo de movimiento, Asignación Inicial, Asignación Parcial, Ampliación, Reducción, Asignación Neta, Abonos, Ejercido y Disponible o Total. En consecuencia el usuario estará informado de manera adecuada que clase de movimiento se realizó.

Estructura de Datos

La única base de datos para conformar los movimientos es Asigna.Dbf y para las consultas de movimientos en un determinado periodo se crea una base de datos aleatoria con los campos de Fecha de movimiento, Tipo de movimiento, Ejercido, Abono, Disponible.

MODULO DEL DOCUMENTO DE EJECUCION PRESUPUESTARIA



Descripción

Se encarga de registrar los movimientos de gasto de una Partida de Gasto mediante una Clave Presupuestal. El documento está dividido en dos partes: *Encabezado* que son los datos de identificación del documento y *Cuerpo* que son los movimientos registrados de una erogación.

Funcionamiento

Tomando como punto de partida el Encabezado del documento mediante el Número de solicitud se relacionará el Cuerpo del documento, además son parte del Encabezado: el Tipo de documento, la Fecha de expedición, U.P.P., Dependencia, Dígito Verificador y el Importe por el cual se realizará la erogación. En tanto el Cuerpo del documento registrará los movimientos: el Número de Secuencia del movimiento, la Clave Presupuestal, el Concepto, el Beneficiario, el Cargo, la Dedución, el Total, el tipo de Justificación y su Número.

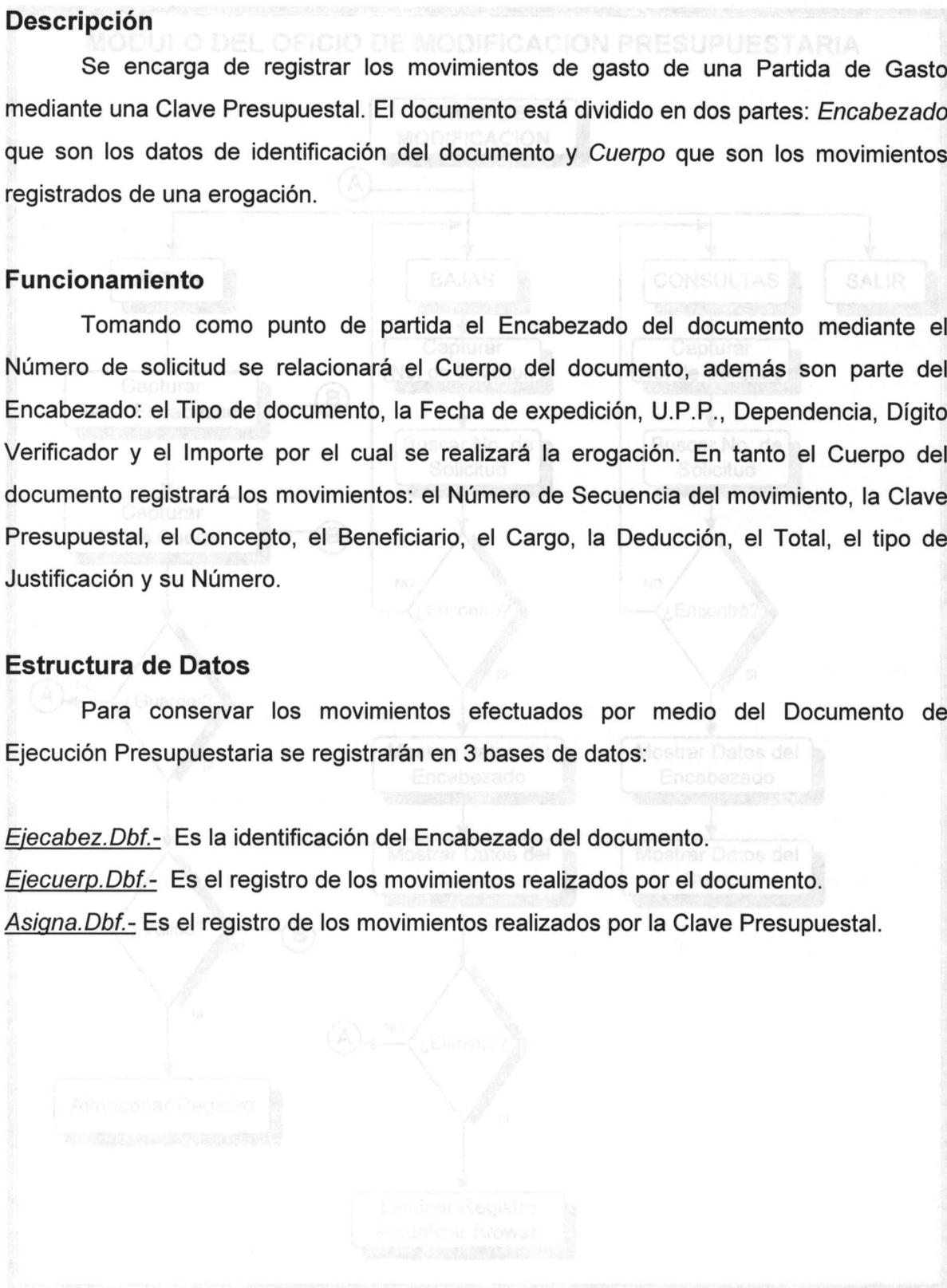
Estructura de Datos

Para conservar los movimientos efectuados por medio del Documento de Ejecución Presupuestaria se registrarán en 3 bases de datos:

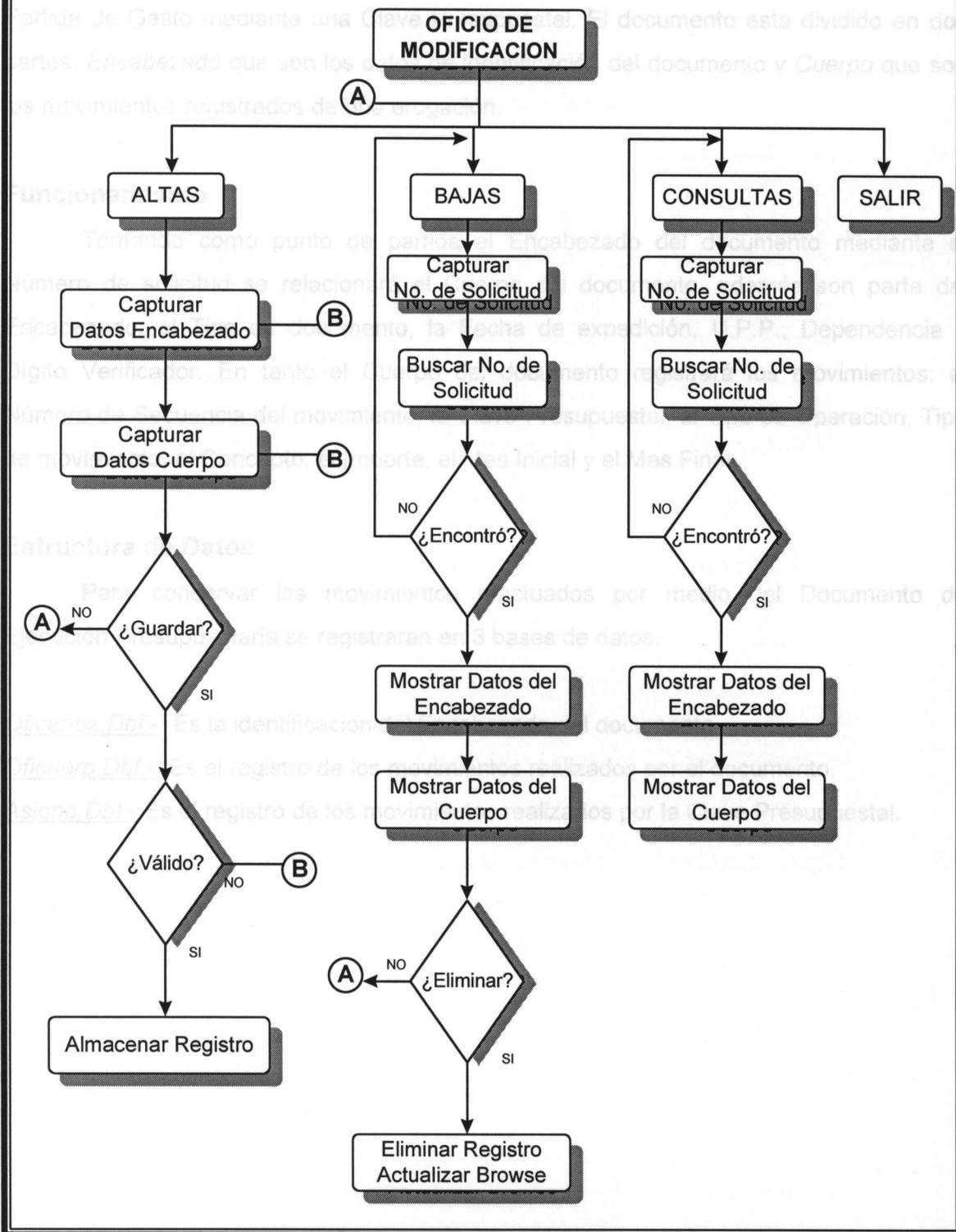
Ejecabez.Dbfc. - Es la identificación del Encabezado del documento.

Ejecuorp.Dbfc. - Es el registro de los movimientos realizados por el documento.

Asigna.Dbfc. - Es el registro de los movimientos realizados por la Clave Presupuestal.



MODULO DEL OFICIO DE MODIFICACION PRESUPUESTARIA



Descripción

Se encarga de registrar los movimientos de ampliación y/o reducción de una Partida de Gasto mediante una Clave Presupuestal. El documento esta dividido en dos partes: *Encabezado* que son los datos de identificación del documento y *Cuerpo* que son los movimientos registrados de una erogación.

Funcionamiento

Tomando como punto de partida el Encabezado del documento mediante el Número de solicitud se relacionará el Cuerpo del documento, además son parte del Encabezado: el Tipo de documento, la Fecha de expedición, U.P.P., Dependencia y Dígito Verificador. En tanto el Cuerpo del documento registrará los movimientos: el Número de Secuencia del movimiento, la Clave Presupuestal, el Tipo de Operación, Tipo de movimiento, el Concepto, el Importe, el Mes Inicial y el Mes Final.

Estructura de Datos

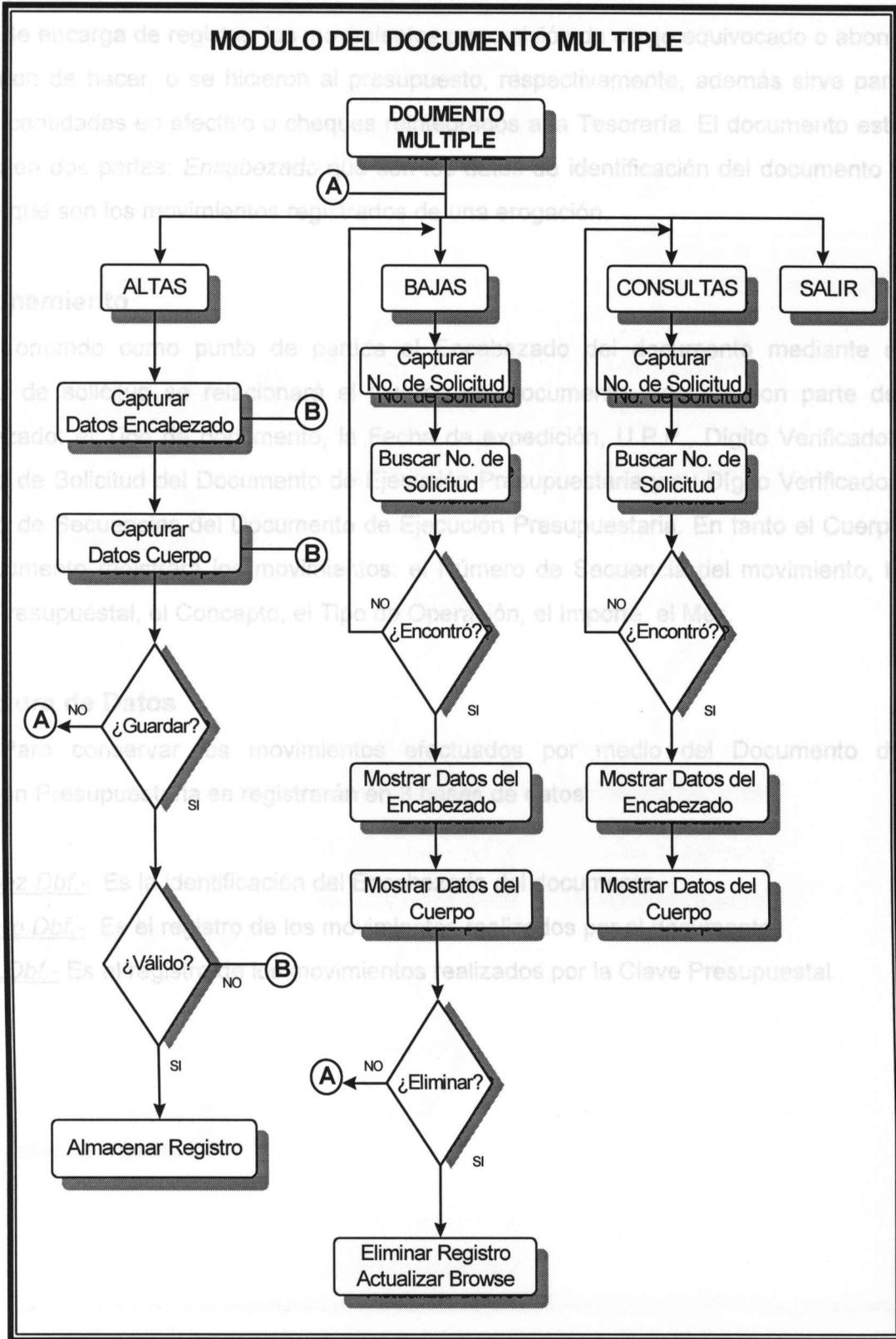
Para conservar los movimientos efectuados por medio del Documento de Ejecución Presupuestaria se registrarán en 3 bases de datos:

Oficabez.Dbf.- Es la identificación del Encabezado del documento.

Oficuerp.Dbf.- Es el registro de los movimientos realizados por el documento.

Asigna.Dbf.- Es el registro de los movimientos realizados por la Clave Presupuestal.

Descripción



Descripción

Se encarga de registrar los movimientos por omisión de cargo equivocado o abono se dejaron de hacer, o se hicieron al presupuesto, respectivamente, además sirve para enterar cantidades en efectivo o cheques reintegrados a la Tesorería. El documento esta dividido en dos partes: *Encabezado* que son los datos de identificación del documento y *Cuerpo* que son los movimientos registrados de una erogación.

Funcionamiento

Tomando como punto de partida el Encabezado del documento mediante el Número de solicitud se relacionará el Cuerpo del documento, además son parte del Encabezado: el Tipo de documento, la Fecha de expedición, U.P.P., Dígito Verificador, Número de Solicitud del Documento de Ejecución Presupuestaria y su Dígito Verificador, Número de Secuencias del Documento de Ejecución Presupuestaria. En tanto el Cuerpo del documento registrará los movimientos: el Número de Secuencia del movimiento, la Clave Presupuestal, el Concepto, el Tipo de Operación, el Importe, el Mes.

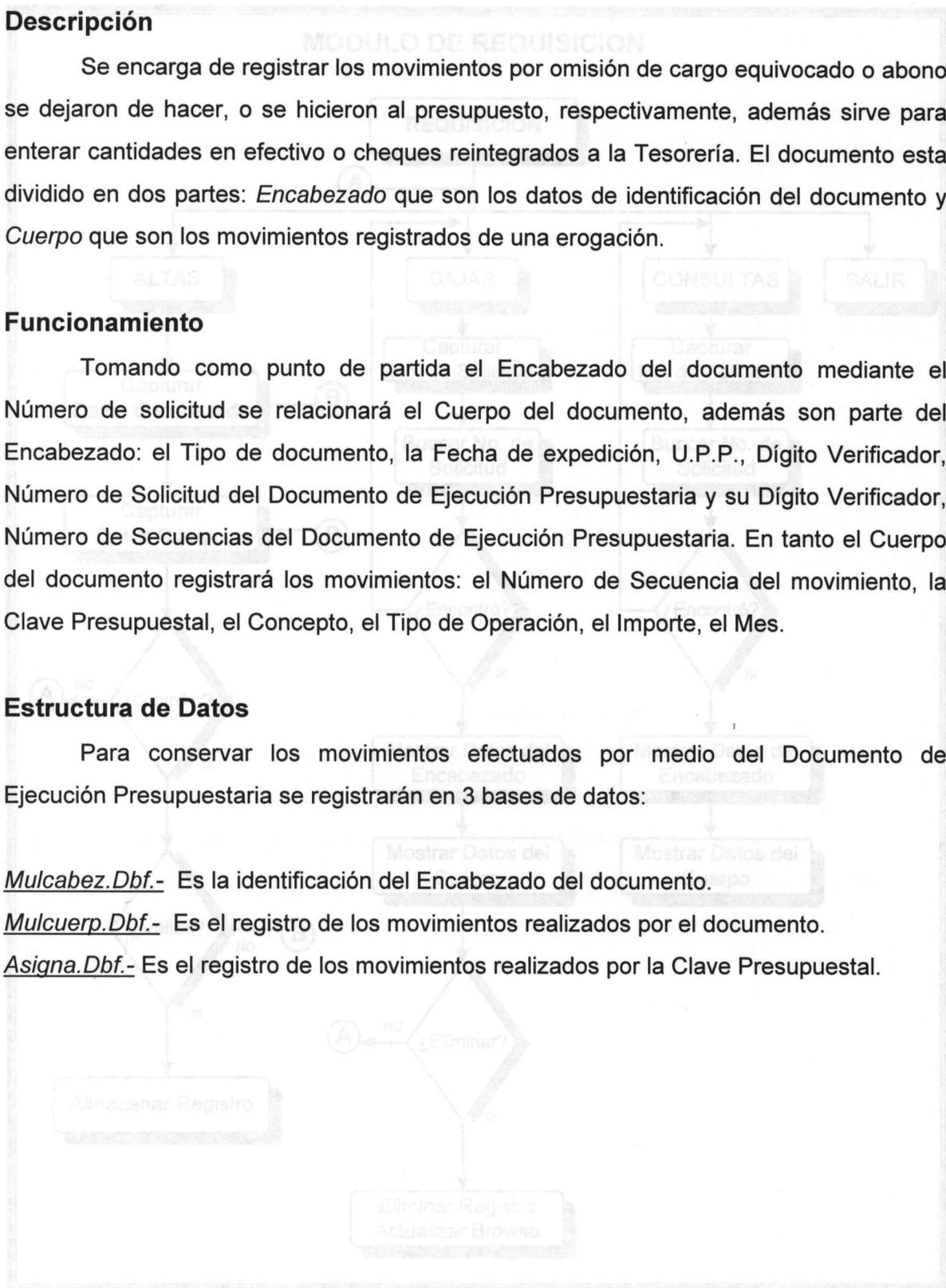
Estructura de Datos

Para conservar los movimientos efectuados por medio del Documento de Ejecución Presupuestaria se registrarán en 3 bases de datos:

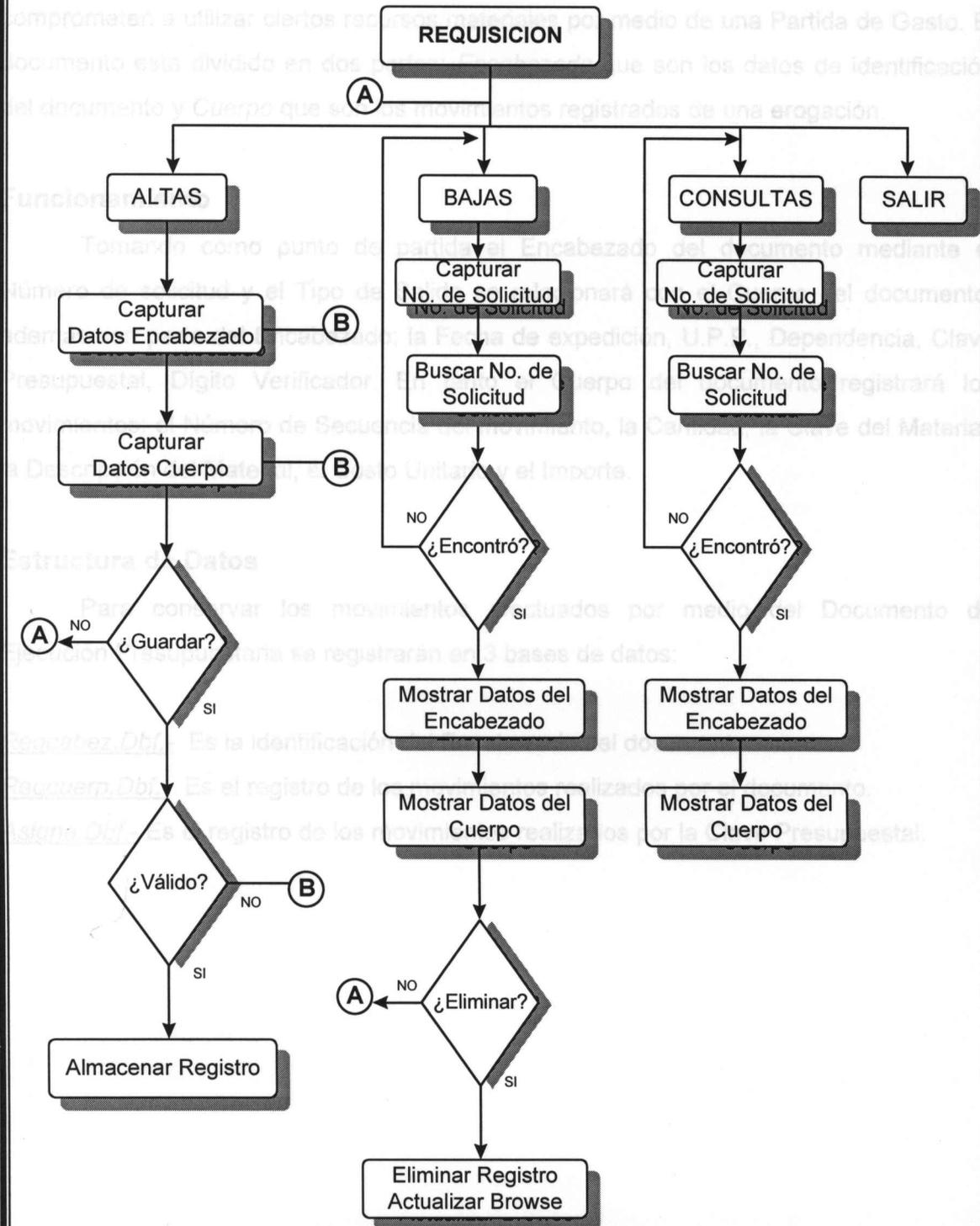
Mulcabez.Dbf.- Es la identificación del Encabezado del documento.

Mulcuerp.Dbf.- Es el registro de los movimientos realizados por el documento.

Asigna.Dbf.- Es el registro de los movimientos realizados por la Clave Presupuestal.



MODULO DE REQUISICION



Descripción

Se encarga de registrar los movimientos en el cual las U.P.P.'s y Dependencias se comprometen a utilizar ciertos recursos materiales por medio de una Partida de Gasto. El documento esta dividido en dos partes: *Encabezado* que son los datos de identificación del documento y *Cuerpo* que son los movimientos registrados de una erogación.

Funcionamiento

Tomando como punto de partida el Encabezado del documento mediante el Número de solicitud y el Tipo de Salida se relacionará con el Cuerpo del documento, además son parte del Encabezado: la Fecha de expedición, U.P.P., Dependencia, Clave Presupuestal, Dígito Verificador. En tanto el Cuerpo del documento registrará los movimientos: el Número de Secuencia del movimiento, la Cantidad, la Clave del Material, la Descripción del Material, el Costo Unitario y el Importe.

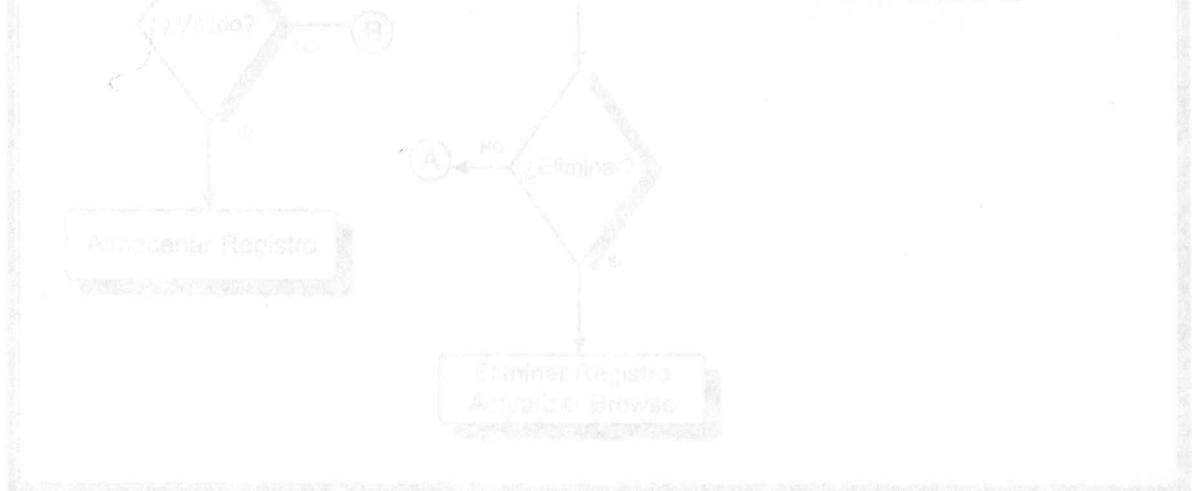
Estructura de Datos

Para conservar los movimientos efectuados por medio del Documento de Ejecución Presupuestaria se registrarán en 3 bases de datos:

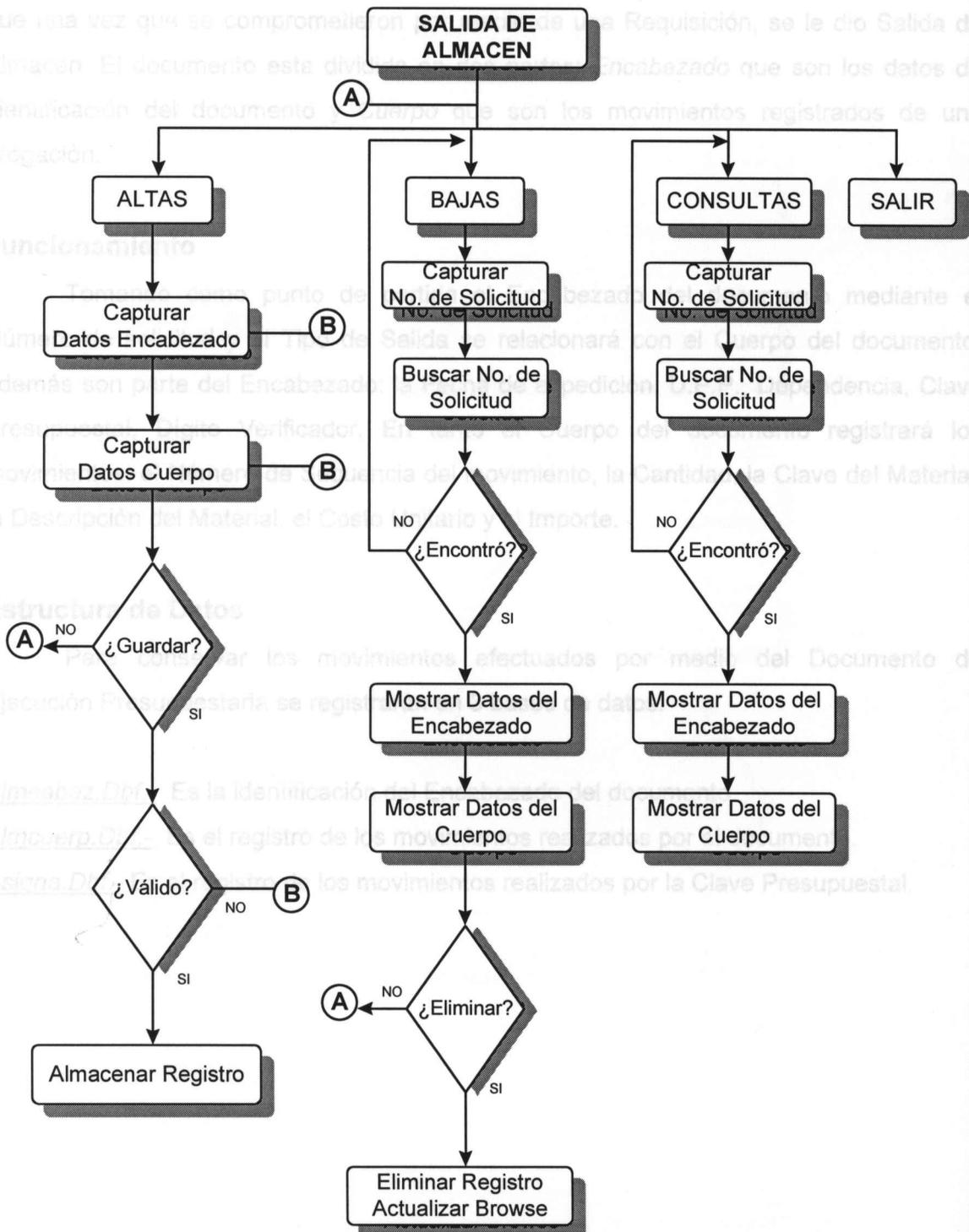
Reqcabez.Dbf.- Es la identificación del Encabezado del documento.

Reqcuerp.Dbf.- Es el registro de los movimientos realizados por el documento.

Asigna.Dbf.- Es el registro de los movimientos realizados por la Clave Presupuestal.



MODULO DE SALIDA DE ALMACEN



Descripción

Se encarga de registrar los movimientos en el cual las U.P.P.'s y Dependencias, que una vez que se comprometieron por medio de una Requisición, se le dio Salida de Almacén. El documento esta dividido en dos partes: *Encabezado* que son los datos de identificación del documento y *Cuerpo* que son los movimientos registrados de una erogación.

Funcionamiento

Tomando como punto de partida el Encabezado del documento mediante el Número de solicitud y el Tipo de Salida se relacionará con el Cuerpo del documento, además son parte del Encabezado: la Fecha de expedición, U.P.P., Dependencia, Clave Presupuestal, Dígito Verificador. En tanto el Cuerpo del documento registrará los movimientos: el Número de Secuencia del movimiento, la Cantidad, la Clave del Material, la Descripción del Material, el Costo Unitario y el Importe.

Estructura de Datos

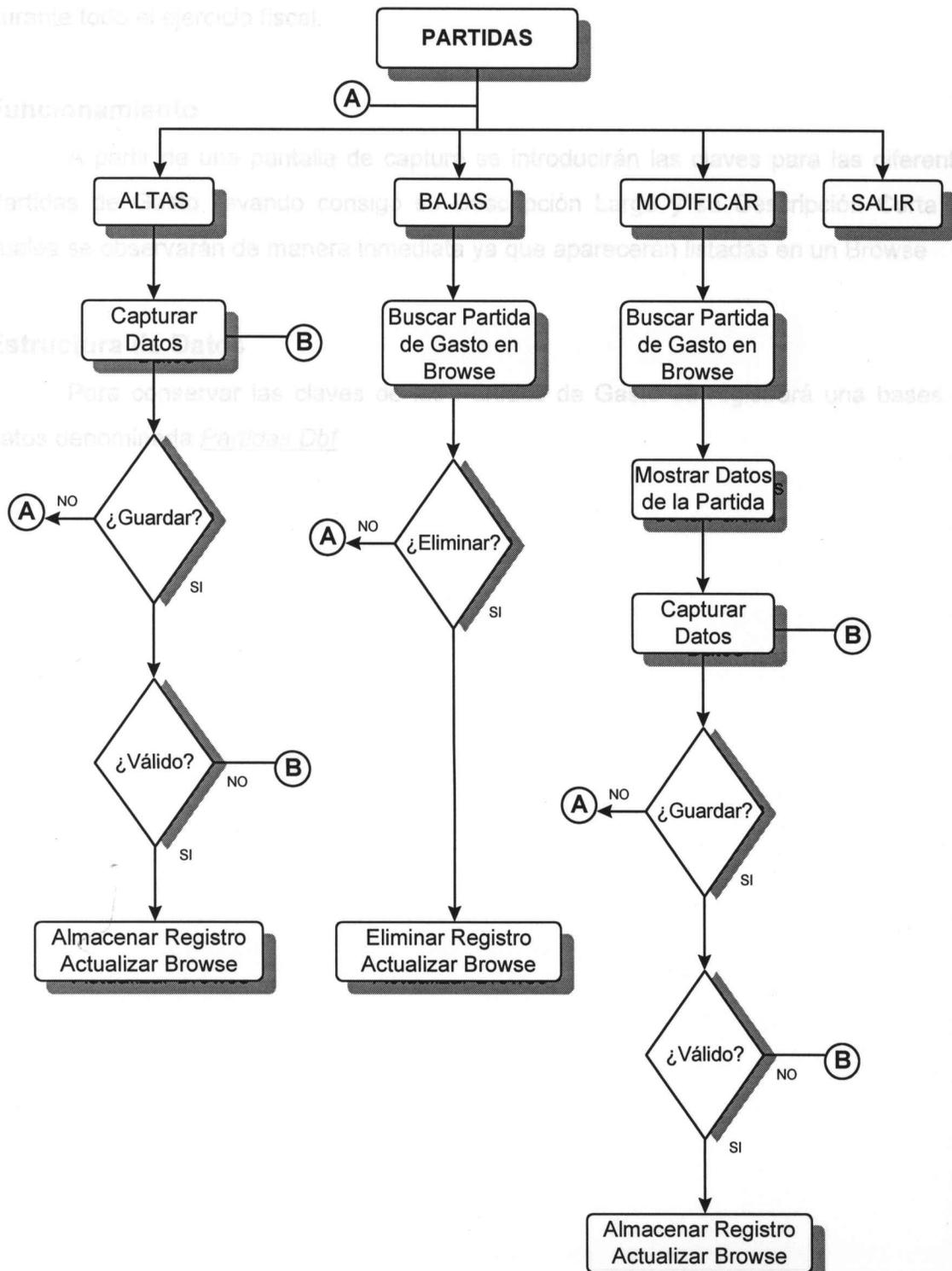
Para conservar los movimientos efectuados por medio del Documento de Ejecución Presupuestaria se registrarán en 3 bases de datos:

Almcabez.Dbf.- Es la identificación del Encabezado del documento.

Almcuerp.Dbf.- Es el registro de los movimientos realizados por el documento.

Asigna.Dbf.- Es el registro de los movimientos realizados por la Clave Presupuestal.

MODULO DEL CATALOGO DE PARTIDAS PRESUPUESTALES



Descripción

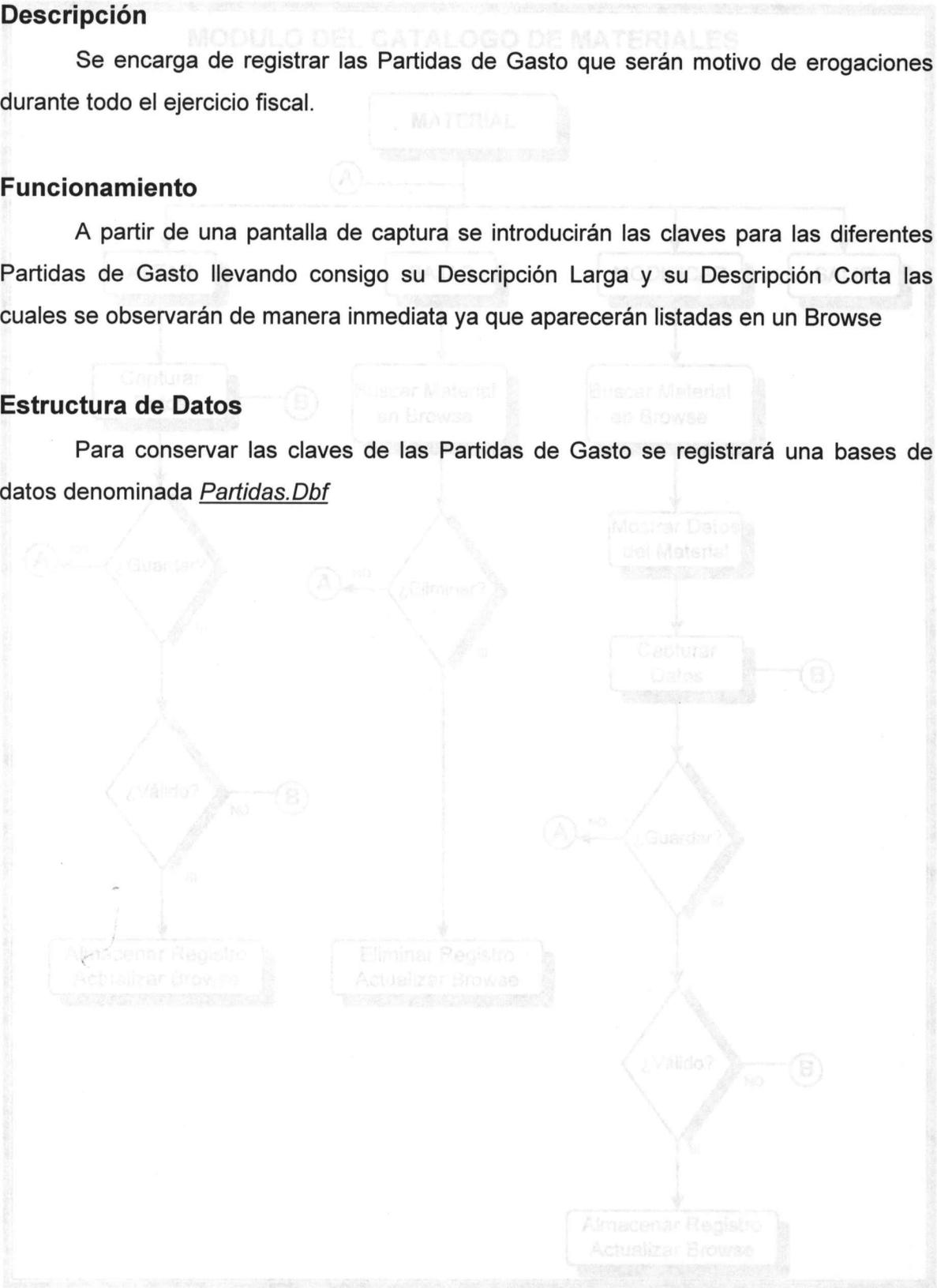
Se encarga de registrar las Partidas de Gasto que serán motivo de erogaciones durante todo el ejercicio fiscal.

Funcionamiento

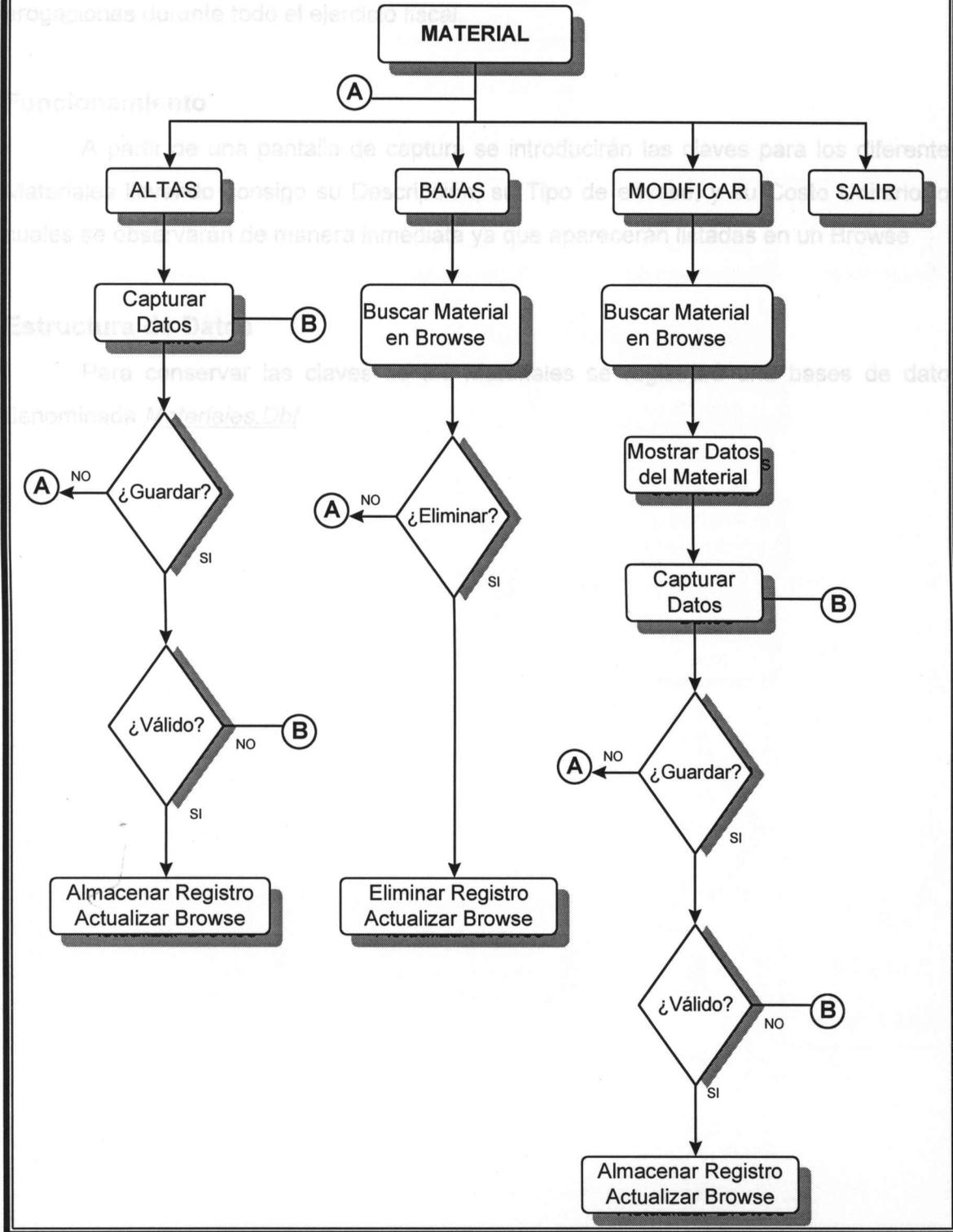
A partir de una pantalla de captura se introducirán las claves para las diferentes Partidas de Gasto llevando consigo su Descripción Larga y su Descripción Corta las cuales se observarán de manera inmediata ya que aparecerán listadas en un Browse

Estructura de Datos

Para conservar las claves de las Partidas de Gasto se registrará una bases de datos denominada Partidas.Dbf



MODULO DEL CATALOGO DE MATERIALES



Descripción

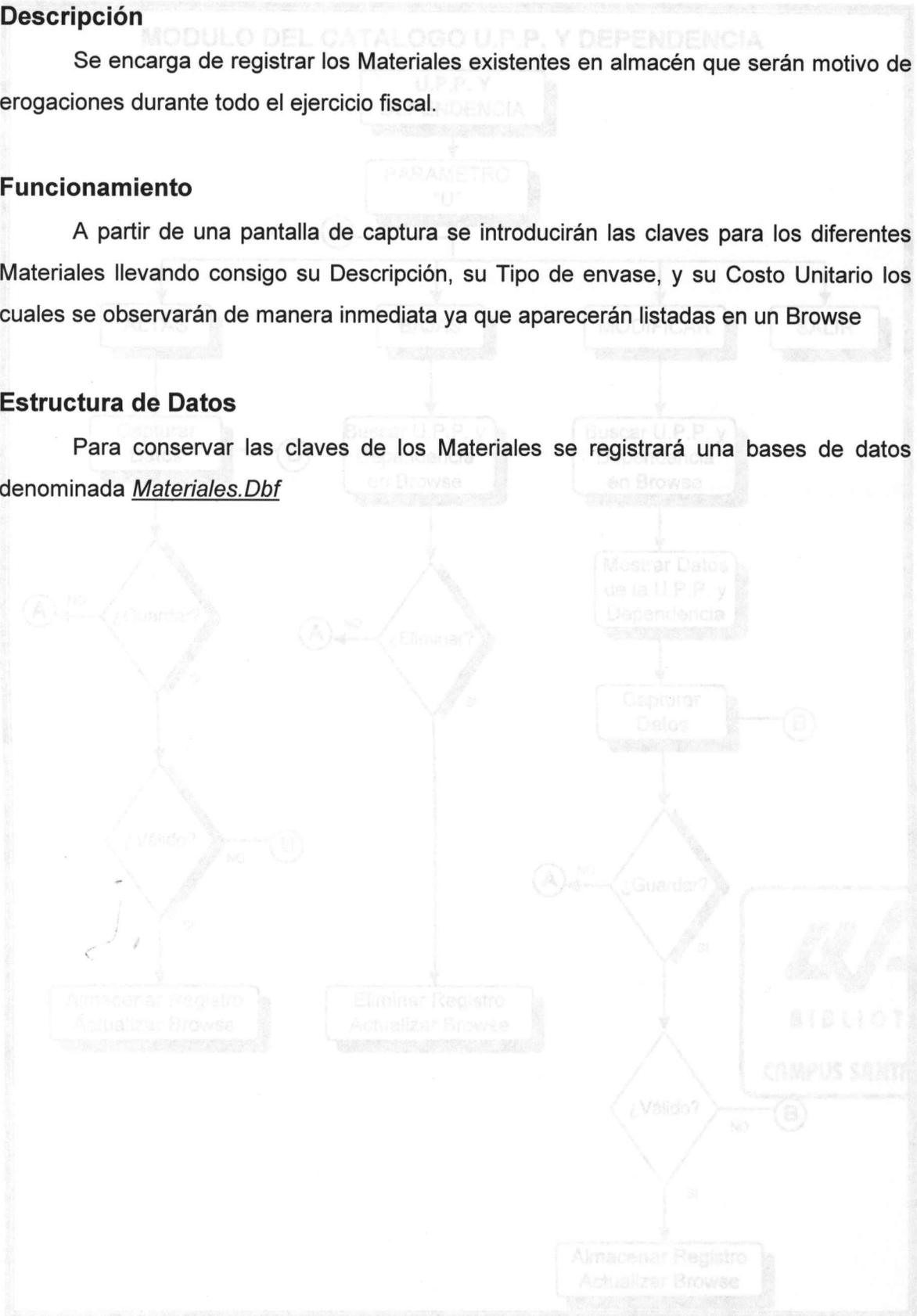
Se encarga de registrar los Materiales existentes en almacén que serán motivo de erogaciones durante todo el ejercicio fiscal.

Funcionamiento

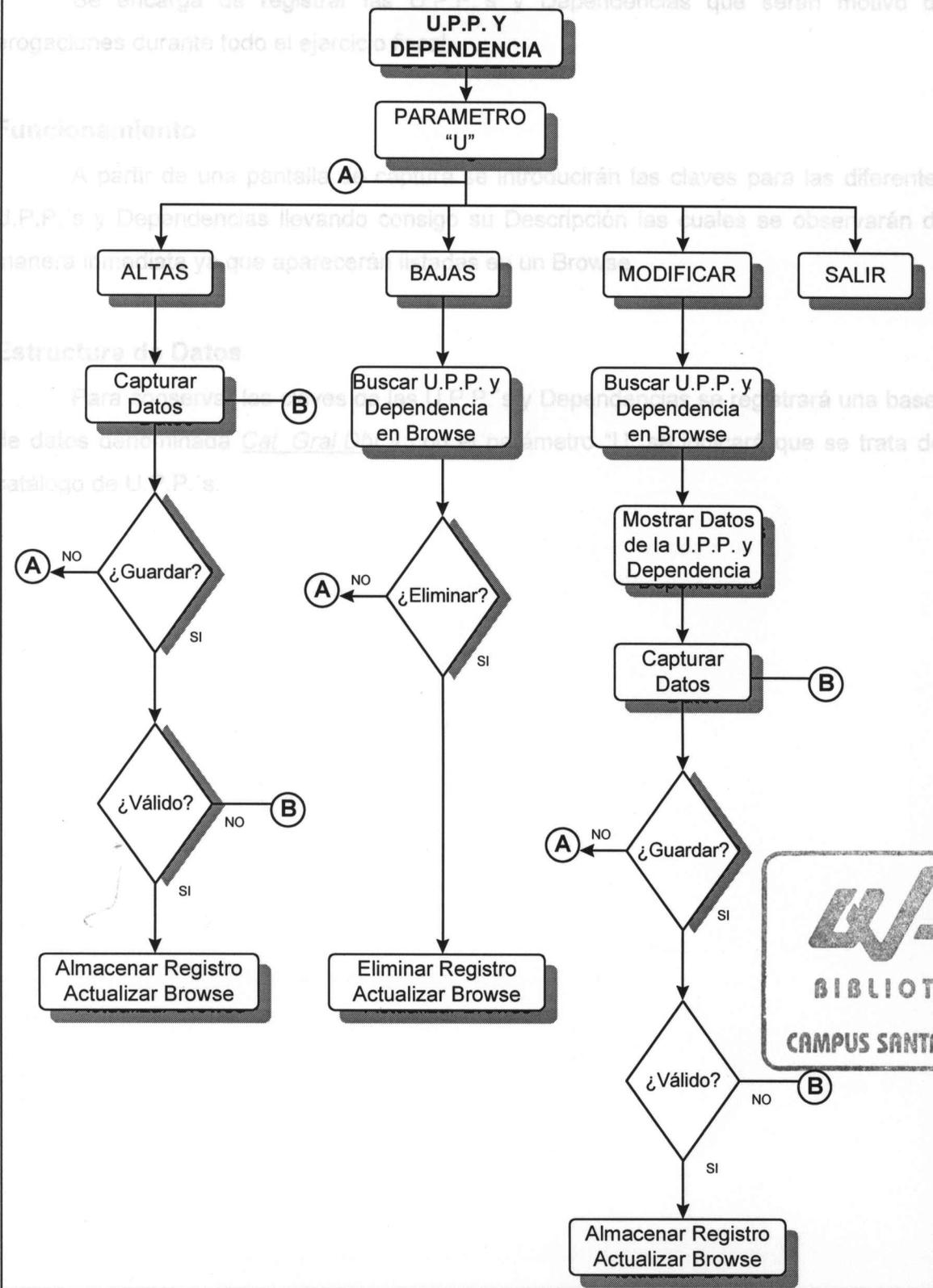
A partir de una pantalla de captura se introducirán las claves para los diferentes Materiales llevando consigo su Descripción, su Tipo de envase, y su Costo Unitario los cuales se observarán de manera inmediata ya que aparecerán listadas en un Browse

Estructura de Datos

Para conservar las claves de los Materiales se registrará una bases de datos denominada Materiales.Dbf



MODULO DEL CATALOGO U.P.P. Y DEPENDENCIA



Descripción

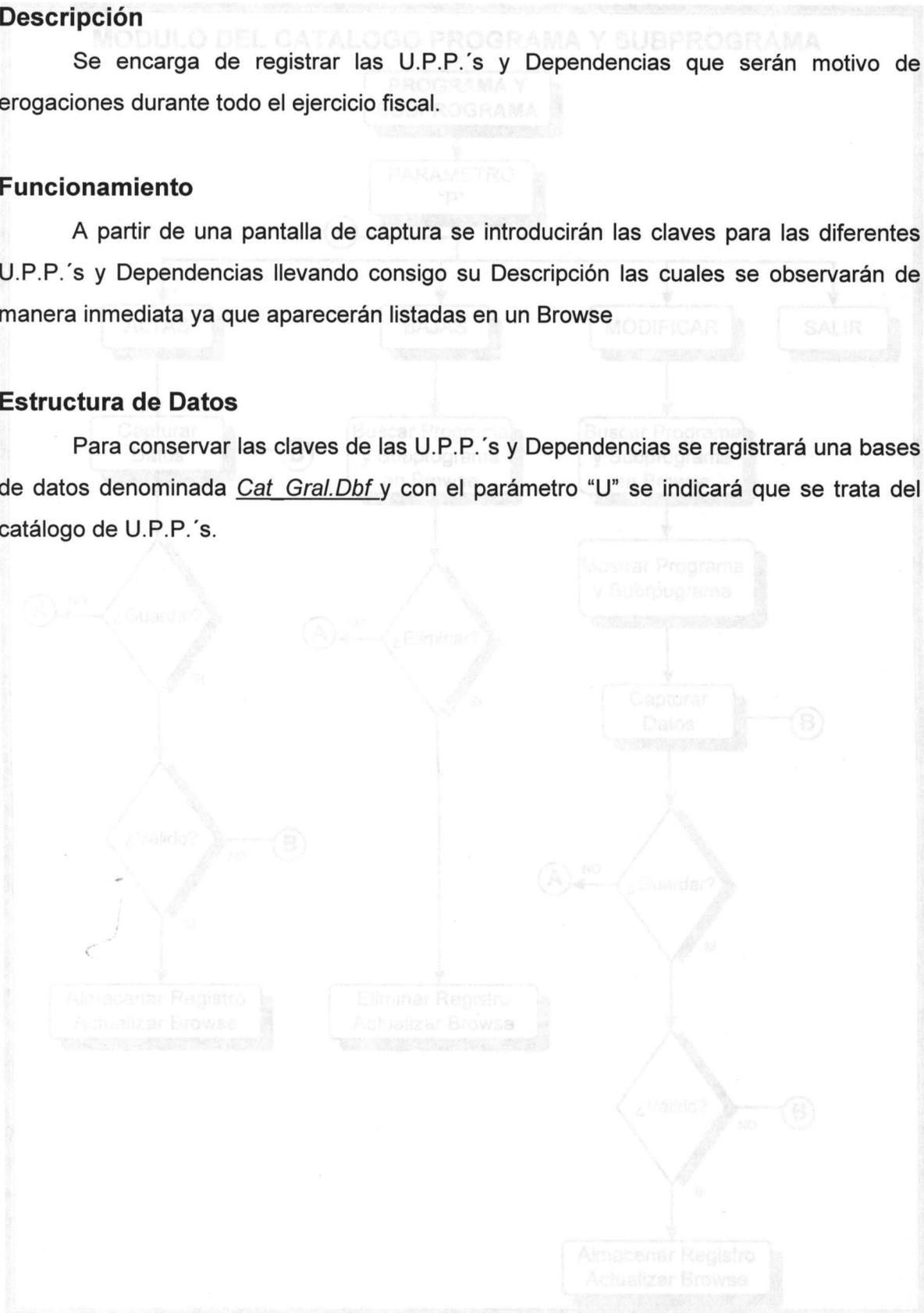
Se encarga de registrar las U.P.P.'s y Dependencias que serán motivo de erogaciones durante todo el ejercicio fiscal.

Funcionamiento

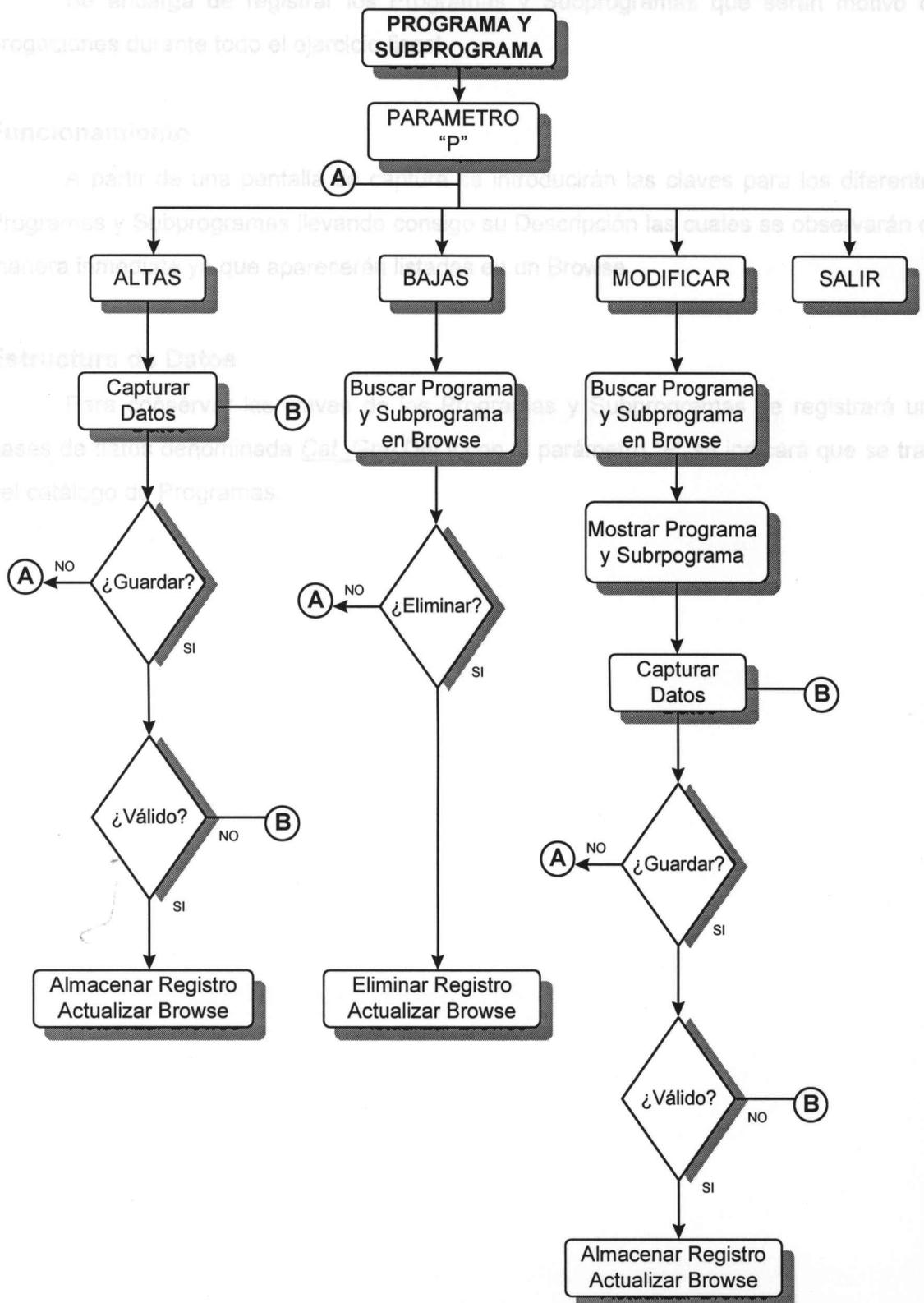
A partir de una pantalla de captura se introducirán las claves para las diferentes U.P.P.'s y Dependencias llevando consigo su Descripción las cuales se observarán de manera inmediata ya que aparecerán listadas en un Browse

Estructura de Datos

Para conservar las claves de las U.P.P.'s y Dependencias se registrará una bases de datos denominada Cat Gral.Dbf y con el parámetro "U" se indicará que se trata del catálogo de U.P.P.'s.



MODULO DEL CATALOGO PROGRAMA Y SUBPROGRAMA



Descripción

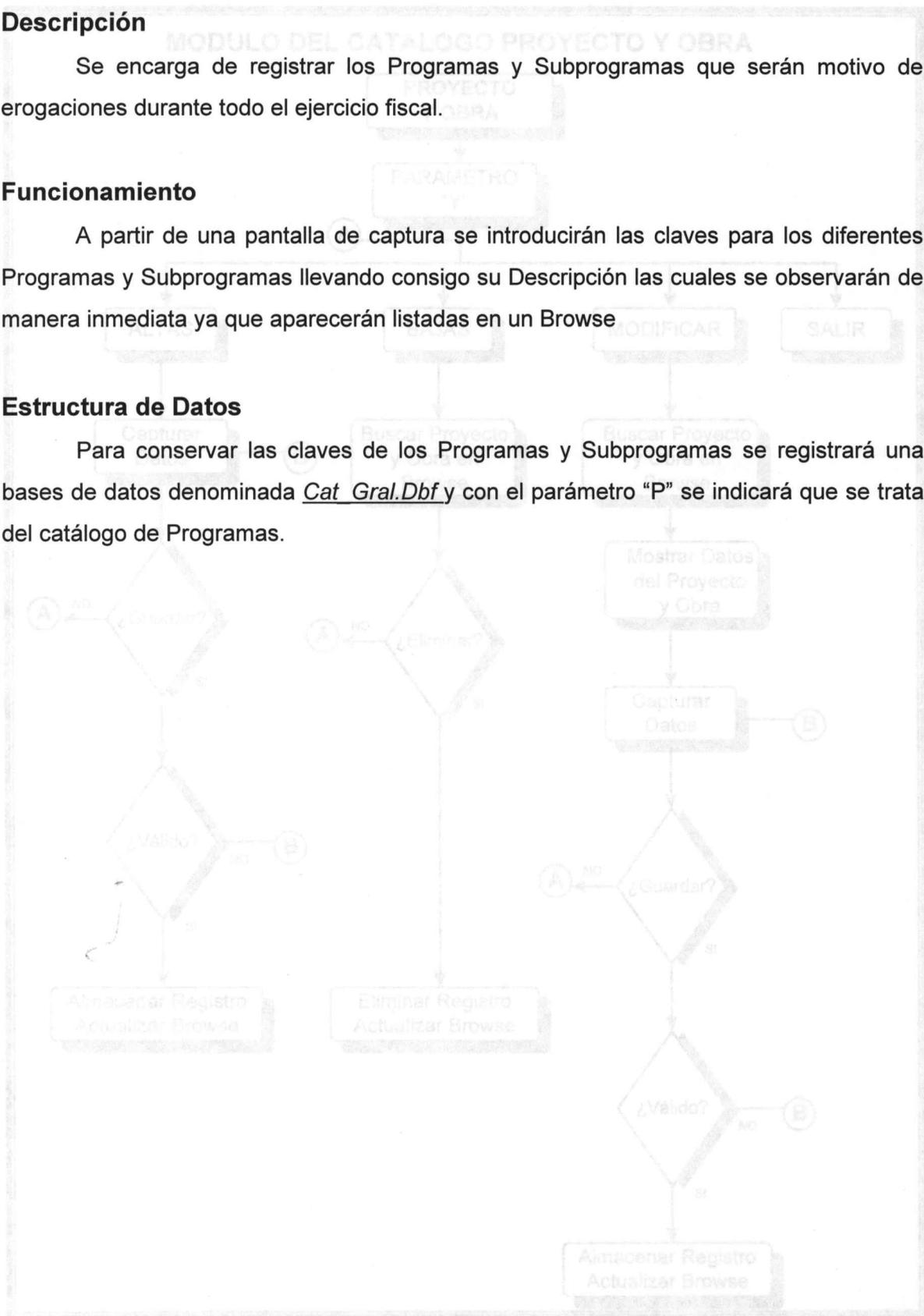
Se encarga de registrar los Programas y Subprogramas que serán motivo de erogaciones durante todo el ejercicio fiscal.

Funcionamiento

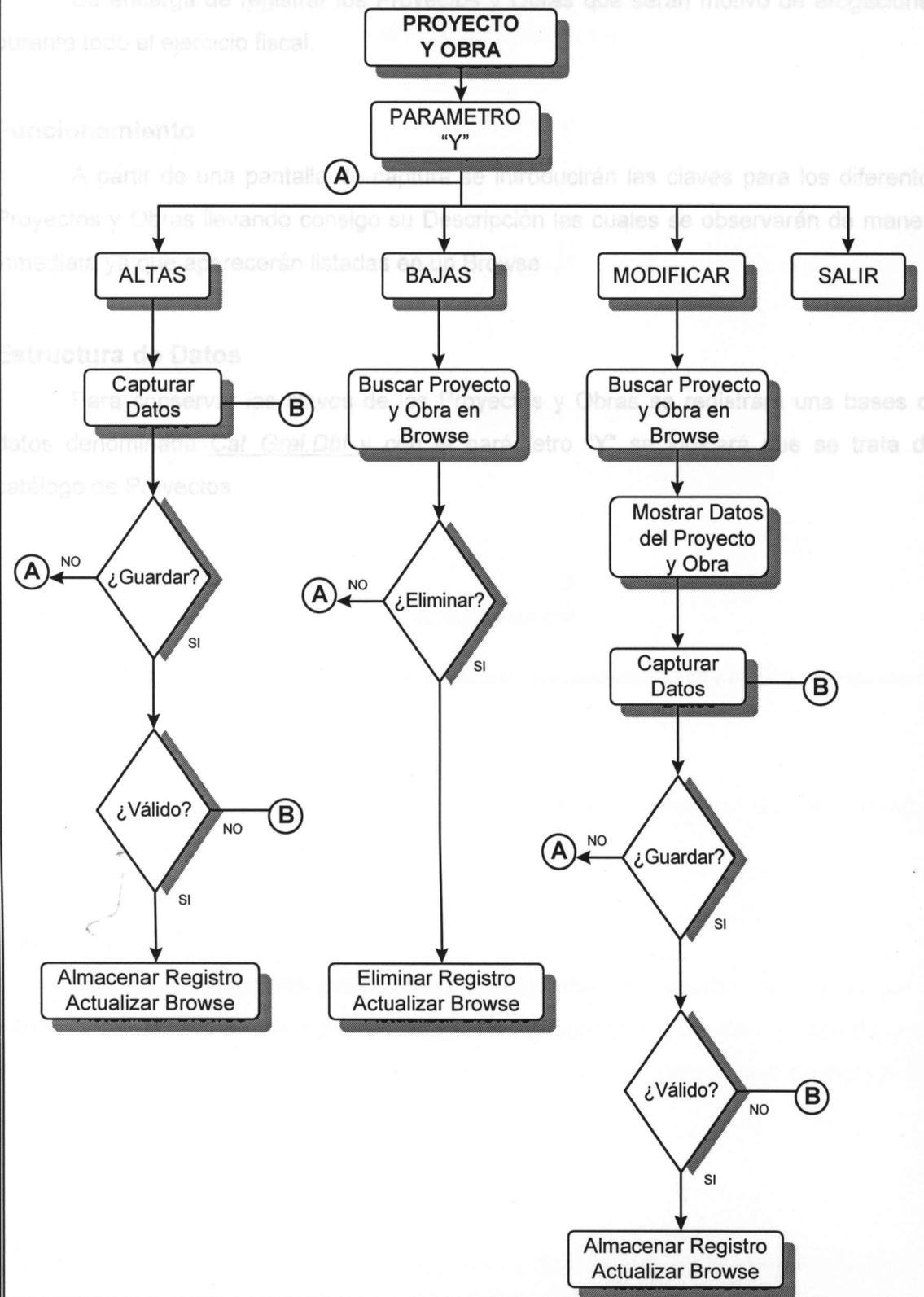
A partir de una pantalla de captura se introducirán las claves para los diferentes Programas y Subprogramas llevando consigo su Descripción las cuales se observarán de manera inmediata ya que aparecerán listadas en un Browse

Estructura de Datos

Para conservar las claves de los Programas y Subprogramas se registrará una bases de datos denominada Cat_Gral.Dbf y con el parámetro "P" se indicará que se trata del catálogo de Programas.



MODULO DEL CATALOGO PROYECTO Y OBRA



Descripción

Se encarga de registrar los Proyectos y Obras que serán motivo de erogaciones durante todo el ejercicio fiscal.

Funcionamiento

A partir de una pantalla de captura se introducirán las claves para los diferentes Proyectos y Obras llevando consigo su Descripción las cuales se observarán de manera inmediata ya que aparecerán listadas en un Browse

Estructura de Datos

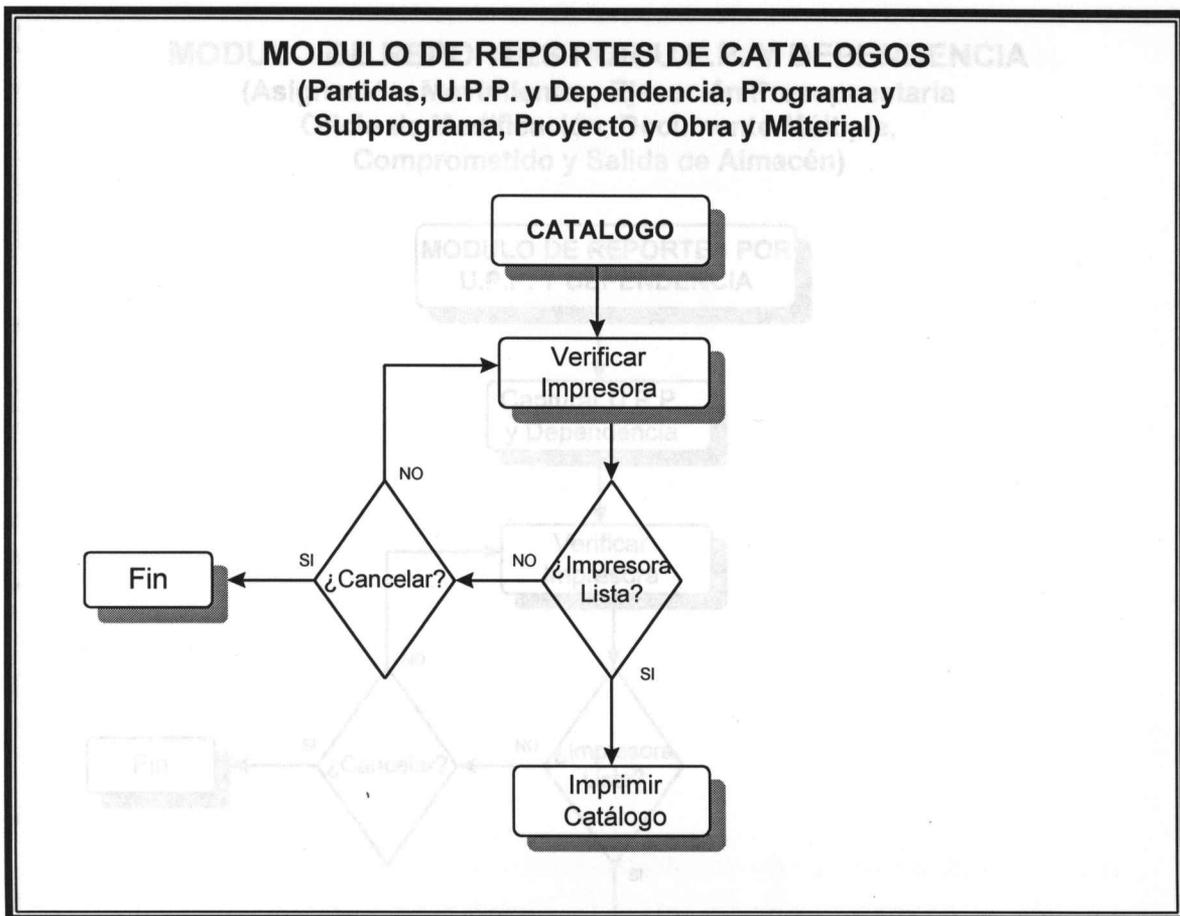
Para conservar las claves de los Proyectos y Obras se registrará una bases de datos denominada Cat Gral.Dbf y con el parámetro "Y" se indicará que se trata del catálogo de Proyectos.

Descripción

Se encarga de mostrar en forma impresa la información de los diferentes catálogos que son parte del sistema

Funcionamiento

A partir del elemento seleccionado en el módulo de reportes para catálogos se imprimirá todos los campos que se detallaron en el capítulo del Diseño de Base de Datos siempre y cuando la impresora se encuentre en línea y este conectada al puerto paralelo número uno de la computadora.



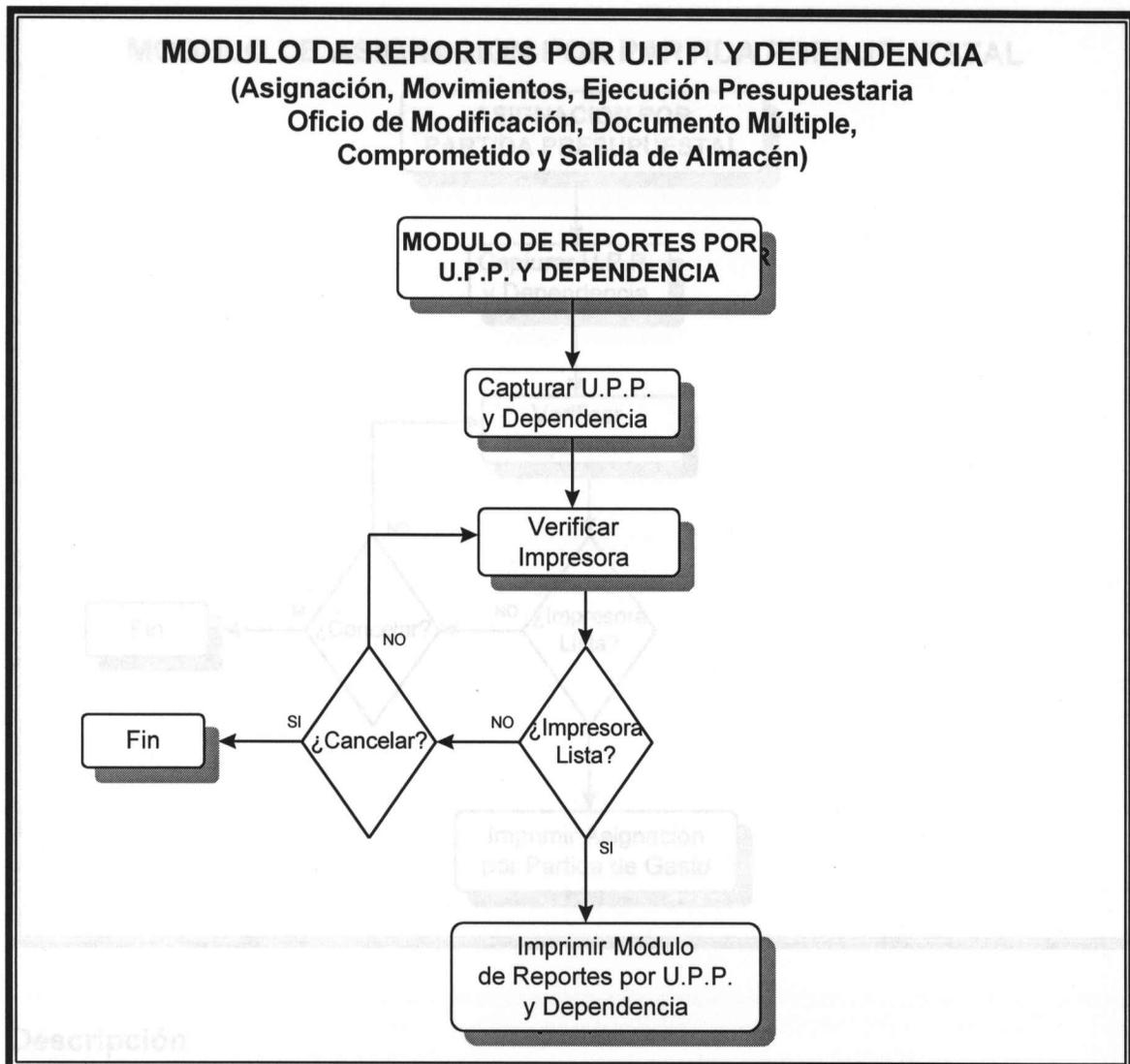
Descripción

Se encarga de mostrar en forma impresa la información de los diferentes catálogos que son parte del sistema

Funcionamiento

A partir del elemento seleccionado en el módulo de reportes para catálogos se imprimirá todos los campos que se detallaron en el capítulo del Diseño de Base de Datos siempre y cuando la impresora se encuentre en línea y este conectada al puerto paralelo número uno de la computadora.

Se extraerá a partir de la U.P.P. y Dependencia la información que será propia de ellas dependiendo del elemento seleccionado en el módulo de reportes y que están señalados en la figura anterior.

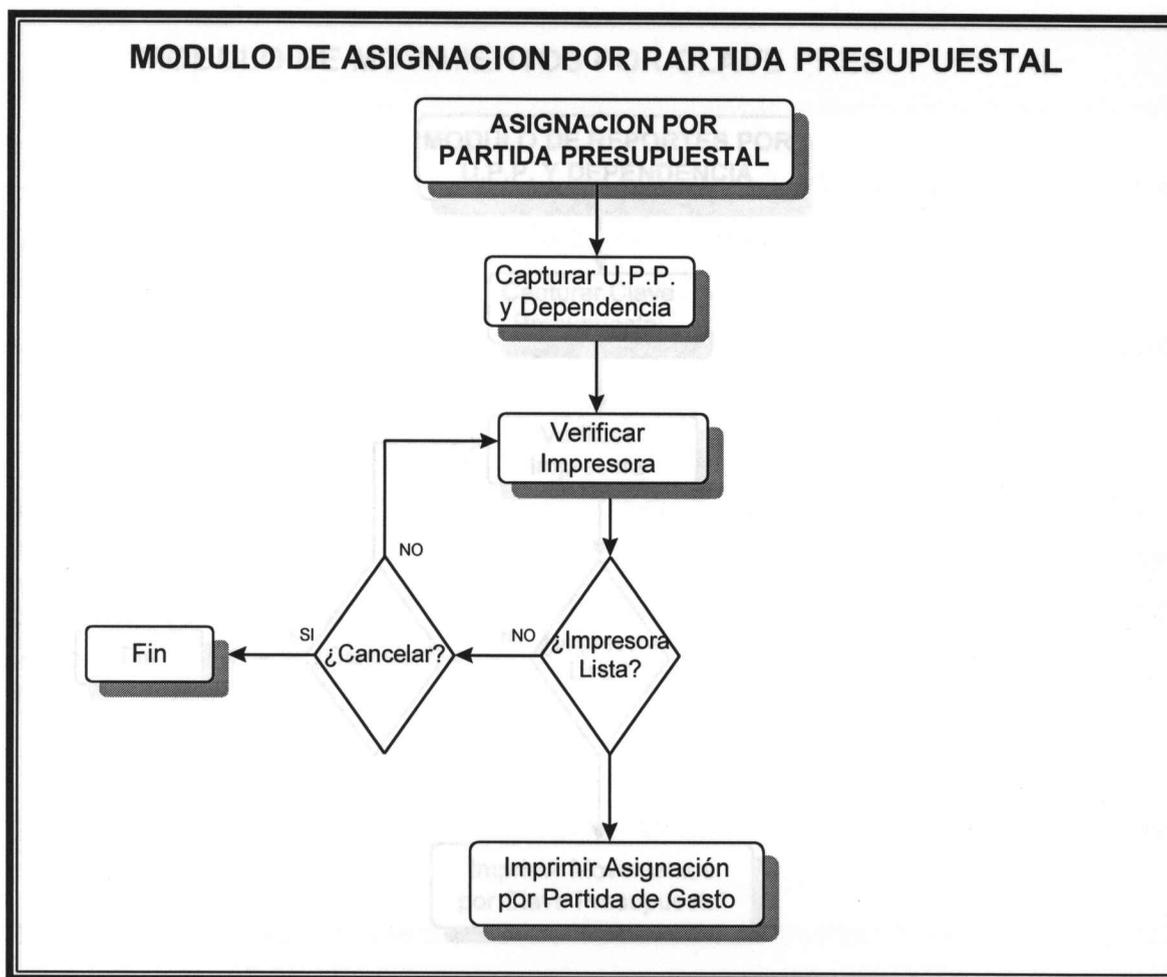


Descripción

Se encarga de mostrar en forma impresa la información que en común tienen una U.P.P. y una Dependencia.

Funcionamiento

Se extraerá a partir de la U.P.P. y Dependencia la información que será propia de ellas dependiendo del elemento seleccionado en el módulo de reportes y que están señalados en la figura anterior.



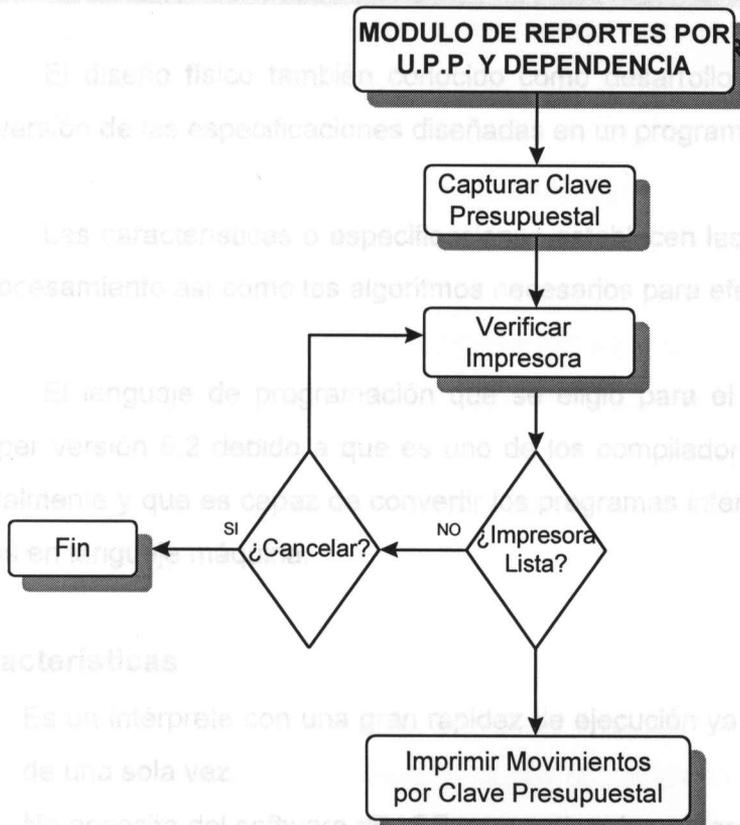
Descripción

Descripción: Se encarga de mostrar la información en forma impresa de las cantidades monetarias que tiene una Partida de Gasto. Se muestra en forma impresa de los movimientos que ha venido sufriendo una Clave Presupuestal cada vez que se realiza un gasto.

Funcionamiento

Funcionamiento: Se pedirá que se capture la U.P.P. y Dependencia y a partir de ellas se buscarán todas las asignaciones monetaria que tiene en común una Clave Presupuestal que a su vez tenga en común la U.P.P. y la Dependencia para ello se extraerá de la Clave Presupuestal fue dada de alta en la módulo de Distribución, su Partida y el monto asignado de tal manera que se sabrá en que se está asignando un gasto y cuanto.

MODULO DE MOVIMIENTOS POR CLAVE PRESUPUESTAL



Descripción

Se encarga de mostrar la información en forma impresa de los movimientos que ha venido sufriendo una Clave Presupuestal cada vez que se realiza un gasto.

Funcionamiento

Se extraerá la Clave Presupuestal que se pide de la Base de Datos que es donde se registran los movimientos de la Clave Presupuestal de tal manera que mostrarán todos sus movimientos y en que fecha los realizó y el monto de las erogaciones.

DISEÑO FISICO

El diseño físico también conocido como desarrollo del software es la etapa de conversión de las especificaciones diseñadas en un programa de computadora.

Las características o especificaciones establecen las funciones de entrada, salida y procesamiento así como los algoritmos necesarios para efectuarlas.

HERRAMIENTAS DE LA ESTRATEGIA DE FLUJO DE DATOS

El lenguaje de programación que se eligió para el desarrollo del SCGP, fue el Clipper versión 5.2 debido a que es uno de los compiladores más potentes que existen actualmente y que es capaz de convertir los programas intérpretes de gestión de base de datos en lenguaje máquina.

El análisis de flujo de datos utiliza las siguientes herramientas:

Características

- 1. ☞ Es un intérprete con una gran rapidez de ejecución ya que analiza las instrucciones de una sola vez.
- ☞ No necesita del software dBASE para activar los programas.
- ☞ Los programas ejecutables de clipper oscilan entre los 140K y 360K de memoria.
- ☞ Es capaz de establecer una conexión con el lenguaje C.
- ☞ Posee una eficiente flexibilidad en el manejo de las bases de datos.
- ☞ Es capaz de adaptar a su lenguaje los formatos de las bases de datos y archivos índices creados por otros sistemas tales como FoxPro, Paradox y dBASE IV.

HERRAMIENTAS PARA EL DESARROLLO DE SISTEMAS

Una herramienta es cualquier dispositivo que, cuando se emplea en forma apropiada mejora el desempeño de una tarea. Una herramienta para el desarrollo de sistema ayuda al analista a trasladar los diseños en aplicaciones funcionales ya que tienen influencia sobre la calidad del resultado final.

Los resultados se ven reflejados de la siguiente manera:

- *Mejora en la productividad.*- Con las herramientas correctas se tiene más potencial de ser más productivo y disminuir el tiempo en que se desarrollan las aplicaciones.
- *Mejora en la eficacia.*- Las herramientas sugieren procedimientos que conducen al empleo de procesos más eficientes. En otras palabras, ayudan a realizar los procesos de la manera correcta.
- *Mejora en la calidad del sistema de información.*- Cuando las herramientas sugieren los procesos, por lo general también ocurre lo mismo con los resultados.

HERRAMIENTAS DE LA ESTRATEGIA DE FLUJO DE DATOS

La estrategia de flujo de datos muestra el empleo de éstos en forma gráfica. Las herramientas utilizadas a seguir esta estrategia muestran todas las características esenciales del sistema y la forman en que se ajustan entre sí.

El análisis de flujo de datos utiliza las siguientes herramientas:

1. Diagrama de Flujo de Datos.- Es una herramienta gráfica que se emplea para describir y analizar el movimiento de datos por medio de un sistema, ya sea que éste fuera manual o automatizado, incluyendo procesos, lugares para almacenar datos y retrasos en el sistema. Los diagramas de flujo de datos son la herramienta más importante y la base sobre la cual se desarrollan otros componentes. La transformación de datos de entrada en salida por medio de procesos puede describirse en forma lógica e independiente de los componentes físicos asociados con el sistema a esto se le conoce como diagramas de flujo de datos lógicos. En contraste los diagramas de flujo de datos físicos muestran la implantación y movimiento real de datos entre las personas, departamentos y estaciones de trabajo.
2. Diccionario de Datos.- El diccionario contiene las características lógicas de los sitios donde se almacenan los datos del sistema, incluyendo nombre, descripción, alias, contenidos y organización. También identifica los procesos donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información. Sirve como punto de partida para identificar los requerimientos de las bases de datos durante el diseño de sistemas.

3. Diagrama de Estructura de Datos.- Este diagrama es una descripción de la relación entre entidades (personas, lugares, eventos y objetos) de un sistema y el conjunto de la información relacionado con la entidad.

4. Gráfica de Estructura.- Herramienta de diseño que muestra con símbolos la relación entre los módulos de procesamiento y el software de la computadora. Describen jerarquía de los módulos componentes y los datos que serán transmitidos entre ellos. Incluye el análisis de las transformaciones entrada-salida y el análisis de transacciones.

SUGERENCIAS EN LA FORMA DE CODIFICAR

Las recomendaciones comúnmente aceptadas para la escritura de programas pretenden simplificar la lectura y comprensión de los programas que continuación se harán mención.

- ◆ Empleo de mayúsculas y minúsculas.- Es conveniente que durante la codificación se siga un mismo criterio para el empleo de mayúsculas y minúsculas, ya que esto ayudará a identificar más rápidamente los comandos, variables, archivos etc.
- ◆ Uso del tipo de dato sobre las variables.- Es importante que cuando se este trabajando con una determinada variable anteponer el tipo de dato que se esta utilizando.
- ◆ Sangrado de las estructuras de control.- Las sentencias Do While... Enddo, If... Else... Endif y Case... EndCase implementan estructuras lógicas de control. Se puede ver sin dificultad cuáles son los enunciados que abarcan si sangramos las líneas contenidas dentro de ellas. El beneficio se incrementa si estas estructuras se encuentran anidadas.
- ◆ Espaciado.- El uso de líneas en blanco ayuda a separa actividades diferentes.
- ◆ Líneas de comentarios.- Cuanto más explícitos sean los comentarios más fácil será el mantenimiento de los programas. Siempre es recomendable incluir unas líneas al principio del programa para recordar su finalidad.
- ◆ Espaciado de las expresiones.- El uso de operaciones aritméticas o relacionales, sobre todo en ecuaciones de tamaño considerable debe hacerse usando un espacio

adecuado de los operandos, de tal manera que no resulte tan complicado reestructurar o revisar las ecuaciones.

EL DISEÑO FÍSICO DEL SISTEMA

En la fase del diseño lógico del sistema se realizó un estudio individual de los módulos principales, se presentó su diagrama de flujo así como una descripción de la forma que opera y, en una forma muy sencilla su funcionamiento interno.

En el diseño físico se pretende explicar de una manera más detallada el funcionamiento del módulo más importante del sistema: Erogaciones (Documento de Ejecución Presupuestaria, Oficio de Modificación, Documento Múltiple, Requisiciones y Salida de almacén) para la mejor comprensión del diseño físico del sistema.

Erogaciones

En líneas anteriores se mencionaron los elementos del módulo de erogaciones y en el capítulo del diseño lógico se hizo una breve descripción y funcionamiento de cada uno.

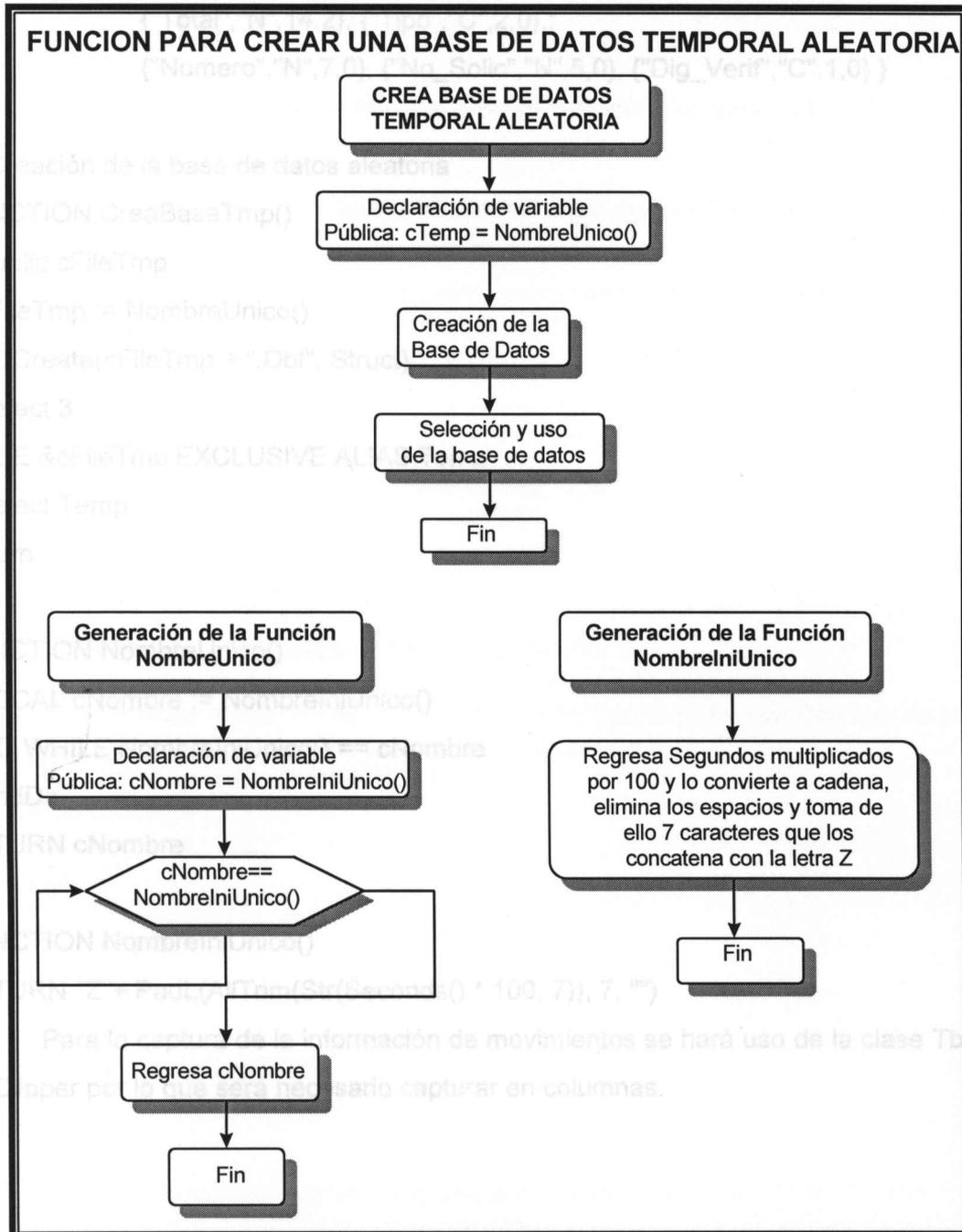
Funcionamiento

Para describir el funcionamiento de cada elemento del módulo de erogaciones partiremos de las siguientes consideraciones:

- 1.- Los documentos de los elementos del módulo de erogaciones están divididos en dos partes: Encabezado y Cuerpo (para conocer su estructura consulte el capítulo del Diseño de Datos).
- 2.- Existe una Clave Presupuestal para realizar erogaciones.

Para el primer punto se creó dos bases de datos, una para guardar la información del encabezado y la otra para guardar la información del cuerpo. La relación entre ambas bases de datos sería un campo el cual fuera único y no fuera ambiguo, por lo cual, después de haber analizado los diferentes documentos, resultó que el número de solicitud es el campo indicado para interrelacionar entre el encabezado y el cuerpo del

documento. Por otra parte para conservar la información de los movimientos que se editan en el cuerpo se creó una base de datos temporal aleatoria (de acuerdo a los campos descritos para la parte del cuerpo de cada documento en el capítulo de Diseño de Datos) para que el usuario no guardará directamente los movimientos en la bases de datos hasta que estuviera seguro de que los movimientos capturados fueran los correctos. Se optó por guardar la información en dos bases de datos ya que se evitaría la duplicidad de información y la densidad de la misma en una sola base de datos.



Estos fueron los pasos para llevar a cabo el diagrama de flujo:
 creo el siguiente objeto.

- 1.-Generación de un arreglo de la estructura de la base de datos tomando como ejemplo el cuerpo del Documento de Ejecución Presupuestaria.

```
Private Struct := { {"No_Sec","N",2,0}, {"Clave_Pres","C",18,0},;
  {"Beneficiar","C",30,0}, {"Concepto","C",30,0},;
  {"Cargo","N",14,2}, {"Deduccion","N",14,2},;
  {"Total","N",14,2}, {"Tipo","C",2,0},;
  {"Numero","N",7,0}, {"No_Solic","N",5,0}, {"Dig_Verif","C",1,0} }
```

- 2.-Creación de la base de datos aleatoria

FUNCTION CreaBaseTmp()

```
Public cFileTmp
cFileTmp := NombreUnico()
DBCreate(cFileTmp + ".Dbf", Struct)
Select 3
USE &cFileTmp EXCLUSIVE ALIAS Temp
Select Temp
Return
```

```
FUNCTION NombreUnico()
LOCAL cNombre := NombreIniUnico()
DO WHILE NombreIniUnico() == cNombre
EndDo
RETURN cNombre
```

```
FUNCTION NombreIniUnico()
RETURN "Z"+ PadL(AllTrim(Str(Seconds() * 100, 7)), 7, "")
```

Para la captura de la información de movimientos se hará uso de la clase Tbrowse de Clipper por lo que será necesario capturar en columnas.

Para realizar la edición de cada uno de los campos del Cuerpo en columnas se creo el siguiente objeto:

Función para Crear un Objeto Get

Descripción.- Permite la edición de datos en la Clase Tbrowse.

Parámetros.-

bBloque.- Es el campo en cual se pretende realizar una edición.

oBrowse.- Es la clase Tbrowse donde se realizará la edición.

Variables.-

xVar.- Guarda la evaluación que hace la función Eval sobre el valor de bBloque (Eval(bBloque))

oGet.- Almacena la edición del campo que fue pasado como parámetro en bBloque.

Static FUNCTION Edicion(bBloque,oBrowse)

LOCAL xVar, oGet

oBrowse:Stabilize()

xVar := Eval(bBloque)

oGet := GetNew(Row(), Col(), { |x| IIF(x == NIL, xVar, xVar := x) }, "xVar")

DO CASE

CASE oBrowse:ColPos = 1

oGet:Picture="@ 99"

CASE oBrowse:ColPos = 2

oGet:Picture="@R 9999-9999-9999-9999-99"

CASE oBrowse:ColPos = 4 .OR. oBrowse:ColPos = 5

oGet:Picture="@S" + Ltrim(Str(oBrowse:GetColumn(oBrowse:ColPos):Width))

CASE oBrowse:ColPos > 5 .AND. oBrowse:ColPos < 9

oGet:Picture="99,999,999,999.99"

CASE oBrowse:ColPos = 9

oGet:Picture="@!!"

ENDCASE

ReadModal({ oGet })

oBrowse:Right()

If Updated()

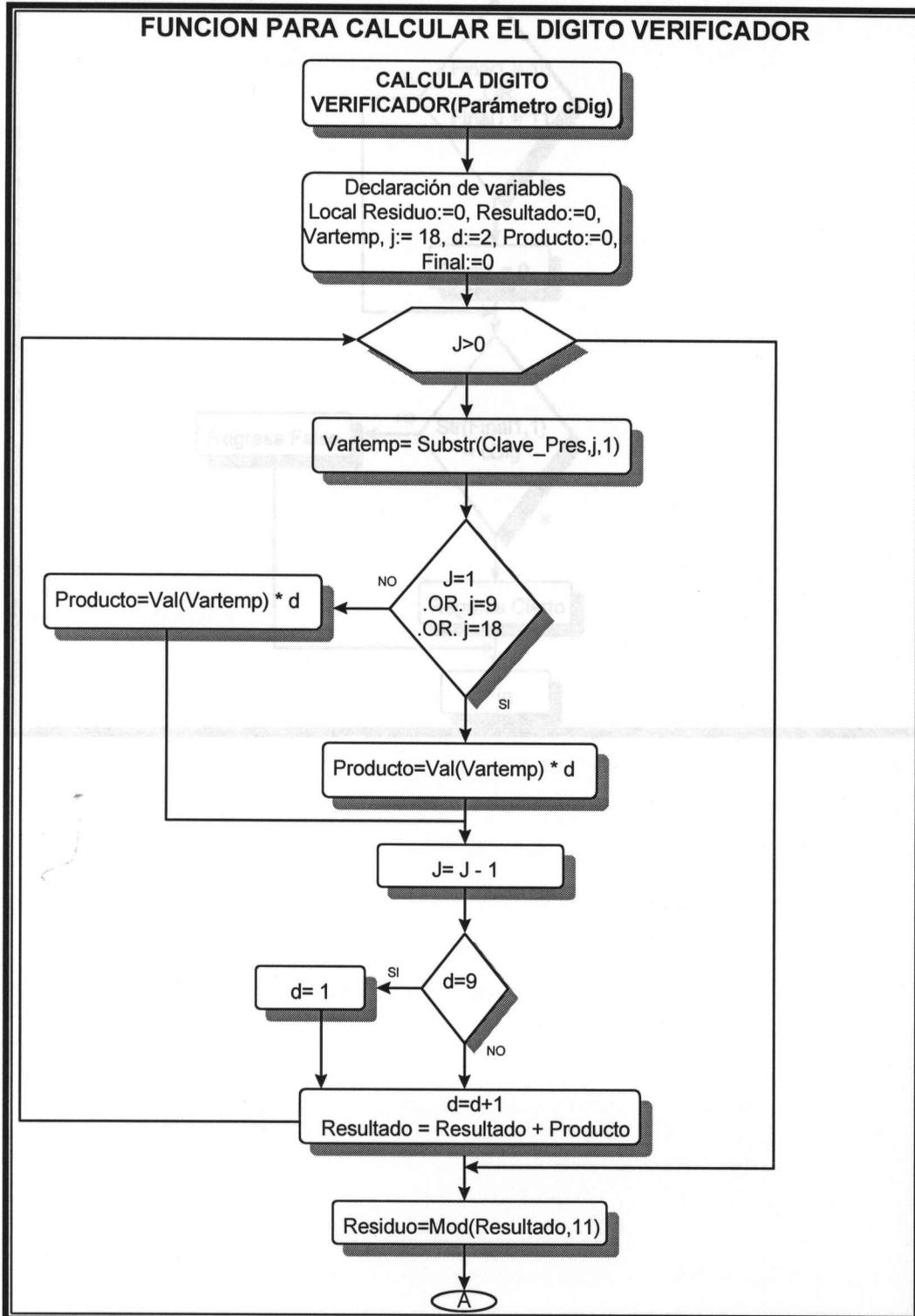
Eval(bBloque, xVar)

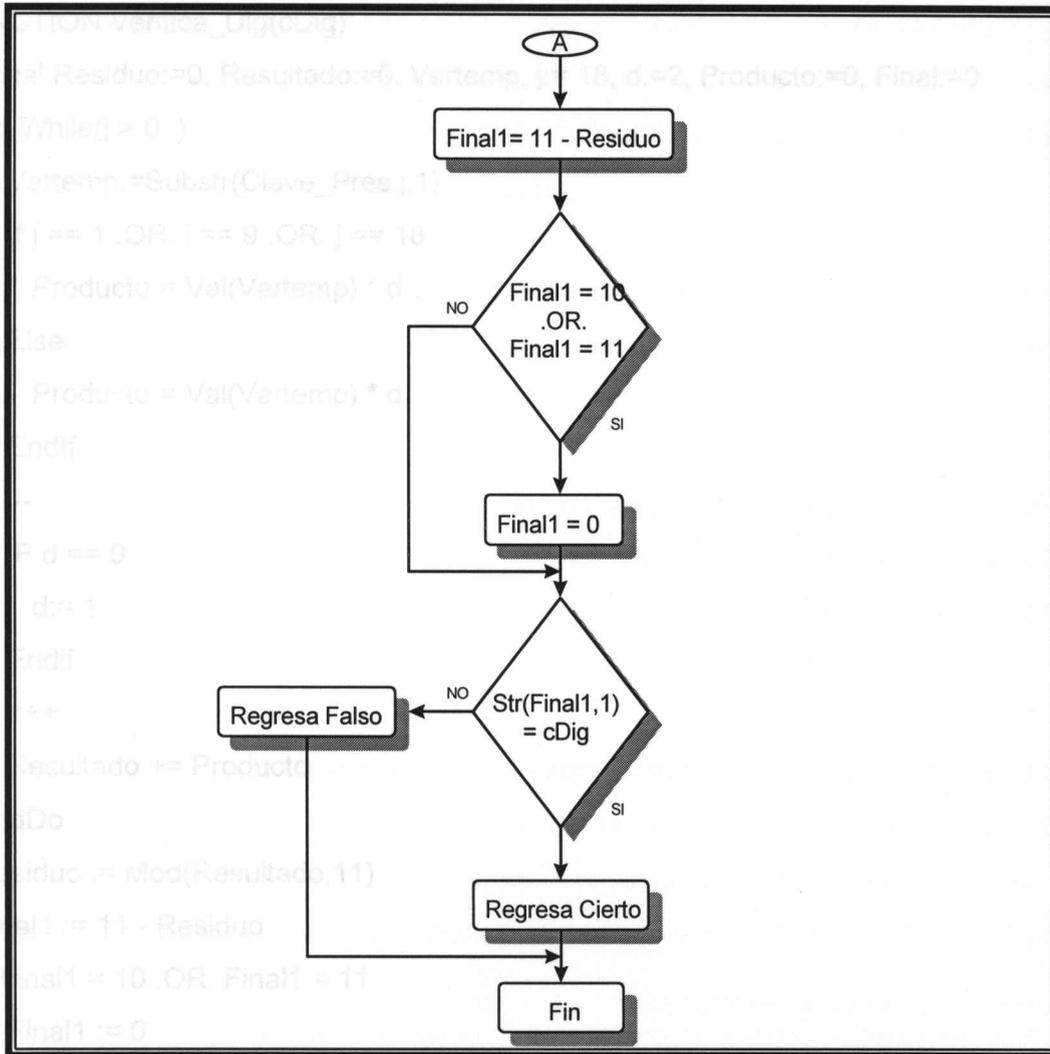
EndIf

oBrowse:RefreshCurrent()

RETURN

Como segundo punto, la Clave Presupuestal es muy importante, dado que si existe en el módulo de distribución se podrá realizar una erogación, además de que por medio del cálculo del dígito verificador se indicará si ésta fue escrita en forma correcta. Para ello se diseñó una función que realice el cálculo de dicho dígito.





```
FUNCTION Verifica_Dig(cDig)
```

```
Local Residuo:=0, Resultado:=0, Vartemp, j:= 18, d:=2, Producto:=0, Final:=0
```

```
Do While(j > 0 )
```

```
Vartemp:=Substr(Clave_Pres,j,1)
```

```
If j == 1 .OR. j == 9 .OR. j == 18
```

```
Producto:= Val(Vartemp) * d
```

```
Else
```

```
Producto:= Val(Vartemp) * d
```

```
EndIf
```

```
j--
```

```
If d == 9
```

```
d:= 1
```

```
Endif
```

```
d++
```

```
Resultado += Producto
```

```
EndDo
```

```
Residuo := Mod(Resultado,11)
```

```
Final1 := 11 - Residuo
```

```
If Final1 = 10 .OR. Final1 = 11
```

```
Final1 := 0
```

```
EndIf
```

```
If Str(Final1,1) = cDig
```

```
Return(.T.)
```

```
Else
```

```
Tone(200)
```

```
G_Dspmsg("El Dígito No Corresponde a la Clave","R")
```

```
oCuerpo:Left()
```

```
Return(.F.)
```

```
EndIf
```

CONECTIVIDAD en Red

Cuando una base de datos tiene que ser compartida por varios usuarios, se precisa un sistema multiusuario que pueda ser instalado bajo la forma de una red o bien por medio de un sistema operativo multiusuario.

Los sistemas para operar en redes locales deben poseer sofisticados medios de comunicación para mantener la seguridad de la base de datos y evitar al máximo los errores.

Si son varios los usuarios que tienen acceso a la base de datos conviene establecer fórmulas de control y restringir ciertos usos con el fin de que los usuarios no interfieran en su trabajo entre si o realicen modificaciones erróneas tomando de datos no actualizados.

Por lo tanto para los movimientos de altas existe la posibilidad de que dos o más usuarios den de alta la misma clave para un catálogo o el mismo número de solicitud de algún documento para realizar una erogación, dando lugar a irregularidades en el sistema. Por este motivo se tomó la decisión de bloquear el archivo y no solo el registro.

Para los movimientos de bajas no existe el mismo problema que en el movimiento de altas, para dar de baja solamente hay que bloquear el registro ya que se le indica cual es el registro a borrar.

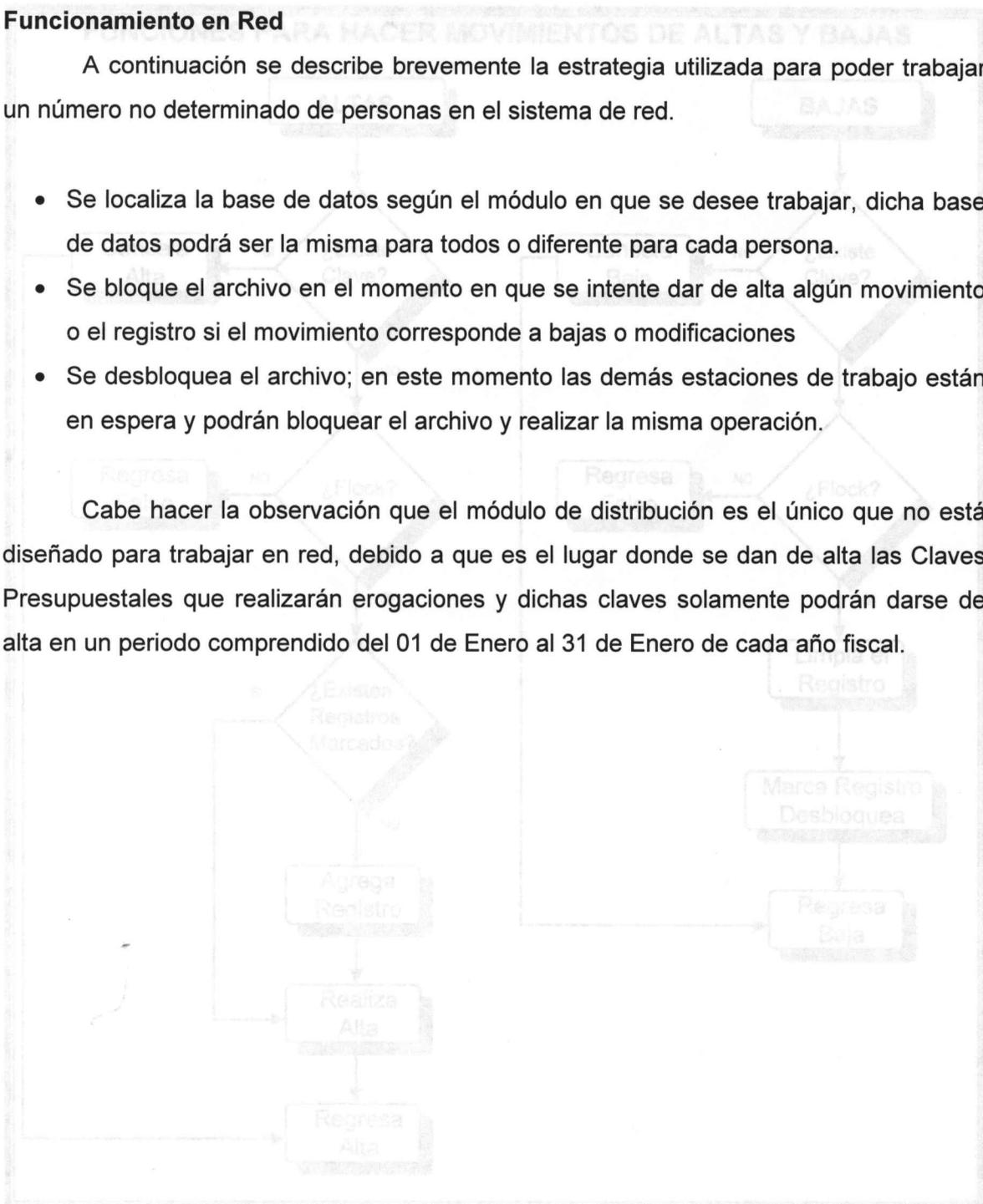
Para los movimientos de modificaciones puede suceder que dos o más usuarios deseen editar simultáneamente el mismo registro por lo que será necesario bloquear únicamente el registro y no el archivo.

Funcionamiento en Red

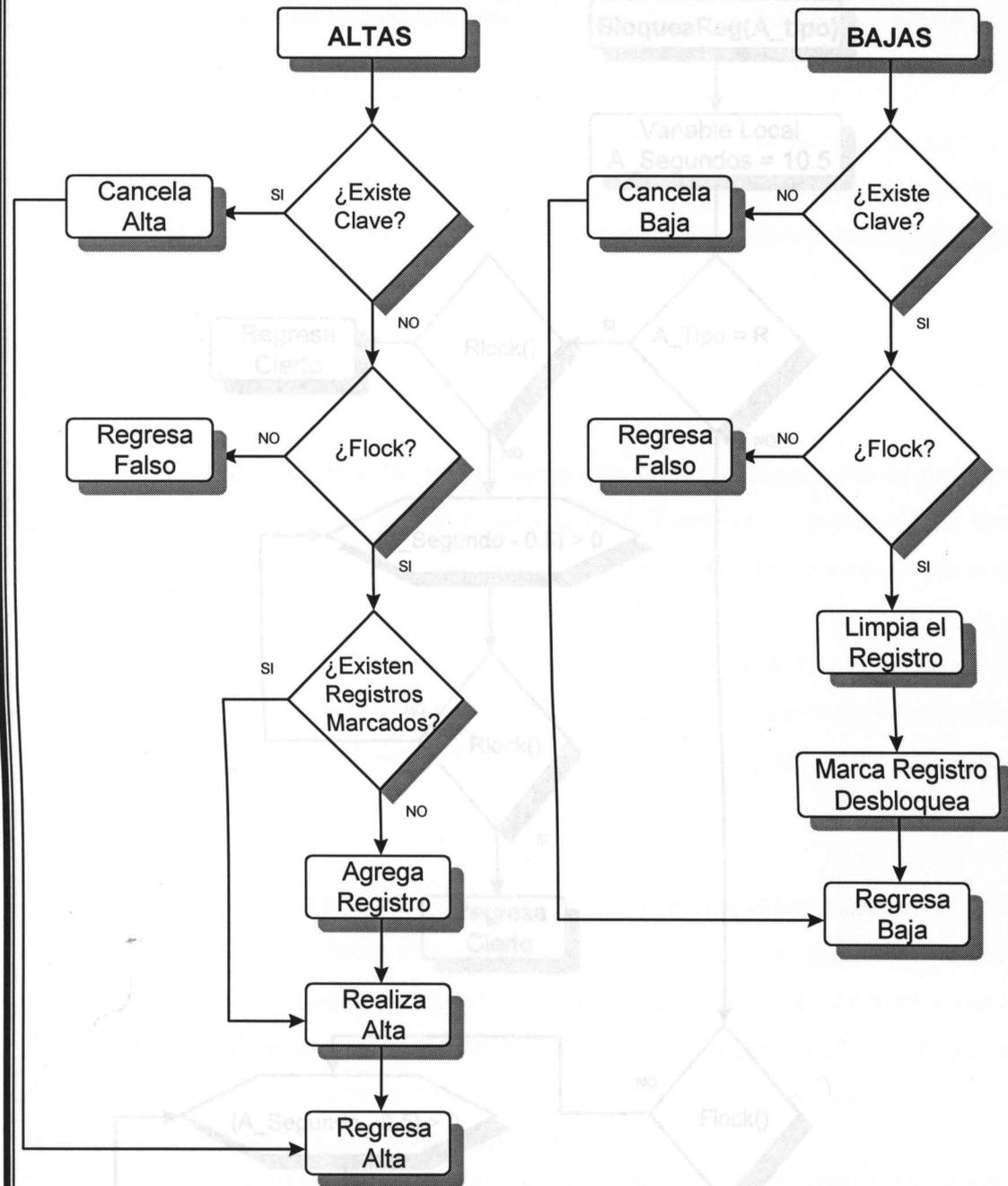
A continuación se describe brevemente la estrategia utilizada para poder trabajar un número no determinado de personas en el sistema de red.

- Se localiza la base de datos según el módulo en que se desee trabajar, dicha base de datos podrá ser la misma para todos o diferente para cada persona.
- Se bloquea el archivo en el momento en que se intente dar de alta algún movimiento o el registro si el movimiento corresponde a bajas o modificaciones
- Se desbloquea el archivo; en este momento las demás estaciones de trabajo están en espera y podrán bloquear el archivo y realizar la misma operación.

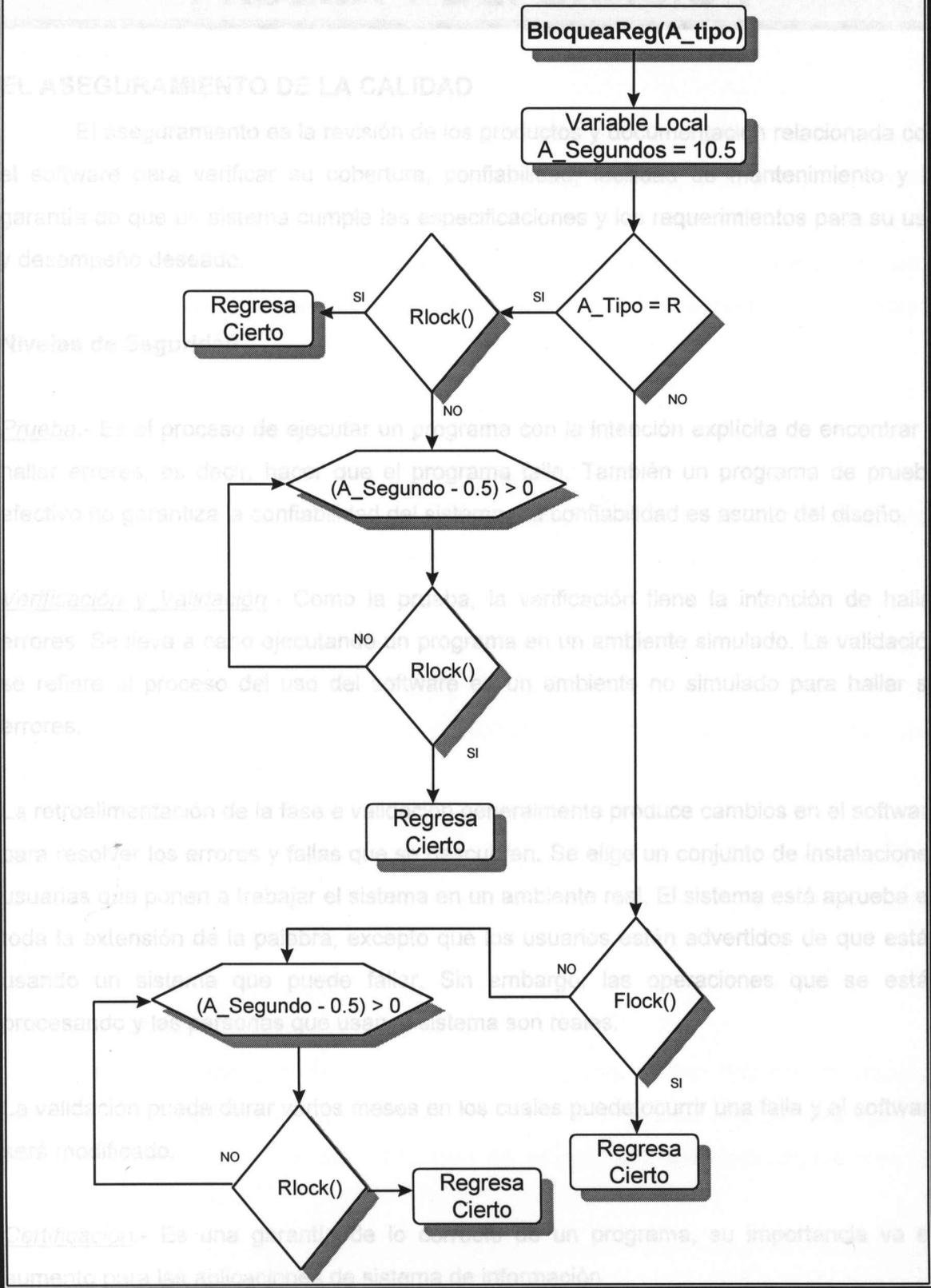
Cabe hacer la observación que el módulo de distribución es el único que no está diseñado para trabajar en red, debido a que es el lugar donde se dan de alta las Claves Presupuestales que realizarán erogaciones y dichas claves solamente podrán darse de alta en un periodo comprendido del 01 de Enero al 31 de Enero de cada año fiscal.



FUNCIONES PARA HACER MOVIMIENTOS DE ALTAS Y BAJAS



FUNCIONES PARA BLOQUEAR ARCHIVOS Y REGISTROS



PRUEBA Y DEPURACION

que el sistema hace y cómo lo lleva a cabo.

EL ASEGURAMIENTO DE LA CALIDAD

ESTR El aseguramiento es la revisión de los productos y documentación relacionada con el software para verificar su cobertura, confiabilidad, facilidad de mantenimiento y la garantía de que un sistema cumple las especificaciones y los requerimientos para su uso y desempeño deseado.

El sistema procesará como entrada normal. Sin embargo, los datos se crean con la intención expresa de determinar si el sistema los procesará

Niveles de Seguridad

Prueba.- Es el proceso de ejecutar un programa con la intención explícita de encontrar o hallar errores, es decir, hacer que el programa falle. También un programa de prueba efectivo no garantiza la confiabilidad del sistema. La confiabilidad es asunto del diseño.

casos que produzcan la ejecución de cada instrucción en el programa. Sin embargo, no

Verificación y Validación.- Como la prueba, la verificación tiene la intención de hallar errores. Se lleva a cabo ejecutando un programa en un ambiente simulado. La validación se refiere al proceso del uso del software en un ambiente no simulado para hallar su errores.

condiciones que señalan lo que el programa debe hacer y cómo lo debe de hacer bajo diferentes condiciones.

La retroalimentación de la fase e validación generalmente produce cambios en el software para resolver los errores y fallas que se descubren. Se elige un conjunto de instalaciones usuarias que ponen a trabajar el sistema en un ambiente real. El sistema está aprueba en toda la extensión de la palabra, excepto que los usuarios están advertidos de que están usando un sistema que puede fallar. Sin embargo, las operaciones que se están procesando y las personas que usan el sistema son reales.

Independientemente de cual sea la estrategia que se siga hay ciertas prácticas

La validación puede durar varios meses en los cuales puede ocurrir una falla y el software será modificado.

junto con las bibliotecas de prueba, son aspectos importantes del proceso real de prueba

Certificación.- Es una garantía de lo correcto de un programa, su importancia va en aumento para las aplicaciones de sistema de información.

Una forma de garantizar el software es cuando una empresa asigna un equipo de especialista para que determinen cuidadosamente si la documentación del sistema es lo que el sistema hace y cómo lo lleva a cabo.

Las pruebas parciales se pueden llevar a cabo de manera ascendente, comenzando de programas pequeños y de nivel inferior. En este caso un pequeño

progr Ya se ha indicado que la filosofía detrás de la prueba es la de encontrar errores. Los casos de prueba se diseñan con este propósito en mente. Un caso de prueba es un conjunto de datos que el sistema procesará como entrada normal. Sin embargo, los datos se crean con la intención expresa de determinar si el sistema los procesará correctamente.

La prueba de sistema no prueba el software en sí, sino la integración de cada módulo en el sistema. También busca la discrepancia entre el sistema y la documentación del sistema. La preocupación principal es la compatibilidad de los módulos individuales.

Hay dos estrategias generales para la prueba de software.

Prueba de Código.- Examina la lógica del programa. Lo que se realiza es desarrollar casos que produzcan la ejecución de cada instrucción en el programa. Sin embargo, no es una garantía en contra de las fallas del software.

Se deben probar a nivel de sistema los procedimientos de ordenamiento y reindexación.

Prueba de especificación.- Para llevar a cabo esta prueba se examinan las especificaciones que señalan lo que el programa debe hacer y cómo lo debe de hacer bajo diferentes condiciones.

Son aquellos datos que se extraen del sistema real. Es difícil obtener datos reales en cantidad suficiente para conducir una prueba extensa. Esta estrategia trata al programa como si fuera una caja negra: no se estudia ni se prueban las instrucciones del programa, simplemente se comprueba que el sistema arroje los resultados que se esperan

Uso de Datos de Prueba Artificial - Son aquellos que se crean con fines de prueba, ya

comprobar todas las combinaciones de formatos y valores.

PRACTICAS DE PRUEBA

Independientemente de cual sea la estrategia que se siga hay ciertas prácticas preferidas para garantizar que la prueba sea útil. Los niveles de prueba y los tipos de datos de prueba, junto con las bibliotecas de prueba, son aspectos importantes del proceso real de prueba.

Pruebas Parciales.- Se prueban los programas que conforman a un sistema. Las pruebas parciales se centran en cada uno de los módulos del sistema, independientes entre sí, para localizar los errores.

Las pruebas parciales se pueden llevar a cabo de manera ascendente, comenzando de uno en uno con los módulos más pequeños y de nivel inferior. En este caso un pequeño programa (programa conductor) ejecuta al módulo y proporciona los datos necesarios, de esta forma se pide que el módulo se desempeñe en la forma en que lo haría al encajarse dentro del sistema.

Prueba de Sistemas.- La prueba de sistema no prueba el software en sí, sino la integración de cada módulo en el sistema. También busca la discrepancia entre el sistema y su objetivo original, especificaciones y documentación del sistema. La preocupación principal es la compatibilidad de los módulos individuales.

2. Procedimientos de conversión.

La prueba de sistemas también debe verificar que los tamaños de los archivos son adecuados y que los índices se han construido en forma adecuada. Se deben probar a nivel de sistema los procedimientos de ordenamiento y reindexación.

Los sistemas técnicamente elegantes y bien diseñados pueden tener éxito si

DISEÑO DE DATOS DE PRUEBA

Uso de Datos de Prueba Reales.- Son aquellos datos que se extraen del sistema real. Es difícil obtener datos reales en cantidad suficiente para conducir una prueba extensa. La inclinación hacia los valores típicos no proporciona entonces una verdadera prueba del sistema y de hecho ignora los casos más probables que causan la falla del mismo.

Uso de Datos de Prueba Artificial.- Son aquellos que se crean con fines de prueba, ya que se pueden generar para probar todas las combinaciones de formatos y valores.

1. Usuarios directos. - Son aquellas personas que van hacer uso del sistema, como la captura Documentos de Erogación (Ejecución Presupuestaria, Modificación, Múltiple, Requisición y Salida de Almacén). Además son quienes aprovechan todas las ventajas que ofrece el sistema y por los mismo deben conocerlo a profundidad de tal manera que deben estar capacitados para poder identificar los problemas, determinando si el problema que surge es causado por el equipo o por el software o por algo hecho por ellos al usar el sistema.

IMPLANTACION Y EVALUACION

La implantación incluye todas aquellas actividades que tienen lugar para realizar la transición del sistema anterior al nuevo. Como se ha mencionado anteriormente, el sistema anterior trabaja en forma manual y el que se pretende implantar nace de una necesidad por parte de los usuarios. Por eso es importante que la implantación del sistema debe dejar convencidos a los usuarios ya que si es adecuada será esencial para lograr un sistema confiable y que cumpla con las necesidades de la organización.

Son tres aspectos que se cubren en la implantación:

1. Capacitación del personal.
2. Procedimientos de conversión.
3. Revisión después de la implantación.

CAPACITACION DEL PERSONAL

Los sistemas técnicamente elegantes y bien diseñados pueden tener éxito o fracasar debido a la forma en que se operan y usan. Por lo tanto, la calidad de la capacitación recibida por el personal relacionado con el sistema ayuda u obstruye, y puede llegar a impedir, la implantación exitosa de un sistema de información. Aquellos que estén relacionados con el sistema deben conocer a detalles cuáles serán sus papeles, cómo pueden usar el sistema y qué hará o que no hará.

Los usuarios del sistema los podemos ubicar en dos categorías:

1. Usuarios directos.- Son aquellas personas que van hacer uso del sistema, como la captura Documentos de Erogación (Ejecución Presupuestaria, Modificación, Múltiple, Requisición y Salida de Almacén). Además son quienes aprovechan todas las ventajas que ofrece el sistema y por los mismo deben conocerlo a profundidad de tal manera que deben estar capacitados para poder identificar los problemas, determinando si el problema que surge es causado por el equipo o por el software o por algo hecho por ellos al usar el sistema.

6. Lo recomendable para este tipo de usuarios es la elaboración de un manual de operación además de una capacitación personal.

EVALUACION

2. Usuarios responsables.- Son aquellos que utilizan las salidas o reportes del sistema y que conocen con profundidad como funciona el sistema. En nuestro caso son las personas que reciben los reportes generados por el sistema de acuerdo al tipo de reporte que fueron descritos en el capítulo Diseño Lógico. De esta manera estos irán realizando una evaluación de flujo de efectivo que esta siguiendo una U.P.P. y/o Dependencia.

METODO DE CONVERSION

Conversión es el proceso de cambiar el sistema anterior al nuevo. El método que se recomienda para la instalación del sistema es el de sistemas paralelos ya que es el más seguro dado que los usuarios siguen operando el sistema anterior de la forma acostumbrada pero también comienzan a usar el sistema nuevo. Este método es el enfoque de conversión más seguro , ya que garantiza que, en caso de surgir problemas, como errores en el procesamiento o incapacidad de manejar ciertos tipos de transacciones en el nuevo sistema, la dependencia puede regresar al sistema anterior sin pérdida de tiempo.

Plan de Conversión

El plan de conversión incluye una descripción de todas las actividades que deben de ocurrir al implantar el sistema nuevo y ponerlo en operación. A continuación se mencionan las actividades a realizar:

1. Listar todos los archivos a convertir.
2. Identificar todos los datos necesarios para construir los archivos nuevos durante la conversión.
3. Listar todos los documentos nuevos y procedimientos que se usarán durante la conversión.
4. Identificar todos los controles a usar durante la conversión.
5. Asignar responsabilidades para cada actividad

6. Verificar los tiempos de conversión.

EVALUACION

Una vez terminadas estas etapas el sistema quedará funcionando durante el tiempo que sea necesario de manera cotidiana, y se estará evaluando periódicamente el funcionamiento para comprobar que el sistema cumple con las expectativas y determinar si es necesario que se realicen algunas mejoras.

El Sistema para el Control del Gasto Presupuestal nace como una necesidad de resolver un problema concreto: organizar y simplificar la labor de captura de las erogaciones que presenta la Dirección de Presupuesto del Gobierno del Estado. El objetivo era diseñar una herramienta computacional que

- A) Facilitara la captura de las erogaciones realizadas por parte de la Dirección de Presupuesto.
- B) Ofrecerá reportes actualizados acerca del estado financiero que guarda su Presupuesto de Egresos.
- C) Ordenará, clasificará, seleccionará y almacenará todas las erogaciones realizadas.

Una vez terminado el sistema se llegó a las siguientes conclusiones:

- ✓ El sistema es propositivo.- Satisface una necesidad real de usuarios reales.
- ✓ Ofrece un manejo eficiente de la información almacenada.- Los procesos de distribución, erogación y reportes están minuciosamente diseñados para el adecuado funcionamiento del sistema.
- ✓ Cumple con los objetivos planteados al principio del proyecto.- Con la evaluación final del sistema se puede decir que el sistema realiza las actividades que se esperaba de él y podemos concluir que la meta del proyecto fue alcanzada en su totalidad.
- ✓ Proporciona una interfaz de fácil manejo.- El diseño de las pantallas fue hecho pensando en la comodidad de los usuarios.

CONCLUSIONES Y RECOMENDACIONES

• El sistema como tal, puede funcionar en otras Dependencias del Gobierno del Estado.

• Estamos viviendo una época de constantes cambios y evoluciones donde la búsqueda del perfeccionamiento es el objetivo a seguir dentro de nuestro entorno, donde las necesidades y los problemas son nuestro quehacer cotidiano y ante ellos debemos de presentar una actitud positiva y propositiva, y aceptar el reto que nos presentan para aportar más hacia la humanidad.

• El sistema como tal, puede funcionar en otras Dependencias del Gobierno del Estado.

El Sistema para el Control del Gasto Presupuestal nace como una necesidad de resolver un problema concreto: organizar y simplificar la labor de captura de las erogaciones que presenta la Dirección de Presupuesto del Gobierno del Estado. El objetivo era desarrollar una herramienta computacional que:

- A) Facilitará la captura de las erogaciones realizadas por parte de la Dirección de Presupuesto.
- B) Ofrecerá reportes actualizados acerca del estado financiero que guarda su Presupuesto de Egresos.
- C) Ordenará, clasificará, seleccionará y almacenará todas las erogaciones realizadas.

Una vez terminado el sistema se llegó a las siguientes conclusiones:

- ✓ **El sistema es propositivo.-** Satisface una necesidad real de usuarios reales.
- ✓ **Ofrece un manejo eficiente de la información almacenada.-** Los procesos de distribución, erogación y reportes están minuciosamente diseñados para el adecuado funcionamiento del sistema.
- ✓ **Cumple con los objetivos planteados al principio del proyecto.-** Con la evaluación final del sistema se puede decir que el sistema realiza las actividades que se esperan de él y podemos concluir que la meta del proyecto fue alcanzada en su totalidad.
- ✓ **Proporciona una interfaz de fácil manejo.-** El diseño de las pantallas fue hecho pensando en la comodidad de los usuarios.

CONSIDERACIONES

- El sistema por si solo funciona como tal y podrá funcionar como subsistema si se une a los demás sistemas mencionados en el capítulo de Metas
- No podrá funcionar para otras instituciones públicas de otras entidades federativas.
- Esta diseñado para cumplir las necesidades de la Dirección de Presupuesto del Gobierno del Estado.
- El sistema como tal, puede funcionar en otras Dependencias del Gobierno del Estado.

PRESUPUESTO DE EGRESOS

Gobierno del Estado de Michoacán

Diciembre 1993

CLIPPER 6.2 A SU ALCANCE

Jose Javier Garcia-Baceli

Sare e McGraw-Hill de Informática Segunda Edición

España Madrid 1994

CLIPPER 3 REFERENCIA BASICA

Grupo EIDOS

Microbit. Segunda Edición.

México D.F. 1993

SOFTWARE DE SISTEMAS. INTRODUCCION A LA PROGRAMACION DE SISTEMAS

Leland L. Beck

Editorial Addison-wesley Iberoamericana

México D.F. 1986

FUNDAMENTOS DE BASES DE DATOS

Henry F. Korth, Abraham Silberschatz

Editorial McGraw-Hill

México D.F. 1993

BIBLIOGRAFIA

ANALISIS Y DISEÑO DE SISTEMAS DE INFORMACION

James A. Seen de haber ejecutado el comando (Presup00), aparecerá un cuadro de
Editorial McGraw -Hill. Segunda Edición
México, D.F. 1994

PRESUPUESTO DE EGRESOS

Gobierno del Estado de Michoacán

Diciembre 1993

MENU PRINCIPAL

CLIPPER 5.2 A SU ALCANCE

José Javier García-Badell
Serie McGraw-Hill de Informática. Segunda Edición
España, Madrid 1994

CLIPPER 5 REFERENCIA BASICA

Grupo EIDOS

Macrobot. Segunda Edición.

México D.F. 1993

SOFTWARE DE SISTEMAS. INTRODUCCION A LA PROGRAMACION DE SISTEMAS

Leland L. Beck

Editorial Addison-wesley Iberoamericana

México D.F. 1988

FUNDAMENTOS DE BASES DE DATOS

Hnery F. Korth, Abraham Silberschatz

Editorial McGraw-Hill

México D.F. 1993

MANUAL DEL USUARIO

primero se encarga de crear las Claves Presupuestales en donde se piensa realizar un gasto. Para dar de alta una Clave Presupuestal se capturará U.P.P., Dependencia,

CLAVE DE ACCESO

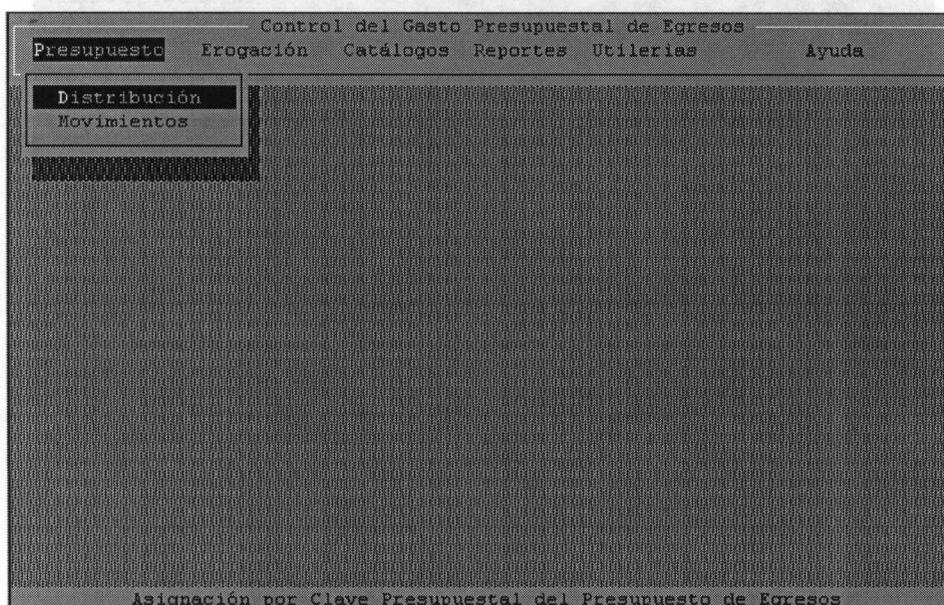
Después de haber ejecutado el comando (**Presup00**), aparecerá un cuadro de diálogo, pidiéndole que introduzca la clave de acceso al sistema. Si la clave es correcta, le pedirá que introduzca el año (de cuatro dígitos) en que desea trabajar, es decir, el año correspondiente al ejercicio del Presupuesto de Egresos. Si corresponde a un ejercicio nuevo, el sistema creará los archivos correspondientes. Si la clave fue incorrecta podrá volver a introducirla hasta dos veces más.

MENU PRINCIPAL

Los módulos del menú principal son: Presupuesto, Erogaciones, Catálogos, Reportes, Utilerías y Ayuda. Al momento ubicarse en alguno de los módulos mediante las teclas de dirección, el módulo automáticamente mostrará cuales son los elementos que lo componen. Se podrá tener acceso a cada uno de los elementos del módulo en que se encuentre la barra azul iluminada presionando la tecla <ENTER>

Para salir de cada elemento se presionará la tecla <ESC>.

Si desea abandonar el sistema deberá presionar la tecla <ESC>.



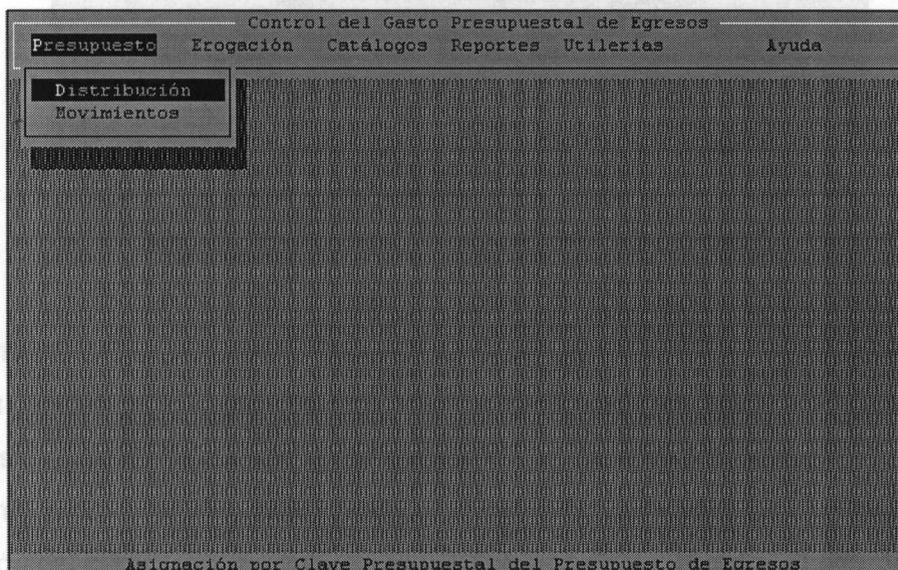
PRESUPUESTO

Los elementos que pertenecen al menú son: Distribución y Movimientos. El primero se encarga de crear las Claves Presupuestales en donde se piensa realizar un gasto. Para dar de alta una Clave Presupuestal se capturará U.P.P., Dependencia, Programa, Subprograma, Proyecto, Obra, Partida, Dígito Identificador, Región, Tipo de Calendario y Asignación una vez realizado si usted está seguro se presionará <F10> para dar de alta la Clave Presupuestal para realizar erogaciones, y por el contrario si usted no esta de acuerdo presionará <ESC> para cancelar o abandonar.

Observación: No se podrán hacer movimientos de alta de Claves Presupuestales fuera del periodo comprendido entre el 01 de Enero al 31 de Enero de cada ejercicio fiscal o al comienzo de cada año.

El segundo elemento se encarga de mostrar como han sido las erogaciones que ha ido realizado determinada Clave Presupuestal y el disponible que le queda. Asimismo pueden realizar consultas de movimientos de una determinada Clave Presupuestal en determinado rango de fecha presionando la tecla <F2> y nuevamente para salir la tecla <ESC>.

Para salir de cada elemento se presionará la tecla <ESC>.



La tecla <F2> sirve para consultar una determinada Clave Presupuestal en un determinado periodo. Si desea abandonar la consulta presione <ESC> para salir.

Distribución

Se encarga de crear las Claves presupuestales que realizarán erogaciones durante el ejercicio fiscal. Para dar de alta una Clave Presupuestal se requiere de presionar la tecla <ALT> más la letra de la palabra "Altas" iluminada de color rojo, en este caso la letra A (ALT + A). Aparecerá un pantalla de captura que le pedirá que vaya introduciendo el valor o la información adecuada para cada uno de los campos que se le indica. Después de haber completado la información de todos los campos y ubicado en el último campo (Asignación) presionará la tecla <ENTER> y aparecerá un cuadro de diálogo indicando si desea **Guardar** o **Cancelar** la acción de dar de alta. Si elige la opción de guarda se dará de alta la Clave Presupuestal en el Presupuesto de Erogaciones. Si elige cancelar volverá a la pantalla de captura para verificar y/o corregir algún dato mal escrito ubicándose en cada campo mediante las teclas de dirección. Volverá a ubicarse en el último campo (Asignación) y presionará la tecla <ENTER> apareciendo nuevamente el cuadro de diálogo descrito anteriormente.

Altas Bajas

DISTRIBUCION PRESUPUESTARIA

Clave Presupuestal	Asignacion	Parcial	Fecha
0101-0101-01			1/01/97
U.P.P.	01	Dependencia 01	1/02/97
Programa	01	Subprograma 01	1/03/97
Proyecto	01	Obra 01	1/04/97
Partida	1200	Subregión 1	1/05/97
Digito Iden		[Grabar]	1/06/97
Asignación		[G] Guardar o Cancelar	1/07/97
			1/08/97
Ene	8.33	Feb 8.33	Mar 8.33
May	8.33	Jun 8.33	Jul 8.33
Sep	8.33	Oct 8.33	Nov 8.33
		Dic 8.37	1/11/97
		ESC -> Salir	1/12/97
0101-0101-0101-1201-91	120,000.00	10,000.00	01/01/97
		10,000.00	01/02/97

Clave Presupuestal 0101-0101-0101-1201-11

Movimientos

Su función es mostrar los movimientos de financieros que ha sufrido una Clave Presupuestal. Las teclas de operación son las teclas de dirección para poder desplazarse de izquierda a derecha y de arriba hacia abajo o en ambos casos a la inversa según sea la ubicación del cursor.

La tecla <F2> sirve para consultar una determinada Clave Presupuestal en un determinado periodo. Si desea abandonar la consulta presione <ESC> para salir.

<F2> -> Consultas

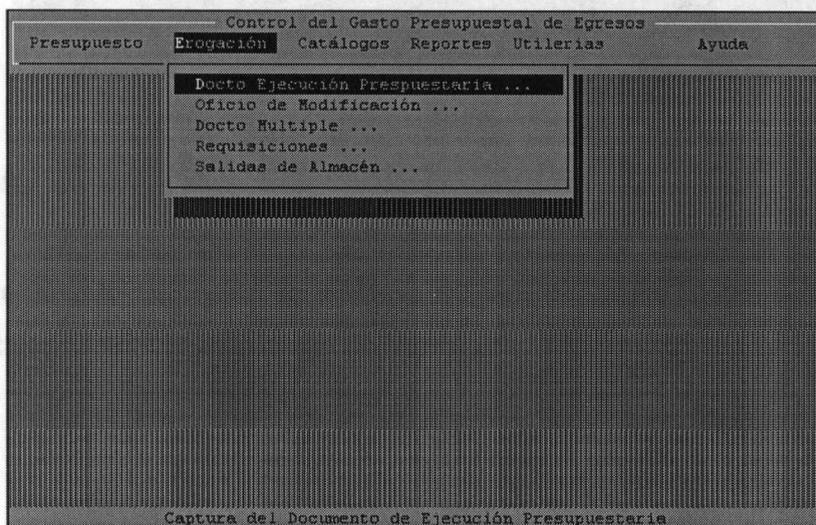
MOVIMIENTOS DE CLAVES PRESUPUESTALES

Clave Presupuestal	Fecha	Asignación
010	Movimientos	
Clave Presupuestal	0101-0101-0101-1201-11	
Fecha Inicial	01/07/1997	
Fecha Final	22/07/1997	
	01/08/1997	
	01/07/1997	
	01/08/1997	
	01/09/1997	
	01/10/1997	
	01/11/1997	
	01/12/1997	
0101-0101-0101-1201-91	01/01/1997	120,000.00
	01/02/1997	

Clave Presupuestal 0101-0101-0101-1201-11

EROGACIONES

Este elemento de menú se divide a su vez en varios elementos: Documento de Ejecución Presupuestaria, Oficio de Modificación, Documento Múltiple, Requisiciones y Salidas de Almacén.



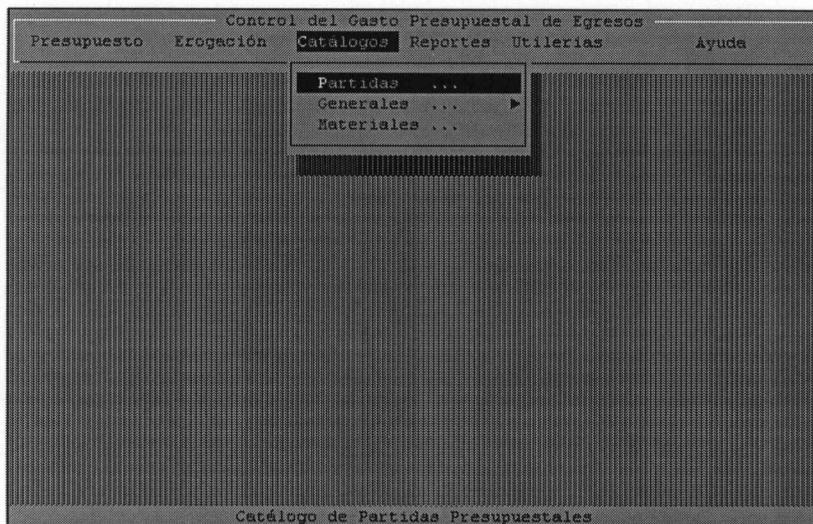
Cada elemento del menú de Erogaciones contiene en su parte superior una barra de proceso: Altas, Bajas y Consultas. Para tener acceso a una de ellas basta con presionar la tecla <ALT> y la letra de color rojo que resalte de la palabra. Ejemplo: Para dar de Alta usted deberá presionar <ALT> + "A" porque es la letra que esta iluminada de color rojo.

Para salir de cualquier pantalla de captura usted presionará <ESC>.

Una vez que esté capturando un documento del menú de Erogaciones, para guardar la captura deberá presionar la tecla <F10> y para cancelar la tecla <ESC> la cual, al momento de presionarse, le preguntará en un cuadro de dialogo de advertencia, si desea guardar los cambios o no.

No	Clave Presupuestal	D	Beneficiario
1	0101-0101-0101-1201-11	8	H. CONGRESO DEL ESTADO

Una vez que esté capturando un documento del menú de Catálogo, para guardar **CATALOGOS** será presionar la tecla <F10> y para cancelar la tecla <ESC> la cual, al momento de presionarse, le preguntará en un cuadro de dialogo de advertencia, si desea guardar. Este elemento de menú se divide a su vez en varios elementos: Catálogo de Partidas, Catálogo General y Catálogo de Materiales.

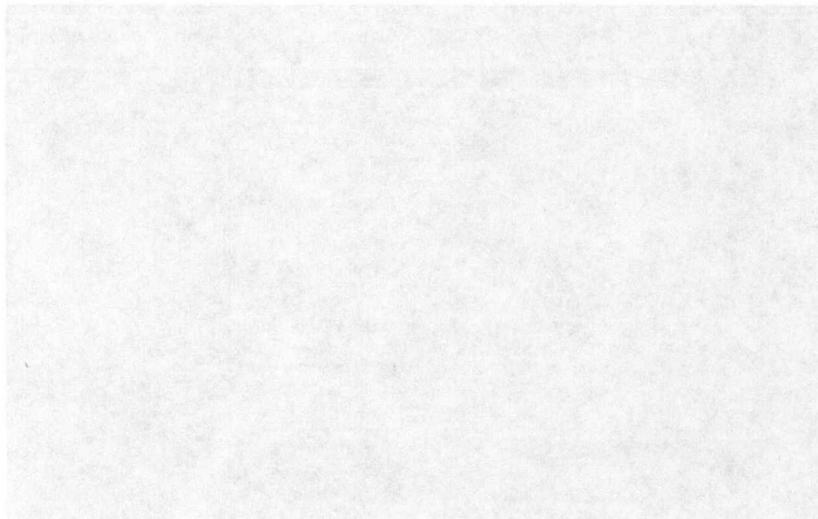


REPORTES Cada elemento del menú de Catálogo contiene en su parte superior una barra de proceso: Altas, Bajas y Modificaciones. Para tener acceso a una de ellas basta con presionar la tecla <ALT> y la letra de color rojo que resalte de la palabra. Ejemplo: Para dar de Alta usted deberá presionar <ALT> + "A" porque es la letra que esta iluminada de color rojo.

Para salir de cualquier pantalla de captura usted presionará <ESC>.

Existen varios tipos de reportes, entre los más comunes son los por U.P.P. y

Depositos Una vez que esté capturando un documento del menú de Catálogo, para guardar la captura deberá presionar la tecla <F10> y para cancelar la tecla <ESC> la cual, al momento de presionarse, le preguntará en un cuadro de dialogo de advertencia, si desea guardar los cambios o no.



PARTIDAS PRESUPUESTALES		
Partidas	Descripción Corta	Descripción Larga
1000	SERVICIOS PERSONALES	SERVICIOS PERSONALES
1100	REM. BASICA AL PERSONAL PERM	REMUNERACIONES BASICAS AL PE
1101		
1102		
1103		
1200		ONAL E
1201		S
1202		VENTUA
1203		IOS
1207		N LICE
2500		ACEUTI
2501	SUBSTANCIAS QUIMICAS	SUBSTANCIAS QUIMICAS

Partida: 500
 Descripción Corta: SUELDO BASE
 Descripción Larga: SUELDO BASE DEL PERSONAL

ESC -> Salir F10 -> Grabar

REPORTES

Cada elemento del menú de Reportes verifica que la impresora se encuentre lista, es decir, su estado sea el encendido. Si la impresora se encontrará apagada el programa manda una advertencia anunciando que se encuentra apagada y preguntando a usted si desea reintentar la impresión encendiendo la impresora.

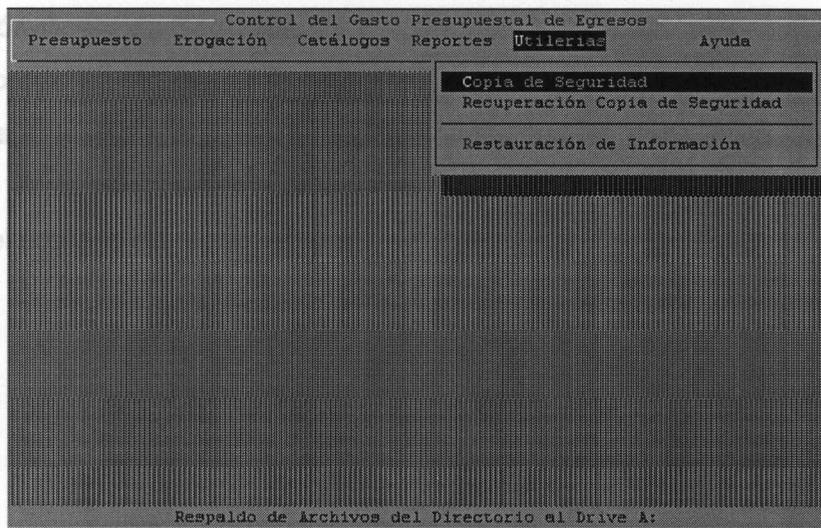
Existen varios tipos de reportes, entre los más comunes son los por U.P.P. y Dependencia en los cuales el usuario elige el tipo de reporte que desea obtener digitando a continuación la U.P.P. y Dependencia por lo que el programa buscará información que sea común a la U.P.P. y Dependencia por la cual se esta preguntando.

Control del Gasto Presupuestal de Egresos	
Presupuesto	Erogación
Catálogos	Reportes
Utilerías	Ayuda
Catálogo de Partidas Catálogo por U.P.P. y Dependencia Catálogo por Programa y Subprograma Catálogo por Proyecto y Obra Catálogo de Materiales Asignación por U.P.P. y Dependencia Asignación por Partida Presupuestal Movimientos por U.P.P. y Dependencia Movimientos por Clave Presupuestal Ejecución Presup. por U.P.P. y Dependencia Oficio de Modificación por U.P.P. y Dependencia Documento Múltiples por U.P.P. y Dependencia Comprometido por U.P.P. y Dependencia Recursos Materiales por U.P.P. y Dependencia Servicios Generales por U.P.P. y Dependencia	
Imprime catálogo de Partidas Presupuestales	

Los reportes de Asignación por Partida Presupuestal, Movimiento Presupuestal por U.P.P. y Dependencia y el reporte de Movimiento Presupuestal por Clave Presupuestal, deberán ser impresos en un tamaño de papel 15 x 11 y los demás reportes en papel 9½ x 11.

UTILERIAS

Los elementos del menú de utilerías son herramientas que permiten al usuario cuidar y proteger su información; el elemento **Copia de Seguridad** sirve para crear una copia de resguardo de la información procesada en la unidad A. La **Recuperación Copia de Seguridad** sirve para tomar la información guardada en un disco flexible y ponerla nuevamente en el disco duro. La **Restauración de Información** ayuda al usuario a reindexar o mantener los índices en buen estado.

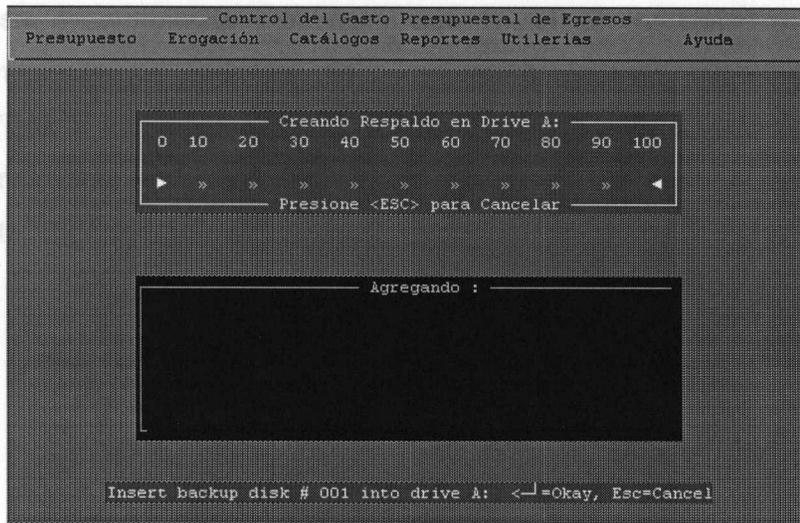


Copia de Seguridad

Crea una copia de todos los archivos con extensión DBF y NTX del sistema en forma de "Backup". Deberá de insertar un disco flexible limpio, previamente formateado, en la unidad de 3½. Una vez hecho lo anterior, presionará la tecla <ENTER> para que inicie el proceso de copia de archivos en la unidad de 3½. Si la longitud o el tamaño de los archivos excediera de la capacidad del almacenamiento del disco flexible, el sistema le pedirá que inserte otro disco flexible para continuar, y así, sucesivamente hasta que los archivos fuera copiados en su totalidad.

Restauración de información

Consiste en indicar al usuario acerca del funcionamiento bajo el que opera el sistema, de modo que se escriba y deba ser guardado para regresar a la pantalla de la pantalla

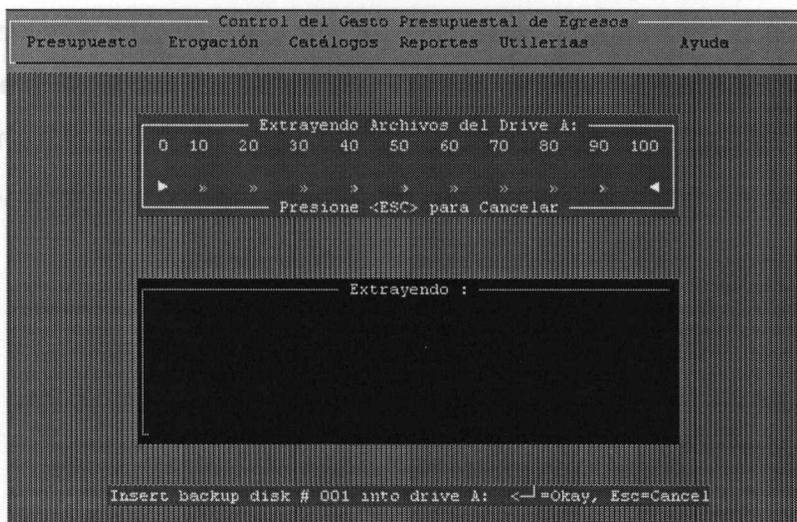


datos que se encuentran en la cual se iniciará el proceso de restauración si elige

Recuperación Copia de Seguridad

El proceso de recuperación se encarga de extraer los archivos, que se encuentran almacenados en el disco flexible, depositándolos en el disco duro en un directorio denominado **Presresp**. Deberá de insertar el primer disco flexible del respaldo hecho para iniciar el proceso de recuperación.

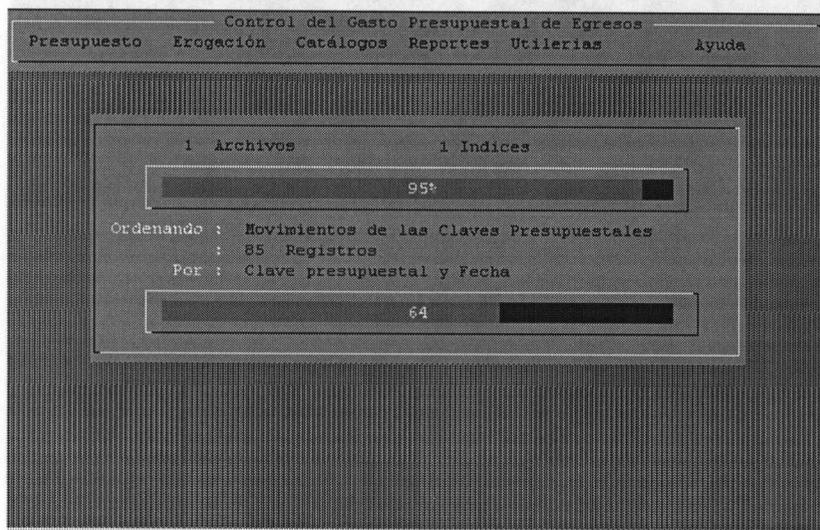
Consiste en indicarle al usuario acerca del funcionamiento bajo el que opera el sistema, de modo que se escriba y deba ser guardado para regresar a la pantalla de la pantalla



aparecerá el mensaje de ayuda y <Re Pág>, parte inferior de la pantalla

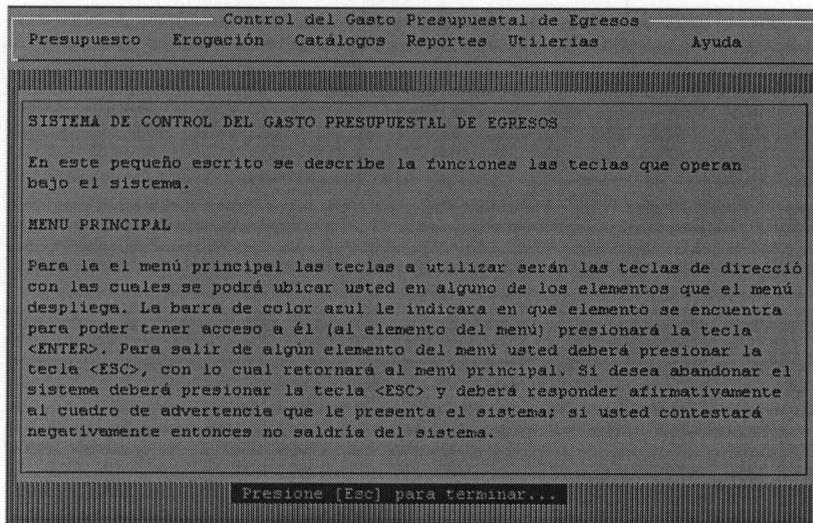
Restauración de Información

Consiste en ordenar la información que existen en las bases de datos que se encuentran indexadas por un índice. Aparecerá una pantalla de advertencia en la cual le preguntará: "Desea dar mantenimiento? Si No". Si elige afirmativamente se iniciará el proceso de restauración o reordenación de la información y por el contrario si elige negativamente no se realizará el proceso.



AYUDA

Consiste en indicarle al usuario acerca del funcionamiento bajo el que opera el sistema, de manera concisa. Para ello se desplegará una pantalla donde aparecerá el escrito y deberá de utilizar las teclas <Av Pág>, para avanzar hacia abajo, y <Re Pág>, para regresar hacia arriba. Para salir de la ayuda existe un mensaje en la parte inferior de la pantalla, al centro, indicándole que mediante la tecla <ESC> abandonará la ayuda.



CODIGO
FUENTE

BIBLIOTECA GENERAL

```

#include "INKEY.CH"
#include "Box.ch"
#define Cierto .T.
#define Falso .F.

Public lControl:= .F.
*-----*
Function LB_DispBox( nExp1, nExp2, nExp3, nExp4, cExp1,
cExp2, nExp5, lExp )
//
LOCAL nColumnaMsg
//
IF cExp1 = NIL; cExp1 := SetColor(); ENDIF
IF lExp = NIL; lExp := .F.; ENDIF

DispBox(nExp1,nExp2,nExp3,nExp4,"ÉÍ»¼ÍÈ° ",cExp1)
IF cExp2#NIL
nColumnaMsg := nExp2+(((nExp4-nExp2)-
(Len(cExp2)+4))/2)
@ nExp1,nColumnaMsg SAY '['+cExp2+' ]' COLOR cExp1
RP_Shadow( nExp1,nExp2,nExp3,nExp4 )
ENDIF
// @ nExp1,nExp4-6 SAY '['+Str(nExp5,2)+' ]' COLOR cExp1
IF lExp; RP_Shadow( nExp1,nExp2,nExp3,nExp4 ); ENDIF
//
RETURN( NIL )

*-----*
Function RP_Shadow(Rens,Coli,Reni,Cold,Exp_C)
//
Local Car2:=Cad1:=",Cont:=0
Local M_C:=SetColor()
//
Reni++;Rens++;Coli+=2;Cold++
setcolor(IIF(Exp_C==NIL,"+N/N",Exp_C))
For Cont:=Coli to Cold
Cad1+=Left(SaveScreen(Reni,Cont,Reni,Cont),1)
Next
@ Reni,Coli Say Cad1
For Cont:=Rens To Reni-1
Car2:=SaveScreen(Cont,Cold,Cont,Cold)
@ Cont,Cold Say Left(Car2,1)
Next
setcolor(M_C)
//
return Nil

*-----*
FUNCTION MSG_Error(cMens)
LOCAL Guarda
Setcursor(0)
Guarda:=SaveScreen(24,0,24,79)
Tone(700,1)
@ 24,0 SAY REPLICATE("Ú",80) COLOR "R"
@ 24,INT((80-LEN(cMens))/2) SAY cMens COLOR "W+R"
Inkey(1.5)
RestScreen(24,0,24,79,Guarda)
RETURN NIL

*-----*
Function Opcion(Cad,M_C)
//
Local C2,C4,Col,Valor,C_A:=SetColor()
//

```

```

If M_C!=Nil
SetColor(M_C)
ENDIF
Col:=INT((79-LEN(Cad))/2)
C2:=Col-3
C4:=Col+LEN(Cad)+3
// Tone(450,1);Tone(700,1)
Vrarp:=SaveScreen(8,C2,15,C4+2)
RP_Shadow(8,C2,14,C4)
A_DispBox( 8,C2,14,C4,"R",.T. )
@ 10,Col Say Cad
@ 12,37 Prompt "Si"
@ 12,43 Prompt "No"
Menu To Opcion
//
Do Case
Case Opcion = 0
Valor := .F.
Case Opcion = 1
Valor := .T.
Case Opcion = 2
Valor := .F.
End Case
//
SetColor(C_A)
RestScreen(8,C2,15,C4+2,Vrarp)
//
Return(Valor)

*****
FUNCTION BusRegMB() // Busca Registro Marcado Con Indice
Nfmerico
Set Deleted Off // Se activan los registros borrados
If IndexOrd() > 0
Seek 0
If Found()
If BloqueaReg('R')
Recall
Set Deleted On
Return(.T.)
ENDIF
ENDIF
ENDIF
Set Deleted On
Return(.F.)

*****
FUNCTION BusRegMB_A() // Busca Registro Marcado Con
Indice Caracter
Set Deleted Off // Se activan los registros borrados
If IndexOrd() > 0
Seek ''
If Found()
If BloqueaReg('R')
Recall
Set Deleted On
Return(.T.)
ENDIF
ENDIF
ENDIF
Set Deleted On
Return(.F.)

*****
* Realiza 20 intentos para bloquear el archivo o registro
especificado *

```

```

* en A_TIPO (correspondiendo 'A' para archivo y 'R' para un
registro) *
* durante 10 segundos. Regresa .T. si tiene exito. *
* Si no se pasa ningun parametro por defecto se bloquear el
archivo. *
*****

```

```

FUNCTION BloqueaReg(A_Tipo)
Local A_Segundos:=10.5

```

```

If A_Tipo='R'
  If Rlock()
    Return(.T.)
  EndIf
  While((A_Segundos=.5)>0)
    If Rlock()
      Return(.T.)
    EndIf
    Inkey(.5)
  EndDo
Else
  If Flock()
    Return(.T.)
  EndIf
  While((A_Segundos=.5)>0)
    IF Flock()
      Return(.T.)
    EndIf
    Inkey(.5)
  EndDo
EndIf
Return(.F.)

```

```

*****
* Intenta agregar un nuevo registro al archivo durante 10
segundos *
* retorna .T. si tiene exito. *
*****

```

```

FUNCTION AgregaReg(Modo)
Local A_Segundos:=10.5
IF Modo == '1'
  If BusRegMB();Return(.T.);EndIf
ENDIF
IF Modo == '2'
  If BusRegMB_A();Return(.T.);EndIf
ENDIF
Append Blank
If !NetErr()
  Return(.T.)
EndIf
While((A_Segundos=.5)>0)
  Append Blank
  If !NetErr()
    Return(.T.)
  EndIf
  Inkey(.5)
EndDo
Return(.F.) // NO HUBO EXITO

```

```

*****
* Función para limpiar los campos del registro que se cala el
puntero, y lo *
* marca para borrar. Retorna .T. si hubo exito. (Area de trabajo
actual) *
*****

```

```

FUNCTION LimpiaReg()
Local A_NomCampo,A_Cont,A_Tipo
If BloqueaReg('R')
  For A_Cont:=1 to FCount()
    A_NomCampo:=Field(A_Cont)
    A_Tipo:=ValType(&A_NomCampo)
    Do Case
      Case A_Tipo = 'C'
        Replace &A_NomCampo With "
      Case A_Tipo = 'N'
        Replace &A_NomCampo With 0
      Case A_Tipo = 'D'
        Replace &A_NomCampo With ctod("")
      Case A_Tipo = 'M'
        Replace &A_NomCampo With "
      Case A_Tipo = 'L'
        Replace &A_NomCampo With .F.
    EndCase
  Next
  Delete
  UnLock
  Return(.T.)
EndIf
UnLock
Return(.F.)

```

```

/*FUNCTION PON_CEROS()
RETURN PadL(LTrim(Str(Val(cExp))))4,0) */

```

```

*****
FUNCTION Saldo(cTemp_cv)
LOCAL nAsig_Net := 0, nAmplia:= 0, nReduce:= 0, nSaldo:=
0, h := 1, Mesprimo := 1,;
nPos := 0, nCargos := 0, nComprom := 0, nAbono := 0,
nTotal := 0
Select Dar
DbSeek(cTemp_cv, Cierto)
Do While (!Eof()) .And. Temp_cv == cTemp_cv
  IF BloqueaReg('R')
    IF Mesprimo != Month(Fecha_Mov)
      nSaldo = 0
      nAmplia = 0
      nReduce = 0
      nCargos = 0
      nAbono = 0
      nComprom= 0
      Mesprimo = Month(Fecha_Mov)
    ENDIF
    nAmplia += Amplia
    nReduce += Reduce
    nCargos += Cargo
    nAbono += Abono
    nComprom += Comprom
    nSaldo := (nAmplia - nReduce) + nSaldo
    nSaldo := nSaldo + Asig_Parci
    nTotal := nSaldo - nCargos - nComprom + nAbono
  REPLACE Asig_Net WITH nSaldo, Total WITH nTotal
  UNLOCK
  COMMIT
  Skip
ELSE
  Tone(800)
  G_Dspmsgg("Registro Ocupado. Intente de Nuevo","R")
  Exit
ENDIF
EndDo

```

```

While(j > 0 )
  Vartemp:=Substr(Clave_Pres,j,1)
  If j == 1 .OR. j == 9 .OR. j == 18
    Producto:= Val(Vartemp) * d
  Else
    Producto:= Val(Vartemp) * d
  EndIf
  j--
  If d == 9
    d:= 1
  Endif
  d++
  Resultado += Producto
EndDo
Residuo := Mod(Resultado,11)
Final1 := 11 - Residuo
If Final1 = 10 .OR. Final1 = 11
  Final1 := 0
EndIf
If Str(Final1,1) = cDig
  Return(cDig)
Else
  Tone(200)
  G_Dspmsg("El Dígito No Corresponde a la Clave","N*")
  oCuerpo:Left()
  Return(cDig)
EndIf
*****
FUNCTION Verifica_Dig(cDig)
Local Residuo:=0, Resultado:=0, Temp, Vartemp, j:= 18, d:=2,
Producto:=0,
Final:=0
While(j > 0 )
  Vartemp:=Substr(Clave_Pres,j,1)
  If j == 1 .OR. j == 9 .OR. j == 18
    Producto:= Val(Vartemp) * d
  Else
    Producto:= Val(Vartemp) * d
  EndIf
  j--
  If d == 9
    d:= 1
  Endif
  d++
  Resultado += Producto
EndDo
Residuo := Mod(Resultado,11)
Final1 := 11 - Residuo
If Final1 = 10 .OR. Final1 = 11
  Final1 := 0
EndIf
If Str(Final1,1) = cDig
  Return(.T.)
Else
  Tone(200)
  G_Dspmsg("El Dígito No Corresponde a la Clave","R")
  oCuerpo:Left()
  Return(.F.)
EndIf
*****
FUNCTION Tipo_Just(cVariab)
If cVariab = "F" .OR. cVariab = "N" .OR. cVariab = "LR" ;
  .OR. cVariab = "R" .OR. cVariab = "E" .OR. cVariab = "O"
  Return(Upper(cVariab))
ELSE

```

```

Tone(200)
  G_Dspmsg("El Tipo de Comprobante no Existe ","N*")
  oCuerpo:Left()
Endif
Return (Upper(cVariab))
*****
FUNCTION Tipo(cExpres)
If cExpres = "C" .OR. cExpres = "c" .OR. cExpres = "L" .OR.
cExpres = "I"
  Return(Upper(cExpres))
Else
  Tone(200)
  G_Dspmsg("El Tipo no Existe","N*")
  oCuerpo:Left()
Endif
Return (Upper(cExpres))
*****
FUNCTION Oper(cExpres)
If cExpres = "02" .OR. cExpres = "21" .OR.
cExpres = "22"
  Return(Upper(cExpres))
Else
  Tone(200)
  G_Dspmsg("El Tipo de Operación no Corresponde","N*")
  oCuerpo:Left()
Endif
Return (Upper(cExpres))
*****
FUNCTION VERIF(cExp)
Local Aux
Select General
Aux = "U"+ cExp
SEEK(Aux)
IF Found()
  Select 2
  Return(.T.)
ELSE
  G_Dspmsg("La U.P.P. no se Encuentra en Cat logo","N*")
  Select 2
  Return(.F.)
ENDIF
Return(.F.)
*****
FUNCTION ITS_OK(cExp)
Local Aux
Select General
Aux = "U"+ cUPP + cExp
SEEK(Aux)
IF Found() .AND. cExp != "00"
  Select 2
  Return(.T.)
ELSE
  G_Dspmsg("La Dependencia no se Encuentra en
Cat logo","N*")
  Select 2
  Return(.F.)
ENDIF
Return(.F.)
*****
FUNCTION Material(cClave)
Select Utencilios
SEEK(cClave)

```

```

IF Found()
  Select Temp
  Return(Utencilios -> Material)
ELSE
  G_Dspmsg("La Clave del Material no se Encuentra en
Cat logo","N*")
  oCuerpo:Left()
  Select Temp
  Return(cClave)
ENDIF
Return(Nil)

*****
FUNCTION MatDesc(cMaterial)
Select Utencilios
SEEK(cMaterial)
Select Temp
Return(Utencilios -> Descrip)

*****
FUNCTION MatCosto(cMaterial)
Select Utencilios
SEEK(cMaterial)
Select Temp
Return(Utencilios -> Costo)

*****
FUNCTION MatTipo(cMaterial)
Select Utencilios
SEEK(cMaterial)
Select Temp
Return(Utencilios -> Tipo)

*****
FUNCTION Check_Material(cClave)
Select Utencilios
SEEK(cClave)
IF Found()
  Select Temp
  Return(.T.)
ELSE
  G_Dspmsg("La Clave del Material no se Encuentra en
Cat logo","R")
  oCuerpo:Left()
  Select Temp
  Return(.F.)
ENDIF

*****
FUNCTION Busca_Material(cExp)
Local Cve1, Cve2
Go Top
Locate For Clave == cExp
Do While Found() .AND. !Eof()
  Continue
  IF Found()
    G_Dspmsg("No se puede dar de Alta la misma Clave del
Material","R")
    Return(Falso)
  ENDIF
ENDDO
Return(Cierto)

*****
FUNCTION NombreUnico()
LOCAL cNombre := NombreIniUnico()
DO WHILE NombreIniUnico() == cNombre

```

```

EndDo
RETURN cNombre

FUNCTION NombreIniUnico()
//RETURN Right(Str(Seconds(), 9, 2), 1) + ;
// PadL(AllTrim(Str(Seconds() * 100, 7)), 7, "0")
RETURN "Z"+ PadL(AllTrim(Str(Seconds() * 100, 7)), 7, "")

*****
FUNCTION NombreCon()
LOCAL cNombre := NombreConsUnico()
DO WHILE NombreConsUnico() == cNombre
EndDo
RETURN cNombre

FUNCTION NombreConsUnico()
//RETURN Right(Str(Seconds(), 9, 2), 1) + ;
// PadL(AllTrim(Str(Seconds() * 100, 7)), 7, "0")
RETURN "AS"+ PadL(AllTrim(Str(Seconds() * 100, 7)), 6, "")

*-----*
FUNCTION G_Dspmsg(cExpMsg,cColorF)
//
LOCAL nCol:= ( 79-Len( cExpMsg ) )/2 , nC2:=nC4:=0,
cPant,;
nCursor := SetCursor( 0 )
//
nC2:=nCol-2
nC4:=nCol+LEN( cExpMsg )+2
cPant := SaveScreen( 8,nC2,13,nC4 )
G_DispBox(8,nC2,13,nC4,cColorF,.T. )
@ 9,nCol SAY cExpMsg COLOR "+W/" +cColorF
G_Boton( 10,36,cColorF,"", " Ok ",.T. )
// Tone( 600,1 )
// Tone( 800,1 )
Inkey( 60 )
G_PBoton( 10,36,cColorF,"", " Ok ",.T. )
SetCursor( nCursor )
RestScreen( 8,nC2,13,nC4,cPant )
//
RETURN( NIL )

/*-----*/
// Función para desplegar una caja con efecto, línea sencilla.
// nX1,nY1,nX2,nY2 - Coordenadas de la ventana.
// cColorFondo - Color de fondo de la ventana.
// IExp - Efecto de la ventana
// (T. Hacia Afuera, .F. hacia adentro)
/*-----*/
FUNCTION G_DispBox( nX1,nY1,nX2,nY2,cColorFondo,IExp )
LOCAL cTColor:=IFF( Right( cColorFondo,1 )
=="W", "+W/", "W/" )
IExp := IIF( IExp == NIL,.F.,IExp )
IF IExp
  DispBegin()
  DispBox(
nX1,nY1,nX2,nY2,B_SINGLE,cTColor+cColorFondo )
  DispBox( nX1+1,nY2-1,nX2,nY2," ÜÄ
",N/" +cColorFondo )
  @ nX2,nY1+1 SAY Replicate('Ä',nY2-nY1-1 ) COLOR
"N/" +cColorFondo
  @ nX1,nY2 SAY '¿' COLOR "N/" +cColorFondo
  DispEnd()
ELSE
  DispBegin()
  DispBox(
nX1,nY1,nX2,nY2,B_SINGLE,"N/" +cColorFondo )

```

```

    DispBox( nX1+1,nY2-1,nX2,nY2," ³³ÜÄ
",cTColor+cColorFondo )
    @ nX2,nY1+1 SAY Replicate('Ä',nY2-nY1-1 ) COLOR
cTColor+cColorFondo
    @ nX1,nY2 SAY '¿' COLOR cTColor+cColorFondo
    DispEnd()
    ENDIF
RETURN( NIL )
/*-----*/
// Función para desplegar una caja con efecto, línea doble.
// nX1,nY1,nX2,nY2 - Coordenadas de la ventana.
// cCF - Color de fondo de la ventana.
// lExp - Efecto de la ventana
// (T. Hacia Afuera, F. hacia adentro)
/*-----*/
FUNCTION A_DispatchBox( nX1,nY1,nX2,nY2,cCF,lExp )
IF lExp
    G_DispatchBox( nX1,nY1,nX2,nY2,cCF,T. )
    G_DispatchBox( nX1+1,nY1+1,nX2-1,nY2-1,cCF,F. )
ELSE
    G_DispatchBox( nX1,nY1,nX2,nY2,cCF,F. )
    G_DispatchBox( nX1+1,nY1+1,nX2-1,nY2-1,cCF,T. )
ENDIF
RETURN( NIL )
/*-----*/
// Función para desplegar un botón no presionado con nombre.
// nX1,nY1 - Coordenadas de la esquina superior izquierda.
// cCF - Color de fondo del botón.
// cExp1 - Letra resaltada.
// cExp2 - Resto de la opción.
/*-----*/
FUNCTION G_Boton( nX1,nY1,cCF,cExp1,cExp2 )
LOCAL nX2 := nX1 + 2, nY2 := nY1 + Len( cExp1+cExp2 ) +
1
    G_DispatchBox( nX1,nY1,nX2,nY2,cCF,T. )
    @ nX1+1,nY1+1 SAY cExp1 COLOR "W/"+cCF
    @ nX1+1,nY1+1+Len(cExp1) SAY cExp2 COLOR
"+W/"+cCF
RETURN( NIL )
/*-----*/
// Función para desplegar un botón presionado con nombre.
// nX1,nY1 - Coordenadas de la esquina superior izquierda.
// cCF - Color de fondo del botón.
// cExp1 - Letra resaltada.
// cExp2 - Resto de la opción.
/*-----*/
FUNCTION G_PushBoton( nX1,nY1,cCF,cExp1,cExp2 )
LOCAL nX2 := nX1 + 2, nY2 := nY1 + Len( cExp1+cExp2 ) +
1
    G_DispatchBox( nX1,nY1,nX2,nY2,cCF,F. )
    @ nX1+1,nY1+1 SAY cExp1 COLOR "R/"+cCF
    @ nX1+1,nY1+1+Len(cExp1) SAY cExp2 COLOR "W/"+cCF
RETURN( NIL )
/*-----*/
// Función para dar el efecto de presión de un botón.
// nX1,nY1 - Coordenadas de la esquina superior izquierda.
// cCF - Color de fondo del botón.
// cExp1 - Letra resaltada.
// cExp2 - Resto de la opción.
/*-----*/
FUNCTION G_PBoton( nX1,nY1,cCF,cExp1,cExp2 )
LOCAL nX2 := nX1 + 2, nY2 := nY1 + Len( cExp1+cExp2 ) +
1
    G_DispatchBox( nX1,nY1,nX2,nY2,cCF,F. )
    @ nX1+1,nY1+1 SAY cExp1 COLOR "R/"+cCF
    @ nX1+1,nY1+1+Len(cExp1) SAY cExp2 COLOR "W/"+cCF
Inkey( 0.2 )

```

```

    G_DispatchBox( nX1,nY1,nX2,nY2,cCF,T. )
    @ nX1+1,nY1+1 SAY cExp1 COLOR "R/"+cCF
    @ nX1+1,nY1+1+Len(cExp1) SAY cExp2 COLOR "W/"+cCF
RETURN( NIL )
/*-----*/
// Función que muestra la ventana para procesos en ejecución.
// bProc - Proceso a ejecutar.
// cMessage - Mensaje informativo del proceso en
ejecución.
/*-----*/
FUNCTION G_VentProceso( bProc,cMessage )
LOCAL cUltRenglon := SaveScreen( 24,0,24,79 )
A_DispatchBox( 12,12,16,68,"W",F. )
@ 14,15 SAY Space(50) COLOR "N/R"
@ 24, 0 SAY Space( 80 ) COLOR "W+B"
@ 24, 3 SAY cMessage COLOR "W+B"
Eval( bProc )
RestScreen( 24,0,24,79,cUltRenglon )
RETURN( NIL )
/*-----*/
// Función que muestra la barra progresiva para procesos en
ejecución.
// nExp1 - Fracción del proceso.
// nExp2 - Total del proceso.
/*-----*/
FUNCTION G_BarraProceso( nExp1, nExp2 )
LOCAL nL := nExp1/nExp2*50,nP:=nP:=0
nP := nL*2
IF nL>26
    nL:=25
    nP:=26
ENDIF
@ 14,15+nP SAY Space(nL) COLOR "N/*R"
@ 14,38 SAY Str( Int( nP ),3) COLOR IIF(
nP>50,"+W/*R",""+W/R" )
RETURN( NIL )
/*-----*/
// Función que muestra la barra progresiva para la regeneración
de índices.
/*-----*/
FUNCTION G_BarraIndex()
LOCAL nL := Recno()/LastRec()*50,nP:=nP:=0
nP := nL*2
IF nL>26
    nL:=25
    nP:=26
ENDIF
@ 14,15+nP SAY Space(nL) COLOR "N/*R"
@ 14,38 SAY Str( Int( nP ),3) COLOR IIF(
nP>50,"+W/*R",""+W/R" )
RETURN( .T. )
/*-----*/
// Función para la generación de índices.
// acExp1 - Arreglo de archivos a indexar.
// { Nombre del archivo, Descripción del archivo, rea de
trabajo,
// indice de comienzo, indice de termino }
// acExp2 - Arreglo de archivos índice.
// { Nombre del archivo índice, descripción del índice,
IndexKey }
// Retorna .T. si hubo éxito, .F. si no.
/*-----*/
FUNCTION G_IndexA( acExp1,acExp2 )

```

```

LOCAL nE := nT := nPG := 0, nArchivo:=0, nContIndex:=0
LOCAL nTotalArch := Len( acExp1 ), nTotalIndex := Len(
acExp2 )

FOR nArchivo:=1 TO nTotalArch
  SELECT (nArchivo)
  USE ( acExp1[ nArchivo,1 ] )
  IF NetErr()
    G_DspMsg("Error ... Imposible abrir el archivo "+
acExp1[ nArchivo,1 ],"R")
  CLOSE DATABASES
  RETURN( .F. )
ENDIF
acExp1[ nArchivo,3 ] := nArchivo
NEXT

G_DispBox( 5,8,16,71,"W",.F.)
@ 6,15 SAY Str( nTotalArch,3 )+" Archivos" COLOR "B/W"
@ 6,40 SAY Str( nTotalIndex,3 )+" Indices" COLOR "B/W"
G_DispBox( 7,13,9,66,"W",.T.)
@ 8,15 SAY Space(50) COLOR "N/R"

FOR nArchivo:=1 TO nTotalArch
  SELECT ( nArchivo )
  @ 10,10 SAY "Ordenando :." COLOR "+W/W"
  @ 10,23 SAY acExp1[ nArchivo,2 ] COLOR "B/W"
  @ 11,10 SAY " :." COLOR "+W/W"
  @ 11,23 SAY LTrim( Transform( LastRec(), "9,999,999" )
)+" Registros" COLOR "B/W"
  @ 12,10 SAY " Por :." COLOR "+W/W"

  FOR nE:=acExp1[ nArchivo,4 ] TO acExp1[ nArchivo,5 ]
    cIndexKey := acExp2[ nE,3 ]
    @ 12,23 SAY acExp2[ nE,2 ] COLOR "B/W"

    G_DispBox( 13,13,15,67,"W",.T.)
    @ 14,15 SAY Space(50) COLOR "N/R"
    INDEX ON &cIndexKey TO (acExp2[ nE,1 ] EVAL
G_BarraIndex()

    nPG:= ++nContIndex*50/nTotalIndex
    @ 8,15 SAY Space( nPG ) COLOR "N/*R"
    @ 8,38 SAY Str( nPG*2,3 )+"%" COLOR IIF(
nPG>25, "+W/*R", "+W/R" )
    @ 6,40 SAY Str( nTotalIndex-nContIndex,3 )+" Indices"
COLOR "B/W"
  NEXT
  @ 6,15 SAY Str( nTotalArch-nArchivo,3 )+" Archivos"
COLOR "B/W"
  NEXT
RETURN( NIL )

FUNCTION Vent( cExp1,cExp2 )
  @ 10,15 SAY "Ordenando :." COLOR "+W/W"
  @ 10,28 SAY cExp1 COLOR "B/W"
  @ 11,15 SAY " :." COLOR "+W/W"
  @ 11,28 SAY LTrim( Transform( LastRec(), "9,999,999" ) )+"
Registros" COLOR "B/W"
  @ 12,15 SAY " Por :." COLOR "+W/W"

  @ 12,28 SAY cExp2 COLOR "B/W"

  G_DispBox( 13,13,15,66,"W",.T.)
  @ 14,15 SAY Space(50) COLOR "N/R"
RETURN

Function G_Alineacion( cCad,nColumnas,cJust )

```

```

LOCAL nCol:=0, cCadena
cCadena = AllTrim( cCad )
DO CASE
  CASE cJust = 'I'
    nCol := 0
  CASE cJust = 'C'.OR. cJust = NIL
    nCol := Int( ( nColumnas - Len( cCadena ) ) / 2 )
  CASE cJust = 'D'
    nCol := nColumnas - Len( cCadena )
ENDCASE
RETURN( Space( nCol )+cCadena )

FUNCTION G_Impresora( cExp )
  WHILE(.T.)
    IF IsPrinter()
      RETURN( .T. )
    ELSE
      IF !Opcion( "Impresora fuera de linea ... Continuar
impresión", cExp )
        EXIT
      ENDIF
    ENDDO
RETURN(.F.)

FUNCTION G_Printer()
  SET CONSOLE OFF
  SET PRINT ON
  SET DEVICE TO PRINTER
RETURN( .F. )

FUNCTION G_Screen()
  SET PRINT OFF
  SET CONSOLE ON
  SET DEVICE TO SCREEN
RETURN( .F. )

FUNCTION Barra(cIzquierda, cDerecha)
  LOCAL cTex := "", nRow := Row(), nCol := Col()
  If cIzquierda = NIL
    cIzquierda := ""
  EndIf
  cDerecha := AllTrim(PadR(IIF(cDerecha = NIL, "", cDerecha),
50))
  cTex := " " + PadR(cIzquierda, MaxCol() - 3 - Len(cDerecha)) +
" " + cDerecha + " "
  @ MaxRow(), 1 SAY "Clave Presupстал" COLOR "R/W"
  @ MaxRow(), 19 SAY cTex COLOR "N/W"
  SetPos(nRow, nCol)
RETURN .T.

FUNCTION Barra_Fecha(cIzquierda, cDerecha)
  LOCAL cTex := "", nRow := Row(), nCol := Col()
  If cIzquierda = NIL
    cIzquierda := ""
  EndIf
  cDerecha := AllTrim(PadR(IIF(cDerecha = NIL, "", cDerecha),
50))
  cTex := " " + PadR(cIzquierda, MaxCol() - 3 - Len(cDerecha)) +
" " + cDerecha + " "
  @ MaxRow(), 50 SAY "Fecha Movimiento" COLOR "R/W"
  @ MaxRow(), 66 SAY cTex COLOR "N/W"
  SetPos(nRow, nCol)
RETURN .T.

FUNCTION CreaBaseTmp()
Public cFileTmp

```

```

cFileTmp := NombreUnico()
DBCCreate(cFileTmp + ".Dbf", Struct)
Select 3
USE &cFileTmp EXCLUSIVE ALIAS Temp
Select Temp
Return

FUNCTION BorraBaseTmp()
Select Temp
Close Temp
DELETE FILE (cFileTmp+".Dbf")
Return

FUNCTION ErrHand(e)
LOCAL nOpcion := G_Dspmsg(TEXError(e:GenCode),"R")
IFError = Cierto
RETURN Falso

FUNCTION TexError(nError)
DO CASE
CASE nError == 11
RETURN "MEMORIA BAJA"
CASE nError == 12
RETURN "FUNCION NO DEFINIDA"
CASE nError == 13
RETURN "MENSAJE NO DEFINIDO"
CASE nError == 14
RETURN "VARIABLE NO EXISTE"
CASE nError == 15
RETURN "ALIAS NO EXISTE"
CASE nError == 20
RETURN "PROBLEMAS EN LA CONSTRUCCION DEL
ARCHIVO"
CASE nError == 21
RETURN "PROBLEMAS EN LA APERTURA DEL
ARCHIVO"
CASE nError == 22
RETURN "PROBLEMAS EN EL CIERRE DEL ARCHIVO"
CASE nError == 23
RETURN "PROBLEMAS EN LA LECTURA DEL
ARCHIVO"
CASE nError == 24
RETURN "PROBLEMAS EN LA IMPRESORA"
CASE nError == 35
RETURN "AREA DE BASE DE DATOS NO ESTA EN
USO"
CASE nError == 36
RETURN "BASE DE DATOS NO INDEXADA"
ENDCASE
RETURN ""

```

DOCUMENTO DE EJECUCION

```

Local lGuarda
#include "Inkey.ch"
#define Cierto .T.
#define Falso .F.
#define Between(x, y, z) ((x) >= (y) .And. (x) <= (z))

```

```

Private Struct := { {"No_Sec","N",2,0},
{"Clave_Pres","C",18,0},;
{"Beneficiar","C",30,0}, {"Concepto","C",30,0},;
{"Cargo","N",14,2}, {"Deduccion","N",14,2},;
{"Total","N",14,2}, {"Tipo","C",2,0},;
{"Numero","N",7,0}, {"No_Solic","N",5,0},;
{"Dig_Verif","C",1,0} }

```

```

Private nNo_Solic:= nImporte := Final1:=0;;
Total:= Total1:= Total2:= Cta:= 0;;
cTipo_Docto := "02", ;
cUPP := cDepend:= Space(2), ;
cDigito := Space(1), ;
dFecha_Exp := DATE(), ;
Compara:=Space(4)

abGet:= { { |x| IIF(x == NIL, Temp -> No_Sec, Temp -> No_Sec
:= Num_Pon(x) ) },;
{ |x| IIF(x == NIL, Temp -> Clave_Pres, Temp ->
Clave_Pres := Clave(x,Compara) ) },;
{ |x| IIF(x == NIL, Temp -> Dig_Verif,Temp -> Dig_Verif
:= Verifica(x) ) },;
{ |x| Temp -> Beneficiar := Trae(x) },;
{ |x| Temp -> Concepto := Poner(x) },;
{ |x| IIF(x == NIL, Temp -> Cargo, Temp -> Cargo := x) },;
{ |x| IIF(x == NIL, Temp -> Deduccion, Temp ->
Deduccion := x) },;
{ |x| IIF(x == NIL, Temp -> Total:= Cargo - Deduccion,
Total:= Cargo - Deduccion) },;
{ |x| IIF(x == NIL, Temp -> Tipo, Temp -> Tipo :=
Tipo_Just(x) ) },;
{ |x| IIF(x == NIL, Temp -> Numero, Temp -> Numero :=
x) } }

IFError := Falso
bError := ErrorBlock( { |e| ErrHand(e) } )
Select 1
USE EJECUERP SHARED INDEX EJECUERP ALIAS
E_Cuerpo
ErrorBlock( bError )
IF lError
CLOSE DATABASE
Return
ENDIF

IFError := Falso
bError := ErrorBlock( { |e| ErrHand(e) } )
Select 2
USE EJECABEZ SHARED INDEX EJECABEZ ALIAS
E_Cabeza
ErrorBlock( bError )
IF lError
CLOSE DATABASE
Return
ENDIF

IFError := Falso
bError := ErrorBlock( { |e| ErrHand(e) } )
Select 4
USE DISTRIB SHARED INDEX DISTRIB ALIAS Reparte
ErrorBlock( bError )
IF lError
CLOSE DATABASE
Return
ENDIF

IFError := Falso
bError := ErrorBlock( { |e| ErrHand(e) } )
Select 5
USE CAT_GRAL SHARED INDEX CAT_GRAL ALIAS
General
ErrorBlock( bError )
IF lError
CLOSE DATABASE
Return

```

```

ENDIF
IError := Falso
bError := ErrorBlock( { |e| ErrHand(e) } )
Select 6
USE PARTIDAS SHARED INDEX PARTIDAS ALIAS Partidas
ErrorBlock( bError )
IF IError
CLOSE DATABASE
Return
ENDIF

IError := Falso
bError := ErrorBlock( { |e| ErrHand(e) } )
Select 7
USE ASIGNA SHARED INDEX ASIGNA, ASIGNA_3 ALIAS
Dar
ErrorBlock( bError )
IF IError
CLOSE DATABASE
Return
ENDIF

CLS
DO PantallaEJ
*****
PROCEDIMIENTO PRINCIPAL
Do Encabezado
Do While .T.
IGuarda:=SAVESCREEN(0,0,1,79)
Set Cursor Off
Opcion := Inkey(0)
Do Case
Case Opcion = K_ALT_A
@ 0, 2 SAY " Altas " COLOR "W+/B"
@ 0,10 SAY " Bajas " COLOR "N+/W"
@ 0,18 SAY " Consultas " COLOR "N+/W"
Altas(Cierto)
Case Opcion = K_ALT_B
@ 0, 2 SAY " Altas " COLOR "N+/W"
@ 0,10 SAY " Bajas " COLOR "W+/B"
@ 0,18 SAY " Consultas " COLOR "N+/W"
Bajas()
Case Opcion = K_ALT_S
@ 0, 2 SAY " Altas " COLOR "N+/W"
@ 0,10 SAY " Bajas " COLOR "N+/W"
@ 0,18 SAY " Consultas " COLOR "W+/B"
Consultas()
Case Opcion = K_ESC
Exit
EndCase
RESTSCREEN(0,0,1,79,IGuarda)
EndDo
Close All
RUN DEL Z*.DBF
Set Color to "W/N"
Cls
***** ALTAS O MODIFICA
STATIC FUNCTION ALTAS(IExp,IExp1)
Local Pant, nObjeto :=1,nTotal:=0, Pant1, cGuarda:="G"
Pant:=SAVESCREEN(0,0,24,79)
SET CURSOR ON
SETCOLOR("W+/B,W+/B,,")
CREABASETMP()
Select E_Cabeza
nImporte:= 0
IF IExp = Cierto

```

```

@ 5,23 GET nNo_Solic PICTURE "99999" VALID
(!Empty(nNo_Solic))
@ 5,44 SAY cTipo_Docto
@ 6, 9 GET cUPP PICTURE "99" VALID
(Verif(cUPP))
@ 6,25 GET cDepend PICTURE "99" VALID
(Its_Ok(cDepend))
@ 6,36 GET cDigito PICTURE "9" VALID
(Checa(cDigito))
@ 6,45 GET dFecha_Exp VALID
(Bien(dFecha_Exp))
@ 7,11 GET nImporte PICTURE "99,999,999,999.99"
VALID(!Empty(nImporte))
READ
Compara:= cUPP
IF Lastkey() == 27
BORRABASETMP()
RESTSCREEN(0,0,24,79,Pant)
Return
ENDIF
SEEK(nNo_Solic)
IF Found()
G_Dspmsg("El N°mero de Solicitud ya Existe !!!", "N*")
@ 5,23 GET nNo_Solic PICTURE "99999"
VALID(!Empty(nNo_Solic))
ELSE
KEYBOARD Chr( K_INS )
Cuerpo_Ejecucion(Cierto)
IF Lastkey() == K_F10
Grabar(Cierto)
ENDIF
IF Lastkey() == K_ESC
Tone(900,0)
Pant1:=SAVESCREEN(11,27,15,53)
SETCOLOR("W+/RB",,,)
LB_DisBox (12,28,14,52,, "Grabar",,,)
@ 13,29 SAY "[G] Guardar o Cancelar" COLOR "W+/RB"
@ 13,33 SAY "G" COLOR "GR+/RB"
@ 13,43 SAY "C" COLOR "GR+/RB"
@ 13,30 GET cGuarda PICTURE "@K !" COLOR
"GR+/RB"
READ
IF cGuarda == "C" .OR. Lastkey() == 27
RESTSCREEN(11,27,15,53,Pant1)
RESTSCREEN(0,0,24,79,Pant)
BORRABASETMP()
Return
ELSE
Cuerpo_Ejecucion(Cierto,Cierto)
IF Lastkey() == K_ESC
BORRABASETMP()
RESTSCREEN(0,0,24,79,Pant)
RETURN
ELSE
Select Grabar(Cierto)
ENDIF
RESTSCREEN(0,0,24,79,Pant)
Return
ENDIF
ENDIF
ENDIF
RESTSCREEN(0,0,24,79,Pant)
RETURN(Nil)

***** B A J A S
STATIC FUNCTION BAJAS()

```

```

Local Pant
Pant:=SAVESCREEN(0,0,24,79)
SET CURSOR ON
SETCOLOR("W+/B,W+/B,,")
CREABASETMP()
Select E_Cabeza
@ 5,23 GET nNo_Solic PICTURE "99999"
VALID(!Empty(nNo_Solic))
READ
IF Lastkey() == 27
  BORRABASETMP()
  RESTSCREEN(0,0,24,79,Pant)
  Return
ENDIF
Seek(nNo_Solic)
IF BloqueaReg('R')
  IF Found()
    @ 5,44 SAY Tipo_Docto
    @ 6, 9 SAY UPP
    @ 6,25 SAY Depend
    @ 6,36 SAY Digito
    @ 6,45 SAY Fecha_Exp
    @ 7,11 SAY Importe
    DO MUEVE
    Keyboard Chr( 27 )
    Go Top
    Cuerpo_Ejecucion(Falso)
    Inkey(0.3)
  IF Opcion("Cancelar el Nmero de Solicitud?", "W+/R") =
  Cierto
    Select E_Cabeza
    LimpiaReg()
    Select E_Cuerpo
    Seek(nNo_Solic)
    While (No_Solic == nNo_Solic .AND. !Eof())
      LimpiaReg()
      Skip
    EndDo
    Select Dar
    Set Order To 2
    Seek(nNo_Solic)
    While (No_Solic == nNo_Solic .AND. !Eof())
      IF Tipo_Oper == "77"
        BloqueaReg('R')
        Delete
        Unlock
      ENDIF
      Skip
    EndDo
    Set Order To 1
    Saldo(Temp -> Clave_Pres)
  ENDIF
ELSE
  Tone(200)
  G_Dspmsg("El Nmero de Solicitud no Existe", "N*")
ENDIF
ELSE
  Tone(200)
  G_Dspmsg("Archivo Ocupado. Intente de Nuevo", "N*")
ENDIF
BORRABASETMP()
RESTSCREEN(0,0,24,79,Pant)
RETURN
*****
STATIC FUNCTION CONSULTAS()
Local Pant
Pant:=SAVESCREEN(0,0,24,79)

```

```

SET CURSOR ON
SETCOLOR("W+/B,W+/B,,")
CREABASETMP()
Select E_Cabeza
@ 5,23 GET nNo_Solic PICTURE "99999"
VALID(!Empty(nNo_Solic))
READ
IF Lastkey() == 27
  BORRABASETMP()
  RESTSCREEN(0,0,24,79,Pant)
  Return
ENDIF
Seek(nNo_Solic)
IF BloqueaReg('R')
  IF Found()
    @ 5,44 SAY Tipo_Docto
    @ 6, 9 SAY UPP
    @ 6,25 SAY Depend
    @ 6,36 SAY Digito
    @ 6,45 SAY Fecha_Exp
    @ 7,11 SAY Importe
    DO MUEVE
    Go Top
    Cuerpo_Ejecucion(Falso)
  ELSE
    Tone(200)
    G_Dspmsg("El Nmero de Solicitud no Existe", "N*")
  ENDIF
ELSE
  Tone(200)
  G_Dspmsg("Archivo Ocupado. Intente de Nuevo", "N*")
ENDIF
BORRABASETMP()
RESTSCREEN(0,0,24,79,Pant)
RETURN
***** MUESTRA EL TBROWSE DEL CATALOGO
STATIC FUNCTION Cuerpo_Ejecucion(lExpres,lCorrige)
LOCAL Tecla, oCol, cGuarda, Sumatotal:= 0, cExpres,
lVer:=Cierto, Cortec
IF lCorrige == NIL
  lCorrige:= Falso
ELSE
  CorTec:= K_F10
ENDIF
IF lExpres == Cierto
  @ 24, 2 SAY "INS ->" COLOR "R/W"
  @ 24,10 SAY "Agregar" COLOR "N/W"
  @ 24,20 SAY "DEL ->" COLOR "R/W"
  @ 24,28 SAY "Borrar" COLOR "N/W"
  @ 24,51 SAY "ESC ->" COLOR "GR+/RB"
  @ 24,59 SAY "Salir " COLOR "W+/RB"
  @ 24,65 SAY "F10 ->" COLOR "GR+/RB"
  @ 24,73 SAY "Grabar " COLOR "W+/RB"
ENDIF
Select Temp
Go Top
//
IF Lastkey() != 27
  oCuerpo := TBrowseDB( 9,3,19,76 )
  oCuerpo:HeadSep := ""
  oCuerpo:ColSep := " 3 "
  oCuerpo:FootSep := ""
  oCuerpo:ColorSpec:="N/W, W+/B, W+/B"
  oCuerpo:AddColumn(TBColumnNew( "No", { |x|
  Temp -> No_Sec:= Num_Pon(x) })

```

```

oCuerpo:AddColumn(TBColumnNew("Clave Presupuestal", {
|| Transform( Temp -> Clave_Pres,"@R 9999-9999-9999-9999-
99" )}))
oCuerpo:AddColumn(TBColumnNew("D", { ||
Temp -> Dig_Verif }))
oCol:=(TBColumnNew("Beneficiario", { |x| Temp -
> Beneficiar:= Trae(x) })
oCol:Width:=30
oCuerpo:AddColumn(oCol)
oCol:=(TBColumnNew("Concepto", { |x| Temp ->
Concepto:= Poner(x) })
oCol:Width:=30
oCuerpo:AddColumn(oCol)
oCuerpo:AddColumn(TBColumnNew("Cargo", { ||
Transform( Temp -> Cargo, "99,999,999,999.99" )}))
oCuerpo:AddColumn(TBColumnNew("Deducción", { ||
Transform( Temp -> Deduccion, "99,999,999,999.99" )}))
oCuerpo:AddColumn(TBColumnNew("Total", { ||
Transform( Temp -> Total:= Cargo - Deduccion,
"99,999,999,999.99" )}))
oCuerpo:AddColumn(TBColumnNew("Tipo", { || Temp ->
Tipo }))
oCuerpo:AddColumn(TBColumnNew("Nºmero", { ||
Temp -> Numero }))

Do While Cierto
oCuerpo:ForceStable()
IF oCuerpo:Stable
IF oCuerpo:HitTop .OR. oCuerpo:HitBottom
Tone(125)
ENDIF
Tecla := 0
Do While Tecla == 0
oCuerpo:RefreshCurrent()
oCuerpo:ColorRect( {
oCuerpo:RowPos,1,oCuerpo:RowPos,oCuerpo:ColCount },{ 1,2
} )
oCuerpo:Hilite()
Tecla := Inkey(1)
IF ICorrige == Cierto
Tecla := Cortec
ICorrige:=Falso
ENDIF
EndDo
ENDIF
DO CASE
CASE Tecla == K_ESC
Exit
CASE Tecla == K_F10 .AND. IExpres == Cierto
IVer:= Cierto
Sum Temp -> Cargo To Sumatotal
Go Top
While !Eof()
cExpre:= Temp -> Clave_Pres
IF Checa_Fin(Temp -> Clave_Pres) == Cierto .AND.
Verifica_Dig(Temp -> Dig_Verif) == Cierto
Skip
Else
Locate For Clave_Pres == cExpre
G_Dspmsg("Clave Presupuestal "+Transform(
cExpre,"@R 9999-9999-9999-9999-99")+ " Inv lida", "R")
oCuerpo:PanHome()
oCuerpo:Right()
nPos := Recno()
Go nPos
oCuerpo:RowPos:=nPos
oCuerpo:Hilite()

```

```

IVer:=Falso
Exit
ENDIF
EndDo
nPos:=0
Go Top
While !Eof() .AND. IVer == Cierto
IF Temp -> Tipo # " "
Skip
ELSE
G_Dspmsg("El Tipo de Justificación no puede ir
vacío", "R")
oCuerpo:PanEnd()
oCuerpo:Left()
nPos:=Recno()
Go nPos
oCuerpo:RowPos:=nPos
oCuerpo:Hilite()
IVer:=Falso
Exit
ENDIF
EndDo
IF IVer == Cierto
Select Temp
Go Top
IF Busca(Temp -> Clave_Pres) == Cierto
IF Sumatotal # nImporte
Tone(200)
Go Top
G_Dspmsg("La Suma del Total no coincide con el
Importe", "R")
ELSE
EXIT
ENDIF
ENDIF
oCuerpo:RefreshAll()
CASE Tecla==K_DEL .AND. IExpres == Cierto
DELETE
PACK
oCuerpo:RefreshCurrent()
oCuerpo:RefreshAll()
CASE Tecla==K_INS .AND. IExpres == Cierto
APPEND BLANK
oCuerpo:PanHome()
@ 20,4 SAY REPLICATE (" ",50) COLOR "N+/W"
Go Top
oCuerpo:Down()
oCuerpo:RefreshAll()
CASE Between(Tecla, 32, 255) .OR. Tecla == K_ENTER
IF IExpres = Cierto
KEYBOARD Chr(Tecla)
Edicion( abGet[ oCuerpo:ColPos ],oCuerpo)
ENDIF
oCuerpo:RefreshAll()
OTHERWISE
TECLAS33(oCuerpo,Tecla)
ENDCASE
EndDo
ENDIF
Return
*****
Static FUNCTION Edicion(bBloque,oBrowse)
// LOCAL b := oBrowse:GetColumn(oBrowse:ColPos):Block,
xVar, oGet
LOCAL xVar, oGet
oBrowse:Stabilize()

```

```

xVar := Eval(bBloque)
oGet := GetNew(Row(), Col(), { |x| IIF(x == NIL, xVar, xVar
:= x) }, "xVar")
DO CASE
CASE oBrowse:ColPos = 1
oGet:Picture:="@ 99"
CASE oBrowse:ColPos = 2
oGet:Picture:="@R 9999-9999-9999-9999-99"
CASE oBrowse:ColPos = 4 .OR. oBrowse:ColPos = 5
oGet:Picture:="@S" +
Ltrim(Str(oBrowse:GetColumn(oBrowse:ColPos):Width))
CASE oBrowse:ColPos > 5 .AND. oBrowse:ColPos < 9
oGet:Picture:="99,999,999,999.99"
CASE oBrowse:ColPos = 9
oGet:Picture:="@!!"
ENDCASE
ReadModal( { oGet } )
oBrowse:Right()
If Updated()
Eval(bBloque, xVar)
EndIf
oBrowse:RefreshCurrent()
RETURN
*****
Static Function Grabar(IVar)
Local Pant, cGuarda:="G", nCarga:=0, nTotal:=0, nReg:=0,
cAux, nSaldo:=0,;
nAbono:=0, IBand:=.T., Auxfec
Select Temp
Auxfec:=CtoD("01"+Str(Month(dFecha_Exp))+Str(Year(dFecha_
Exp)))
Go Top
While !Eof()
Select Dar
Seek(Temp -> Clave_Pres+Dtos(Auxfec))
While (Temp_cv == Temp -> Clave_Pres .AND.
Month(Fecha_Mov) == Month(dFecha_Exp) .AND. !Eof())
Skip
Enddo
Skip -1
IF ( Dar -> Temp_Cv == Temp -> Clave_Pres .AND. Dar ->
Total > 0)
IF Temp -> Cargo <= Dar -> Total
nTotal:= Dar -> Total - Temp -> Total
IF AgregaReg('2')
REPLACE Total WITH nTotal, Tipo_Oper WITH "77",
No_Solic WITH nNo_Solic,;
Cargo WITH Temp -> Cargo, Fecha_Mov WITH
dFecha_Exp,;
Temp_cv WITH Temp -> Clave_Pres, Abono
WITH Temp -> Deduccion
COMMIT
UNLOCK
ELSE
LimpiaReg()
G_Dspmsg("Error al Dar de Alta el Registro !!", "R")
ENDIF
ELSE
RESTSCREEN(11,27,15,53,Pant)
G_Dspmsg("El Cargo es Mayor que el Fondo de la
Partida", "R")
@ 7,11 GET nImporte PICTURE "99,999,999,999.99"
VALID(!Empty(nImporte))
READ
IF Lastkey() == K_ESC
IBand:=Falso

```

```

Exit
Endif
Cuerpo_Ejecucion(Cierto)
Grabar(Cierto)
IBand:=Falso
Exit
ENDIF
ELSE
RESTSCREEN(11,27,15,53,Pant)
G_Dspmsg("La Partida no tiene suficientes fondos para
cubrir el gasto", "R")
@ 7,11 GET nImporte PICTURE "99,999,999,999.99"
VALID(!Empty(nImporte))
READ
IF Lastkey() == K_ESC
IBand:=Falso
Exit
Endif
Cuerpo_Ejecucion(Cierto)
Grabar(Cierto)
IBand:=Falso
Exit
ENDIF
Select Temp
Skip
Enddo
IF IBand == Cierto
Select E_Cabeza
Seek(nNo_Solic)
IF Found()
G_Dspmsg("El N£mero de Solicitud ya Existe !!!", "R")
Return
ENDIF
IF IBand = Cierto
IF AgregaReg('1')
REPLACE No_Solic WITH nNo_Solic, Tipo_Docto
WITH cTipo_Docto,;
UPP WITH cUPP, Depend WITH cDepend, Digito
WITH cDigito,;
Fecha_Exp WITH dFecha_Exp, Importe WITH
nImporte
COMMIT
UNLOCK
ELSE
LimpiaReg()
G_Dspmsg("Error al Dar de Alta el Registro !!", "R")
ENDIF
ENDIF
Select Temp
Go Top
While !Eof() .AND. IBand = Cierto
Select E_Cuerpo
IF AgregaReg('1')
REPLACE No_Solic WITH nNo_Solic, No_Sec WITH
Str(TEMP -> No_Sec,2),;
Clave_Pres WITH TEMP -> Clave_Pres,;
Beneficiario WITH TEMP -> Beneficiario, Concepto
WITH TEMP -> Concepto,;
Cargo WITH TEMP -> Cargo, Deduccion WITH
TEMP -> Deduccion,;
Total WITH TEMP -> Total, Tipo WITH TEMP ->
Tipo,;
Numero WITH TEMP -> Numero, Dig_Verif WITH
TEMP -> Dig_Verif
COMMIT
UNLOCK
ELSE

```

```

LimpiaReg()
G_Dspmsg("Error al Dar de Alta el Registro !!","R")
ENDIF
Saldo(Temp -> Clave_Pres)
Select Temp
Skip
EndDo
BORRABASETMP()
ENDIF
RESTSCREEN(11,27,15,53,Pant)
Return(Nil)
***** ADICIONALES
PROCEDURE PANTALLAEJ
SETCOLOR("N/W,,")
disbox(0,0,24,79," ° °°")//acColor[11]
@ 0, 0 SAY Space(80)//COLOR acColor[10]
@ 2,20 SAY "DOCUMENTO DE EJECUCION
PRESUPUESTARIA " COLOR "B/W"
A_DisBox( 3,1,22,78,"W",.T. )
@ 0, 2 SAY " Itas ajas Con ultas " COLOR "N/W"
@ 0, 3 SAY "A" COLOR "R/W"
@ 0,11 SAY "B" COLOR "R/W"
@ 0,22 SAY "s" COLOR "R/W"
RETURN

PROCEDURE ENCABEZADO
@ 5, 3 SAY "NEmero de Solicitud"
@ 5,30 SAY "Tipo de Docto"
@ 6, 3 SAY "U.P.P."
@ 6,13 SAY "Dependencia"
@ 6,29 SAY "Dígito"
@ 6,39 SAY "Fecha"
@ 7, 3 SAY "Importe"
@ 8, 2 SAY "Á" COLOR "N/W"
@ 8, 3 SAY REPLICATE("Á",74)
@ 8,77 SAY "" COLOR "W+/W"
RETURN

PROCEDURE MUEVE
Select E_Cuerpo
Seek(nNo_Solic)
While(No_Solic == nNo_Solic .AND. !Eof())
IF BloqueaReg("R")
Select Temp
IF AgregaReg()
REPLACE No_Solic WITH E_Cuerpo -> No_Solic, ;
No_Sec WITH Val(E_Cuerpo -> No_Sec);
Clave_Pres WITH E_Cuerpo -> Clave_Pres, ;
Beneficiario WITH E_Cuerpo -> Beneficiario, ;
Concepto WITH E_Cuerpo -> Concepto, ;
Cargo WITH E_Cuerpo -> Cargo, ;
Deduccion WITH E_Cuerpo -> Deduccion, ;
Total WITH E_Cuerpo -> Total, ;
Tipo WITH E_Cuerpo -> Tipo, ;
Numero WITH E_Cuerpo -> Numero, ;
Dig_Verif WITH E_Cuerpo -> Dig_Verif
COMMIT
UNLOCK
ELSE
LimpiaReg()
G_Dspmsg("Error al Dar de Alta el Registro","R")
ENDIF
ELSE
Tone(200)
G_Dspmsg("Registro Ocupado. Intente de Nuevo","N*")
ENDIF
Select E_Cuerpo

```

```

Skip
EndDo
RETURN

STATIC FUNCTION TECLAS33(O,T)
//
Do Case
Case T == K_DOWN
O:Down()
Case T == K_UP
O:Up()
Case T == K_PGDN
O:PageDown()
Case T == K_PGUP
O:PageUp()
Case T == K_CTRL_HOME
O:GoTop()
Case T == K_CTRL_END
O:GoBottom()
Case T == K_RIGHT .OR. T == K_TAB
O:Right()
Case T == K_LEFT .OR. T == K_SH_TAB
O:Left()
Case T == K_HOME
O:Home()
Case T == K_END
O:End()
EndCase
O:RefreshCurrent()
Return

```

CREACION DE CLAVES PRESUPUETALES PARA DISTRIBUCION

```

Local lGuarda, Opcion
#include "Inkey.ch"
#Define Cierto .T.
#Define Falso .F.

DECLARE anPORCENTAJE [12]
DECLARE anTotal[0]
DECLARE anFinal[12]
AFILL(anPORCENTAJE,0)
AFILL(anTotal,0)
AFILL(anFinal,0)

Private Struct := { {"Tipo","C",1,0}, {"Clave","C",2,0}, ;
{"Subclave","C",2,0}, {"Descrip","C",50,0} }

cFileTmp := NombreUnico()
DBCreate(cFileTmp + ".Dbf", Struct)

Private dFecha := DATE(), ;
cUpp := cDepend := AuxCve:= Space(2), ;
cProg := cSubprog := Space(2), ;
cProy := cObra := Space(2), ;
nAsignacion := nMes := sum := Sumar := 0, ;
cTipo := cSub_Reg := cTipo_Cal := Space(1), ;
cTemp_cv := Space(18), cGuarda := "G", cDig_Iden :=
Space(1), ;
lBand := .T., UPP := Space(2), cPartida:=Space(4)

lError := Falso
bError := ErrorBlock( { |e| ErrHand(e) } )
Select 1
USE DISTRIB EXCLUSIVE INDEX DISTRIB Alias Reparte
ErrorBlock( bError )

```

```

IF IError
  CLOSE DATABASES
  Return
ENDIF

IError := Falso
bError := ErrorBlock( {e| ErrHand(e) } )
Select 2
USE CAT_GRAL EXCLUSIVE INDEX CAT_GRAL Alias
General
IF IError
  CLOSE DATABASE
  Return
ENDIF

IError := Falso
bError := ErrorBlock( {e| ErrHand(e) } )
Select 3
USE PARTIDAS EXCLUSIVE INDEX PARTIDAS Alias
Partidas
ErrorBlock( bError )
IF IError
  CLOSE DATABASE
  Return
ENDIF

Select 4
USE ASIGNA SHARED INDEX ASIGNA Alias Dar

Select 5
USE &cFileTmp EXCLUSIVE ALIAS Ayuda

CLS
*****
PROCEDIMIENTO PRINCIPAL
DO Pantallaa
IF Month(Date()) = 7
  Despliega_Distrib()
ELSE
  G_Dspmsg("Expiro el Período de Altas y/o Bajas. No se
  pueden hacer Modificaciones","R")
ENDIF
CLOSE ALL
Set Cursor Off
DELETE FILE (cFileTmp + ".Dbf")
RELEASE ALL
Set Color to "W/N"
CLS
RETURN

***** MUESTRA EL TBROWSE DE LA PARTIDA
FUNCTION Despliega_Distrib()
  LOCAL Tecla, oColumna, oColumna1, oColumna2,
  oColumna3, lExp, reng:=6
  //
  Select Reparte
  oDistrib:=TBrowseDB( 5,3,21,76 )
  oDistrib:HeadSep := "ÄÄÄ"
  oDistrib:ColSep := " "
  oDistrib:FootSep := "ÄÄÄ"
  oDistrib:ColorSpec:="N/W, W+/GB, W+/B, GR/W, B/W"
  //
  oColumna := TBColumnNew( "Clave Presupuestal", { || IIF
  (Tipo_Oper = "01" .OR. Tipo_Oper = NIL,Transform(Temp_CV,
  "@R 9999-9999-9999-9999-99"),Nil)} )
  oColumna:Width := 22
  oDistrib:AddColumn( oColumna )

```

```

  oColumna2:= TBColumnNew( "Asignacion", { || IIF
  (Tipo_Oper = "01",Transform(Asignacion,
  "99,999,999,999.99"),Nil)} )
  oColumna2:ColorBlock := { || IIF (Tipo_Oper =
  "01",{4,2},{1,2}})
  oDistrib:AddColumn( oColumna2 )
  oColumna3:= TBColumnNew( "Parcial", { ||
  Transform(Total, "99,999,999,999.99") } )
  oColumna3:ColorBlock := { || IIF (Tipo_Oper = "01".OR.
  Tipo_Oper = "NA",{4,2},{1,2}})
  oDistrib:AddColumn( oColumna3 )
  oColumna := TBColumnNew( "Fecha", { || Arregla(Day(Fecha))
  + "/" + Arregla(Month(Fecha)) + "/" + Arregla(Year(Fecha) %
  100) } )
  oColumna:Width := 8
  oDistrib:AddColumn(oColumna)
  @ 6, 2 SAY "Ä" COLOR "N/W"
  @ 6,77 SAY "" COLOR "W+/W"
  //
  WHILE .T.
    oDistrib:ForceStable()
    IF oDistrib:Stable
      IF oDistrib:HitTop .OR. oDistrib:HitBottom()
        Tone(800,0)
      ENDIF
      Tecla := 0
      Barra(Transform(Temp_CV, "@R 9999-9999-9999-9999-
      99"), "")
      Do While Tecla == 0
        oDistrib:RefreshCurrent()
        oDistrib:ColorRect( {
        oDistrib:RowPos,1,oDistrib:RowPos,oDistrib:ColCount }, { 3,2
        } )
        oDistrib:Hilite()
        Tecla := Inkey(1)
      EndDo
    ENDIF
    lExp := SAVESCREEN(0,0,1,79)
    DO CASE
      CASE Tecla==K_ESC
        SETCOLOR("W/N,,")
        EXIT
      CASE Tecla==K_ALT_A
        WHILE LastKey()# K_ESC
          @ 0, 2 SAY " Altas " COLOR "W+/B"
          @ 0, 9 SAY " Bajas " COLOR "N+/W"
        ALTAS()
        ENDDO
        oDistrib:RefreshAll()
        CASE Tecla==K_ALT_B // Da de Baja Distribs
          @ 0,2 SAY " Altas " COLOR "N+/W"
          @ 0,9 SAY " Bajas " COLOR "W+/B"
          IF Opcion("Desea dar de Baja esta Clave
          Presupuestal?", "W+/R") = Cierto
            cTemp_cv:=Temp_cv
            Seek(cTemp_cv)
            While(Temp_cv = cTemp_cv .AND. !EOF())
              IF BloqueaReg('A')
                LimpiaReg()
              ELSE
                Tone(800)
                G_Dspmsg("Registro Ocupado. Intente de
                Nuevo","R")
              ENDIF
            Select Dar
            IF BloqueaReg('A')
              LimpiaReg()
            ENDIF

```

```

Skip
ELSE
  Tone(800)
  G_Dspmsg("Registro Ocupado. Intente de
Nuevo", "R")
ENDIF
  Select Reparte
  Skip
  Enddo
ENDIF
  oDistrib:RefreshAll()
OTHERWISE
  TECLAS33(oDistrib, Tecla)
ENDCASE
  RESTSCREEN(0,0,1,79, lExp)
ENDDO
RETURN ( NIL )

```

***** ALTAS O MODIFICA

```

FUNCTION ALTAS()
Local Pant, nCant_mod:=0, lGuarda
SET CURSOR ON
Select General
lGuarda:=SAVESCREEN(6,16,22,69)
G_DispBox( 7,15,18,68,"B",.T. )
Part_Ele()
Clave_Pres()
RETURN (Nil)

```

```

FUNCTION CLAVE_PRES()
Local cTemp_cv:=Space(17), nObjeto:=1, lGuarda, suma:=0, i:=0
Select General
@ 8,30 GET cUpp PICTURE "99" VALID
Valida_C("U",cUpp,"00")
@ 8,53 GET cDepend PICTURE "99" VALID
Valida_C("U",cUpp, cDepend)
@ 9,30 GET cProg PICTURE "99" VALID
Valida_C("P",cProg,"00")
@ 9,53 GET cSubprog PICTURE "99" VALID
Valida_C("P",cProg, cSubprog)
@ 10,30 GET cProy PICTURE "99" VALID
Valida_C("Y",cProy,"00")
@ 10,53 GET cObra PICTURE "99" VALID
Valida_C("Y",cProy, cObra)
@ 11,30 GET cPartida PICTURE "9999" VALID
Check_Part(cPartida)
@ 11,53 GET cSub_Reg PICTURE "9"
VALID(cSub_Reg$"0123456789")
@ 12,30 GET cDig_Iden PICTURE "9"
VALID(VerIden(cDig_Iden))
@ 12,53 GET cTipo_Cal PICTURE "9"
VALID(cTipo_Cal$"12")
@ 13,30 GET nAsignacion PICTURE "99,999,999,999.99"
VALID(!Empty(nAsignacion))
READ
cTemp_cv := cUpp + cDepend + cProg + cSubprog + cProy +
cObra + cPartida;
+ cSub_Reg + cDig_Iden
IF Lastkey() == K_ESC
  Select Reparte
  Go Top
  RETURN
ELSE
  Select Reparte
  SEEK(cTemp_cv)
  IF Found()

```

```

G_Dspmsg("La Clave Presupuestal ya existe !!!", "R")
Return
ELSE
  IF cTipo_Cal = "1"
    AFILL(anPorcentaje,8.33)
    anPorcentaje[12]:= 8.37
    DO PORCENT
    CLEAR GETS
    INKEY(0.5)
  ENDIF
  IF cTipo_Cal = "2"
    AFILL(anPorcentaje,0)
    DO PORCENT
    READ
    i:=1
    While i < 13
      Suma+=anPorcentaje[i]
      i++
    EndDo
    IF Suma < 100 .OR. Suma > 101
      G_Dspmsg("La Distribución no es del 100%, Vuelva a
Capturar", "R")
      Return
    ENDIF
  ENDIF
  IF Lastkey() = 27 .OR. lBand == Falso
    Return
  ELSE
    Guardar(cTemp_cv,Cierto)
  ENDIF
ENDIF
RETURN ( Nil )

```

***** ADICIONALES

```

PROCEDURE PANTALLAA
SETCOLOR("N/W,,")
dispbox(0,0,24,79," ° °°",)/acColor[11])
@ 0, 0 SAY Space(80) //COLOR acColor[10]
@ 2,24 SAY " DISTRIBUCION PRESUPUESTARIA "
COLOR "B/W"
A_DispBox( 3,1,22,78,"W",.T. )
@ 0, 2 SAY " Itas ajas " COLOR "N/W"
@ 0, 3 SAY "A" COLOR "R/W"
@ 0,10 SAY "B" COLOR "R/W"
RETURN

```

```

STATIC FUNCTION VALIDA_C(cTipo, cExp, cExp1)
Select General
Seek(cTipo+cExp+cExp1)
IF Found()
  Return (.T.)
ELSE
  lGuarda:= SaveScreen(5,34,10,76)
  Go Top
  Seek(cTipo)
  While !Eof() .AND. Tipo == cTipo
    Select Ayuda
    APPEND BLANK
    REPLACE Clave WITH General -> Clave, Subclave WITH
General -> Subclave,;
    Descrip WITH General -> Descrip
  COMMIT
  UNLOCK
  IF cExp1 == "00"
    Select Ayuda

```

```

DELETE For Subclave <> "00"
ELSE
  Select Ayuda
  DELETE For Subclave == "00"
  DO CASE
    CASE cTipo = "U"
      DELETE For Clave <> cUpp
    CASE cTipo = "P"
      DELETE For Clave <> cProg
    CASE cTipo = "Y"
      DELETE For Clave <> cProy
  ENDCASE
ENDIF
Select General
Skip
EndDo
Obj:=GetActive()
Obj:Varput(Visual(cExp1))
ENDIF
Select General
RETURN(.T.)

*****
STATIC FUNCTION Check_Part(cExp)
Select Partidas
Seek(cExp)
IF Found()
  Return (.T.)
ELSE
  lGuarda:= SaveScreen(5,34,10,76)
  Obj:=GetActive()
  Obj:Varput(Visual("PA"))
ENDIF
Select Partidas
RETURN(.T.)

***** MUESTRA EL TBROWSE DE AYUDA
FUNCTION Visual(cExp1)
  LOCAL Tecla, oColumna, lExp, reng:=6
  //
  Select Ayuda
  Set Delete On
  Go Top
  @ 5,34,10,76 BOX "ÉÍ»º¼½°" COLOR "W+/GB"
  oVisual:=TBrowseDB( 5,35,10,75 )
  oVisual:HeadSep := "ÁÁÁ"
  oVisual:ColSep := " 3 "
  oVisual:FootSep := "ÁÁÁ"
  oVisual:ColorSpec:="W+/GB, W+/B, W+/B, N/GB, B/W"
  //
  IF cExp1 == "PA"
    Select Partidas
    Go Top
    oVisual:AddColumn(TbColumnNew("", { || Partida } ))
    oColumna:=(TbColumnNew("", { || Desc_Corta } ))
    oColumna:Width:=30
    oVisual:AddColumn( oColumna )
  ELSEIF cExp1 == "00"
    oVisual:AddColumn(TbColumnNew( "", { || Clave } ))
    oColumna:=(TbColumnNew( "", { || Descrip } ))
    oColumna:Width:=30
    oVisual:AddColumn( oColumna )
  ELSE
    oVisual:AddColumn(TbColumnNew( "", { || Subclave } ))
    oColumna:=(TbColumnNew( "", { || Descrip } ))
    oColumna:Width:=30
    oVisual:AddColumn( oColumna )
  ENDIF

```

```

ENDIF
@ 6, 2 SAY "Á" COLOR "N/W"
@ 6,77 SAY "" COLOR "W+/W"
//
WHILE .T.
  oVisual:ForceStable()
  IF oVisual:Stable
    IF oVisual:HitTop .OR. oVisual:HitBottom()
      Tone(800,0)
    ENDIF
    oVisual:ColorRect( {
oVisual:RowPos,1,oVisual:RowPos,oVisual:ColCount }, { 3,2 } )
    oVisual:Hilite()
    Tecla:=Inkey(0)
  ENDIF
  DO CASE
    CASE Tecla == K_ESC .OR. Tecla == K_ENTER
      IF cExp1 == "PA"
        Auxcve:= Partida
      ELSEIF cExp1 == "00"
        Auxcve := Clave
      ELSE
        Auxcve := Subclave
      ENDIF
      RESTSCREEN(5,34,10,76,lGuarda)
      Exit
      oVisual:RefreshAll()
    OTHERWISE
      TECLAS33(oVisual,Tecla)
    ENDCASE
  ENDDO
  Select Ayuda
  Delete All
  RETURN ( Auxcve )

STATIC FUNCTION PART_ELE(cQue)
SETCOLOR("W/B","N/W")
@ 8,17 SAY "U.P.P. "
@ 8,40 SAY "Dependencia "
@ 9,17 SAY "Programa "
@ 9,40 SAY "Subprograma "
@ 10,17 SAY "Proyecto "
@ 10,40 SAY "Obra "
@ 11,17 SAY "Partida "
@ 11,40 SAY "Subregión "
@ 12,17 SAY "Dígito Ident"
@ 12,40 SAY "Tipo Calend"
@ 13,17 SAY "Asignación "
@ 18,54 SAY "ESC -> Salir " Color "+W/R"
Return( Nil )

*****
FUNCTION VerIden(cExp)
IF cExp == "1" .OR. cExp == "2" .AND. UPP == "12"
  Return (.T.)
ELSEIF cExp == "1" .AND. UPP != "12"
  Return (.T.)
ELSE
  G_Dspmsg("El Dígito Identificador No corresponde a la UPP","R")
ENDIF
Return (.F.)

*****
FUNCTION Guardar(cTemp_cv,lVar)
Local lKeep, Suma:=0, Result:=0, Total1:=0, nAsigna:=0, m:=0

```

```

IKeep:= SAVESCREEN(12,19,14,52)
Tone(750)
SETCOLOR("W+/RB",,,)
LB_DispBox( 12,28,14,52,,"Grabar",,)
@ 13,29 SAY "[G] Guardar o Cancelar" COLOR "W+/RB"
@ 13,33 SAY "G" COLOR "GR+/RB"
@ 13,43 SAY "C" COLOR "GR+/RB"
@ 13,30 GET cGuarda PICTURE "@K !" COLOR "GR+/RB"
READ
IF cGuarda <> "G" .OR. LastKey() = 27
  RESTSCREEN(12,28,14,52,IKeep)
  Return
ELSE
  i:=1
  j:=1
  nAsigna:=nAsignacion
  WHILE (i <= LEN(anPorcentaje) .AND. j <= LEN(anFinal))
    sum+=anPorcentaje[i]
    IF Empty(Porcentaje) .AND. anPorcentaje[i] > 0
      DO CONV_FECHA
      IF cTipo_Cal = "1"
        Result = nAsigna / 12
        Total1 = Round(Result, 2)
        nAsigna = Round((nAsigna - Total1),2)
        AADD(anTotal,Total1)
      //      anTotal[j] := nAsignacion / 12
      ENDIF
      IF cTipo_Cal = "2" .AND. anPorcentaje[i] > 0
      //      Total1 := nAsignacion * (anPorcentaje[i]/100)
      //      AADD(anTotal,Total1)
      anFinal[j] := nAsignacion * (anPorcentaje[i]/100)
      ENDIF
      IF AgregaReg()
        REPLACE Asignacion WITH nAsignacion, Tipo_Oper
        WITH "01",;
        Temp_cv WITH cTemp_cv, Porcentaje WITH
        anPorcentaje[i],;
        Mes WITH i, Fecha WITH dFecha,Tipo_Cal WITH
        cTipo_Cal
      IF cTipo_Cal = "1"
        REPLACE Total WITH anTotal[j]
      ELSE
        REPLACE Total WITH anFinal[j]
      ENDIF
      COMMIT
      UNLOCK
      ELSE
        LimpiaReg()
        G_Dspmsg("Error al Dar de Alta el Registro !!","R")
        Return
      ENDIF
      Select Dar
      IF AgregaReg()
        REPLACE Asignacion WITH nAsignacion, Tipo_Oper
        WITH "01",;
        Temp_cv WITH cTemp_cv, Fecha_Mov WITH
        dFecha
      IF cTipo_Cal = "1"
        REPLACE Total WITH anTotal[j], Asig_Parci WITH
        anTotal[j],;
        Asig_Net WITH anTotal[j]
      ELSE
        REPLACE Total WITH anFinal[j], Asig_Parci WITH
        anFinal[j],;
        Asig_Net WITH anFinal[j]
      ENDIF
      ELSE
        LimpiaReg()
        G_Dspmsg("Error al Dar de Alta el Registro !!","R")
      ENDIF
      Select Reparte
      ENDIF
      i++
      j++
      ENDDO
      ENDIF
      ASIZE(anTotal,0)

```

```

UNLOCK
ELSE
  LimpiaReg()
  G_Dspmsg("Error al Dar de Alta el Registro !!","R")
  Return
ENDIF
Select Reparte
ELSEIF anPorcentaje[i] > 0
  DO CONV_FECHA
  IF cTipo_Cal = "1"
    m:= 13 - i
    For h = m to 1 Step - 1
      Result = nAsigna / h
      Total1 = Round(Result, 2)
      nAsigna = Round((nAsigna - Total1),2)
      AADD(anTotal,Total1)
    Next
    anTotal[j] := nAsignacion / 12
  ENDIF
  IF cTipo_Cal = "2" .AND. anPorcentaje[i] > 0
  //      Total1 := nAsignacion * (anPorcentaje[i]/100)
  //      AADD(anTotal,Total1)
  anFinal[j] := nAsignacion * (anPorcentaje[i]/100)
  ENDIF
  IF AgregaReg()
    REPLACE Asignacion WITH nAsignacion, Tipo_Oper
    WITH "76",;
    Fecha WITH dFecha, Temp_cv WITH cTemp_cv,;
    Porcentaje WITH anPorcentaje[i], Mes WITH i,;
    Tipo_Cal WITH cTipo_Cal
  IF cTipo_Cal = "1"
    REPLACE Total WITH anTotal[j]
  ELSE
    REPLACE Total WITH anFinal[j]
  ENDIF
  COMMIT
  UNLOCK
  ELSE
    LimpiaReg()
    G_Dspmsg("Error al Dar de Alta el Registro !!","R")
  ENDIF
  Select Dar
  IF AgregaReg()
    REPLACE Asignacion WITH nAsignacion, Tipo_Oper
    WITH "76",;
    Temp_cv WITH cTemp_cv, Fecha_Mov WITH
    dFecha
  IF cTipo_Cal = "1"
    REPLACE Total WITH anTotal[j], Asig_Parci WITH
    anTotal[j],;
    Asig_Net WITH anTotal[j]
  ELSE
    REPLACE Total WITH anFinal[j], Asig_Parci WITH
    anFinal[j],;
    Asig_Net WITH anFinal[j]
  ENDIF
  ELSE
    LimpiaReg()
    G_Dspmsg("Error al Dar de Alta el Registro !!","R")
  ENDIF
  Select Reparte
  ENDIF
  i++
  j++
  ENDDO
  ENDIF
  ASIZE(anTotal,0)

```

```
RESTSCREEN(12,19,14,52,IKeep)
RETURN( cGuarda )
```

```
*****
```

```
PROCEDURE PORCENT
```

```
@ 15,17 SAY "Ene" GET anPorcentaje[1] PICTURE "999.99"
@ 15,29 SAY "Feb" GET anPorcentaje[2] PICTURE "999.99"
@ 15,40 SAY "Mar" GET anPorcentaje[3] PICTURE "999.99"
@ 15,51 SAY "Abr" GET anPorcentaje[4] PICTURE "999.99"
@ 16,17 SAY "May" GET anPorcentaje[5] PICTURE "999.99"
@ 16,29 SAY "Jun" GET anPorcentaje[6] PICTURE "999.99"
@ 16,40 SAY "Jul" GET anPorcentaje[7] PICTURE "999.99"
@ 16,51 SAY "Ago" GET anPorcentaje[8] PICTURE "999.99"
@ 17,17 SAY "Sep" GET anPorcentaje[9] PICTURE "999.99"
@ 17,29 SAY "Oct" GET anPorcentaje[10] PICTURE "999.99"
@ 17,40 SAY "Nov" GET anPorcentaje[11] PICTURE "999.99"
@ 17,51 SAY "Dic" GET anPorcentaje[12] PICTURE "999.99"
RETURN
```

```
PROCEDURE CONV_FECHA
```

```
nMes := i
IF nMes <= 9
  nMess:=PADL(nMes,1,0)
  dTemp := "01"+Str(nMes)+Str(Year(DATE()))
ELSE
  dTemp := "01"+Str(nMes)+Str(Year(DATE()))
ENDIF
dFecha := CTOD(dTemp)
RETURN
```

```
FUNCTION Arregla(nVal)
```

```
RETURN IIF( nVal < 10, "0" + Str(nVal, 1), Str(nVal, 2) )
```

```
FUNCTION Sale()
```

```
IF Lastkey() = 27
```

```
  Select Reparte
```

```
  Go Top
```

```
  Return(Nil)
```

```
ENDIF
```

```
STATIC FUNCTION TECLAS33(O,T)
```

```
//
```

```
Do Case
```

```
  Case T == K_DOWN
```

```
    O:Down()
```

```
  Case T == K_UP
```

```
    O:Up()
```

```
  Case T == K_PGDN
```

```
    O:PageDown()
```

```
  Case T == K_PGUP
```

```
    O:PageUp()
```

```
  Case T == K_CTRL_HOME
```

```
    O:GoTop()
```

```
  Case T == K_CTRL_END
```

```
    O:GoBottom()
```

```
  Case T == K_RIGHT .OR. T == K_TAB
```

```
    O:Right()
```

```
  Case T == K_LEFT .OR. T == K_SH_TAB
```

```
    O:Left()
```

```
  Case T == K_HOME
```

```
    O:Home()
```

```
  Case T == K_END
```

```
    O:End()
```

```
EndCase
```

```
O:RefreshCurrent()
```

REQUISICION DE BIENES

```
Local lGuarda
```

```
#Include "Inkey.ch"
```

```
#Define Cierto .T.
```

```
#Define Falso .F.
```

```
#Define Between(x, y, z) ((x) >= (y) .And. (x) <= (z))
```

```
Private Struct := { {"No","N",2,0}, {"Cantidad","N",3,0},;
  {"Clave","C",7,0}, {"Descrip","C",40,0},;
  {"Costo","N",7,2}, {"Importe","N",9,2},;
  {"No_Solic","N",5,0}, {"Tipo","C",5,0},;
  {"Req_Tipo","C",1,0} }
```

```
Private Ayuda := { {"Material","C",7,0}, {"Descrip","C",40,0} }
```

```
cFileTmp := NombreUnico()
```

```
DBCcreate(cFileTmp + ".Dbf", Ayuda)
```

```
Private      nNo_Solic:= nTotal := 0,      ;
             cUPP := cDepend:= Space(2),    ;
             cDigito := cReq_Tipo:= Space(1), ;
             cClave_Pres := Space(18),      ;
             dFecha_Exp := DATE(),          ;
             Compara:=Space(2), Auxcve:=Space(7)
```

```
abGet:= { { |x| IIF(x == NIL, Temp -> No, Temp -> No :=
  Req_Sec(x)) },;
  { |x| IIF(x == NIL, Temp -> Cantidad, Temp -> Cantidad :=
  x) },;
  { |x| IIF(x == NIL, Temp -> Clave, Temp -> Clave :=
  Val_Material(x)) },;
  { || Temp -> Descrip := MatDesc(Temp -> Clave) },;
  { || Temp -> Tipo := MatTipo(Temp -> Clave) },;
  { || Temp -> Costo := MatCosto(Temp -> Clave) },;
  { || Temp -> Importe:= Cantidad * Costo } }
```

```
lError := Falso
```

```
bError := ErrorBlock( {e| ErrHand(e) } )
```

```
Select 1
```

```
USE REQUERP SHARED INDEX REQUERP ALIAS
```

```
R_Cuerpo
```

```
ErrorBlock( bError )
```

```
IF lError
```

```
  CLOSE DATABASE
```

```
  Return
```

```
ENDIF
```

```
lError := Falso
```

```
bError := ErrorBlock( {e| ErrHand(e) } )
```

```
Select 2
```

```
USE REQCAEZ SHARED INDEX REQCAEZ ALIAS
```

```
R_Cabeza
```

```
ErrorBlock( bError )
```

```
IF lError
```

```
  CLOSE DATABASE
```

```
  Return
```

```
ENDIF
```

```
lError := Falso
```

```
bError := ErrorBlock( {e| ErrHand(e) } )
```

```
Select 4
```

```
USE DISTRIB SHARED INDEX DISTRIB ALIAS Reparte
```

```
ErrorBlock( bError )
```

```
IF lError
```

```
  CLOSE DATABASE
```

```
  Return
```

```

ENDIF

IError := Falso
bError := ErrorBlock( { |e| ErrHand(e) } )
Select 5
USE CAT_GRAL SHARED INDEX CAT_GRAL ALIAS
General
ErrorBlock( bError )
IF IError
  CLOSE DATABASE
  Return
ENDIF

IError := Falso
bError := ErrorBlock( { |e| ErrHand(e) } )
Select 6
USE ASIGNA SHARED INDEX ASIGNA, ASIGNA_3 ALIAS
Dar
ErrorBlock( bError )
IF IError
  CLOSE DATABASE
  Return
ENDIF

IError := Falso
bError := ErrorBlock( { |e| ErrHand(e) } )
Select 7
USE MATERIAL SHARED INDEX MATERIAL ALIAS
Utencilios
ErrorBlock( bError )
IF IError
  CLOSE DATABASE
  Return
ENDIF

Select 8
USE &cFileTmp EXCLUSIVE ALIAS Ayuda

CLS
DO PantallaREQ
***** PROCEDIMIENTO
PRINCIPAL
Do Cabezareq
Do While .T.
  lGuarda:=SAVESCREEN(0,0,1,79)
  Set Cursor Off
  Opcion := Inkey(0)
  Do Case
    Case Opcion = K_ALT_A
      @ 0, 2 SAY " Altas " COLOR "W+/B"
      @ 0,10 SAY " Bajas " COLOR "N+/W"
      @ 0,18 SAY " Consultas " COLOR "N+/W"
      Altas(Cierto)
    Case Opcion = K_ALT_B
      @ 0, 2 SAY " Altas " COLOR "N+/W"
      @ 0,10 SAY " Bajas " COLOR "W+/B"
      @ 0,18 SAY " Consultas " COLOR "N+/W"
      Bajas()
    Case Opcion = K_ALT_S
      @ 0, 2 SAY " Altas " COLOR "N+/W"
      @ 0,10 SAY " Bajas " COLOR "N+/W"
      @ 0,18 SAY " Consultas " COLOR "W+/B"
      Consultas()
    Case Opcion = K_ESC
      Exit
  EndCase
  RESTSCREEN(0,0,1,79,lGuarda)

```

```

EndDo
Close All
RUN DEL Z*.DBF
Set Color to "W/N"
Cls
***** ALTAS O MODIFICA
STATIC FUNCTION ALTAS(lExp)
Local Pant, nObjeto :=1, cGuarda:="G", nTotal:=0,Pant1
Pant:=SAVESCREEN(0,0,24,79)
SET CURSOR ON
SETCOLOR("W+/B,W+/B,..")
CREABASETMP()
Select R_Cabeza
nImporte:= 0
IF lExp = Cierto
  @ 5, 9 GET dFecha_Exp
  @ 5,38 GET nNo_Solic PICTURE "99999"
  VALID(!Empty(nNo_Solic))
  @ 5,44 GET cReq_Tipo PICTURE "!"
  VALID(cReq_Tipo$"ADI")
  @ 6, 9 GET cUPP PICTURE "99" VALID
  (Verif(cUPP))
  READ
  IF Lastkey() == 27
    BORRABASETMP()
    RESTSCREEN(0,0,24,79,Pant)
    Return
  ENDIF
  Select General
  Seek ("U"+cUPP)
  IF Found()
    @ 6,12 SAY General -> Descrip PICTURE "@S25"
  ENDIF
  Select R_Cabeza
  @ 6,53 GET cClave_Pres PICTURE "@R 9999-9999-9999-
9999-99" VALID(Checa(cClave_Pres))
  @ 6,75 GET cDigito PICTURE "9"
  VALID(Verifica(cDigito))
  @ 7,15 GET cDepend PICTURE "99" VALID
  (Its_Ok(cDepend))
  READ
  IF Lastkey() == K_ESC
    BORRABASETMP()
    RESTSCREEN(0,0,24,79,Pant)
    Return
  ENDIF
  Select General
  Seek ("U"+cUPP+cDepend)
  IF Found()
    @ 7,18 SAY General -> Descrip PICTURE "@S35"
  ENDIF
  Select R_Cabeza
  Compara:= cUPP
  SEEK(nNo_Solic)
  IF Found()
    G_Dspmsg("El Número de Solicitud ya Existe !!!", "N*")
    @ 5,23 GET nNo_Solic PICTURE "99999"
    VALID(!Empty(nNo_Solic))
  ELSE
    KEYBOARD Chr( K_INS )
    Cuerpo_Req(Cierto)
    IF Lastkey() == K_F10
      Grabar(Cierto)
    ENDIF
    IF Lastkey() == K_ESC
      Tone(900,0)
      Pant1:=SAVESCREEN(11,27,15,53)

```

```

SETCOLOR("W+/RB",,,)
LB_DispBox (12,28,14,52,,"Grabar",,)
@ 13,29 SAY "[G] Guardar o Cancelar" COLOR "W+/RB"
@ 13,33 SAY "G" COLOR "GR+/RB"
@ 13,43 SAY "C" COLOR "GR+/RB"
@ 13,30 GET cGuarda PICTURE "@K !" COLOR
"GR+/RB"
READ
IF cGuarda == "C" .OR. Lastkey() == 27
  RESTSCREEN(11,27,15,53,Pant1)
  RESTSCREEN(0,0,24,79,Pant)
  BORRABASETMP()
  Return
ELSE
  Cuerpo_Req(Cierto,Cierto)
  IF Lastkey() == K_ESC
    BORRABASETMP()
    RESTSCREEN(0,0,24,79,Pant)
    RETURN
  ELSE
    Grabar(Cierto)
    BORRABASETMP()
  ENDIF
  RESTSCREEN(0,0,24,79,Pant)
  Return
ENDIF
ENDIF
ENDIF
RESTSCREEN(0,0,24,79,Pant)
RETURN(Nil)
***** B A J A S
STATIC FUNCTION BAJAS()
Local Pant,cClave,cSubcve,cCve_Pres
Pant:=SAVESCREEN(0,0,24,79)
SET CURSOR ON
SETCOLOR("W+/B,W+/B,,")
CREABASETMP()
Select R_Cabeza
@ 5,38 GET nNo_Solic PICTURE "99999"
VALID(!Empty(nNo_Solic))
@ 5,44 GET cReq_Tipo PICTURE "!"
VALID(cReq_Tipo$"ADI")
READ
IF Lastkey() == 27
  BORRABASETMP()
  RESTSCREEN(0,0,24,79,Pant)
  Return
ENDIF
Seek(Str(nNo_Solic,5)+Req_Tipo)
IF BloqueaReg('R')
  IF Found()
    @ 5, 9 SAY Fecha_Exp
    @ 6, 9 SAY UPP
    cClave:=UPP
    Select General
    Seek ("U"+cClave)
    IF Found()
      @ 6,12 SAY General -> Descrip PICTURE "@S25"
    ENDIF
    Select R_Cabeza
    @ 6,53 SAY Clave_Pres PICTURE "@R 9999-9999-9999-
9999-99"
    cCve_Pres:=Clave_Pres
    @ 6,75 SAY Digito PICTURE "9"
    @ 7,15 SAY Depend PICTURE "99"
    cSubcve:=Depend

```

```

Select General
Seek ("U"+cClave+cSubcve)
IF Found()
  @ 7,18 SAY General -> Descrip PICTURE "@S25"
ENDIF
DO CAMBIAREQ
Keyboard Chr(27)
Go Top
Cuerpo_Req(Falso)
Inkey(0.5)
IF Opcion("Cancelar el Número de Solicitud?", "W+/R") =
Cierto
  Select R_Cabeza
  LimpiaReg()
  Select R_Cuerpo
  Seek(Str(nNo_Solic,5)+Req_Tipo)
  While (No_Solic == nNo_Solic .AND. !Eof())
    LimpiaReg()
    Skip
  EndDo
  Select Dar
  Set Order To 2
  Seek(nNo_Solic)
  While (No_Solic == nNo_Solic .AND. !Eof())
    IF Tipo_Oper == "78"
      BloqueaReg('R')
      Delete
      Unlock
    ENDIF
    Skip
  EndDo
  Set Order to 1
  Saldo(cCve_Pres)
ENDIF
ELSE
  Tone(200)
  G_Dspmsg("El Número de Solicitud no Existe","N*")
ENDIF
ELSE
  Tone(200)
  G_Dspmsg("Archivo Ocupado. Intente de Nuevo","N*")
ENDIF
BORRABASETMP()
RESTSCREEN(0,0,24,79,Pant)
RETURN
*****
STATIC FUNCTION CONSULTAS()
Local Pant,cClave,cSubcve
Pant:=SAVESCREEN(0,0,24,79)
SET CURSOR ON
SETCOLOR("W+/B,W+/B,,")
CREABASETMP()
Select R_Cabeza
@ 5,38 GET nNo_Solic PICTURE "99999"
VALID(!Empty(nNo_Solic))
@ 5,44 GET cReq_Tipo PICTURE "!"
VALID(cReq_Tipo$"ADI")
READ
IF Lastkey() == 27
  BORRABASETMP()
  RESTSCREEN(0,0,24,79,Pant)
  Return
ENDIF
Seek(Str(nNo_Solic,5)+Req_Tipo)
IF BloqueaReg('R')
  IF Found()
    @ 5, 9 SAY Fecha_Exp

```

```

@ 6, 9 SAY UPP
cClave:=UPP
Select General
Seek ("U"+cClave)
IF Found()
  @ 6,12 SAY General -> Descrip PICTURE "@S25"
ENDIF
Select R_Cabeza
@ 6,53 SAY Clave_Pres PICTURE "@R 9999-9999-9999-9999-9"
@ 6,75 SAY Digito PICTURE "9"
@ 7,15 SAY Depend PICTURE "99"
cSubcve:=Depend
Select General
Seek ("U"+cClave+cSubcve)
IF Found()
  @ 7,18 SAY General -> Descrip PICTURE "@S25"
ENDIF
DO CAMBIAREQ
Go Top
Cuerpo_Req(Falso)
ELSE
  Tone(200)
  G_Dspmsg("El Número de Solicitud no Existe","N*")
ENDIF
ELSE
  Tone(200)
  G_Dspmsg("Archivo Ocupado. Intente de Nuevo","N*")
ENDIF
BORRABASETMP()
RESTSCREEN(0,0,24,79,Pant)
RETURN
***** MUESTRA EL TBROWSE
STATIC FUNCTION Cuerpo_Req(lExpres,lCorrige)
  LOCAL Tecla, oCol, cGuarda, Sumatotal:= 0, lVer:= Cierto,
cExpre, Cortec
  IF lCorrige == NIL
    lCorrige:= Falso
  Else
    CorTec:= K_F10
  ENDIF
  IF lExpres == Cierto
    @ 24, 2 SAY "INS ->" COLOR "R/W"
    @ 24,10 SAY "Agregar" COLOR "N/W"
    @ 24,20 SAY "DEL ->" COLOR "R/W"
    @ 24,28 SAY "Borrar" COLOR "N/W"
    @ 24,51 SAY "ESC ->" COLOR "GR+/RB"
    @ 24,59 SAY "Salir " COLOR "W+/RB"
    @ 24,65 SAY " F10 ->" COLOR "GR+/RB"
    @ 24,73 SAY "Grabar " COLOR "W+/RB"
  ENDIF
  Select Temp
  Go Top
//
IF Lastkey() != 27
  oCuerpo := TBrowseDB( 9,3,19,76 )
  oCuerpo:HeadSep := "----"
  oCuerpo:ColSep := "| "
  oCuerpo:FootSep := "----"
  oCuerpo:ColorSpec:="N/W, W+/B, W+/B"

  oCuerpo:AddColumn(TBColumnNew( "No", { |x|
Temp -> No:= Req_Sec(x) })
oCuerpo:AddColumn(TBColumnNew( "Cant.", { ||
Temp -> Cantidad })
oCuerpo:AddColumn(TBColumnNew( "Clave", { ||
Transform( Temp -> Clave,"@R 9999-999" ) })

```

```

oCol:=(TBColumnNew( "Descripción", { || Temp ->
Descrip := MatDesc(Temp -> Clave) })
oCol:Width:=30
oCuerpo:AddColumn( oCol )
oCuerpo:AddColumn(TBColumnNew( "Forma", { ||
Temp -> Tipo:=MatTipo(Temp -> Clave) })
oCuerpo:AddColumn(TBColumnNew( "Costo", { ||
Temp -> Costo:=MatCosto(Temp -> Clave) })
oCuerpo:AddColumn(TBColumnNew( "Importe", { ||
Transform( Temp -> Importe:=Cantidad * Costo, "9,999.99" ) }) )

Do While Cierto
  oCuerpo:ForceStable()
  IF oCuerpo:Stable
    IF oCuerpo:HitTop .OR. oCuerpo:HitBottom()
      Tone(125)
    ENDIF
    Tecla := 0
    Do While Tecla == 0
      oCuerpo:RefreshCurrent()
      oCuerpo:ColorRect( {
oCuerpo:RowPos,1,oCuerpo:RowPos,oCuerpo:ColCount },{ 1,2
})
      oCuerpo:Hilite()
      Tecla := Inkey(1)
    IF lCorrige == Cierto
      Tecla := Cortec
      lCorrige:=Falso
    ENDIF
  EndDo
ENDIF
DO CASE
  CASE Tecla == K_ESC
    Exit
  CASE Tecla== K_F10
    lVer:= Cierto
    Go Top
    While !Eof()
      cExpre:= Temp -> Clave
      IF Check_Material(Temp -> Clave) == Cierto
        Skip
      Else
        Locate For Clave == cExpre
        G_Dspmsg("Clave del Material"+Transform(
cExpre,"@R 9999-999")+ " Inválida", "R")
        oCuerpo:PanHome()
        oCuerpo:Right()
        oCuerpo:Right()
        nPos := Recno()
        Go nPos
        oCuerpo:RowPos:=nPos
        oCuerpo:Hilite()
        lVer:=Falso
      EndIf
    EndDo
    nPos:=0
    Go Top
    While !Eof() .AND. lVer == Cierto
      IF Temp -> Cantidad # 0
        Skip
      ELSE
        G_Dspmsg("El Cantidad no puede ir vacía", "R")
        oCuerpo:PanHome()
        oCuerpo:Right()
        nPos:=Recno()
        Go nPos

```

```

oCuerpo:RowPos:=nPos
oCuerpo:Hilite()
IVer:=Falso
Exit
EndIF
EndDo
IF IVer == Cierto
  Select Temp
  Go Top
  IF Busca_Material(Temp -> Clave) == Cierto
    EXIT
  ENDIF
ENDIF
oCuerpo:RefreshAll()
CASE Tecla==K_DEL .AND. IExpres == Cierto
  DELETE
  PACK
  Suma()
  oCuerpo:RefreshAll()
CASE Tecla==K_INS .AND. IExpres == Cierto
  APPEND BLANK
  Go Top
  oCuerpo:PanHome()
  oCuerpo:Down()
  oCuerpo:RefreshAll()
CASE Between(Tecla, 32, 255) .OR. Tecla == K_ENTER
  IF IExpres = Cierto
    KEYBOARD Chr(Tecla)
    Edicion( abGet[ oCuerpo:ColPos ],oCuerpo)
  ENDIF
  Suma()
  oCuerpo:RefreshAll()
OTHERWISE
  Suma()
  Go Top
  oCuerpo:RefreshAll()
  TECLAS33(oCuerpo,Tecla)
ENDCASE
EndDo
ENDIF
Return

STATIC FUNCTION Suma()
Go Top
nTotal:=0
While !Eof()
  nTotal+= Temp -> Importe
  IF oCuerpo:ColPos > 6 .AND. oCuerpo:ColPos <= 8
    oCuerpo:RefreshAll()
    @ 20,59 SAY "Total" COLOR "N/W"
    @ 20,65 SAY nTotal PICTURE "9,999.99" COLOR
    "N/W"
  ENDIF
  Skip
EndDo
Skip -1
RETURN
*****
STATIC FUNCTION Edicion(bBloque,oBrowse)
// LOCAL bBloque :=
oBrowse:GetColumn(oBrowse:ColPos):Block
  LOCAL xVar, oGet
  oBrowse:Stabilize()
  xVar := Eval(bBloque)
  oGet := GetNew(Row(), Col(), { |x| IIF(x == NIL, xVar, xVar
:= x) }, "xVar")
DO CASE

```

```

CASE oBrowse:ColPos = 1
  oGet:Picture:="99"
  oGet:postBlock:={|x| !Empty(x)}
CASE oBrowse:ColPos = 2
  oGet:Picture:="999"
  oGet:postBlock:={|x| !Empty(x)}
CASE oBrowse:ColPos = 3
  oGet:Picture:="@R 9999-999"
  oGet:postBlock:={|x| !Empty(x)}
CASE oBrowse:ColPos = 4
  oGet:Picture:="@S" +
  Ltrim(Str(oBrowse:GetColumn(oBrowse:ColPos):Width))
CASE oBrowse:ColPos > 5 .AND. oBrowse:ColPos < 8
  oGet:Picture:="9,999.99"
  oGet:postBlock:={|x| !Empty(x)}
ENDCASE
ReadModal( { oGet } )
oBrowse:Right()
If Updated()
  Eval(bBloque, xVar)
  EndIf
oBrowse:RefreshCurrent()
RETURN
*****
Static Function Grabar(IVar)
Local Pant, cGuarda:="G", nCargo:=0, nTotal:=0, nReg:=0,
cAux, nSaldo:=0,;
nAbono:=0, lBand:=Cierto, Auxfec
  Select Temp
  Auxfec:=CtoD("01"+Str(Month(dFecha_Exp))+Str(Year(dFecha_
Exp)))
  Go Top
  While !Eof()
    Select Dar
    Seek(cClave_Pres + Dtos(Auxfec))
    While (Temp_cv == cClave_Pres .AND. Month(Fecha_Mov)
== Month(dFecha_Exp) .AND. !Eof())
      Skip
    Enddo
    Skip -1
    IF ( Dar -> Temp_Cv == cClave_Pres .AND. Dar -> Total > 0
)
      IF Temp -> Importe <= Dar ->Total
        nTotal:= Dar -> Total - Temp -> Importe
        IF AgregaReg("2")
          REPLACE Total WITH nTotal, Tipo_Oper WITH "78",
No_Solic WITH nNo_Solic,;
          Comprom WITH Temp -> Importe, Fecha_Mov
WITH dFecha_Exp,;
          Temp_cv WITH cClave_Pres
        COMMIT
        UNLOCK
      ELSE
        LimpiaReg()
        G_Dspmsg("Error al Dar de Alta el Registro !!", "R")
      ENDIF
    ELSE
      G_Dspmsg("El Cargo es Mayor que el Fondo de la
Partida", "R")
    Enddo
    Cuerpo_Req(Cierto)
    IF Lastkey() == K_ESC
      lBand:= .F.
      Exit
    ELSE
      Grabar(Cierto)
      lBand:=F.
    Enddo
  EndWhile

```

```

Exit
ENDIF
ENDIF
ELSE
G_Dspmsg("La Partida no tiene suficientes fondos para
cubrir el gasto","R")
Cuerpo_Req(Cierto)
IF Lastkey() = K_ESC
IBand:=.F.
Exit
ELSE
Grabar(Cierto)
IBand:=.F.
Exit
ENDIF
ENDIF
Select Temp
Skip
EndDo
IF IBand = Cierto
Select R_Cabeza
Seek(Str(nNo_Solic,5) + Req_Tipo)
IF Found()
G_Dspmsg("El Número de Solicitud ya Existe !!!","R")
IBand:=.F.
Return
ENDIF
IF IBand = Cierto
IF AgregaReg('2')
REPLACE No_Solic WITH nNo_Solic, UPP WITH
cUPP, Depend WITH cDepend,;
Digito WITH cDigito, Fecha_Exp WITH
dFecha_Exp, Total WITH nTotal,;
Clave_Pres WITH cClave_Pres, Req_Tipo WITH
cReq_Tipo
COMMIT
UNLOCK
ELSE
LimpiaReg()
G_Dspmsg("Error al Dar de Alta el Registro !!","R")
ENDIF
ENDIF
Selec Temp
Go Top
While !Eof() .AND. IBand = Cierto
Select R_Cuerpo
IF AgregaReg('2')
REPLACE No_Solic WITH nNo_Solic, No WITH
Str(Temp -> No,2), Req_Tipo WITH cReq_Tipo,;
Cantidad WITH Temp -> Cantidad, Clave WITH
Temp -> Clave,;
Descrip WITH Temp -> Descrip, Costo WITH
Temp -> Costo,;
Importe WITH Temp -> Importe, Tipo WITH Temp
-> Tipo
COMMIT
UNLOCK
ELSE
LimpiaReg()
G_Dspmsg("Error al Dar de Alta el Registro !!","R")
ENDIF
Select Temp
Skip
EndDo
Saldo(cClave_Pres)
BORRABASETMP()
ENDIF

```

```

RESTSCREEN(11,27,15,53,Pant)
Return(Nil)
***** ADICIONALES
PROCEDURE PANTALLAREQ
SETCOLOR("N/W,")
dispbox(0,0,24,79," | |",//acColor[11])
@ 0, 0 SAY Space(80) //COLOR acColor[10]
@ 2,23 SAY" REQUISICION DE BIENES DE CONSUMO "
COLOR "B/W"
A_Dispatch( 3,1,22,78,"W",.T. )
@ 0, 2 SAY " Itas ajas Con ultas " COLOR "N/W"
@ 0, 3 SAY "A" COLOR "R/W"
@ 0,11 SAY "B" COLOR "R/W"
@ 0,22 SAY "s" COLOR "R/W"
RETURN

PROCEDURE CABEZAREQ
@ 5, 3 SAY "Fecha"
@ 5,21 SAY "Número Solicitud"
@ 6, 3 SAY "U.P.P."
@ 6,40 SAY "Clave Presup."
@ 7, 3 SAY "Dependencia"
@ 8, 2 SAY "+" COLOR "N/W"
@ 8, 3 SAY REPLICATE("-",74)
@ 8,77 SAY "!" COLOR "W+W"
RETURN

PROCEDURE CAMBIAREQ
Select R_Cuerpo
Seek(Str(nNo_Solic,5)+cReq_Tipo)
While(No_Solic = nNo_Solic .AND. Req_Tipo = cReq_Tipo
.AND. !Eof())
IF BloqueaReg('R')
Select Temp
IF AgregaReg()
REPLACE No_Solic WITH R_Cuerpo -> No_Solic, ;
No WITH Val(R_Cuerpo -> No), ;
Clave WITH R_Cuerpo -> Clave, ;
Cantidad WITH R_Cuerpo -> Cantidad, ;
Descrip WITH R_Cuerpo -> Descrip, ;
Costo WITH R_Cuerpo -> Costo, ;
Importe WITH R_Cuerpo -> Importe, ;
Tipo WITH R_Cuerpo -> Tipo
COMMIT
UNLOCK
ELSE
LimpiaReg()
G_Dspmsg("Error al Dar de Alta el Registro","R")
ENDIF
ELSE
Tone(800)
G_Dspmsg("Registro Ocupado. Intente de Nuevo","N*")
ENDIF
Select R_Cuerpo
Skip
EndDo
RETURN

STATIC FUNCTION Checa(cClave)
Select Reparte
Seek(cClave)
IF Found() .AND. (Substr(cClave,13,4) > "1999" .AND.
Substr(cClave,13,4) < "3001");
.OR. (Substr(cClave,13,4) > "4999" .AND.
Substr(cClave,13,4) < "5001")
SELECT R_Cabeza
Return(.T.)

```

```

ELSE
  G_Dspmsg("La Clave Presupuestal o el Capitulo o no
Coincide", "N*")
  Return(.F.)
ENDIF
Return(.F.)
*****
STATIC FUNCTION Verifica(cDig)
Local Residuo:=0, Resultado:=0, Tempor, Vartemp, j:= 17, d:=2,
Producto:=0,;
  Final:=0
While(j > 0 )
  Vartemp:=Substr(cClave_Pres,j,1)
  If j == 1 .OR. j == 9 .OR. j == 17
    Producto:= Val(Vartemp) * d
  Else
    Producto:= Val(Vartemp) * d
  EndIf
  j--
  If d == 9
    d:= 1
  Endif
  d++
  Resultado += Producto
EndDo
Residuo := Mod(Resultado,11)
Final1 := 11 - Residuo
If Final1 = 10 .OR. Final1 = 11
  Final1 := 0
EndIf
If Str(Final1,1) = cDig
  Return(.T.)
Else
  Tone(200)
  G_Dspmsg("El Dígito No Corresponde a la Clave", "N*")
EndIf
Return(.F.)
*****
STATIC FUNCTION Val_Material(cExp)
Select Utencilios
Seek(cExp)
IF Found()
  Return (.T.)
ELSE
  lGuarda:= SaveScreen(5,34,10,76)
  Go Top
  While !Eof()
    Select Ayuda
    APPEND BLANK
    REPLACE Material WITH Utencilios -> Material,;
      Descrip WITH Utencilios -> Descrip
    Select Utencilios
    Skip
  EndDo
  Obj:=GetActive()
  Obj:Varput(Ayuda_Mat())
ENDIF
Select Utencilios
RETURN(.T.)
***** MUESTRA EL TBROWSE DE
AYUDA
FUNCTION Ayuda_Mat()
LOCAL Tecla, oColumna, lExp, reng:=6
//
  Select Ayuda
  Set Delete On
  Go Top

```

```

@ 5,34,10,76 BOX "+-!+-+!" COLOR "W+/GB"
oAyuda:=TBrowseDB( 5,35,10,75 )
oAyuda:HeadSep := "----"
oAyuda:ColSep := " | "
oAyuda:FootSep := "----"
oAyuda:ColorSpec:="W+/GB, W+/B, W+/B, N/GB, B/W"
//
oAyuda:AddColumn(TBColumnNew( "", { || Material } ))
oColumna:=(TBColumnNew( "", { || Descrip } ))
oColumna:Width:=30
oAyuda:AddColumn( oColumna )
//
WHILE .T.
  oAyuda:ForceStable()
  IF oAyuda:Stable
    IF oAyuda:HitTop .OR. oAyuda:HitBottom()
      Tone(800,0)
    ENDIF
    oAyuda:ColorRect( {
oAyuda:RowPos,1,oAyuda:RowPos,oAyuda:ColCount }, { 3,2 } )
    oAyuda:Hilite()
    Tecla:=Inkey(0)
  ENDIF
  DO CASE
    CASE Tecla == K_ESC .OR. Tecla == K_ENTER
      Auxcve:= Material
      RESTSCREEN(5,34,10,76,lGuarda)
      Exit
      oAyuda:RefreshAll()
    OTHERWISE
      TECLAS33(oAyuda,Tecla)
  ENDCASE
ENDDO
Select Ayuda
Delete All
RETURN
*****
STATIC FUNCTION TECLAS33(O,T)
//
Do Case
  Case T == K_DOWN
    O:Down()
  Case T == K_UP
    O:Up()
  Case T == K_PGDN
    O:PageDown()
  Case T == K_PGUP
    O:PageUp()
  Case T == K_CTRL_HOME
    O:GoTop()
  Case T == K_CTRL_END
    O:GoBottom()
  Case T == K_RIGHT .OR. T == K_TAB
    O:Right()
  Case T == K_LEFT .OR. T == K_SH_TAB
    O:Left()
  Case T == K_HOME
    O:Home()
  Case T == K_END
    O:End()
EndCase
O:RefreshCurrent()
Return

```

REPORTE EJECUCION PRESUPUESTAL

```
#INCLUDE "Inkey.Ch"
#INCLUDE "Box.ch"
#INCLUDE "Fileio.ch"

Private cUPP:= cDep:=Space(2), cTipo:= Space(1)

Private Struct := { {"Tipo","C",1,0}, {"Clave","C",2,0},;
                  {"Subclave","C",2,0}, {"Descrip","C",50,0} }

cFileTmp := NombreUnico()
DBCcreate(cFileTmp + ".Dbf", Struct)

!Error := .F.
bError := ErrorBlock( { |e| ErrHand(e) } )
SELECT 1
USE EJECABEZ INDEX EJECABE1 EXCLUSIVE ALIAS
E_Cabeza
IF !Error
    DELETE FILE (cFileTmp + ".Dbf")
    CLOSE DATABASES
    Return
ENDIF

!Error := .F.
bError := ErrorBlock( { |e| ErrHand(e) } )
SELECT 2
USE EJECUERP INDEX EJECUERP EXCLUSIVE ALIAS
E_Cuerpo
IF !Error
    DELETE FILE (cFileTmp + ".Dbf")
    CLOSE DATABASES
    Return
ENDIF

!Error := .F.
bError := ErrorBlock( { |e| ErrHand(e) } )
SELECT 3
USE CAT_GRAL INDEX CAT_GRAL EXCLUSIVE ALIAS
General
IF !Error
    DELETE FILE (cFileTmp + ".Dbf")
    CLOSE DATABASES
    Return
ENDIF

Select 4
USE &cFileTmp EXCLUSIVE ALIAS Ayuda

IF G_Impresora("W+/R")
    Set Cursor On
    G_DispBox( 9,19,11,65,"B",.T. )
    @ 9,24 SAY "Documento de Ejecución Presupuestaria"
    COLOR"GR+/B"
    Setcolor("W+/B,N/W")
    Select General
    cTipo:="U"
    @ 10,21 SAY "U.P.P. "
    @ 10,44 SAY "Dependencia "
    @ 10,34 GET cUPP PICTURE "99" VALID
    Valida_C(cUPP,"00")
    @ 10,57 GET cDep PICTURE "99" VALID
    Valida_C(cUPP, cDep)
    READ
    IF Lastkey() = K_ESC
        Return
    
```

```
ENDIF
G_VentProceso( { || Imp_Ejecucion() }, "Imprimiendo Resumen
...." )
ENDIF
DELETE FILE (cFileTmp + ".Dbf")
CLOSE DATABASES
RETURN
*****
FUNCTION Imp_Ejecucion( )
    LOCAL nHoja:= 0, fp:=FOpen( "LPT1",FO_WRITE ),
    nRen:=7, nCont:=0;
    nTot := 0, Aux, Auxtemp, Solic:=0
    Set Cursor Off
    IF cDep # "00"
        Titulo( fp,++nHoja )
        Aux:= Alltrim(cUPP+cDep)
        nTot:= LastRec()
        SEEK(cUPP+cDep)
        Auxtemp := ""
        WHILE( !Eof() .AND. Upp+Depend == Aux)
            G_BarraProceso( ++nCont,nTot )
            IF nRen > 60
                FWrite( fp,Chr( 12 ) )
                Titulo( fp,++nHoja )
                nRen := 7
            ENDIF
            nRen++
            Select E_Cabeza
            FWrite( fp,Chr( 18 )+Chr( 27 )+Chr( 70 )+Chr(15))
            FWrite( fp,Space(2)+Upp+Space(3)+Depend+Space(5);
                +Str(No_Solic,5)+Space(8)+Tipo_Docto+Space(12);
                +Digito+Space(6)+Dtoc(Fecha_Exp)+Space(3);
                +Transform(Importe, "99,999,999,999.99")+
            Chr(10))
            Solic:= No_Solic
            Select E_Cuerpo
            Seek(Solic)
            While(No_Solic == Solic .AND. !Eof())
                FWrite(
            fp,Space(4)+No_Sec+Space(4)+Clave_Pres+Space(8);
                +Concepto+Space(4);
                +Transform(Cargo,
            "99,999,999,999.99")+Space(4);
                +Transform(Deducion,
            "99,999,999,999.99")+Space(4);
                +Transform(Total,
            "99,999,999,999.99")+Chr(10))
                Skip
            EndDo
            Select E_Cabeza
            SKIP
        ENDDO
    ENDIF
    FClose( fp )
    EJECT
RETURN( NIL )

STATIC FUNCTION Titulo( nfp,nExp )
    FWrite( nfp,Chr( 18 )+Chr( 27 )+Chr( 69 ) )
    FWrite( nfp,"D.I.E. DIRECCION DE
PROGRAMACION Y PRESUPUESTO HOJA: "+Str(
nExp,3 )+Chr( 10 ) )
    FWrite( nfp,G_Alineacion( "RESUMEN DE DOCUMENTOS
DE EJECUCION PRESUPUESTARIA",80,'C' )+Chr( 10 ) )
    FWrite( nfp,G_Alineacion( "POR U.P.P. Y
DEPENDENCIA",80,'C' )+Chr( 10 ) )
    FWrite( nfp,G_Alineacion( Dtoc(Date( )),79,'D' )+Chr( 10 ) )

```

```

Select General
Seek(cTipo+cUPP)
IF Found()
  FWrite( nfp, "U.P.P." + Space(2) + Descrip + Chr( 10 ) )
ENDIF
Seek(cTipo+cUPP+cDep)
IF Found()
  FWrite( nfp, "DEPENDENCIA" + Space(2) + Descrip + Chr( 10 ) )
)
ENDIF
Select E_Cabeza
FWrite( nfp, Replicate("-",80) + Chr( 10 ) )
FWrite( nfp, Chr( 18 ) + Chr( 27 ) + Chr( 69 ) + Chr(15) )
FWrite( nfp, Space( 1 ) + "UPP DEP FOLIO TIPO DOCTO
DIGITO FECHA IMPORTE
" + Chr( 10 ) )
FWrite( nfp, Space( 3 ) + "SEC CLAVE PRESUPUESTAL
CONCEPTO CARGO
DEDUCCION LIQUIDO" + Chr( 10 ) )
FWrite( nfp, Chr( 18 ) + Chr( 27 ) + Chr( 69 ) )
FWrite( nfp, Replicate("-",80) + Chr( 10 ) )
FWrite( nfp, Chr( 27 ) + Chr( 70 ) )
RETURN( NIL )
*****
STATIC FUNCTION VALIDA_C(cExp, cExp1)
Select General
Seek(cTipo+cExp+cExp1)
IF Found()
  Return (.T.)
ELSE
  lGuarda:= SaveScreen(5,34,10,76)
  Go Top
  Seek(cTipo)
  While !Eof() .AND. Tipo == cTipo
    Select Ayuda
      APPEND BLANK
      REPLACE Clave WITH General -> Clave, Subclave
WITH General -> Subclave,;
      Descrip WITH General -> Descrip
    COMMIT
    UNLOCK
    IF cExp1 == "00"
      DELETE For Subclave <> "00"
    ELSE
      DELETE For Subclave == "00"
      DELETE For Clave <> cUPP
    ENDIF
  Select General
  Skip
  EndDo
  Obj:=GetActive()
  Obj:Varput(Visual(cExp1))
ENDIF
Select General
RETURN(.T.)
***** MUESTRA EL TBROWSE DE AYUDA
STATIC FUNCTION Visual(cExp1)
LOCAL Tecla, oColumna, lExp, reng:=6
//
Select Ayuda
Go Top
@ 5,34,10,76 BOX "Í-À|ç-È!" COLOR "W+/GB"
oVisual:=TBrowseDB( 5,35,10,75 )
oVisual:HeadSep := "----"
oVisual:ColSep := "| "
oVisual:FootSep := "----"
oVisual:ColorSpec:="W+/GB, W+/B, W+/B, N/GB, B/W"

```

```

//
IF cExp1 == "00"
  oVisual:AddColumn(TBColumnNew( "", { || Clave } ))
  oColumna:=(TBColumnNew( "", { || Descrip } ))
  oColumna:Width:=30
  oVisual:AddColumn( oColumna )
ELSE
  oVisual:AddColumn(TBColumnNew( "", { || Subclave } ))
  oColumna:=(TBColumnNew( "", { || Descrip } ))
  oColumna:Width:=30
  oVisual:AddColumn( oColumna )
ENDIF
//
WHILE .T.
  oVisual:ForceStable()
  IF oVisual:Stable
    IF oVisual:HitTop .OR. oVisual:HitBottom()
      Tone(800,0)
    ENDIF
    oVisual:ColorRect( {
oVisual:RowPos,1,oVisual:RowPos,oVisual:ColCount }, { 3,2 } )
    oVisual:Hilite()
    Tecla:=Inkey()
  ENDIF
  DO CASE
    CASE Tecla == K_ESC .OR. Tecla == K_ENTER
      IF cExp1 == "00"
        Auxcve := Clave
      ELSE
        Auxcve := Subclave
      ENDIF
      RESTSCREEN(5,34,10,76,lGuarda)
      Exit
      oVisual:RefreshAll()
    OTHERWISE
      TECLAS33(oVisual,Tecla)
    ENDCASE
  ENDDO
  Select Ayuda
  Zap
  RETURN ( Auxcve )
STATIC FUNCTION TECLAS33(O,T)
//
Do Case
  Case T == K_DOWN
    O:Down()
  Case T == K_UP
    O:Up()
  Case T == K_PGDN
    O:PageDown()
  Case T == K_PGUP
    O:PageUp()
  Case T == K_CTRL_HOME
    O:GoTop()
  Case T == K_CTRL_END
    O:GoBottom()
  Case T == K_RIGHT
    O:Right()
  Case T == K_LEFT
    O:Left()
  Case T == K_HOME
    O:Home()
  Case T == K_END
    O:End()
  EndCase
  O:RefreshCurrent()

```

CATALOGO DE PARTIDAS

```
#Include "Inkey.ch"
#Define Cierto .T.
#Define Falso .F.
#Define Between(x, y, z) ((x) >= (y) .And. (x) <= (z))

Private cClave:=Space(0), i := 1

IError := Falso
bError := ErrorBlock( { |e| ErrHand(e) } )
Select 1
USE PARTIDAS SHARED INDEX PARTIDAS ALIAS Partidas
ErrorBlock( bError )
IF IError
    CLOSE DATABASES
    Return
ENDIF

CLS
*****
PROCEDIMIENTO PRINCIPAL
DO Pantalla
Despliega_Partida()
CLOSE ALL
RELEASE ALL
Set Color to "W/N"
CLS
RETURN

***** MUESTRA
EL TBROWSE DE LA PARTIDA
STATIC FUNCTION Despliega_Partida()
LOCAL Tecla, oColumna2, oColumna3, oColumna1, IPant
//
Select Partidas
oPartida:=TBrowseDB( 5,3,21,76 )
oPartida:HeadSep := "ÄÄÄ"
oPartida:ColSep := " "
oPartida:FootSep := "ÄÄÄ"
oPartida:ColorSpec:="N/W, W+/GB, W+/B, GR/W"
//
oColumna1:= ( TBColumnNew( "Partidas", { || Partida } ) )
oColumna1:ColorBlock := { || IIF(Right(Partida,2) =
"00",{4,2},{1,2}) }
oPartida:AddColumn(oColumna1)
oColumna2:= TBColumnNew( "Descripción Corta", { ||
Desc_Corta } )
oColumna2:Width:=28
oPartida:AddColumn(oColumna2)
oColumna3:= TBColumnNew( "Descripción Larga", { ||
Desc_Larga } )
oColumna3:Width:=28
oPartida:AddColumn(oColumna3)
@ 6,2 SAY "Ä" COLOR "N/W"
@ 6,77 SAY "" COLOR "W+/W"
//
WHILE T.
oPartida:ForceStable()
IF oPartida:Stable
    IF oPartida:HitTop .OR. oPartida:HitBottom()
        Tone(800,0)
    ENDIF
    Tecla := 0
    Do While Tecla == 0
        oPartida:RefreshCurrent()
```

```
oPartida:ColorRect( {
oPartida:RowPos,1,oPartida:RowPos,oPartida:ColCount }, { 3,2 }
)
oPartida:Hilite()
Tecla := Inkey(1)
EndDo
ENDIF
IPant := SAVESCREEN(0,0,1,79)
DO CASE
CASE Tecla==K_ESC
    SETCOLOR("W/N,," )
    EXIT
CASE Tecla==K_ALT_A
    @ 24,51 SAY " ESC -> " COLOR "GR+/RB"
    @ 24,59 SAY " Salir " COLOR "W+/RB"
    @ 24,65 SAY " F10 -> " COLOR "GR+/RB"
    @ 24,73 SAY " Grabar " COLOR "W+/RB"
    @ 0,2 SAY " Altas " COLOR "W+/B"
    @ 0,9 SAY " Bajas " COLOR "N+/W"
    @ 0,16 SAY " Modificaciones " COLOR "N+/W"
    ALTAS(Cierto)
    i := 1
    cClave:=Space(0)
    oPartida:RefreshAll()
CASE Tecla==K_ALT_B // Da de Baja Partidas
    @ 0,2 SAY " Altas " COLOR "N+/W"
    @ 0,9 SAY " Bajas " COLOR "W+/B"
    @ 0,16 SAY " Modificaciones " COLOR "N+/W"
    IF BloqueaReg("R")
        IF Opcion("Desea Eliminar el Registro?", "W+/R") =
Cierto
            LimpiaReg()
        ENDIF
    ELSE
        Tone(800)
        G_Dspmsg("Registro Ocupado. Intente de Nuevo","R")
        i:= 1
        cClave:=Space(0)
        Return
    ENDIF
    oPartida:RefreshAll()
CASE Tecla==K_ALT_M
    @ 24,51 SAY " ESC -> " COLOR "GR+/RB"
    @ 24,59 SAY " Salir " COLOR "W+/RB"
    @ 24,65 SAY " F10 -> " COLOR "GR+/RB"
    @ 24,73 SAY " Grabar " COLOR "W+/RB"
    @ 0,2 SAY " Altas " COLOR "N+/W"
    @ 0,9 SAY " Bajas " COLOR "N+/W"
    @ 0,16 SAY " Modificaciones " COLOR "W+/B"
    ALTAS(Falso)
    i := 1
    cClave:=Space(0)
    oPartida:RefreshAll()
CASE Between(Tecla, 45, 57)
    Set SoftSeek On
    IF i <= Len(Partida)
        cClave+= Chr(Tecla)
        Seek(cClave)
        oPartida:RefreshAll()
        i++
    ENDIF
    Set SoftSeek Off
    oPartida:RefreshAll()
OTHERWISE
    TECLAS33(oPartida,Tecla)
ENDCASE
RESTSCREEN(0,0,1,79,IPant)
```

```

ENDDO
RETURN ( NIL )

***** ALTAS O
MODIFICA
STATIC FUNCTION ALTAS(IExp)
Local Pant, Tecla, cGuarda:="G", Pant1
cPartida:=Space(4); cDesc_Corta:=Space(40);
cDesc_Larga:=Space(80)
SET CURSOR ON
//SETCOLOR("W+/B,W+/B,,")

IF IExp = Cierto
  G_DispatchBox( 9,15,17,68,"B",.F. )
  Part_Ele()
  @ 11,26 GET cPartida PICTURE "9999"
VALID(!Empty(cPartida))
  READ
  IF Lastkey() == K_ESC
    Go Top
    @ 24, 0 SAY REPLICATE(" ",80) COLOR "W+/W"
    Return
  ENDIF
  SEEK(cPartida)
  IF Found()
    G_Dspmsg("Partida ya se Encuentra en Catalogo !!!","R")
    Return(Nil)
  ELSE
    @ 13,36 GET cDesc_Corta PICTURE "@!S28"
VALID(!Empty(cDesc_Corta))
    @ 15,36 GET cDesc_Larga PICTURE "@!S28"
VALID(!Empty(cDesc_Larga))
    READ
    IF Lastkey() == K_ESC
      Go Top
      @ 24, 0 SAY REPLICATE(" ",80) COLOR "W+/W"
      Return
    ENDIF
    Set Cursor Off
    Tecla:=Inkey(0)
    Set Cursor On
    DO CASE
      CASE Tecla == K_F10
        Grabar(Cierto)
      CASE Tecla == K_ESC
        Tone(900,0)
        Pant1:=SAVESCREEN(11,27,15,53)
        SETCOLOR("W+/RB",,,)
        LB_DispatchBox( 12,28,14,52,,"Grabar",,)
        @ 13,29 SAY "[G] Guardar o Cancelar" COLOR
"W+/RB"
        @ 13,33 SAY "G" COLOR "GR+/RB"
        @ 13,43 SAY "C" COLOR "GR+/RB"
        @ 13,30 GET cGuarda PICTURE "@K !" COLOR
"GR+/RB"
        READ
        IF cGuarda == "C" .OR. Lastkey() == 27
          Go Top
          @ 24, 0 SAY REPLICATE(" ",80) COLOR "W+/W"
          RESTSCREEN(11,27,15,53,Pant1)
          Return (Nil)
        ELSE
          @ 24, 0 SAY REPLICATE(" ",80) COLOR "W+/W"
          Grabar(Falso)
          Return
        ENDIF
      ENDCASE
    ELSE
      Tone(800)
      G_Dspmsg("Registro Ocupado. Intente de Nuevo","R")
      Return
    ENDIF
  UNLOCK
ENDIF
SET CURSOR OFF
Return (Nil)

*****
Static Function Grabar(IExp)
IF IExp == Cierto
  Seek(cPartida)
  IF Found()
    G_Dspmsg("El N°mero de Solicitud ya Existe !!!","R")
    Return
  ENDIF

```

```

ENDIF
ENDCASE
ENDIF
ELSE
  IF BloqueaReg('R')
    G_DispatchBox( 9,15,17,68,"B",.F. )
    Part_Ele()
    cDesc_Corta:=Desc_Corta; cDesc_Larga:=Desc_Larga;
cPartida:=Partida
    @ 11,26 SAY cPartida PICTURE "9999" COLOR "W+/B"
    @ 13,36 GET cDesc_Corta PICTURE "@!S28"
VALID(!Empty(cDesc_Corta))
    @ 15,36 GET cDesc_Larga PICTURE "@!S28"
VALID(!Empty(cDesc_Larga))
    READ
    IF Lastkey() == K_ESC
      Go Top
      @ 24, 0 SAY REPLICATE(" ",80) COLOR "W+/W"
      Return
    ENDIF
    Set Cursor Off
    Tecla:=Inkey(0)
    Set Cursor On
    DO CASE
      CASE Tecla == K_F10
        Grabar(Falso)
      CASE Tecla == K_ESC
        Tone(900,0)
        Pant1:=SAVESCREEN(11,27,15,53)
        SETCOLOR("W+/RB",,,)
        LB_DispatchBox( 12,28,14,52,,"Grabar",,)
        @ 13,29 SAY "[G] Guardar o Cancelar" COLOR
"W+/RB"
        @ 13,33 SAY "G" COLOR "GR+/RB"
        @ 13,43 SAY "C" COLOR "GR+/RB"
        @ 13,30 GET cGuarda PICTURE "@K !" COLOR
"GR+/RB"
        READ
        IF cGuarda == "C" .OR. Lastkey() == 27
          Go Top
          @ 24, 0 SAY REPLICATE(" ",80) COLOR "W+/W"
          RESTSCREEN(11,27,15,53,Pant1)
          Return (Nil)
        ELSE
          @ 24, 0 SAY REPLICATE(" ",80) COLOR "W+/W"
          Grabar(Falso)
          Return
        ENDIF
      ENDCASE
    ELSE
      Tone(800)
      G_Dspmsg("Registro Ocupado. Intente de Nuevo","R")
      Return
    ENDIF
  UNLOCK
ENDIF
SET CURSOR OFF
Return (Nil)

*****
Static Function Grabar(IExp)
IF IExp == Cierto
  Seek(cPartida)
  IF Found()
    G_Dspmsg("El N°mero de Solicitud ya Existe !!!","R")
    Return
  ENDIF

```

```

ENDIF
IF IExp = Cierto
  IF AgregaReg('2')
    REPLACE Partida WITH cPartida, Desc_Corta WITH
cDesc_Corta,;
    Desc_Larga WITH cDesc_Larga
    COMMIT
    UNLOCK
  ELSE
    LimpiaReg()
    G_Dspmsg("Error al Dar de Alta el Registro !!", "R")
  ENDIF
ELSE
  REPLACE Desc_Corta WITH cDesc_Corta, Desc_Larga
WITH cDesc_Larga
  COMMIT
  UNLOCK
ENDIF
RETURN
***** ADICIONALES
PROCEDURE PANTALLA
SETCOLOR("N/W,,,")
dispbox(0,0,24,79," ° °°",)
@ 0, 0 SAY Space(80)
@ 2,26 SAY" PARTIDAS PRESUPUESTALES " COLOR
"B/W"
A_DispBox( 3,1,22,78,"W",.T. )
@ 0, 2 SAY " Itas ajas odificaciones " COLOR "N/W"
@ 0, 3 SAY "A" COLOR "R/W"
@ 0,10 SAY "B" COLOR "R/W"
@ 0,17 SAY "M" COLOR "R/W"
RETURN

STATIC FUNCTION PART_ELE()
SETCOLOR("W+/B")
@ 11,17 SAY "Partida:"
@ 13,17 SAY "Descripción Corta:"
@ 15,17 SAY "Descripción Larga:"
Return( Nil )

STATIC FUNCTION TECLAS33(O,T)
//
i:= 1
cClave:=Space(0)
Do Case
  Case T = K_DOWN
    O:Down()
  Case T = K_UP
    O:Up()
  Case T = K_PGDN
    O:PageDown()
  Case T = K_PGUP
    O:PageUp()
  Case T = K_CTRL_HOME
    O:GoTop()
  Case T = K_CTRL_END
    O:GoBottom()
  Case T = K_RIGHT .OR. T = K_TAB
    O:Right()
  Case T = K_LEFT .OR. T = K_SH_TAB
    O:Left()
  Case T = K_HOME
    O:Home()
  Case T = K_END
    O:End()
EndCase
O:RefreshCurrent()

```

