



UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO

FACULTAD DE INGENIERIA

SISTEMA DE ACTIVO FIJO

T E S I S
QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
PRESENTA :
EDGARDO TAPIA RANGEL



DIRECTOR: ING. ADRIAN ERIK PEREZ VARGAS
CODIRECTOR: M.I. ADOLFO MILLAN NAJERA

MEXICO, D. F.

1999

TESIS CON FALLA DE ORIGEN

27 15 17



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

INTRODUCCION

La elaboración de este trabajo de tesis, obedece a las circunstancias que se presentan en el Instituto Mexicano del Petróleo (IMP, instituto). En el instituto por mucho tiempo se ha venido trabajando en la parte administrativa; específicamente en el área de inventarios, con sistemas y actividades desconectadas en muchos aspectos. Lo anterior aunado al crecimiento que presenta el instituto, obligan a buscar nuevas soluciones más benéficas. Una de esas soluciones es la que se aborda en el presente trabajo.

El trabajo consta de 4 capítulos en los que se plantea y se desarrolla la solución al problema. En el primer capítulo, se explican algunos fundamentos necesarios para la comprensión del tema.

El segundo capítulo, llamado análisis, mostrará cuál es la problemática dentro del instituto y plantea los nuevos requerimientos de los usuarios, así como las especificaciones necesarias en cada actividad realizada.

El tercer capítulo, correspondiente al diseño, describe las bases informáticas que se crearán para darle solución al problema, esto es, toda la estructura interna que resuelve el problema, pero que el usuario final nunca ve.

El último capítulo, o implementación, llevará a cabo la ejecución de la solución propuesta, así como las pruebas necesarias para medir el desempeño y eficacia del sistema, y las correcciones que pudieran surgir al momento de las pruebas.

Al final de este trabajo se pretende haber logrado el análisis, diseño y desarrollo de un sistema institucional, para los Activos Fijos del Instituto Mexicano del Petróleo. Al mismo tiempo, unificar los Sistemas Informáticos del instituto, todo esto, con el fin de dejar una plataforma de datos adecuada y un mejor manejo de la información que se utiliza en el área Administrativa y Contable del IMP.

Indice

Capítulo 1	Fundamentos	1
1.1	Bases de Datos	2
1.1.1	Componentes de un Sistema de Base de Datos	3
1.1.1.1	Datos	3
1.1.1.2	Hardware	3
1.1.1.3	Software	3
1.1.1.4	Usuarios	4
1.1.2	Características	5
1.1.2.1	Independencia de Datos	5
1.1.2.2	Términos	6
	Tabla	6
	Registro	6
	Campo	6
	Llave Primaria	6
	Llave Foránea	6
	Vista	6
	Diccionario de datos	7
	Indice	7
	Administrador de la base de datos	7
1.1.3	Arquitectura de un sistema de base de datos	7
1.1.4	Enfoques para Bases de Datos	9
1.1.4.1	Enfoque Relacional	9
1.1.4.2	Enfoque Jerárquico	9
1.1.4.3	Enfoque de Redes	10
1.2	Programación	10
1.2.1	Panorama General	10
1.2.2	Clasificación	13
1.2.2.1	Lenguajes declarativos	14
1.2.2.2	Lenguajes de alto nivel	14
1.2.2.3	Lenguaje ensamblador y lenguaje máquina	14
1.2.2.4	Lenguajes para generar aplicaciones científicas	15
1.2.2.5	Lenguajes para generar aplicaciones de proceso de datos	16
1.2.2.6	Lenguajes para generar aplicaciones en Inteligencia Artificial	16
1.2.2.7	Lenguajes para generar aplicaciones de sistemas	17
1.2.3	Evolución	17
1.2.4	Programación Orientada a Objetos	19
1.2.5	Programación Orientada a Eventos	20
Capítulo 2	Análisis	22
2.1	Problemática y Aproximación a su Resolución	23
2.2	Requisitos	25
2.2.1	Contemplar los Movimientos más Importantes y Comunes de los Bienes	26

2.2.2	Fácil Manejo	26
2.2.3	Agilizar las Operaciones en el Inventario	26
2.2.4	Realizar Procesos Contables a Escala Institucional y de Forma Rápida	27
2.2.5	Automatizar Procesos	27
2.2.6	Reducir al Máximo el Papeleo Excesivo	27
2.2.7	Estandarizar los Catálogos	27
2.2.8	Homogeneizar los Datos	28
2.2.9	Homogeneizar Esquemas	28
2.2.10	Centralizar la Información	28
2.2.11	Migrar la Información Histórica	28
2.2.12	Seguridad en los Datos	28
2.3	Especificaciones	29
2.3.1	Operaciones	29
2.3.1.1	Notas de Entrada	29
	Altas	29
	Bajas	30
	Cambios y Consultas	30
2.3.1.2	Registro de Bienes	31
	Altas	31
	Bajas	32
	Cambios	33
	Consultas	33
2.3.1.3	Resguardos	33
	Altas	33
	Bajas	34
	Consultas	34
2.3.1.4	Notas de Traspaso	34
	Altas	34
	Cambios	35
2.3.1.5	Catálogos	35
2.3.1.6	Proceso Depreciación	36
2.3.1.7	Proceso Póliza Contable	36
2.3.1.8	Seguridad	37
2.3.2	Especificaciones Informáticas	37
2.3.2.1	Especificaciones de Software	37
2.3.2.2	Especificaciones de Hardware	38
2.4	Diagramas	39
2.5	Revisión	44
Capítulo 3	Diseño	45
3.1	Modular	46
3.2	Diseño de los Datos	48
	Catálogos	50
	Registro de Bienes	53
	Notas de Traspaso	55

Notas de Entrada	56
Resguardos	59
Seguridad	60
3.3 Diseño Arquitectónico	62
Catálogos	64
Notas de Entrada	66
Registro de Bienes	68
Resguardos	74
Gestión	76
Procesos	77
Utilerías	77
3.4 Procedimental	78
Capítulo 4 Implementación	104
4.1 Lenguaje	105
4.2 Algoritmos	105
Inicio	106
Notas de Entrada	108
Registro de Bienes	113
Resguardo	127
Gestión	129
Procesos	132
Utilerías	133
4.3 Pruebas y Correcciones	134
Conclusiones	137
Bibliografía	139

CAPITULO 1

1 Fundamentos

- 1.1 Bases de Datos**
- 1.2 Programación**

1.1 Bases de Datos

En la vida cotidiana de una empresa siempre ha sido de importancia el almacenar los registros de actividades realizadas o de información específica. Por ejemplo, en una tienda, se tiene el inventario de cosas que están en bodegas, que salieron o que llegan, todo esto es sencillo cuando se tienen pocos volúmenes de trabajo, el problema surge cuando el crecimiento de los datos es muy rápido y no es tan sencillo el control; para este tipo de situaciones se han creado diversas alternativas administrativas que agilizan los procesos. Otro tipo de alternativas la presenta la rama informática, en este caso tenemos archivos de texto, hojas de cálculo y bases de datos, estas últimas son el tema que abordaremos con más detalle.

La base de datos se define como una colección de datos relacionados y almacenados en conjunto, sin redundancias perjudiciales o innecesarias, su finalidad es la de servir a uno o más usuarios de la mejor manera posible, los datos se almacenan de modo que resulten independientes de aplicaciones que los usan, se emplean métodos bien determinados para incluir datos nuevos y para modificar o extraer los almacenados. La base de datos es la colección ordenada de datos, tablas, índices, diccionarios, vistas, etc. La figura 1.1 muestra un enfoque global de las bases de datos.

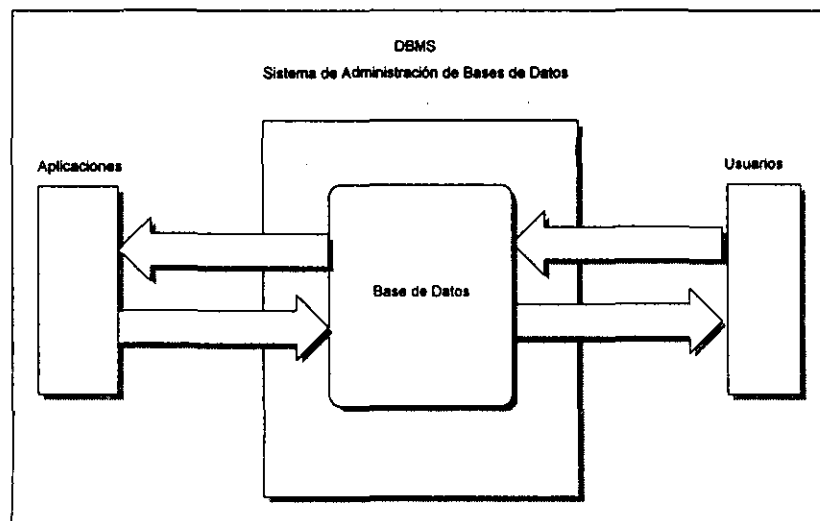


Fig 1.1 Esquema global de flujo en una base de datos

1.1.1 Componentes de un Sistema de Base de Datos

La tecnología de base de datos llegó a remplazar sistemas de archivos a mediados de los 60's. Una base de datos está integrada por cuatro elementos básicos, éstos son el hardware, software, usuarios y datos.

1.1.1.1 Datos

Los datos son lo más importante, son todos los registros e información que el usuario quiere mantener. El éxito de una base de datos lo marcará el manejo de los datos con poco y mucho volumen. Los datos llegan a tener relaciones muy estrechas y complejas, esto implica que para poder acceder la información de manera rápida y oportuna, la base de datos debe ser capaz de consultar óptimamente la información.

1.1.1.2 Hardware

En lo correspondiente a este rubro, los medios secundarios de almacenamiento dentro de una computadora son los importantes, en conjunto a ellos, se manejan también los sistemas físicos de conexión como las redes para comunicar computadoras, las unidades de control de dispositivos y los servidores como equipos de procesamiento de datos.

1.1.1.3 Software

El software es un nivel en la base de datos que se ubica entre el almacenamiento físico y el usuario, es decir, es la interfaz entre los datos y el usuario, a este software se le conoce con el nombre de Sistema de Administración de Bases de Datos (DBMS Data Base Management System). El DBMS es el encargado de manejar todas las solicitudes de acceso a la información y a su vez, de restringir la entrada a aquellas personas que no tengan permiso de trabajar los datos, el DBMS aplica la seguridad en cada elemento que existe en la base de datos, así como su mantenimiento preventivo y correctivo, el DBMS tiene las herramientas que un usuario necesita para mantener la integridad de la base de datos.

Un DBMS es una colección de numerosas rutinas de software relacionadas, cada una de las cuales es responsable de alguna tarea específica. Las funciones principales de un DBMS son:

- Creación, organización, almacenamiento, recuperación, independencia, y privacidad en la base de datos.
- Establecer y mantener las trayectorias de acceso a la base de datos de tal manera que los datos en cualquier parte de la base de datos se pueden acceder rápidamente.
- Manejar los datos de acuerdo con las peticiones de los usuarios.
- Mantener la integridad y seguridad de los datos.
- Registrar el uso de la base de datos.

El DBMS interpreta y procesa las peticiones del usuario para recobrar información de la base, estas peticiones de I/O del usuario, las manda al sistema operativo para la transferencia de datos entre la unidad de memoria secundaria y la memoria principal.

1.1.1.4 Usuarios

Los usuarios de la base de datos son las personas que manejan los datos para darles un sentido, en las bases de datos pueden intervenir tres clases de usuarios: los usuarios finales, los programadores de aplicaciones y el Administrador de la Base de Datos (DBA Data Base Administrator). El primer tipo de usuario es el que ve los datos por medio de una interfaz de fácil manejo y en la que no se requieran instrucciones muy técnicas para traer lo deseado.

El programador de aplicaciones es el creador de las interfaz finales que utilizará el usuario final, la diferencia entre los usuarios finales y los programadores de aplicaciones es que los segundos utilizan lenguajes especiales para lograr obtener de la base de datos los resultados que el usuario final requiere.

Finalmente el DBA es el encargado de diseñar, crear y administrar la base de datos por medio de aplicar las herramientas del DBMS para contar siempre con una base de datos confiable y protegida.

1.1.2 Características

Ya que hemos entendido que la base de datos es una propuesta informática para los problemas de almacenamiento de datos, la principal razón por la cual usaríamos una base de datos es por la centralización de la información en un solo lugar y porque además puede presentar las siguientes ventajas:

- Disminuir la redundancia
- Evitar la inconsistencia
- Cumplir las normas establecidas
- Aplicar restricciones de seguridad
- Conservar la integridad
- Equilibrar los requerimientos contradictorios
- Los datos pueden compartirse

1.1.2.1 Independencia de Datos

Otro concepto importante que manejan las bases de datos es el de independencia de datos, el significado de esto es que existirá inmunidad de las aplicaciones a los cambios de la estrategia de almacenamiento y de la estrategia de acceso, es decir, que al realizar una aplicación en determinado lenguaje y manejando una base de datos específica, si se tiene, por alguna causa, que modificar la forma de almacenar o acceder los datos, a la aplicación no se tendrá que modificarle el código, en contraparte, al trabajar directamente la forma de almacenar datos con el código de la aplicación, si es requerida una nueva forma de estructurar los datos, se implicaría una modificación en el código de la aplicación.

1.1.2.2 Términos

La base de datos permite no sólo la lectura de los datos almacenados, sino la continua modificación de los que son necesarios para el control de las operaciones. Algunos términos importantes para adentrarse en el conocimiento de las bases de datos son:

Tabla: Una tabla es una colección de datos en un disco al que se tiene acceso por un nombre singular. Suele contener una secuencia de registros de formato idéntico, cada uno conteniendo una serie de campos.

Registro: Un registro es un grupo de campos afines de información considerados como una unidad dentro de una tabla. Un registro es una fila en la tabla.

Campo: Un campo identifica una posición en un registro, en donde un dato elemental es almacenado. Este campo tiene características como largo y tipo (una cadena de caracteres de longitud fija, tipo numérico, etc.) y se identifican también como una columna en la tabla.

Llave Primaria: La llave primaria dentro de una tabla es la composición de uno o más campos que en su combinación nunca se repetirán, también estará asociado a la llave primaria un índice.

Llave Foránea: Una llave foránea es un campo que existe en una tabla y que hace referencia a una llave primaria con las mismas características, provocando que no se escriban datos diferentes a los que contempla la llave primaria referenciada.

Vista: Una vista se puede considerar como una tabla virtual, es decir, una tabla que no existe en realidad por derecho propio, sino que se deriva de una o más tablas. En otras palabras, no hay un archivo almacenado que de modo directo represente la vista por sí misma.

Diccionario de datos: El diccionario de datos es una descripción completa de los campos en una base de datos. Este diccionario describe las relaciones entre los diversos campos en una base de datos, también describe cada campo por nombre, extensión, tipo de datos, si se trata de una llave primaria o foránea y la interpretación del nombre del campo. Este puede contener datos adicionales como el rango de valores, índices asociados, etc.

Índice: Un índice es una tabla de números de registros. Estos números están dispuestos para ayudar a encontrar rápidamente un registro particular mediante una clave interna. El índice permite la recuperación de registros en orden de secuencia y permite la inserción de nuevos registros.

Administrador de la base de datos (DBA Data Base Administrador): Es la persona responsable del diseño físico de la base de datos, además de la evaluación, selección e implementación del DBMS. El DBA es el encargado de aplicar las herramientas del DBMS para contar siempre con una base de datos confiable y protegida.

1.1.3 Arquitectura de un sistema de base de datos

El esbozar la arquitectura de un sistema de bases de datos es muy útil para describir conceptos generales de las bases de datos y para explicar la estructura de los sistemas individuales; con esto no se pretende que los sistemas correspondan fielmente a la arquitectura que se plantea, sin embargo, la arquitectura es muy recurrida en el análisis de sistemas.

La arquitectura de un sistema de bases de datos consta de tres niveles que son:

- Nivel Interno. Es el más cercano al almacenamiento físico. Se refiere a la forma de almacenamiento de los datos.
- Nivel Externo. Es el más cercano al usuario, se refiere a la forma en que el usuario final ve los datos.

- Nivel Conceptual. Es un nivel de dirección indefinida entre los otros dos niveles, se refiere a la forma en que los diseñadores de sistemas ven los datos. Trata de expresar como son los datos realmente.

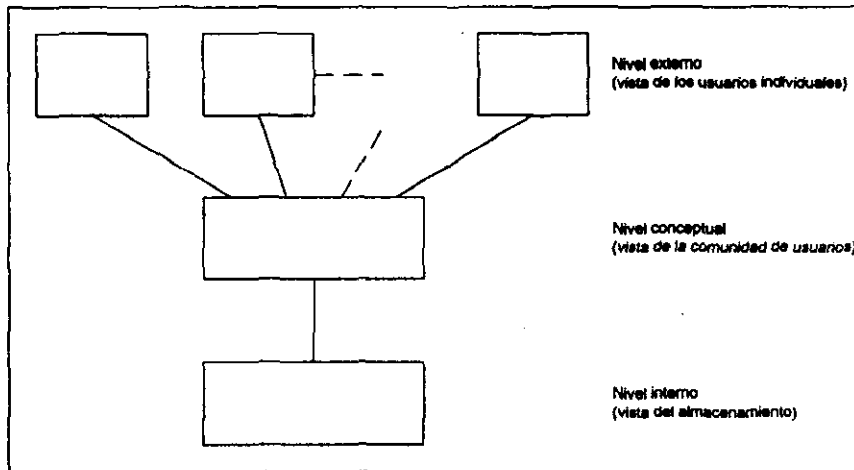


Fig. 1.2 Los tres niveles de la arquitectura

La figura 1.2 presenta la arquitectura básica de un sistema de bases de datos y como están ligados entre los niveles, una explicación más clara la encontramos en la siguiente figura 1.3 que nos presenta los equivalentes reales que se usarían en cada nivel.

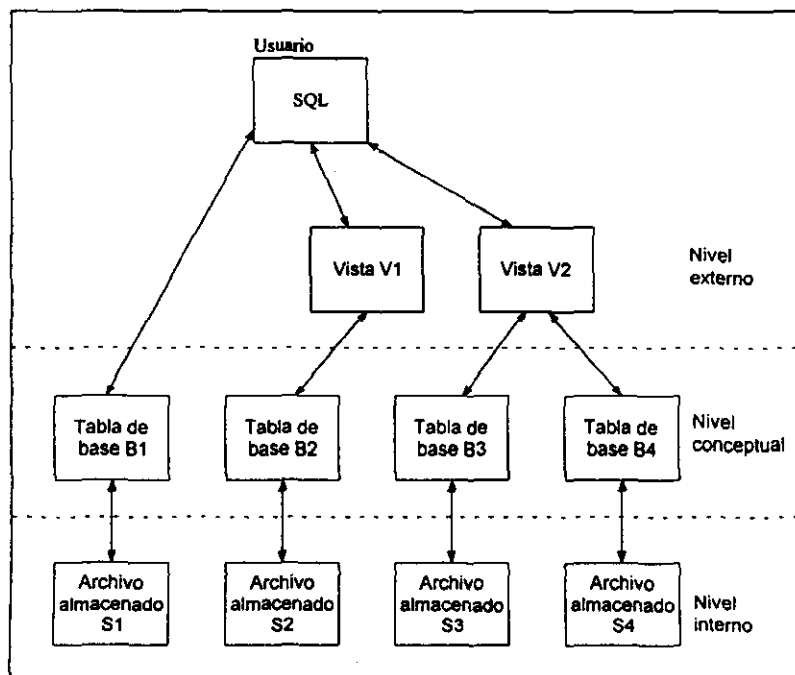


Fig 1.3 Equivalentes reales de cada nivel

1.1.4 Enfoques para Bases de Datos

Un problema que se presenta es el ver a que debería parecerse la base de datos para el usuario, es normal presentarle al usuario una vista de los datos en la que deliberadamente se omiten los detalles sobre la forma en que están representados esos datos en el almacenamiento, esta vista es la externa. El rango de estructuras de datos soportadas al nivel del usuario es un factor que afecta de manera decisiva a muchos componentes del sistema. En particular, impone el diseño del lenguaje de manipulación de datos correspondiente, porque cada operación debe definirse en términos de su efecto sobre esas estructuras de datos. Por eso la pregunta de, ¿qué estructura de datos y operaciones debe soportar el sistema?, es crucial. Es conveniente clasificar a los sistemas de bases de datos de acuerdo con el enfoque que adoptan para dar respuesta a esta pregunta.

Existen 3 enfoques principales para el manejo de bases de datos:

- Enfoque relacional
- Enfoque jerárquico
- Enfoque de redes

1.1.4.1 Enfoque Relacional

El enfoque relacional ve a los datos como un conjunto de tablas(archivos) formadas de un conjunto de renglones (registros) y cada renglón constituida de un conjunto de columnas (campos). La relación entre las tablas se establece sólo por nombres de atributos comunes.

1.1.4.2 Enfoque Jerárquico

Permite ver una base de datos como una relación entre padre e hijo, en la que no puede haber hijos sin padre.

1.1.4.3 Enfoque de Redes

Modela la información como conjuntos, donde hay dos tipos de conjuntos: propietarios (Equivale a un padre en el enfoque jerárquico) y miembros (Equivale a un hijo dentro del enfoque jerárquico).

En el enfoque de redes la redundancia se reduce al máximo, no se pierde información. Es el enfoque más eficiente para representar relaciones N a M, sin embargo, tiene la desventaja de ser muy complejo.

De los tres enfoques el más usado y el que tiene bases más formales es el enfoque relacional, motivo por el cual casi el 100% de los manejadores son construidos con este enfoque.

1.2 Programación

1.2.1 Panorama General

En la evolución de la tecnología informática, tenemos dos grandes rubros, uno es el que compete a toda la parte física y el otro a la parte lógica, estamos hablando del hardware y software respectivamente. Estas ramas han crecido, pero el hardware lo ha hecho más rápidamente que el software. En este trabajo, por hardware estaremos refiriendo equipos personales o PC's. En principio vamos a decir que la relación que guardan el software con el hardware es que el primero se encarga de explotar adecuadamente los recursos del segundo, esto nos lleva a que el software es el medio por el cual las personas (usuarios) pueden lograr que el hardware realice las tareas deseadas.

En la creación de los lenguajes se ha buscado que la forma de expresión entre las máquinas y los usuarios sea más fácil y mejor organizada. Un lenguaje de programación es una herramienta para producir programas y su valor se debe juzgar según el modo en que afecta a la producción del software y dependiendo de la facilidad con que se puede integrar

con otras herramientas. Existen cientos de lenguajes diferentes y muchos de ellos son dialectos de otros. La historia nos muestra como los lenguajes han ido evolucionando por familias, influidos por los cambios en el diseño de las arquitecturas de las computadoras y por las fallas de diseño cometidos en el pasado.

La mayoría de los lenguajes han sido diseñados para resolver un determinado tipo de problema, pero con el tiempo muchos de ellos han ido evolucionando y ampliándose hasta convertirse en lenguajes de propósito general.

La arquitectura de Von Neumann (1946), basada de alguna forma en el diseño de la máquina de Turing (1936), ha sido el modelo básico de arquitectura durante mucho tiempo y ha influido enormemente en el diseño de los primeros lenguajes (programación imperativa), así como en la programación de sistemas.

Los lenguajes, a través del tiempo, han sido motivados para su evolución, en los primeros años de la Informática los programas se realizaban casi exclusivamente para uso personal. El proceso de desarrollo de programas consistía únicamente en la fase de codificación. La PC sólo se utilizaba en aplicaciones científicas y una aplicación era desarrollada por una sola persona, normalmente la misma que después utilizaba el programa. El problema a resolver era muy preciso, estaba bien definido y era bien conocido por el programador. Por tanto, un lenguaje de programación era utilizado principalmente por un solo programador que realizaba aplicaciones que hoy consideraríamos muy sencillas.

El deseo y la conveniencia de utilizar la PC para otro tipo de aplicaciones hizo que se comenzara a utilizar en entornos más sofisticados. Los programadores empezaron a trabajar en grupo, y los resultados debían ser comunicados de una persona a otra. Ya que se gastaba tanto dinero y esfuerzo en el desarrollo del software, no debía desecharse el trabajo antiguo cuando fuera necesario uno nuevo, era conveniente reutilizarlo, al menos en parte, era preferible mejorar un sistema ya existente para adaptarlo a las nuevas necesidades, que

construir uno totalmente nuevo. De esta forma, el mantenimiento de los programas se convirtió también en una cuestión a tener en cuenta.

La fiabilidad de un sistema es una cuestión que ha ido también ganando en importancia, por dos factores fundamentales. Por una parte, las aplicaciones se diseñan para ser utilizadas por usuarios que no tienen una gran formación informática y, por tanto, estos usuarios no son tan tolerantes a las fallas del sistema como los programadores que los han realizado. Por otra parte, los sistemas se utilizan en áreas como centrales nucleares, monitorización de pacientes, etc., donde una falla puede tener consecuencias muy graves.

En la década de los 60's, conforme los sistemas de cómputo se multiplican, se hacen más complejos y penetran con mayor profundidad en la sociedad moderna, se evidencia la necesidad de enfoques sistemáticos para el desarrollo del software, así como para su mantenimiento. Durante esta década, aparecieron las computadoras de la tercera generación y se desarrollaron técnicas como la multiprogramación y el tiempo compartido, estas nuevas capacidades aportaron la tecnología necesaria para el establecimiento de sistemas interactivos, multiusuarios, en tiempo real, etc., y así surgieron nuevas aplicaciones para la computación como las reservaciones en líneas aéreas, bancos de información médica, control de procesos, control y dirección de equipos militares, etc.

Aunque la mayor parte de estos primeros y grandes sistemas fueron llevados a cabo correctamente, algunos de ellos nunca llegaron a terminarse, o sufrieron de un gran sobrecosto, una terminación tardía, ineficacia, poco confiable, y sobre todo de poca aceptación por parte de los usuarios.

Los primeros programadores consideraban que la base de la programación era la experiencia. Cuando los programas se complican y resulta muy difícil depurarlos y modificarlos, los programadores empiezan a pensar en una metodología sistemática de programación, esto hace que la evolución en los lenguajes de programación se haya visto muy marcada por un uso cada vez mayor de la abstracción, tanto de datos como de estructuras de control. En cuanto a la abstracción de las estructuras de control podemos

mencionar el siguiente ejemplo: en los lenguajes más antiguos la única forma de programar ciclos de repetición era utilizando una instrucción de salto (GOTO) combinada con una instrucción condicional (IF), luego un momento en que los lenguajes que iban apareciendo desaconsejaban o incluso prohibían el uso del GOTO para adaptarse a la filosofía de la programación estructurada, incorporando estructuras de control más sofisticadas y más abstractas. La programación estructurada, y después la programación modular, han influido decisivamente en el diseño de los lenguajes de programación.

De la misma forma también se ha producido un aumento en la abstracción de los datos, debido al uso de datos organizados de forma cada vez más compleja. Si al principio sólo se usaban estructuras del tipo arreglo y registro, con el tiempo aparecieron los tipos abstractos de datos (1972-74), en los que lo importante era el comportamiento de los datos y no su codificación interna.

Más tarde aparece la programación orientada a objetos (nace en 1968, pero hasta 1983 toma fuerza), que abstrae aún más el concepto de dato haciendo que los objetos puedan heredar propiedades unos de otros, e incluso comportarse de diferente forma según convenga (polimorfismo).

Las deficiencias de los primeros lenguajes en estas áreas, han dirigido una gran cantidad de esfuerzo en el campo del diseño de nuevos lenguajes. Los lenguajes han evolucionado para proporcionar al programador características que apoyen el desarrollo y mantenimiento de los productos software. Por ejemplo, verificación robusta de tipos, compilación por separado, tipos de datos definidos por el usuario, encapsulación y abstracción de datos, manejo de excepciones, mecanismos de concurrencia, etc.

1.2.2 Clasificación

Para la clasificación de los lenguajes existen dos formas diferentes de hacerlo: por su nivel y por el tipo de aplicaciones para las que ha sido diseñado.

En la clasificación por niveles se tienen en cuenta las facilidades que el lenguaje ofrece al programador que lo va a utilizar y es un hecho que las facilidades para el programador decrecen al disminuir el nivel. En este sentido podemos encontrar los siguientes lenguajes.

1.2.2.1 Lenguajes declarativos

Son lo que más se parecen al lenguaje natural (Inglés) por su poder expresivo y por su funcionalidad. Fundamentalmente son lenguajes de comandos, en los que abundan instrucciones que expresan "qué hay que hacer", en lugar de "cómo hay que hacerlo". Hay que incluir en este grupo, lenguajes estadísticos, y lenguajes para gestionar bases de datos. Estos lenguajes han sido desarrollados como herramientas profesionales fáciles de asimilar y utilizar.

1.2.2.2 Lenguajes de alto nivel

Es el grupo más amplio de lenguajes de programación. Aunque no son declarativos, permiten escribir programas fáciles de entender por otros programadores. La mayoría son lenguajes que se caracterizan por su portabilidad, es decir, un programa puede ser utilizado en diferentes máquinas sin cambios de importancia.

1.2.2.3 Lenguajes ensamblador y lenguajes máquina

Estos dos últimos grupos de lenguajes son dependientes de la máquina; es decir, cada máquina tiene su propio lenguaje máquina y un lenguaje ensamblador asociado a él. En general, el lenguaje ensamblador es una representación simbólica de su lenguaje máquina asociado. Cuanto más bajo es el nivel de un lenguaje, más cerca se encuentra de la forma de trabajar de la máquina y más difícil es de comprender por el hombre. Existe una correspondencia de 1 a 1 entre las instrucciones ensamblador y las instrucciones máquina.

La diferencia principal entre ellos está en el uso de símbolos (A para sumar, M para multiplicar, X, Y, W, Z para las variables) por parte del lenguaje ensamblador.

La programación en un lenguaje ensamblador, de alto nivel o declarativo, hace necesaria la intervención de algún traductor al lenguaje máquina (que es el único que entiende directamente la máquina) para que los programas puedan ejecutarse. Para realizar esta tarea utilizamos ensambladores, compiladores e intérpretes. Los ensambladores y compiladores realizan un paso de traducción previo al paso de ejecución de programa, sin embargo, el intérprete, ejecuta directamente cada instrucción escrita en un lenguaje de alto nivel, en general se considera que el proceso de compilación es más eficiente que el del intérprete, por ejemplo, un intérprete debe traducir varias veces la misma instrucción si está incluida en un ciclo de repetición.

Otra forma de clasificar los lenguajes se puede llevar a cabo teniendo en cuenta el tipo de aplicaciones para las que los lenguajes han sido especialmente diseñados. A veces las fronteras entre estos grupos de aplicaciones están muy difusas, pero tienen gran importancia desde el punto de vista histórico, ya que la mayoría de los lenguajes de alto nivel nacieron para resolver problemas dentro de alguno de los campos que estudiaremos a continuación, aunque con el tiempo la mayoría de ellos hayan ido ampliándose hasta convertirse en lenguajes de propósito general.

1.2.2.4 Lenguajes para generar aplicaciones científicas

Se caracterizan porque predominan las operaciones de manipulación de números y arreglos de números. La base de los algoritmos utilizados con estos lenguajes se basa en principios matemáticos y estadísticos. Al principio estos problemas utilizaban una cantidad pequeña de datos, estructurados de forma relativamente sencilla, aunque actualmente algunos científicos resuelven problemas que manipulan grandes cantidades de información (por ejemplo, la información recibida por un satélite) que a veces debe ser reducida para poder ser procesada de forma eficaz.

Una característica de estas aplicaciones es la gran complejidad matemática de los algoritmos utilizados. Los programadores deben tener amplios conocimientos matemáticos para poder depurar sus programas y para saber cómo deben ser los datos de entrada para estos programas. En estos programas el tiempo de computación utilizado suele ser, en proporción, mayor que el tiempo utilizado en operaciones de entrada y salida. Todas estas características hacen que estas aplicaciones difieran de las demás.

1.2.2.5 Lenguajes para generar aplicaciones de proceso de datos

Se caracterizan porque en ellos predomina el interés por la creación, mantenimiento, borrado y consulta de datos, en ficheros y bases de datos. Estas aplicaciones suelen tener gran importancia en la organización de una empresa y permiten controlar nóminas, transacciones comerciales, contabilidad, clientes, proveedores, almacenes, etc. Normalmente el volumen de datos a tratar suele ser alto y las operaciones aritméticas que se realizan con estos datos suelen ser muy sencillas, por esto el tiempo que estos programas dedican a las operaciones de entrada/salida suele ser mayor que el dedicado por el procesador a la computación. Estas aplicaciones deben además cuidar la integridad de los datos y su seguridad, es decir, a determinados datos sólo pueden tener acceso determinadas personas.

1.2.2.6 Lenguajes para generar aplicaciones en Inteligencia Artificial

Estos programas han sido diseñados para simular el comportamiento de la inteligencia humana. Podemos incluir en este grupo: juegos, programas que entienden el lenguaje natural, visión artificial, robótica, sistemas expertos, reconocimiento de patrones, etc.

1.2.2.7 Lenguajes para generar aplicaciones de sistemas

Estos programas actúan de intermediarios entre la PC (hardware) y el programador o el usuario del sistema, por ejemplo, programas traductores, rutinas de entrada/salida, programas de utilidades de sistema, sistemas operativos, etc. Hay dos características que distinguen a estos programas de los demás, la primera es la necesidad de coordinar actividades entre varias tareas o programas, y el control de excepciones (errores de I/O). Tradicionalmente la programación de sistemas se realizaba en lenguaje ensamblador, pero con el tiempo se diseñaron lenguajes de alto nivel como C que sirve para este propósito y que eran mucho más fáciles de utilizar por el programador.

1.2.3 Evolución

Hay dos aspectos que debemos considerar para entender la evolución de los lenguajes de programación en los últimos años: la programación orientada a objetos y el paralelismo.

En primer lugar, hay que tener en cuenta que las metodologías de programación abogan por la construcción de programas legibles y fáciles de modificar; por eso, cuando de forma lógica aumentan la complejidad y el tamaño de los programas, estas metodologías apuntan hacia un aumento en la abstracción, llegándose así a la programación orientada a objetos.

Repasando brevemente la evolución en cuanto a la abstracción de datos, encontramos que los lenguajes de nivel máquina ven los datos simplemente como cadenas de bits. Un primitivo paso se dio al incluir los tipos de datos, aquí la información ya no se ve como una secuencia de bits sino como un valor entero, real o lógico. La siguiente generación permitió al programador usar tipos de datos incorporados y le ofrece formas para construir nuevos tipos (matrices, registros, etc.). Esto mejora considerablemente la legibilidad de los programas ya que agrupa las entidades que están envueltas en una determinada abstracción y así llegamos al concepto de tipo abstracto de datos que serán tipos definidos por el usuario que satisfagan dos propiedades:

- a) la representación de los datos con operaciones asociadas en una unidad adecuada del lenguaje.
- b) el ocultamiento de la representación del nuevo tipo a las unidades que lo usan.

El concepto de tipo abstracto de datos se deriva de un principio más general: el ocultamiento de la información. Los tipos abstractos de datos esconden los detalles de la representación y encaminan los accesos a los objetos abstractos a través de los procedimientos, enfatizando así en la distinción entre los diferentes niveles de abstracción. Finalmente llegamos a la Programación Orientada a Objetos, el concepto de objeto se parece al de tipo abstracto de datos pero teniendo en cuenta características como la herencia, el polimorfismo, etc.

Por otro lado al analizar la evolución en las arquitecturas de hardware, vemos como éste es cada vez más potente y con tarjetas cada vez más baratas, con varios procesadores, lo que permite la ejecución de programas paralelos, pero para construirlos hacen falta lenguajes de programación diseñados de tal forma que permitan una programación paralela.

Parece lógico pensar, a la hora de diseñar nuevos lenguajes, que hay que integrar los dos paradigmas creando lenguajes concurrentes orientados a objetos. Otro aspecto a tener en cuenta a la hora de diseñar lenguajes es el de permitir que el programador construya de forma sencilla, aplicaciones muy complejas y utilizando un entorno amigable, por esta razón a un grupo de nuevos lenguajes les llamamos visuales. Estos lenguajes ponen a disposición del programador todos los elementos que le permiten ir construyendo el esqueleto del programa, ésta es una programación orientada por windows que propicia la programación orientada a eventos. No existe un flujo secuencial de instrucciones, sino una colección de eventos pasivos (pueden ser botones) que reaccionan cuando se produce un evento (el clic del ratón, entrada de datos, llegada de un mensaje por la red).

1.2.4 Programación Orientada a Objetos

El enfoque que se presenta de la programación orientada a objetos, es el manejado en SQLWindows. Para comenzar la introducción al tema definimos el elemento básico que debe existir, este elemento es una clase, una clase es un molde el cual define las propiedades y comportamiento de un objeto, sin ser el objeto en sí. Al momento de crearse un objeto es cuando se toma la clase o molde, en ese momento se dice que se esta creando una instancia de la clase. Para entender con más claridad el concepto, tomemos una cadena de restaurantes, la clase es el conjunto de características que identifican a cierta cadena de restaurantes: su nombre, logotipo, menú, arquitectura, etc. Y las instancias son las diferentes sucursales de la cadena de restaurantes.

En el caso de SQLWindows, las clases son modelos de objetos utilizados en SQLWindows como Form Windows, Dialog Box, Data Field, Combo Box, etc.

Las clases nos permiten que el código generado sea 100% reutilizable, ya que el código que resuelve una problemática, se coloca en una clase y automáticamente todas las instancias de esa clase también resuelven el problema.

Encapsulamiento. Existen varios principios para comprender y utilizar correctamente los objetos, el primero es denominado encapsulamiento de datos, éste consiste en que un objeto puede modificar alguno de sus atributos, pero no podrá cambiar los atributos de otro objeto. Retomando el ejemplo de la cadena de restaurantes, cada sucursal tiene su menú, y solo su menú es el que modifica.

Un concepto vinculado con el encapsulamiento es el de comunicación por mensajes, que es la forma en que un objeto solicita una acción de otro objeto, en el ejemplo tomado la dirección de la cadena de restaurantes sería otro objeto que en determinado momento decide que los precios de los menús deben incrementarse, pero la dirección general no modificará los precios de los menús de todas las sucursales, sino que enviará a cada sucursal una notificación para que realicen el aumento.

Herencia. La herencia es de las propiedades más importantes dentro de la programación orientada a objetos ya que permite combinar propiedades de dos o más clases, en una nueva clase. La herencia se basa en generar una nueva clase a partir de una o varias ya existentes, la nueva clase posee automáticamente la mismas propiedades y comportamientos de las clases que la generaron. Tomando un caso con dos clases, una llamada DiaHabil, que valida fechas como días hábiles y otra llamada FontColor que define el tipo de letra y color que tendrá la información al ser desplegada en pantalla. Si yo quiero un objeto con ambas cosas lo único que se requiere es crear una nueva clase FechaHabil y definir que sus antecesores son las clases DiaHabil y FontColor. La ventaja de manejar esto así, es que si en un futuro el algoritmo para validar que una fecha sea hábil cambia, sólo se necesita modificar el código de la clase DiaHabil y gracias a la herencia se tendrá la modificación en la clase FechaHabil.

Polimorfismo. El polimorfismo es la habilidad de que diferentes objetos respondan al mismo acontecimiento en diferente forma. Esta propiedad se utiliza con los objetos estándar de SQLWindows, veamos el caso de tener varios pushbottons, todos registrarán el acontecimiento de haber sido presionados (SAM_Click) pero uno de ellos se moverá al siguiente registro, otro retrocederá un registro y tal vez otro imprima la información, aunque todos están recibiendo el mismo acontecimiento, realizan diferentes tareas.

El polimorfismo facilita la independencia entre objetos, ya que cuando un objeto va a enviar un mensaje a otro objeto, no necesita conocer la naturaleza del receptor, simplemente le envía el mensaje y el objeto responderá en la forma apropiada.

1.2.5 Programación Orientada a Eventos

El ambiente gráfico provisto por Windows requiere de una programación diferente de la empleada normalmente, la secuencia de las acciones, las posibilidades del usuario y el estilo establecido por la misma interfaz, obligan al programador a cambiar su mentalidad al respecto, ahora se debe aplicar la programación por eventos.

La programación por eventos representa un cambio radical en la lógica de programación, ya no se puede conceptualizar el programa como un flujo de instrucciones, en su lugar debe considerarse que cada elemento de la interfaz, es un ente con identidad y función propia, responsable de actividades específicas, y que la colaboración entre estas entidades permitirá alcanzar el propósito de la aplicación.

Aunado a la programación por eventos, el ambiente de SQLWindows cuenta con características de Programación Orientada a Objetos, si bien puede prescindirse de ellas, utilizarlas proporciona grandes beneficios en cuanto al desarrollo y mantenimiento de la aplicación, pues se ahorra tiempo al generar código modular y reutilizable.

Entrando en materia, tenemos que un evento se define como la notificación de un acontecimiento o requerimiento que se suscita en el sistema o en un objeto, como ejemplos tenemos: el mover un ratón, cambiarle el tamaño a una ventana, tecla oprimida por un usuario, creación de una pantalla, llegada de información al puerto serial, etc., y como ejemplos de requerimientos tenemos: arrancar un programa, leer un archivo, cerrar una ventana, etc.

Los eventos son identificados por una constante numérica, Microsoft definió todos los mensajes que las aplicaciones pueden recibir en ambiente Windows listándolos y asignando un número único a cada uno de ellos.

El valor asignado al mensaje no tiene relevancia salvo que debe identificar un sólo evento, el único atributo del mensaje de importancia es su significado. Por ejemplo: WM_MOUSEMOVE es una notificación de que el usuario movió el ratón, si el valor del mensaje fuera otro número 300, 320, ó 450, no importaría porque sigue significando lo mismo.

CAPITULO 2

2 Análisis

- 2.1 Problemática y Aproximación a su Resolución
- 2.2 Requisitos
- 2.3 Especificaciones
- 2.4 Diagramas
- 2.5 Revisión

El análisis dentro de la Ingeniería de Software es la primera etapa que se abordará para la realización del Sistema de Activo Fijo, en esta etapa definimos todas las necesidades de los usuarios, algunas evaluaciones técnicas, restricciones, volumen de datos, funciones de cada elemento del sistema, desempeño, confiabilidad, número de usuarios, etc. El análisis aplicado está basado en la experiencia de los programadores que plantean primeramente la problemática existente, estableciendo los requisitos y especificaciones del usuario llegando finalmente a diagramas de flujo de la información y una revisión. Para esto, llevan a cabo reuniones de trabajo directamente con las áreas involucradas en el proyecto y con la información obtenida se realizan las estimaciones.

2.1 Problemática y Aproximación a su Resolución

Administrar una empresa es sinónimo de administrar recursos materiales, humanos y financieros. En una empresa, la administración es la encargada de tomar decisiones que van a permitir realizar sus objetivos de una manera eficiente, basándose en información veraz, confiable y oportuna. Los recursos materiales son aquellos bienes con los cuales cuenta la empresa para operar, entre ellos se encuentran el mobiliario, el equipo de oficina, la maquinaria y los vehículos entre otros, a este tipo de recursos o bienes materiales se les conoce también con el nombre de activos fijos.

Por mucho tiempo, dentro de la Empresa en el Area de Administración y Finanzas, se ha tenido el apoyo de recursos informáticos para facilitar los procesos sobre los bienes materiales o activos fijos existentes y de nueva adquisición, estos recursos informáticos son sistemas creados dentro de la misma institución, pero con el paso del tiempo, el desarrollo de nueva tecnología de hardware y software y el crecimiento de la institución misma, ha provocado que los sistemas sean obsoletos en el sentido de manejo y eficiencia para el procesamiento de información, aparte de que existen nuevos métodos administrativos y mayor volumen de datos, razón por la cual se genera duplicidad de información y funciones.

En el área de Contabilidad y de Inventarios se maneja un mismo sistema en donde encontramos una parte para cada área, los datos ahí almacenados son los básicos requeridos por ambas partes, el inventario contiene la información más relevante de los bienes tales como la descripción, marca, modelo, grupo, dependencia, número de inventario, etc., la contable contiene los datos de compra, factura, proveedor, fecha de factura, valor de adquisición, etc., cada área se encarga de llenar la información que le corresponde y al final los datos quedan aislados.

Los sistemas informáticos que actualmente se manejan están hechos con lenguajes de cómo Clipper y Dbase los cuales están limitados en su actualización provocando algunas incompatibilidades con las nuevas tecnologías, así como rigidez en su manejo.

Existen otras operaciones que se realizan sobre los bienes como son su depreciación, revaluación, asignarlos en resguardo o traspasarlos entre áreas o dependencias. El primero (depreciación) consiste en disminuir el valor del bien con el transcurso del tiempo por razón de su uso. La revaluación es actualizar el valor del bien de modo que muestre su valor real al tiempo presente. Asignar un bien en resguardo significa definir a que persona se le otorga la custodia de dicho bien, y traspasar los bienes entre áreas o dependencias se refiere a los cambios de ubicación física y de área o dependencia responsable de uno o más bienes. Estas operaciones no están contempladas en el sistema actual, en el caso de los resguardos y los trasposos se realizan sin ningún apoyo informático y para la revaluación y depreciación es necesario pasar por medio de un disco flexible los datos para ser procesados en otra área. El manejar la información de esta manera, ha venido generando muchos problemas de redundancia, control, confiabilidad, precisión y oportunidad, así como mantener las operaciones aisladas a pesar de la vinculación estrecha que existe entre ellas.

Con todo lo anterior, el problema consiste en erradicar la independencia de cada proceso y los errores generados en los mismos, retomar el control, disminuir el tiempo de procesamiento de información, suprimir la falta de confiabilidad y veracidad de los datos y desplazar las plataformas existentes de programación.

Partiendo de la problemática señalada en los párrafos anteriores, la solución propuesta es la integración en un solo sistema institucional los procesos más importantes en los cuales se ve involucrado un bien (activo fijo), centralizar la información, crear un ambiente amigable con el usuario, agilizar los procesos, robustecer el sistema, darle confiabilidad y veracidad, tener en línea la información, tomar el control, llevar a una nueva plataforma los sistemas existentes y optimizar el rendimiento hora-hombre.

2.2 Requisitos

La manera en que entraremos en el análisis a fondo de la problemática a resolver, será comenzando por conocer cuáles son las necesidades reales del usuario, al mismo tiempo que definiremos los conceptos, parámetros, operaciones, etc., que se manejan.

Una vez que las pláticas con los usuarios finales han comenzado, las necesidades generales más importantes que se detectan son las siguientes:

- considerar los movimientos más importantes y comunes de los bienes
- fácil manejo
- agilizar operaciones en el inventario
- realizar procesos contables a nivel institucional y de forma rápida
- automatizar procesos
- reducir al máximo el papeleo excesivo
- estandarizar los catálogos
- homogeneizar los datos
- homogeneizar esquemas
- centralizar la información
- migrar la información histórica
- seguridad en los datos

Una vez listadas las necesidades generales de los usuarios finales empezaremos a particularizar en cada una de éstas.

2.2.1 Contemplar los Movimientos más Importantes y Comunes de los Bienes

El usuario, al referir esto, nos pide que se integren las operaciones sobre las notas de entrada como son altas, bajas, cambios y consultas, así como las operaciones sobre los bienes, esto es altas, bajas, cambios, consulta de bienes, resguardos, consulta de resguardos, notas de traspaso entre áreas y dependencias, depreciación, revaluación y póliza contable.

2.2.2 Fácil Manejo

En muchas ocasiones el usuario final no tiene los conocimientos informáticos básicos en el manejo de las PC's, esto se refleja cuando quieren comenzar a manejar una herramienta informática, si existe experiencia no tendrá problemas, pero cuando se habla del uso de esta herramienta por primera vez, su aprendizaje será más rápido con un ambiente amigable que lo lleve de la mano sin la necesidad de instrucciones, a comparación de un ambiente en donde se requieren muchas instrucciones y capacitación especializada en el manejo del lenguaje de una PC. Por el fácil manejo hablamos de una interfaz gráfica amistosa con el usuario final.

2.2.3 Agilizar las Operaciones en el Inventario

En condiciones normales, la operación en el inventario requiere de mucha inversión de tiempo y papel, las personas en este sentido tienden en condiciones normales a retener mucho los documentos o los guardan hasta completar un número de necesidades, en este punto se busca que el tiempo invertido para llevar a cabo un movimiento u operación sea el óptimo y sin errores.

2.2.4 Realizar Procesos Contables a Escala Institucional y de Forma Rápida

En la problemática hemos planteado el hecho de tener procesos contables que tienen que realizarse en otras áreas, el requerimiento en este sentido es que no se tenga que mover la información en discos para ser procesada, sino que sea una opción dentro del sistema, esto es que la operación deberá ser en línea con las áreas que interactúan.

2.2.5 Automatizar Procesos

Dentro de los movimientos aplicables al inventario podemos encontrar algunos cálculos que pueden ser realizados directamente por el sistema, aquí es donde se quiere que se contemplen estos eventos para su integración, esto es, tener la mayoría de los posibles cálculos sobre un bien, implementados en el programa y listos para ser ejecutados.

2.2.6 Reducir al Máximo el Papeleo Excesivo

En la mayoría de los casos en los que se tiene que generar un reporte o lista, el consumo de papel es muy grande, por lo que se solicita que el sistema evite esto por medio de un fácil manejo de consultas sobre las características de cada bien y así sólo imprimir los reportes mínimos.

2.2.7 Estandarizar los Catálogos

Los catálogos que existen no tienen un estándar por lo que hay muchos con la misma información, para erradicar el problema el sistema aportará un catálogo único para cada tipo de datos que lo requieran y para todas las dependencias.

2.2.8 Homogeneizar los Datos

El requerimiento del usuario al mencionar el término homogeneizar, es el de tener una información uniforme y que sea introducida de igual forma en todas las dependencias basándose en los mismos parámetros de validación de datos.

2.2.9 Homogeneizar Esquemas

Por esquemas entendemos los formatos que se manejan para los trámites de los bienes, como son las notas de traspaso, notas de entrada y resguardos. Una vez integrados en el sistema, los usuarios no podrán presentar otros esquemas.

2.2.10 Centralizar la Información

La petición es contar con la información concentrada en un solo lugar (en una Base de Datos) y no que se tenga la información en cada dependencia, esto es desde el punto de vista del usuario; informáticamente esto se contempla con la nueva plataforma que maneja la filosofía cliente-servidor.

2.2.11 Migrar la Información Histórica

Se requiere que los datos que actualmente son trabajados, se pasen al nuevo sistema, manteniendo la normatividad de la empresa.

2.2.12 Seguridad en los Datos

La seguridad en los datos implicará que los usuarios no podrán realizar cualquier operación en los datos. Asimismo, el usuario pide que los datos almacenados tengan la mayor seguridad y confiabilidad en su integridad.

2.3 Especificaciones

Una vez planteada la definición de los requisitos generales del sistema, nos enfocaremos en las especificaciones que surgen de cada necesidad, no en todas las necesidades existe algo que especificar, así que sólo tomaremos las correspondientes.

2.3.1 Operaciones

2.3.1.1 Notas de Entrada

Altas

Las primeras operaciones que aparecen son en las notas de entrada. La nota de entrada es el documento contable donde queda avalado el bien como propiedad del Instituto Mexicano del Petróleo (IMP, empresa, instituto). Las notas de entrada son información única, esto quiere decir, que nunca se encontrarán dos notas de entrada con el mismo nombre. Las altas de la nota de entrada involucran otros movimientos, estos son los desgloses de la misma que de acuerdo al número de bienes que incluya serán los movimientos a realizar.

Los datos que se involucran en una alta para la nota de entrada son:

El identificador de la nota de entrada

La fecha en que se esta creando la nota de entrada

La requisición del bien

El identificador del pedido

Proyecto que lo requirió

El número de factura

La fecha de la factura

Proveedor

Su origen, nacional o importado

Nombre o número de la póliza contable

Póliza contable de diario o informativo

Partida a la que se incluiría contablemente

Dependencia solicitante

Importe

Impuesto al Valor Agregado (IVA)

Pacto de Estabilidad y Crecimiento Económico (PECE)

Persona que recibe

Fecha en que recibe

Observaciones

El consecutivo del bien dentro de la factura

El número de bienes idénticos dentro del pedido

Los que quedan pendientes dentro de una factura

Los bienes que se entregaron averiados o defectuosos

El precio unitario por cada bien

Gastos adicionales

Bajas

En las bajas de las notas de entrada hay que verificar que no se tengan bienes asignados a esa nota, si ese es el caso entonces se tiene que deshacer ese vínculo entre la nota y él o los bienes.

Cambios y Consultas

Los cambios y consultas de notas de entrada, especifican que para llegar a la nota de entrada deseada se haga por su identificador.

2.3.1.2 Registro de Bienes

Altas

En el registro de los bienes tenemos en primer lugar las altas, donde observamos varias etapas. Para que un bien se incorpore en la empresa, como paso inicial se requiere generar la nota de entrada con su desglose correspondiente, posteriormente se da el levantamiento en el área de almacén e inventario. Ha quedado definido que se tomará un número consecutivo e irrepetible para la identificación del bien, desplazando la forma actual de realizar el inventario, esto es, tomando la partida, determinante y un consecutivo. El área de almacén e inventario captura las características físicas del bien y queda especificado que los datos de identificador del bien, estado, partida presupuestal y determinante son obligatorios.

Los datos requeridos para dar de alta un bien en el área de almacén e inventario son:

- Número consecutivo que se le asigna al bien
- El estado de uso o propiedad en que se encuentra el bien
- El número de serie
- El modelo
- La marca
- La fecha en que entra al inventario
- La nota de entrada que ya debe de existir previamente
- La descripción
- Observaciones
- La partida presupuestal a la que pertenece
- El determinante o subgrupo de la partida presupuestal
- La dependencia
- La ubicación geográfica dentro del país
- La ubicación física dentro de las instalaciones del instituto
- Empleado que tendrá asignado ese bien
- Número de control para la relación del empleado contra bien

Valores de revaluación y depreciación para los que sean el caso

Fechas en que se calculó la revaluación y depreciación

En el caso de vehículos:

Número económico

Modelo

Número de motor

Placas

Que uso se le da al vehículo

Para los bienes que tienen características técnicas especiales

La(s) característica(s) especial(es)

Bajas

El dar de baja un bien dentro de la empresa, no implica deshacerse del mismo, el dar de baja un bien inventariado es trasladarlo a un almacén especial para esos bienes, ahí seguirá con los procedimientos marcados por la normatividad que establece la empresa y se decidirá si el bien es enajenado, donado, destruido, etc., pero eso, es un trabajo independiente a la baja. La baja afecta dentro del área de inventarios y almacén al momento de cambiarle el estado (status) de uso o propiedad, para la parte contable se genera un reporte de los bienes dados de baja y con el status se identifica para que no afecte en los movimientos contables. Al llevarse a cabo la operación, se genera una nota de traspaso a la bodega de la empresa donde se explica la razón de la baja y quien lo autoriza, así como también es cambiado su status. Los datos necesarios con los que hay que contar, son los siguientes:

El número identificador del bien

Fecha en que se da de baja

Motivo por el que se está dando de baja el bien

Quién lo autoriza

El visto técnico del estado físico y funcional del bien

La nota de traspaso para dar de baja

Cambios

La siguiente operación a tratar es la de aplicar cambios, y para esto, se especifica que se haga por medio del número consecutivo asignado al bien o por medio de la selección dentro de un listado de bienes filtrados.

Consultas

Al ser éste, un sistema en el cual las consultas de bienes pueden ser realizadas por una gran variedad de posibilidades, se ha definido la creación de un filtro general en el que se tomen en cuenta los campos más comunes de filtrado y las opciones adicionales más usadas. El resultado del filtro mostrará los campos más importantes de los bienes.

Una consulta independiente, es la de los bienes con características especiales; para esta consulta sólo se requiere que aparezcan todos aquellos bienes que tengan estas cualidades y sin ningún filtrado previo.

2.3.1.3 Resguardos

Altas

Cuando un bien ya es reconocido como parte de la empresa, entonces es asignado a un empleado para que éste tenga la custodia y responsabilidad del mismo, el otorgar esta responsabilidad se conoce como resguardo. Un empleado puede tener varios bienes a su cargo pero nunca se comparte esta responsabilidad, esto es, si un empleado tiene el resguardo de un bien, ningún otro puede tener ese resguardo, ya sea por el identificador del resguardo o por el bien o bienes que involucra ese resguardo. La información necesaria para tener un resguardo es:

El identificador del bien

La clave del empleado

La clave del resguardo

Fecha en la que comenzó la asignación del bien a un empleado

Fecha de terminación que tendrá la asignación del bien a ese empleado

Bajas

Solo es necesario saber que bien ya no estará en custodia del empleado para dar de baja el resguardo, pero hay que tener en cuenta que el bien no puede quedar sin resguardo, por lo que se asigna a otro empleado. Otro tipo de baja son los resguardos que llegan a su fecha de vencimiento.

Consultas

Las consultas de los resguardos se rigen por cuatro variables de búsqueda, éstas son por el identificador del bien, la clave del empleado, la dependencia o el identificador del resguardo, siendo éstos los únicos criterios a considerar.

2.3.1.4 Notas de Traspaso

Altas

Las notas de traspaso son los movimientos de ubicación física, de área o dependencia que sufren los bienes. Cuando tenemos una nota de traspaso entre áreas de una misma dependencia, el control se lleva internamente dentro de la dependencia y el documento generado es meramente informativo, pero cuando tenemos estas operaciones entre dependencias, el documento generado tiene que pasar por firma del departamento de almacén e inventarios. El identificador de la nota de traspaso es un carácter formado por tres partes, la primera es la dependencia, la segunda es un consecutivo y el último es el año. Lo necesario para realizar una nota de traspaso es:

El identificador de la nota de traspaso.

Fecha en la que se efectúa la nota de traspaso

Si es entre dependencias o entre áreas el movimiento

El o los bienes involucrados en el movimiento

La nueva ubicación física del bien

Dictamen técnico sobre el bien

Motivo del movimiento

La dependencia que remite

La firma de quién dio el visto bueno del área que remite

La firma del área operativa que remite

La dependencia que recibe

La firma del área operativa que recibe

En el caso de notas entre dependencias la firma del área de almacén e inventarios

Cambios

Las modificaciones a las notas de traspaso no tienen ninguna restricción exceptuando el identificador de la nota que no es modificable, esto es, como volver a hacer otra nota pero con el mismo identificador.

2.3.1.5 Catálogos

Los catálogos son parte importante dentro del sistema, ya que con la ayuda de ellos se validarán muchos de los datos que se capturen en la interfaz final con el usuario, existen muchos datos que podrían ser validados a través de catálogo, pero a la vez el que existan muchos catálogos puede ser contraproducente, esto en el sentido de operatividad por parte de los usuarios; con este antecedente, se analizan los casos en donde un catálogo es realmente necesario.

Los catálogos necesarios en el sistema son los de las dependencias, proveedores de la empresa, empleados de la empresa, partidas presupuestales, componentes, origen, marca y modelo del bien, zona geográfica en que se encuentra el bien dentro de instalaciones de la empresa en la República Mexicana, motivos de baja y status de un bien, así como tipo de

uso que se le da a un vehículo. Todos los catálogos anteriores requieren su clave de identificación y el nombre o descripción asociado a la clave.

Tenemos también el caso del catálogo de determinantes o subgrupos de las partidas presupuestales, que aparte de tener su clave y descripción cuenta con la clave de la partida presupuestal a la que pertenece y los CABMS que son identificadores del tipo de bienes registrados en la Secretaría de Hacienda y Crédito Público. Otro caso, es el del catálogo de los edificios que tiene la clave de la zona geográfica en donde se localiza el edificio, y por último, tenemos el catálogo de los índices de depreciación-revaluación que contiene los años en que se deprecia la partida presupuestal, el año en el que se ingresó ese índice, la clave de la partida presupuestal y el valor del índice.

2.3.1.6 Proceso Depreciación

Este proceso es aplicar una depreciación y una revaluación a los bienes; para ejecutarlo se requieren los datos del bien como son su valor de adquisición, el año de compra y su partida de adquisición, estos datos son consignados en el registro de los bienes, por lo que sólo es necesario tener actualizado un catálogo de índices de revaluación.

La depreciación de un bien, es el valor calculado basándose en el valor de adquisición del bien, un factor que considera los años en que se deprecia totalmente ese bien y el año en que se compró. La revaluación por su parte se efectúa de acuerdo a índices preestablecidos para el tipo de bien.

2.3.1.7 Proceso Póliza Contable

Para procesar una póliza contable, se requiere tener conocimiento del tipo de movimiento que se procesa y la cuenta contable en la que se integra el mismo, esta información no cambia por lo que se incluye en el proceso como constante, con estos datos y las notas de entrada al almacén se genera la póliza contable.

2.3.1.8 Seguridad

La seguridad tiene dos niveles generales, el primero permite todas las operaciones mientras que el segundo tiene restricciones en los catálogos, ya que no pueden realizar bajas, altas y cambios, excepto en el catálogo de componentes. En el registro de bienes, las altas y bajas están restringidas completamente y los cambios sólo permitidos en la ubicación física del bien dentro de las instalaciones del instituto, el empleado que tendrá asignado el bien, el número de control para la relación del empleado contra bien y las características técnicas especiales. Los resguardos y notas de traspaso no tienen ninguna restricción, y finalmente, los procesos de la depreciación y la póliza contable son restringidos.

2.3.2 Especificaciones Informáticas

2.3.2.1 Especificaciones de Software

Este sistema que se desarrollará, tiene estimado trabajar un volumen de datos aproximado de 110,000 bienes, para los cuales se ha considerado que tengan acceso 9 usuarios, entendiendo como usuarios a las unidades operativas, con restricciones de operación a nivel de tablas en la base de datos y dos niveles generales de seguridad dentro del sistema. El sistema incluye una parte para el manejo de las cuentas en el nivel de seguridad de sistema. Por la complejidad de la seguridad dentro del sistema, ésta es considerada a partir de la etapa de diseño.

Los catálogos que se manejarán en la validación de datos, estarán integrados en el sistema y podrán ser manipulados por los usuarios. En todos los catálogos se aplicarán altas, bajas y cambios, con las excepciones de los catálogos de determinantes y empleados.

Obviamente todos los datos estarán almacenados en una base de datos y la manera de consultarlos será por medio de reportes, los cuales son la otra parte de los resultados esperados por el sistema; los reportes serán los más generales y útiles, esto es porque podemos tener una infinidad de reportes diferentes, pero no todos con utilidad general, por

eso, sólo se contemplaran reportes particulares en los casos que sean estrictamente necesarios y que la misma operación requiere de manera rutinaria para todas las operaciones.

Los procesos de depreciación y revaluación serán externos al sistema, programados en PL/SQL, esto es, estarán almacenados en los servidores dentro de la base de datos y ejecutados por el sistema; el definir esto, es por las ventajas que presenta hacerlo de esta manera, estamos hablando de ventajas de velocidad de proceso y aprovechamiento de los recursos de software con que cuenta la base de datos.

El software con el cual se está trabajando en estos momentos dentro de la empresa, es una programación con SQLWindows y por medio de conexión TCP/IP se maneja la base de datos Oracle, la cual está montada en un servidor AlphaServer de Digital, que es administrado con sistema operativo UNIX, por ser éstos los recursos con los que cuenta la empresa, la interfaz final será programada con SQLWindows.

2.3.2.2 Especificaciones de Hardware

Dentro de la institución se han preocupado por tener y actualizar, en la medida de lo posible, todas las herramientas de trabajo con las que cuenta para su desarrollo.

Recientemente el IMP realizó la compra de un número importante de equipos personales (PC's), los cuales además de reemplazar muchos de los equipos (PC's) ya obsoletos, servirán para el apoyo de más personal; estamos hablando de que los equipos donde correrá el sistema contarán con procesador 80486 a 100MHz o pentium a 166MHz o superior. Los equipos ofrecen una buena tecnología y aún con los avances que se dan muy rápidamente en la actualidad, estas PC's mantendrán un buen desempeño de aquí a un tiempo razonable.

El servidor que se encuentra en operación, es un servidor Digital AlphaServer que trabaja con dos procesadores de 64 bit's a 250MHz con el apoyo de una memoria principal

de 1GB (gigabyte), 50GB de memoria secundaria y administrado con DigitalUNIX como sistema operativo. La comunicación con los usuarios se realiza por medio de TCP/IP.

Las comunicaciones dentro de la empresa, se basan en una red institucional con topología de anillo y cableado de fibra óptica en su nivel más externo, esta red institucional se conecta al exterior a otras WAN's por medio de microondas a red UNAM. Aquí los nodos de la capa más externa son conformados por los edificios de la institución.

Un nivel más adentro, las redes se comportan con la misma topología para conectar todas las LAN's dentro de los edificios. Todo está montado en un cableado estructurado, que cubre a cada uno de los edificios que conforman a la empresa.

El último nivel lo ocupan las pequeñas LAN's que tienen topología Estrella en su mayoría, y topología de Bus algunas. Las Estrellas trabajan con cable par trenzado y las LAN's de Bus utilizan coaxial.

Todos los elementos anteriormente expuestos conforman la base de hardware en la que se implantará el sistema. Por esto, en el momento de la ejecución del sistema, el tiempo de respuesta se presume adecuado; con este tipo de equipos y para el tipo de datos e interfaz al que se pretende llegar, siempre tendremos buena respuesta.

2.4 Diagramas

Los diagramas que se presentan a continuación, son una extensión más de todo lo anterior; retomaremos los puntos ya tratados por medio de una breve explicación y con soporte en un diagrama de flujo de información. Cabe mencionar que estos diagramas, están fundamentados con base en las necesidades y requisiciones del usuario anteriormente expuestas.

La figura 2.1 describe el comportamiento actual del manejo de las operaciones sobre los activos fijos. Podemos observar que el sistema informático lo alimentan las áreas de contabilidad e inventarios, mientras que los resguardos y notas de traspaso, son operaciones que sólo guardan una relación de consulta con el sistema de activo fijo. También tenemos el proceso de depreciación y revaluación que implica el manejo de información en discos flexibles para poder ser ejecutado.

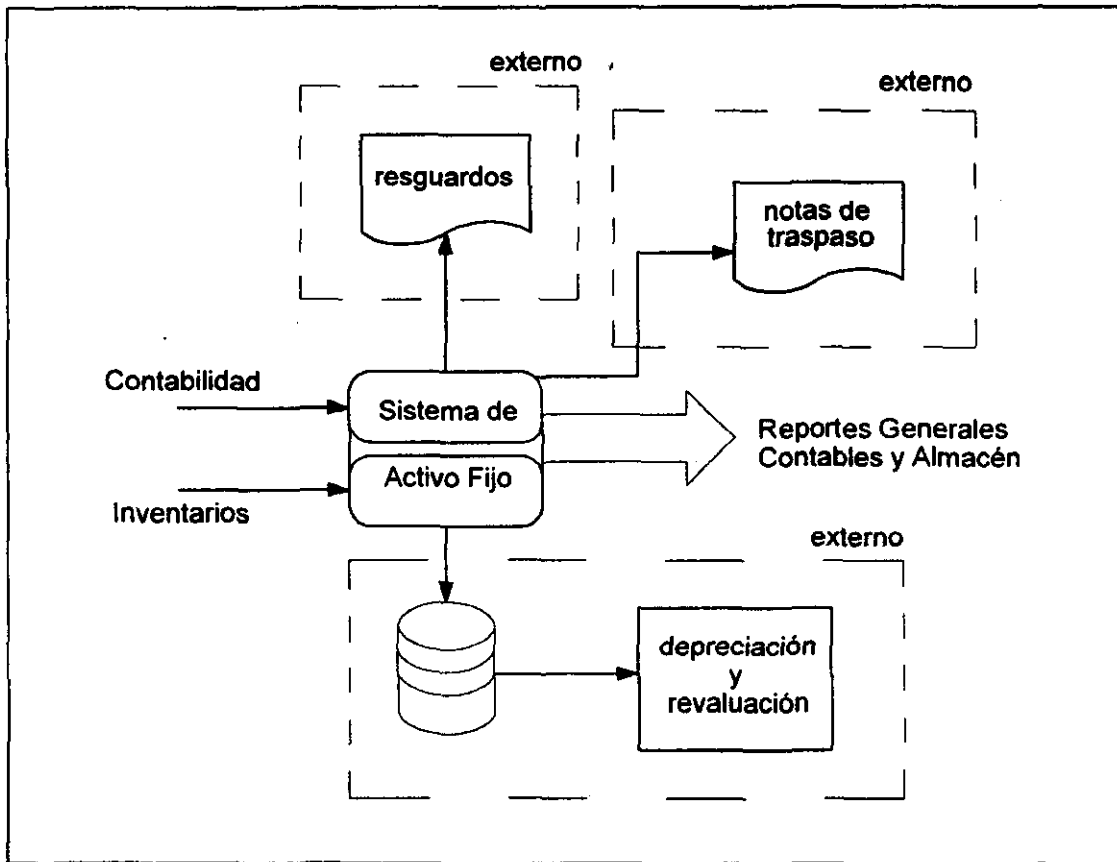


Fig. 2.1 Manejo actual de las operaciones sobre los bienes

De acuerdo a los planteamientos del usuario, se pretende llegar a un sistema de activo fijo que nos brinde todos los movimientos en una sola herramienta, esto lo podemos apreciar en la figura 2.2 que nos muestra todas las operaciones incluidas en el sistema de activo fijo.

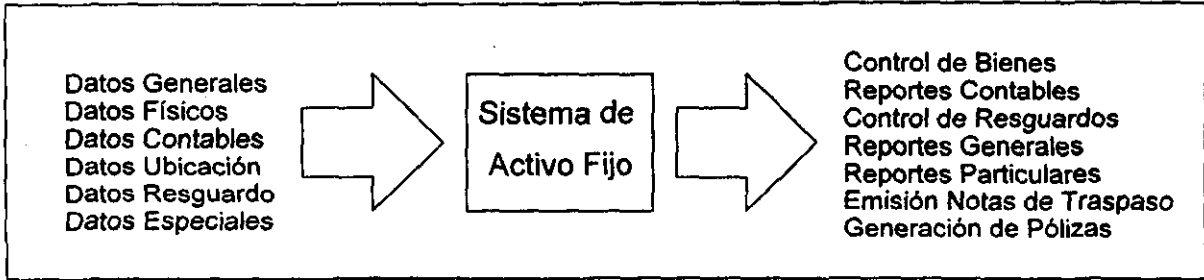


Fig. 2.2 Operaciones integradas en un solo sistema

Al tener la base de lo que realizará el sistema, tomamos las consideraciones básicas en cada operación. En el caso de las notas de entrada, el diagrama de la figura 2.3 nos muestra como la información proviene de cuatro fuentes principales, y ya conformada la nota de entrada, la podemos desglosar por proyectos o por la factura. Los resultados que obtenemos son los reportes contables y la generación de pólizas.

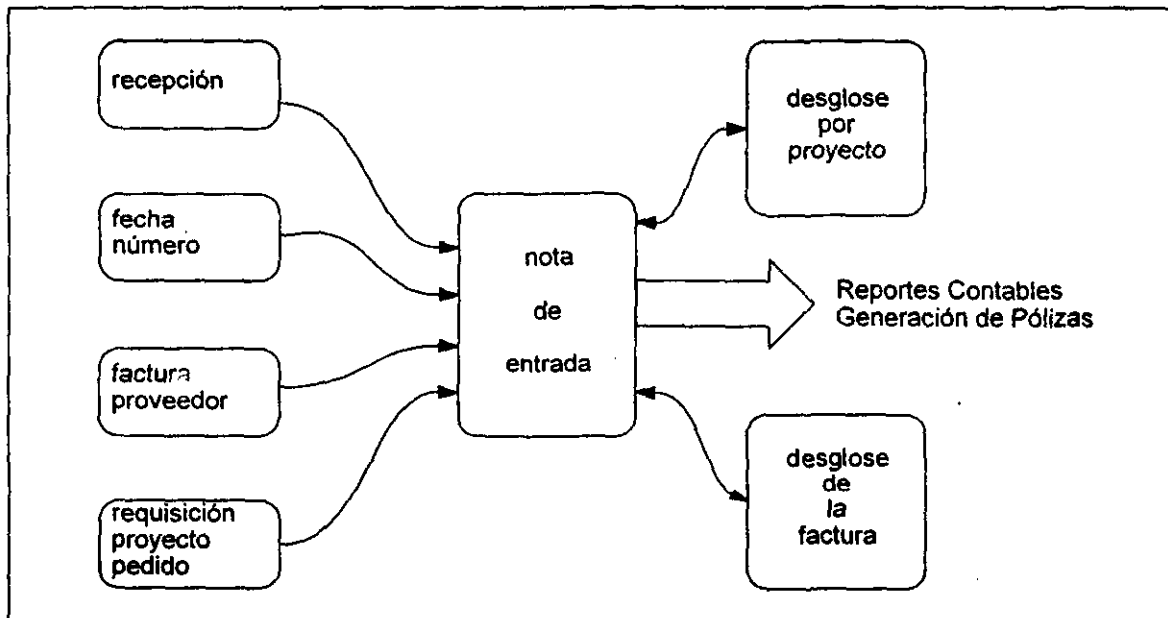


Fig. 2.3 Flujo en la nota de entrada al almacén

La figura 2.4 muestra el ciclo de los activos fijos, la información que está contenida para un activo fijo, está dada por siete fuentes principales de donde se toman los datos más

importantes de cada una y donde también tiene que ver el tipo de bien en cuestión; esta última consideración es el caso de los vehículos que llevan información particular.

Los resultados esperados del sistema, al ser alimentado con estos datos son los de controlar los bienes, entendiendo por control el tener acceso a toda la información generada por los bienes, llevándonos esto, a que si se conoce esta información, se fortalece una buena administración. La manera de consultar la información será por medio de los reportes, los cuales son parte de los resultados esperados por el sistema, los reportes serán generales y solo algunos particulares.

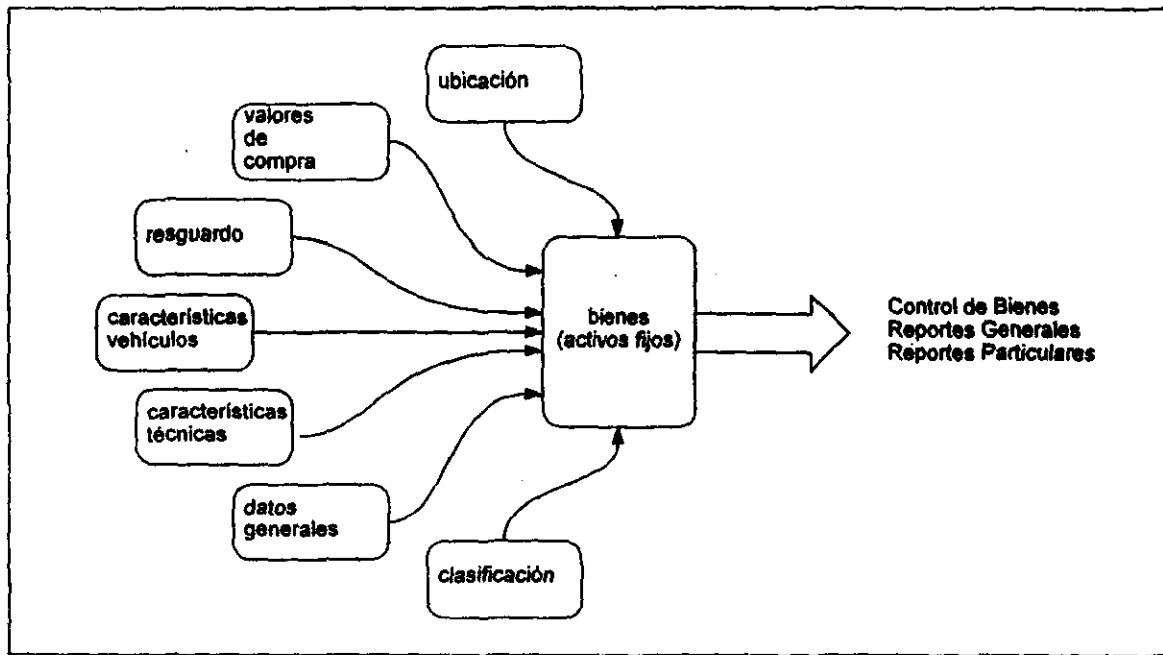


Fig. 2.4 Flujo en los activos Fijos

Cuando tenemos un bien con todas sus características, el operar un resguardo sobre un bien, lo podemos realizar al dar de alta el bien o posteriormente a esto; la figura 2.5 describe las fuentes necesarias para aplicar un resguardo y el beneficio que nos aporta el sistema al introducir esta información.

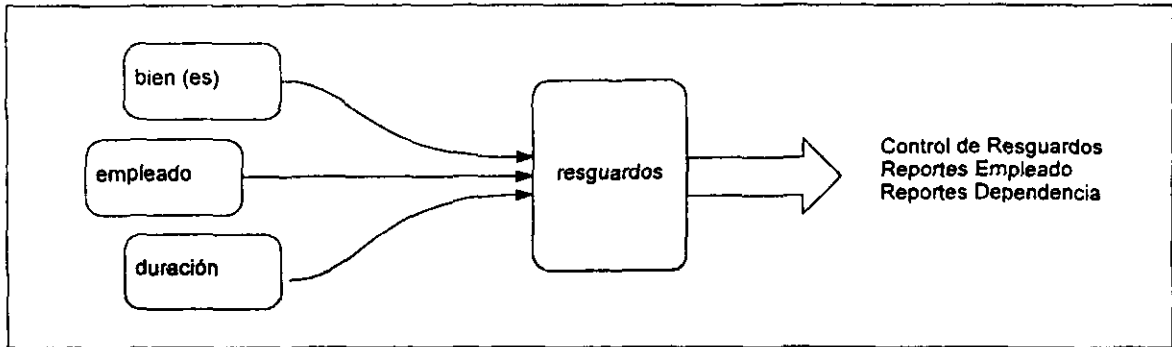


Fig. 2.5 Flujo en los resguardos

La nota de traspaso, es otra operación realizable sobre los bienes y tiene como fin la reubicación del bien, por eso es que la figura 2.6 nos presenta las fuentes que forman una nota de traspaso y el resultado que obtenemos. El diagrama presenta cuatro fuentes principales y la emisión de la nota de traspaso como resolución.

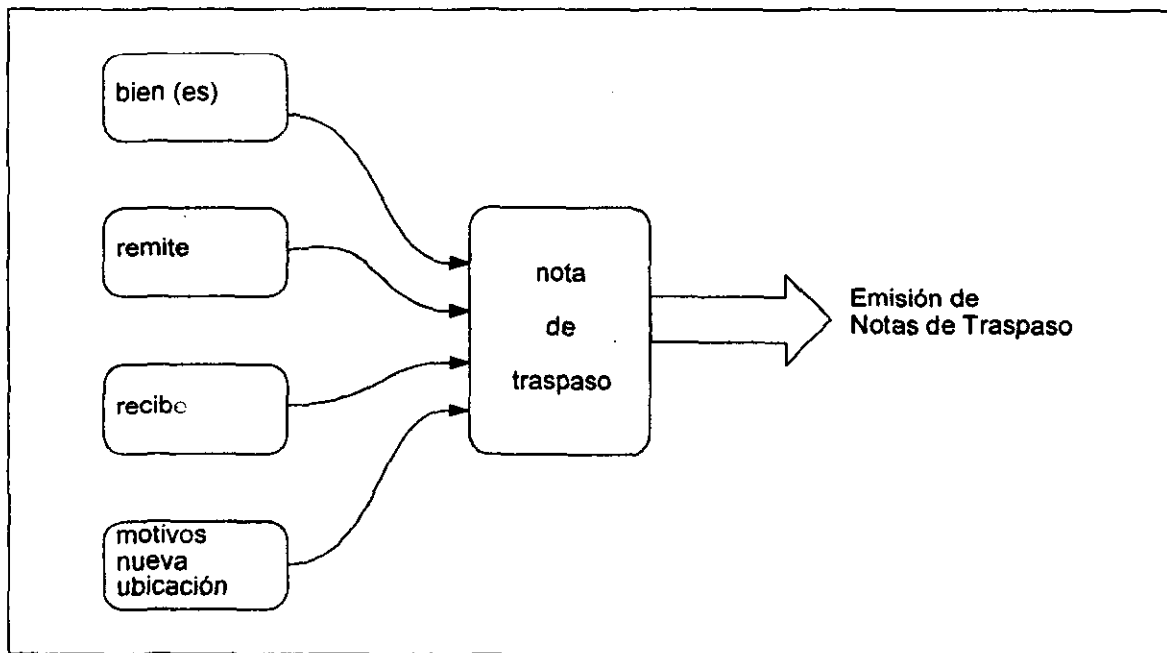


Fig. 2.6 Flujo en la nota de traspaso

CAPITULO 3

3 Diseño

- 3.1 Modular
- 3.2 Diseño de los datos
- 3.3 Diseño Arquitectónico
- 3.4 Procedimental

Una vez que ya se tiene toda la información de los requerimientos, definiciones y especificaciones que debe contemplar el sistema. El siguiente paso es el diseño. Por diseño entendemos el crear la base de la solución propuesta al problema, aquí está el alma del sistema y la clave entre tener la solución correcta o no tenerla. Al momento de arrancar esta etapa, si la información recaudada es excesiva, su manejo directo se torna difícil; para evitar esto y tener un diseño más óptimo y confiable, es recomendable fragmentar el problema y reagruparlo de acuerdo a características comunes, este proceso es denominado modular la Información.

Si ya tenemos la información modulada, entonces podemos enfocarnos a dar comienzo con la estructura de la base de datos, y una vez concluida, se continúa con el diseño arquitectónico, esto es, la presentación final que el usuario trabajará. El último paso de diseñar, lo tenemos en los algoritmos informáticos que aplicaremos para la resolución del problema.

3.1 Modular

Cuando tenemos la recopilación de la información, ésta existe como se muestra en la figura 3.1, en la cual observamos que no están definidos grupos que asocien los datos del sistema, es decir, bloques especiales para las operación.



Fig. 3.1 Universo de datos del sistema de activo fijo

De acuerdo a sus principales funciones y características, las actividades en los bienes llevan una cierta relación que puede llegar a ser grande, tomando esto en cuenta tenemos una primera agrupación de las operaciones; la figura 3.2 describe los principales grupos del sistema como el de registro de bienes, notas de entrada, notas de traspaso, resguardos y catálogos. Aunque entre ellos también existe una relación, la finalidad en estos momentos es verlos modularmente.

**Agrupamiento Principal
del
Sistema de Activo Fijo**

Notas de Entrada	Notas de Traspaso
Registro de Bienes	Resguardo
Depreciación Revaluación	Catálogos

Fig. 3.2 Grupos de datos

Profundizando en cada uno de los módulos principales, desprendemos las actividades que se agrupan en cada uno de éstos, de esta manera la figura 3.3 presenta el árbol resultante de los grupos y sus ramificaciones.

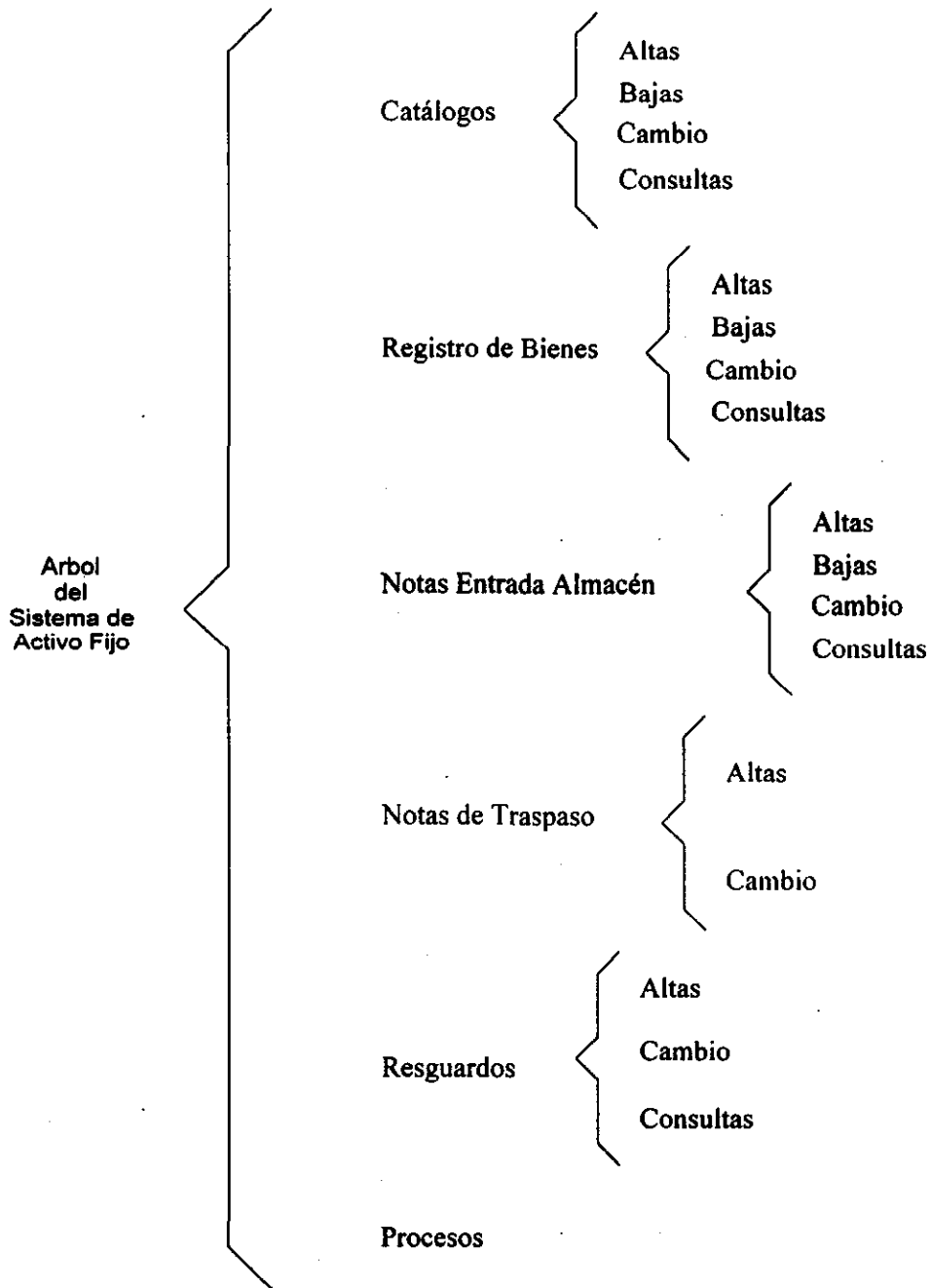


Fig. 3.3 Subgrupos de datos

3.2 Diseño de los Datos

Al concluir esta primera etapa, hemos logrado visualizar fácilmente las relaciones que existen entre cada uno de los módulos, esto nos permite estructurar con mayor eficiencia la base de datos y sus relaciones.

Un módulo importante pero que se aplica directamente en los algoritmos es el de la seguridad, éste se aplica directamente porque sólo se requieren las especificaciones de donde están los permisos, por eso únicamente se describe su entidad dentro del diccionario de datos.

El diccionario de datos, es la descripción completa de una base de datos internándose a partir de sus entidades, sus relaciones entre ellas y llegando al detalle de cada campo. La información que presenta para cada campo dentro de una entidad es su nombre completo, tipo de dato y longitud, la opción de ser nulo o no, si es llave primaria, si es llave foránea y una descripción de los datos que guardará bajo ese nombre. De esta manera tenemos el Diccionario:

CATALOGOS

Nombre de la Entidad MAF_DEPENDENCIA

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
CVE_DEPENDENCIA	CHAR(6)	NOT NULL	(PK)		Clave de la dependencia donde se tiene el bien
DEPENDENCIA	VARCHAR2(100)	NULL			Nombre de la dependencia donde se tiene el bien

Nombre de la Entidad MAF_PARTIDA

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
CVE_PARTIDA	NUMBER(4)	NOT NULL	(PK)		La clave de la partida presupuestal a la que pertenece el bien
PARTIDA	VARCHAR2(60)	NULL			Nombre de la partida presupuestal a la que pertenece el bien

Nombre de la Entidad MAF_GRUPO

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
CVE_PARTIDA	NUMBER(4)	NOT NULL	(PK)	(FK)	La clave de la partida presupuestal a la que pertenece el bien
CVE_GRUPO	CHAR(4)	NOT NULL	(PK)		La clave del determinante o subgrupo de la partida presupuestal en el que entra el bien
GRUPO	CHAR(80)	NULL			Nombre del subgrupo donde se integra el bien
CVE_CADMS	CHAR(10)	NULL			Identificador del tipo de bien registrado en la Secretaría de Hacienda y Crédito Público

Nombre de la Entidad MAF_ZONA

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
CVE_ZONA	CHAR(5)	NOT NULL	(PK)		Clave de la ubicación geográfica dentro del país de las instalaciones donde se encuentra el bien
ZONA	CHAR(20)	NULL			El nombre de la ubicación geográfica dentro del país de las instalaciones donde se encuentra el bien

Nombre de la Entidad MAF_EDIFICIO

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
CVE_ZONA	CHAR(5)	NOT NULL	(PK)	(FK)	Clave de la ubicación geográfica dentro del país de las instalaciones donde se encuentra el bien
CVE_EDIFICIO	CHAR(5)	NOT NULL	(PK)		La clave del edificio donde se encuentra ubicado el bien
EDIFICIO	VARCHAR2(60)	NULL			El nombre del edificio donde se encuentra ubicado el bien

Nombre de la Entidad MAF_MARCA

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
CVE_MARCA	CHAR(10)	NOT NULL	(PK)		Clave de la marca del bien
MARCA	CHAR(35)	NULL			Nombre de la marca del bien

Nombre de la Entidad MAF_MODELO

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
CVE_MARCA	CHAR(10)	NOT NULL			Clave de la marca del bien
CVE_MODELO	CHAR(10)	NOT NULL			Clave del modelo del bien
MODELO	CHAR(35)	NULL			Nombre del modelo del bien

Nombre de la Entidad MAF_STATUS_BIEN

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
CVE_STATUS_BIEN	CHAR(2)	NOT NULL	(PK)		La clave del status del bien
STATUS_BIEN	CHAR(28)	NULL			El tipo de status del bien

Nombre de la Entidad MAF_EQUIPO

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
CVE_INVENTARIO	CHAR(7)	NOT NULL	(PK)		Identificador del bien
CVE_COMPONENTE	CHAR(10)	NOT NULL	(PK)	(FK)	Clave del componente específico
CANTIDAD	CHAR(10)	NULL			Las unidades registradas para el componente específico
UNIDAD	CHAR(10)	NULL			Las unidades en que esta medido el componente específico

Nombre de la Entidad MAF_COMPONENTE

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
CVE_COMPONENTE	CHAR(10)	NOT NULL	(PK)		Clave del componente específico de un bien
COMPONENTE	CHAR(40)	NULL			Nombre completo del componente

Nombre de la Entidad MAF_INDICE_DEP

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
ANIO	CHAR(4)	NOT NULL	(PK)		Fecha en la que se adquirió el bien
ANIOS_DEPRECIACION	CHAR(2)	NOT NULL	(PK)		Tiempo en que quedará depreciado totalmente
CVE_PARTIDA	NUMBER(4)	NOT NULL	(PK)	(FK)	Partida presupuestal correspondiente al bien
INDICE_DEP	NUMBER(7 5)	NULL			Porcentaje en que se deprecia por año

Nombre de la Entidad MAF_INDICE_REV

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
ANIO	CHAR(4)	NOT NULL	(PK)		Fecha en la que se adquirió el bien
MES	CHAR(2)	NOT NULL	(PK)		Mes en que se aplica
CVE_PARTIDA	NUMBER(4)	NOT NULL	(PK)	(FK)	Partida presupuestal correspondiente al bien
INDICE_REV	NUMBER(9 6)	NULL			Porcentaje de incremento aplicado por mes

Nombre de la Entidad MAF_MOTIVO

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
CVE_MOTIVO	CHAR(4)	NOT NULL	(PK)		Clave del motivo
MOTIVO	CHAR(60)	NULL			Descripción del motivo

Nombre de la Entidad MAF_USO

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
CVE_USO	CHAR(4)	NOT NULL	(PK)		Clave del tipo de uso que tiene un vehículo
USO	CHAR(50)	NULL			Descripción del uso que se le da al vehículo

Nombre de la Entidad MAF_AFECTACION

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
CVE_AFECTACION	CHAR(3)	NOT NULL	(PK)		Clave del tipo de origen del bien
AFECTACION	CHAR(20)	NOT NULL			La procedencia del bien, nacional o importado

Nombre de la Entidad MAF_FIRMA

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
FIRMANTE	VARCHAR2(40)	NULL			Persona que firma

REGISTRO DE BIENES**Nombre de la Entidad MAF_ACTIVO_FIJO**

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
CVE_INVENTARIO	CHAR(7)	NOT NULL	(PK)		Número consecutivo único que se le asigna al bien
CVE_DEPENDENCIA	CHAR(6)	NOT NULL		(FK)	La dependencia que tiene a cargo el bien
CVE_PARTIDA	NUMBER(4)	NOT NULL		(FK)	La clave de la partida presupuestal a la que pertenece el bien
CVE_GRUPO	CHAR(4)	NOT NULL		(FK)	La clave del determinante o subgrupo de la partida presupuestal en el que entra el bien
CVE_ZONA	CHAR(5)	NULL		(FK)	La clave de la ubicación geográfica dentro del país de las instalaciones donde se encuentra el bien
CVE_EDIFICIO	CHAR(5)	NULL		(FK)	La clave del edificio donde se encuentra ubicado el bien
NUM_NOTA_ENTRADA	CHAR(10)	NOT NULL		(FK)	El identificador de la nota de entrada en donde se ampara el bien
CVE_STATUS_BIEN	CHAR(2)	NULL		(FK)	Estado del bien
CVE_MARCA	CHAR(10)	NULL			La clave de la marca comercial del bien
CVE_MODELO	CHAR(10)	NULL			La clave del modelo del bien
NUM_SERIE	CHAR(20)	NULL			El número de serie que corresponde al bien
DESCRIPCION	VARCHAR2(150)	NULL			La descripción de las características generales del bien
OBSERVACIONES	VARCHAR2(80)	NULL			Observaciones adicionales acerca del bien
VALOR_ADQUISICION	NUMBER(12 2)	NULL			El valor del importe inicial del bien registrado en la respectiva factura
VALOR_ACTUAL	NUMBER(12 2)	NULL			El valor actual del bien después de sufrir depreciaciones y revaluaciones
ORIGEN	CHAR(11)	NOT NULL			La procedencia del bien esto es nacional o importado
FECHA_ACTUALIZACION	DATE	NULL			La última fecha en la que fue actualizado el bien
UBICACION	VARCHAR2(60)	NULL			La descripción de la ubicación física de un bien dentro de los edificios de la Empresa
FECHA_PROCESO	DATE	NULL			Fecha en la que se llevó a cabo el proceso de la revaluación y depreciación sobre el bien
REVALUACION	NUMBER(12 2)	NULL			El monto de la revaluación a aplicar al bien

DEPRECIACION	NUMBER(12 2)	NULL			El monto de la depreciación a aplicar al bien
DEPRECIACION_REVAL	NUMBER(12 2)	NULL			El monto de la depreciación revaluada a aplicar al bien
PARTIDA_ADQUISICION	NUMBER(4)	NULL			La partida presupuestal contable en la que entra el bien
INV_ANTERIOR	CHAR(12)	NULL			El identificador histórico o anterior del bien
RESGUARDO	CHAR(7)	NULL			El identificador del resguardo en que entra el bien
BAJA_LOGICA	CHAR(1)	NOT NULL			Bandera que indica la baja del bien de manera lógica, o sea, sin deshacerse del bien
ANIO_COMPRA	DATE	NULL			La fecha en que se adquirió el bien
CONSECUTIVO	CHAR(7)	NOT NULL		(FK)	El número que identifica al bien dentro de una factura

Nombre de la Entidad MAF_VEHICULO

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
CVE_INVENTARIO	CHAR(7)	NOT NULL	(PK)		Identificador del vehículo como parte del activo fijo
CVE_USO	CHAR(4)	NOT NULL		(FK)	Clave del tipo de uso que tiene el vehículo
MODELO_ANIO	NUMBER(4)	NULL			El año del modelo del vehículo
NUM_MOTOR	CHAR(20)	NULL			El número del motor del vehículo
PLACAS	CHAR(8)	NULL			Las placas del vehículo
NUM_ECONOMICO	NUMBER(4)	NULL			El número de identificación como vehículo

Nombre de la Entidad MAF_GTOADICIONAL

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
CVE_INVENTARIO	CHAR(7)	NOT NULL	(PK)	(FK)	Identificador del bien que tiene gastos adicionales
AAMM_AUXILIAR	CHAR(10)	NOT NULL	(PK)		Identificador del documento contable que resume los movimientos de las pólizas
NUM_POLIZA	CHAR(10)	NOT NULL	(PK)		Número de la póliza que registró esos gastos adicionales
GTO_ADICIONAL	NUMBER(12 2)	NOT NULL			El monto del gasto adicional
DEPRECIACION	NUMBER(12 2)	NULL			El monto de lo depreciado
REVALUACION	NUMBER(12 2)	NULL			El monto de lo revaluado
DEPRECIACION_REVAL	NUMBER(12 2)	NULL			El monto de la depreciación revaluada
FECHA_PROCESO	DATE	NULL			La fecha en que se aplica el proceso de depreciación revaluación

NOTAS DE TRASPASO

Nombre de la Entidad MAF_ENTRA_SALE

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
NO_NOTA_TRASPASO	CHAR(12)	NOT NULL	(PK)		El identificador de la nota de traspaso.
CVE_INVENTARIO	CHAR(7)	NOT NULL	(PK)	(FK)	El identificador del bien que entra en el traspaso
CVE_MOTIVO	CHAR(4)	NULL		(FK)	La clave del tipo de causa por la que se realiza el traspaso
NO_DICTAMEN	CHAR(7)	NULL			Número que identifica el dictamen técnico
FECHA_TRASPASO	DATE	NULL			La fecha en que se esta realizando el traspaso
DESC_MOTIVO	VARCHAR2(120)	NULL			Explicación del motivo del traspaso
REFERENCIA	CHAR(15)	NULL			Una descripción general y concreta del tipo de traspaso que se maneja
AREA_OPER_REMITE	VARCHAR2(30)	NULL			Jefe del área operativa que esta remitiendo
AREA_OPER_RECIBE	VARCHAR2(30)	NULL			Jefe del área operativa que esta recibiendo
UNICOTA_REMITE	VARCHAR2(30)	NULL			La persona que dará el visto bueno de la unidad administrativa que remite
UNICOTA_RECIBE	VARCHAR2(30)	NULL			La persona de unidad administrativa que queda enterada del traspaso
AUTORIZADA	CHAR(1)	NOT NULL			En el caso de que sea autorizado o no el traspaso se tiene una S o N
TIPO_TRASPASO	CHAR(12)	NOT NULL			Clasifica si es un traspaso entre dependencias o entra áreas
CVE_DEP_REMITE	CHAR(6)	NULL		(FK)	La clave de la dependencia que remite los bienes
CVE_DEP_RECIBE	CHAR(6)	NULL		(FK)	La clave de la dependencia que recibe los bienes
CVE_ZONA	CHAR(5)	NULL			Clave de la zona de la nueva ubicación
CVE_EDIFICIO	CHAR(5)	NULL			Clave del edificio de la nueva ubicación
CVE_ZONA_REMITE	CHAR(5)	NULL			Clave de la zona de la ubicación anterior
CVE_EDIF_REMITE	CHAR(5)	NULL			Clave del edificio de la ubicación anterior

NOTAS DE ENTRADA

Nombre de la Entidad MAF_NOTA_ENTRADA

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
NUM_NOTA_ENTRADA	CHAR(10)	NOT NULL	(PK)		El identificador de la nota de entrada
FECHA_NOTAENTRADA	DATE	NOT NULL			La fecha en que se esta creando la nota de entrada
CVE_DEPENDENCIA	CHAR(6)	NOT NULL		(FK)	La clave de la dependencia que genera la nota de entrada
CVE_PARTIDA	NUMBER(4)	NOT NULL		(FK)	La clave de la partida presupuestal donde pertenecen los bienes que entran
CVE_PROVEEDOR	CHAR(8)	NOT NULL			La clave del proveedor que distribuyó el o los bienes
PEDIDO	CHAR(10)	NULL			El identificador del pedido hecho al proveedor
OFICIO	CHAR(10)	NULL			El identificador del oficio donde se autorizó la adquisición
REQUISICION	CHAR(10)	NULL			El identificador de la requisición hecha para la adquisición de bienes
CVE_PROYECTO	CHAR(13)	NULL			La clave del proyecto al que se le carga la nota de entrada
FECHA_FACTURA	DATE	NOT NULL			La fecha en que se facturaron el o los bienes
NUM_FACTURA	CHAR(10)	NOT NULL			El identificador de la factura que ampara a el o los bienes
IMPORTE	NUMBER(12 2)	NOT NULL			El valor total sin desglose de la factura
IVA	NUMBER(12 2)	NULL			El valor del IVA generado por el importe de la factura
PECE	NUMBER(12 2)	NULL			El valor del PECE generado por el importe de la factura
OBS_NOTA_ENTRADA	VARCHAR2(60)	NULL			Observaciones adicionales que se requieran agregar a la nota de entrada
ORIGEN	CHAR(11)	NOT NULL			Si es nacional o importado
NUM_POLIZA	NUMBER(5)	NULL			Número de la póliza en donde queda registrada la nota de entrada contablemente
TIPO_POLIZA	CHAR(1)	NOT NULL			El tipo de póliza que se registra, póliza de diario o informativa
CVE_EMPLEADO	CHAR(7)	NOT NULL			La clave del empleado que recibió el o los bienes
FECHA_RECEPCION	DATE	NOT NULL			Fecha en que se recibe el o los bienes

TIPO_IVA	CHAR(8)	NULL			El tipo de IVA que se aplicó en la factura
FECHA_POLIZA	DATE	NULL			La fecha en que se genera la póliza contablemente

Nombre de la Entidad MAF_MOV_NOTA

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
NUM_NOTA_ENTRADA	CHAR(10)	NOT NULL	(PK)	(FK)	El identificador de la nota a la que se le aplican los movimientos
CONSECUTIVO	CHAR(7)	NOT NULL	(PK)		Número que identifica el bien dentro de la lista de bienes de la factura
SURTIDAS	NUMBER(4)	NULL			El número de bienes del mismo tipo que fueron surtidos
AVERIADAS	NUMBER(4)	NULL			El número de bienes que llegaron en mal estado
PENDIENTES	NUMBER(4)	NULL			El número de bienes que quedaron pendientes por entregar
PRECIO_UNITARIO	NUMBER(12 2)	NULL			El precio unitario de cada bien

Nombre de la Entidad MAF_PRORRATEO

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
NUM_NOTA_ENTRADA	CHAR(10)	NOT NULL	(PK)		El identificador de la nota de entrada que se esta desglosando por proyecto
NUM_PROYECTO	CHAR(13)	NOT NULL	(PK)		El identificador del proyecto
IMPORTE	NUMBER(11 2)	NOT NULL			El monto que tiene ese proyecto asignado del importe total de la nota

Nombre de la Entidad MAF_POLIZA

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
FECHA_POLIZA	DATE	NOT NULL			Fecha de la póliza de diario
NUM_POLIZA	NUMBER(5)	NOT NULL			El número que identifica la póliza
CONCEPTO	CHAR(30)	NULL			Descripción del registro contable
PARTIDA	CHAR(5)	NULL			Partida presupuestal afectada
REFERENCIA	CHAR(7)	NULL			Referencia asociada con la factura
PROYECTO	CHAR(13)	NULL			Identificador del proyecto
DEPENDENCIA	CHAR(6)	NULL			La dependencia que tiene ese proyecto a cargo
CUENTA	CHAR(8)	NULL			Cuenta contable afectada
DEUDOR	NUMBER(12 2)	NULL			Monto del movimieto contable deudor llamado también cargo
ACREEDOR	NUMBER(12 2)	NULL			Monto del movimieto contable acreedor llamado también abono

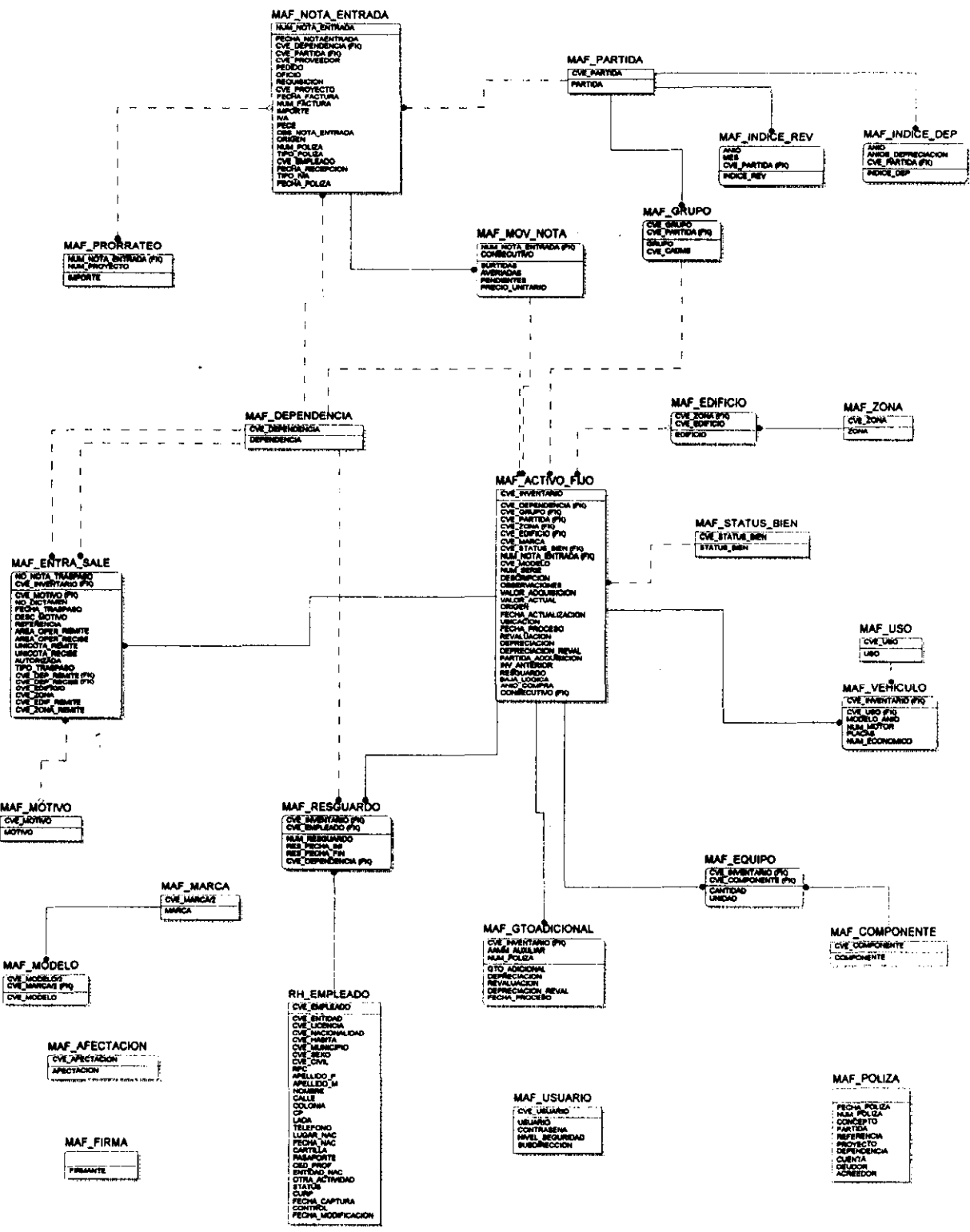
RESGUARDOS**Nombre de la Entidad MAF_RESGUARDO**

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
CVE_INVENTARIO	CHAR(7)	NOT NULL	(PK)	(FK)	El identificador del bien que queda en resguardo de un empleado
CVE_EMPLEADO	CHAR(7)	NOT NULL	(PK)		La clave del empleado que tiene asignado el bien
NUM_RESGUARDO	CHAR(11)	NOT NULL			El identifiador del resguardo
RES_FECHA_INI	DATE	NOT NULL			La fecha en que empieza el resguardo
RES_FECHA_FIN	DATE	NULL			La fecha en que termina el resguardo
CVE_DEPENDENCIA	CHAR(6)	NOT NULL		(FK)	La clave de la dependencia a la que pertenece el bien

SEGURIDAD**Nombre de la Entidad MAF_USUARIO**

Nombre Columna	Tipo Columna	Opción Null	PK	FK	Descripción del Atributo
CVE_USUARIO	NUMBER(5)	NOT NULL	(PK)		Clave del usuario que tiene acceso al sistema
USUARIO	CHAR(60)	NULL			Nombre del usuario que tiene acceso al sistema
CONTRASENA	CHAR(20)	NULL			Palabra clave o contraseña del usuario para entrar al sistema
NIVEL_SEGURIDAD	NUMBER(5)	NULL			El tipo de seguridad que es aplicado al usuario
SUBDIRECCION	CHAR(6)	NULL			Nombre del área a la que pertenece el usuario

El diccionario de datos generado para esta base de datos, nos lleva al siguiente diagrama de entidad-relación, en el que se muestran las tablas y sus relaciones entre ellas, así como sus llaves primarias y foraneas.



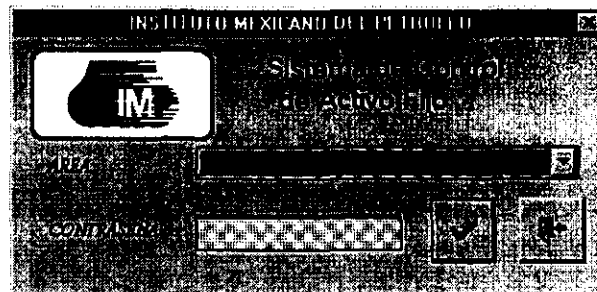
3.3 Diseño Arquitectónico

El diseño arquitectónico del sistema es crear la interfaz gráfica de la base de datos con el usuario final, o sea, las pantallas con las que trabajará el usuario final. Las pantallas se crean de acuerdo a estándares que deben ser constantes en todo el sistema, estas constantes son las formas de la pantalla, la organización en grupos de las mismas, los botones utilizados para operaciones comunes, el formato de los reportes, la ubicación de los botones, la ruta de pantallas dentro del sistema, la presentación de los catálogos existentes.

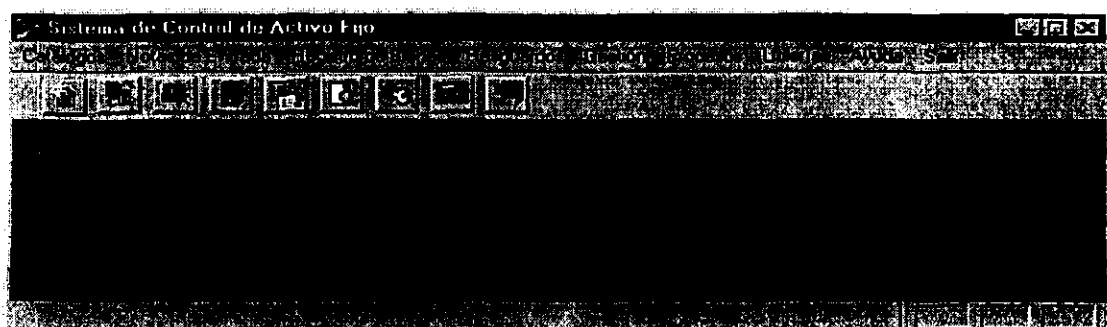
Introducción al sistema

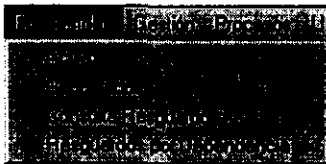
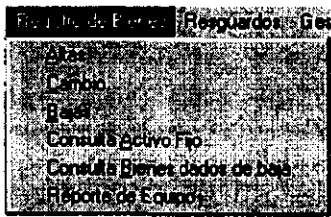
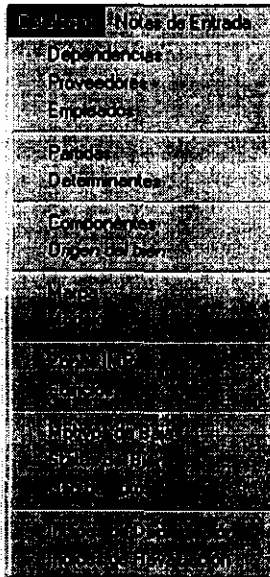


Datos de acceso



Menú Principal





Nota de Entrada

Pantalla general de notas de entrada y pantalla de detalle de una nota de entrada.

The screenshot shows a window titled "Notas de Entrada al Almacén". At the top, there is a header bar with standard window controls and a toolbar. Below the header, there is a label "Nota de Entrada:" followed by a text input field. The main area of the window is occupied by a large table with approximately 10 columns and 15 rows. The table is currently empty.

Pantalla desglose de nota de entrada y pantalla desglose por proyecto.

The screenshot shows a window titled "Nota de Entrada" with a detailed breakdown form. The form includes several sections:

- Header:** A toolbar with various icons.
- Form Fields:** Multiple input fields for "Número", "Fecha", "Proveedor", "Partida", "Importe", "T.IVA", "PECE", "Clave", and "Fecha".
- Radio Buttons:** A group of radio buttons labeled "Institución" and "Sector".
- Section Headers:** "Soporte" and "Proyecto".
- Text Areas:** A large text area for "Observaciones" at the bottom.

Registro de Bienes

The screenshot shows a software window titled "Datos Generales" with a standard Windows-style title bar. The window contains several input fields and a date field. The date field is pre-filled with "19-02-1999". The fields are arranged in a structured layout with labels on the left side.

No. S...	[]	[]	[]
No. S...	[]	19-02-1999	[]
Partida Presup...	[]	[]	[]
Descripción...	[]		
Marca	[]	Modelo	[]
Genero	[]		
[]	[]		

Arriba pantalla de datos generales de un bien.

The screenshot shows a software window titled "Datos de Ubicación". The window contains several input fields and a large text area. The fields are arranged in a structured layout with labels on the left side.

[]	[]	[]
[]	[]	[]
[]	[]	[]
[]	[]	

Datos de Compra y Revaluación

DATOS DE REVALUACIONES DE CLASE

Fecha de Revaluación:

Datos de Revaluación

Valor Actual: Fecha de Adquisición: Valor Revaluado:

Datos de Depreciación Motor

Depreciación: Por Depreciar:

Datos de Depreciación Otros

Depreciación: Por Depreciar:

Vehículo

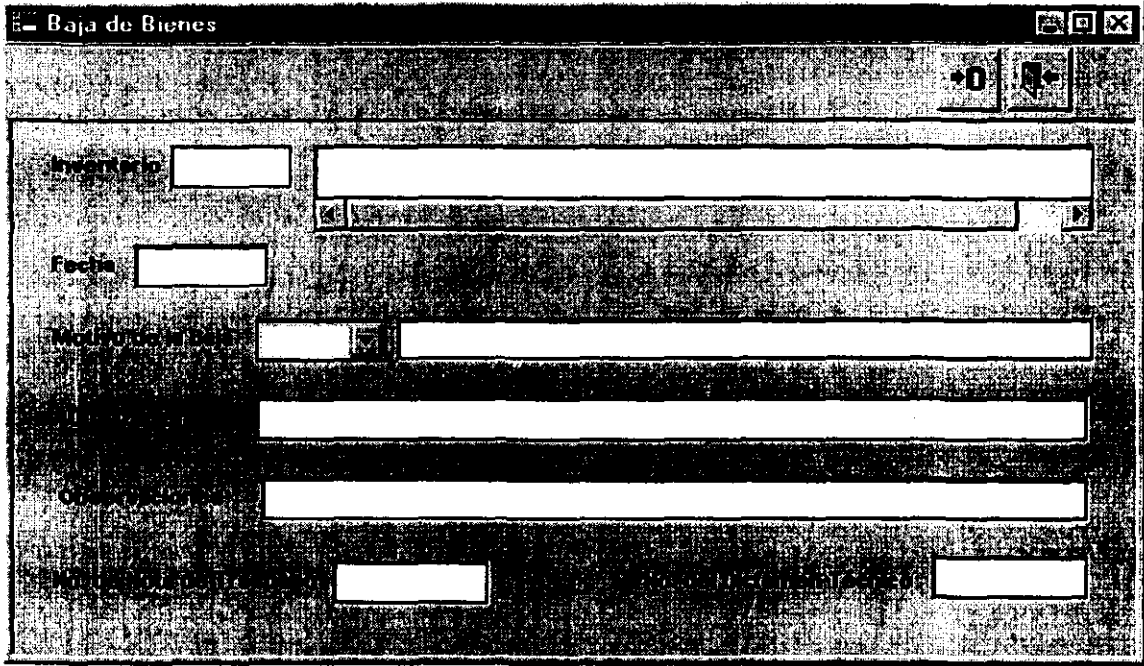
Datos de Vehículo

Vehículo

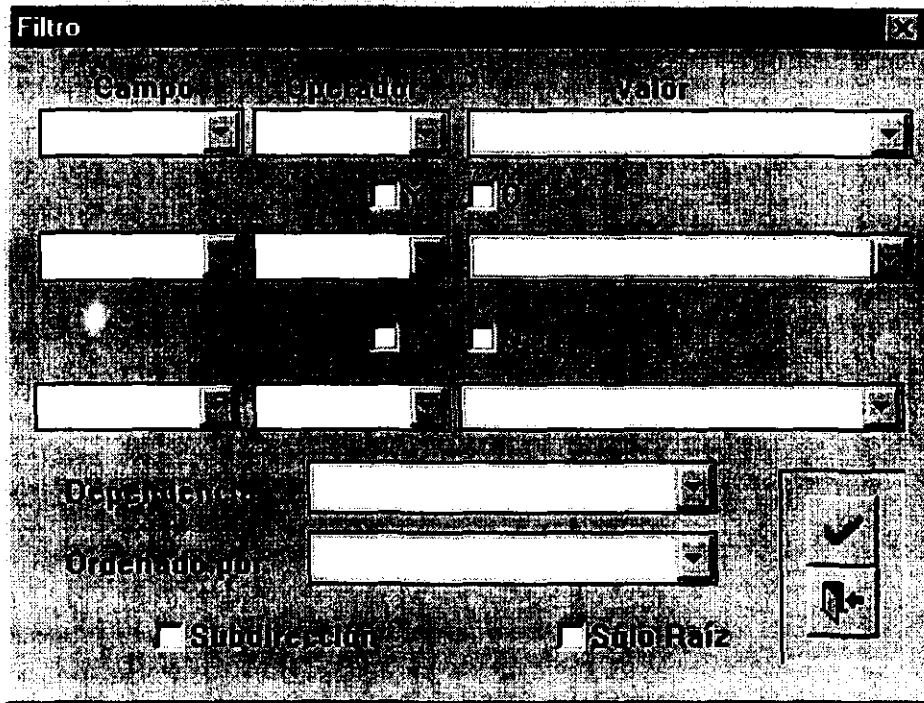
Nº Económico: Modelo Año:

Nº de Motor: Piezas: Uso:

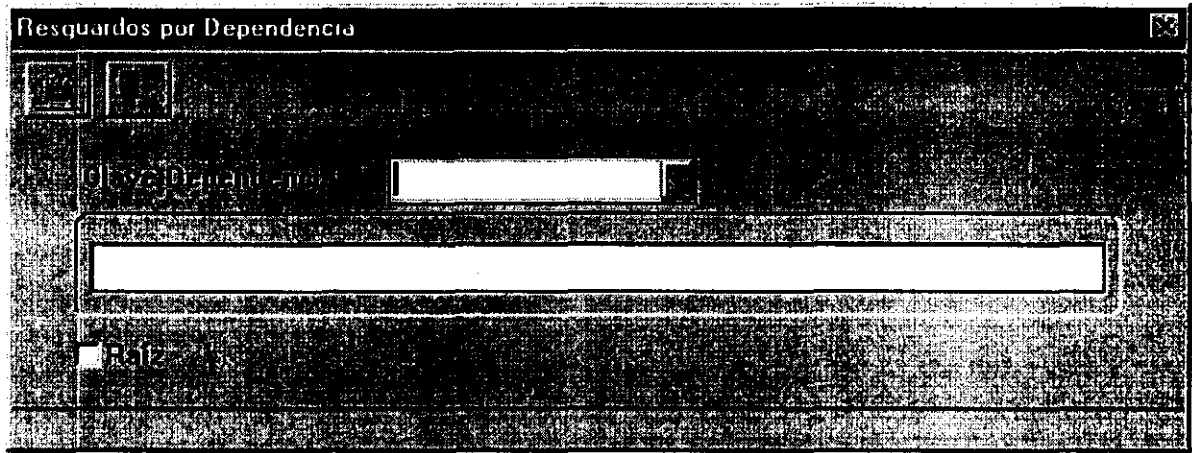
Pantalla de baja de un bien



Pantalla de filtro genérico



Pantalla filtro de resguardos por dependencia



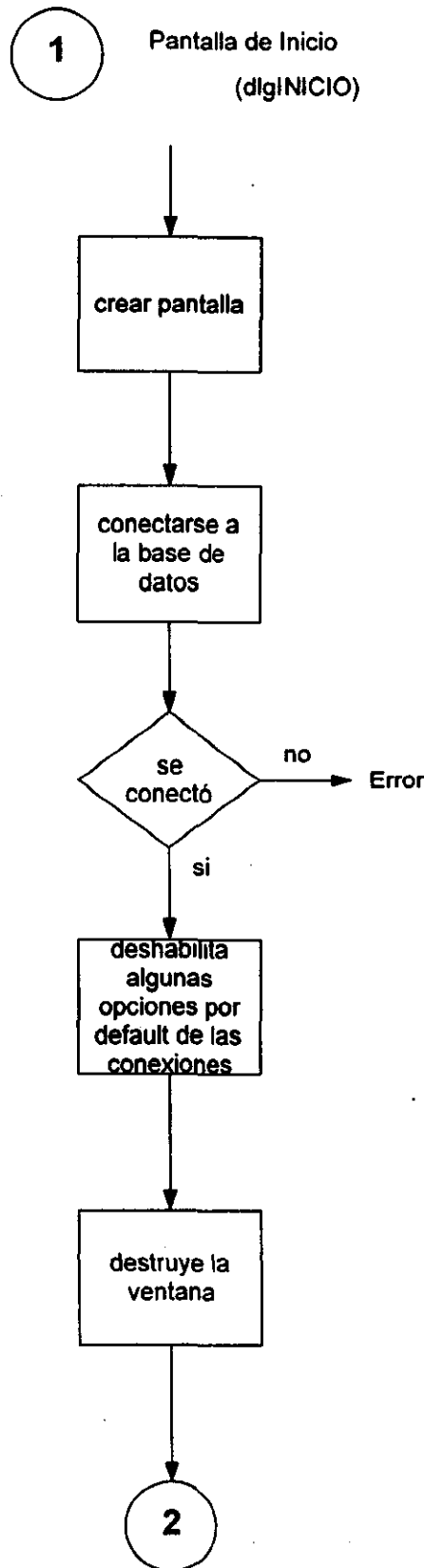
3.4 Procedimental

La parte del diseño procedimental, involucra lo que son los flujos generales de la información, para cada pantalla que se programará dentro del sistema.

Los diagramas de flujo ven los puntos principales a considerar en cada ventana, esto nos proporciona una visión más exacta de a donde se pretende llegar. Todo lo generado en los diagramas dará paso a los algoritmos o pseudocódigos de las pantallas. En los algoritmos encontraremos más consideraciones, pero esto será por la perspectiva de programación que se tomará.

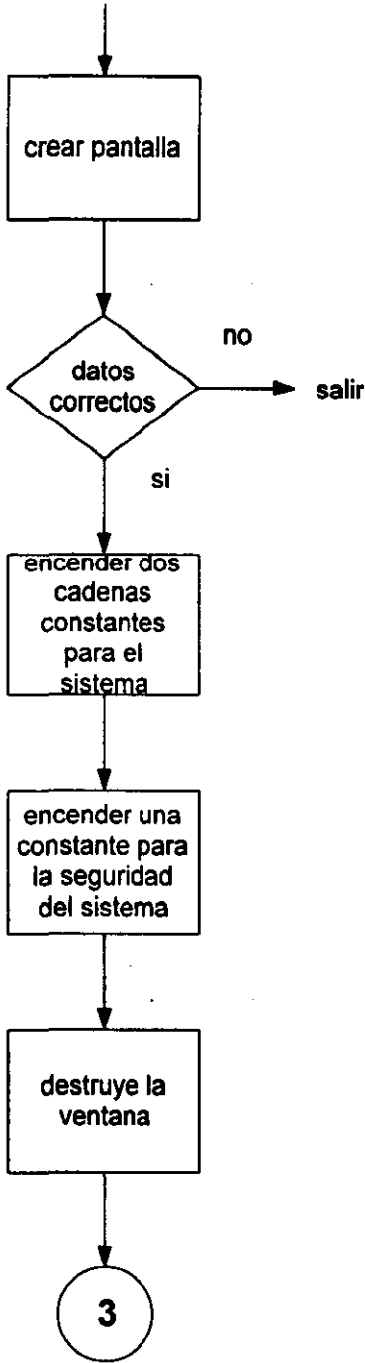
A continuación se presentan los flujos de información de las pantallas del sistema y sus vínculos con otras pantallas.

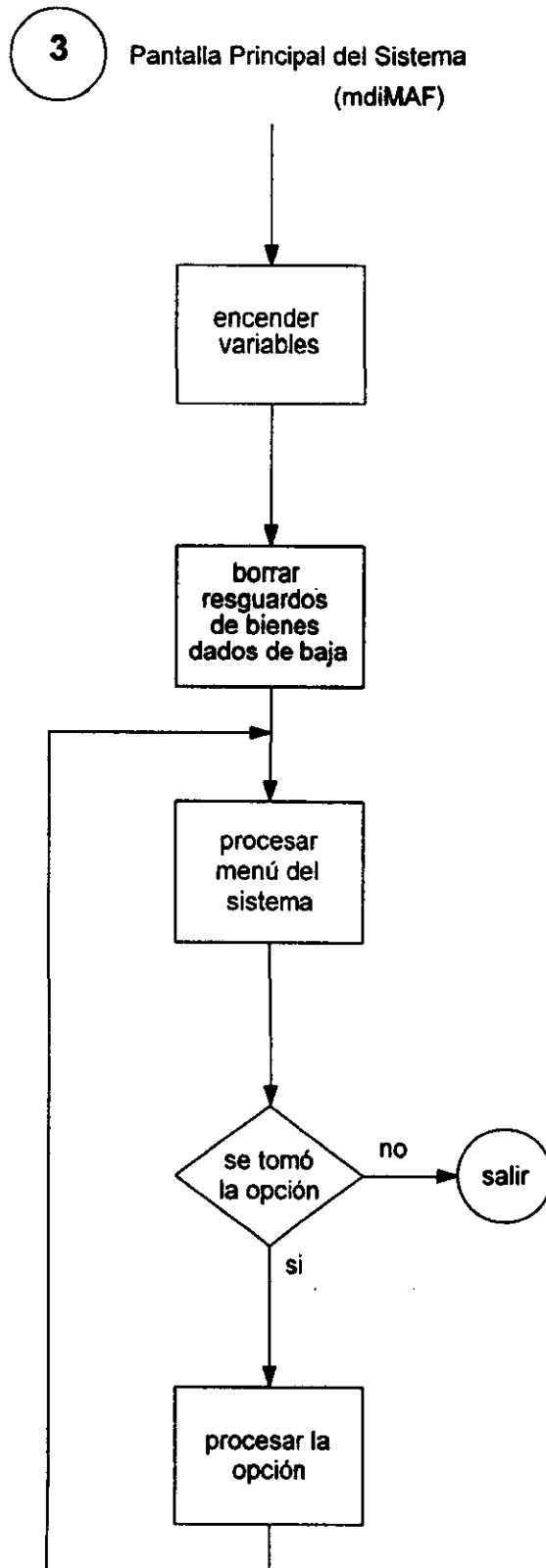
ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

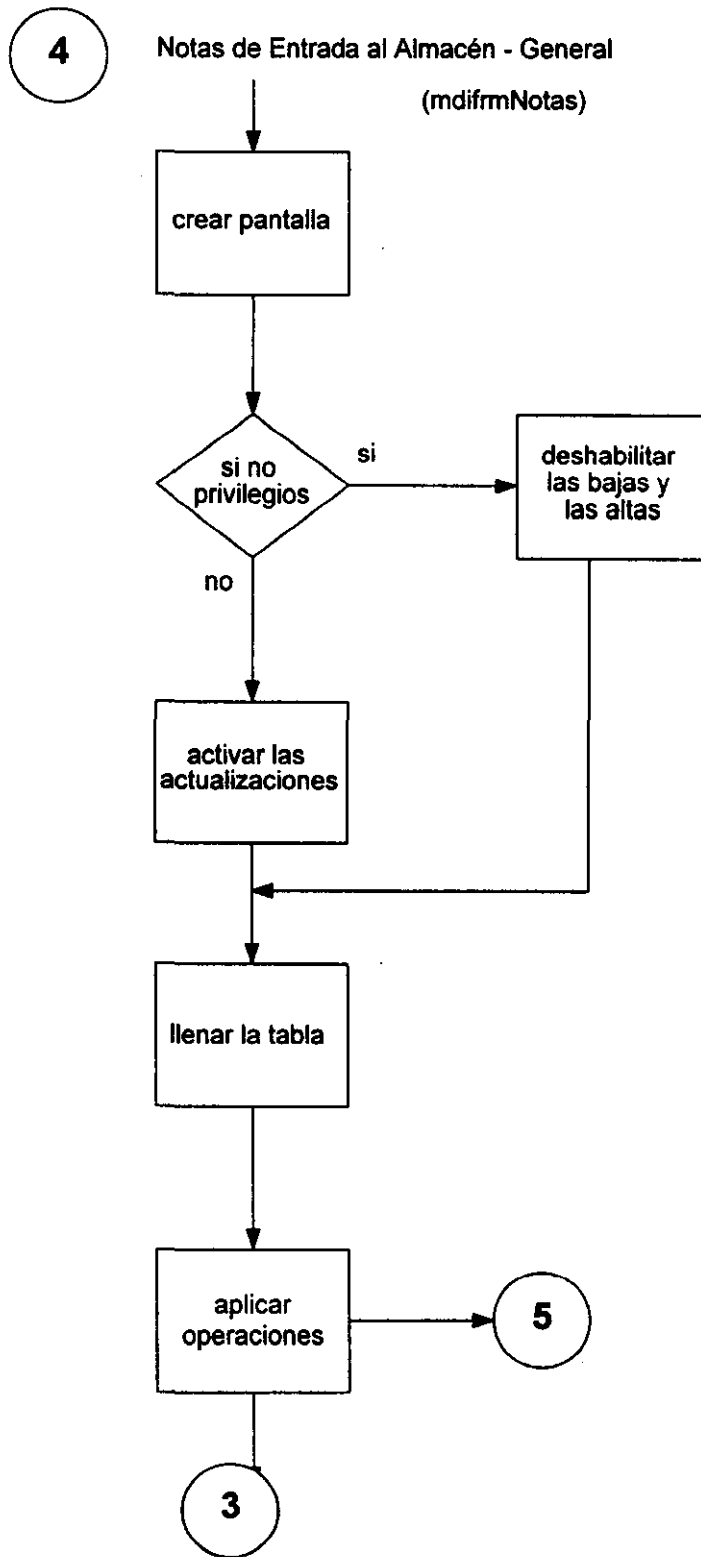


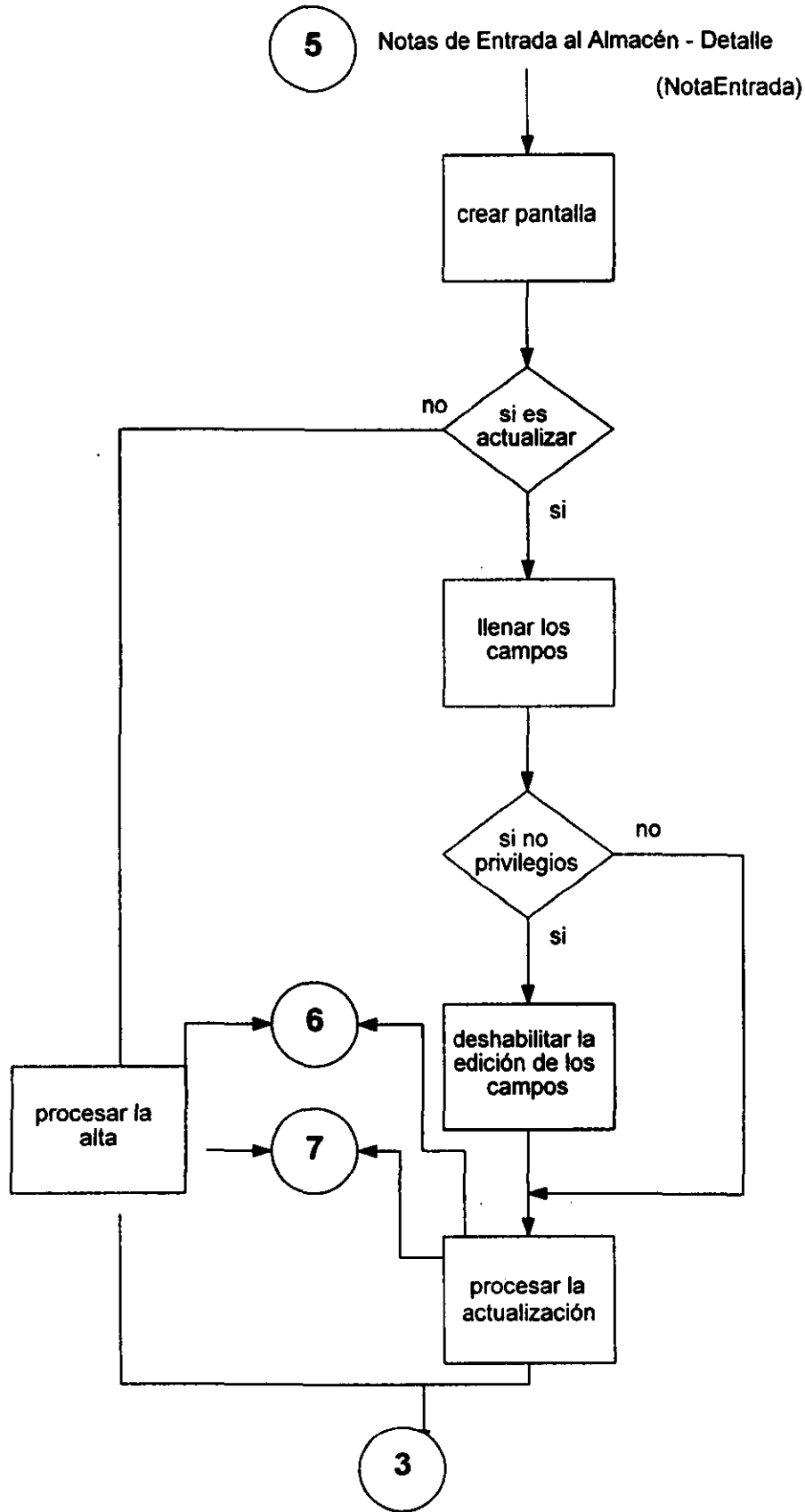
2

Pantalla de Password
(dlgPassword)





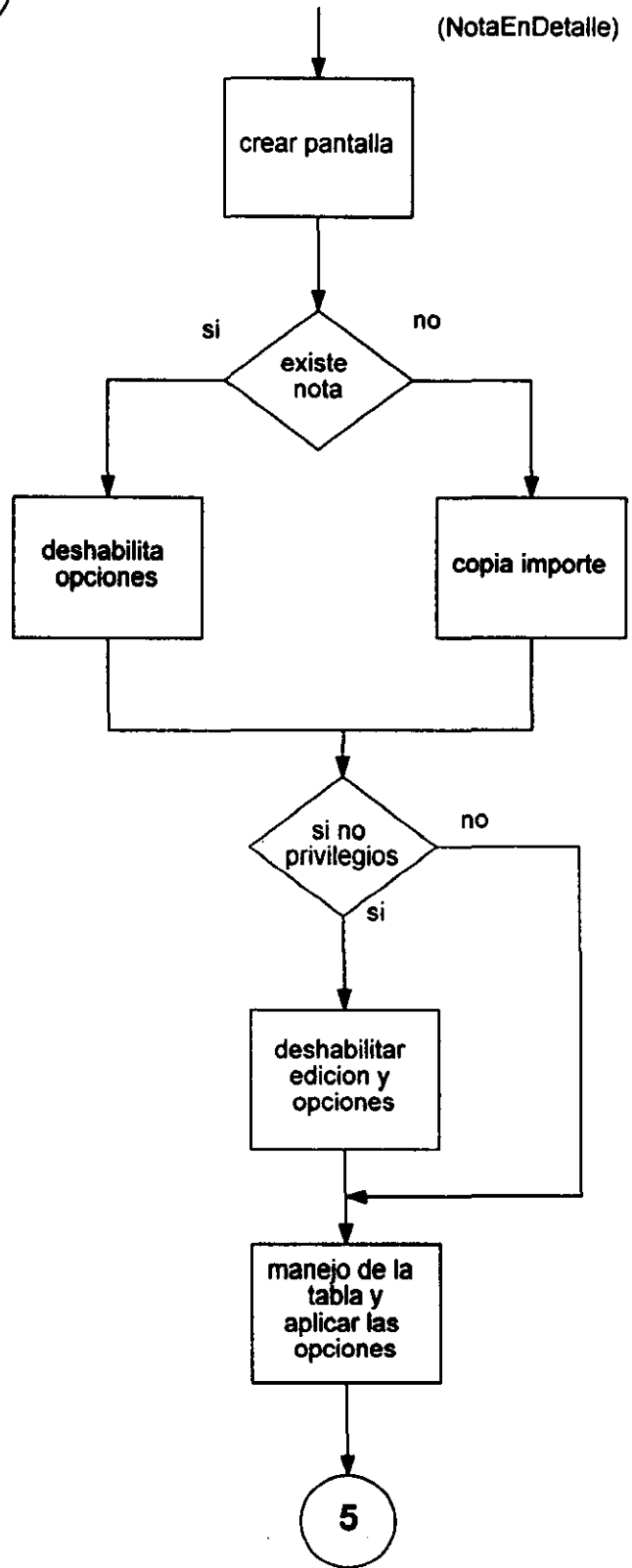




6

Notas de Entrada al Almacén - Notas en Detalle

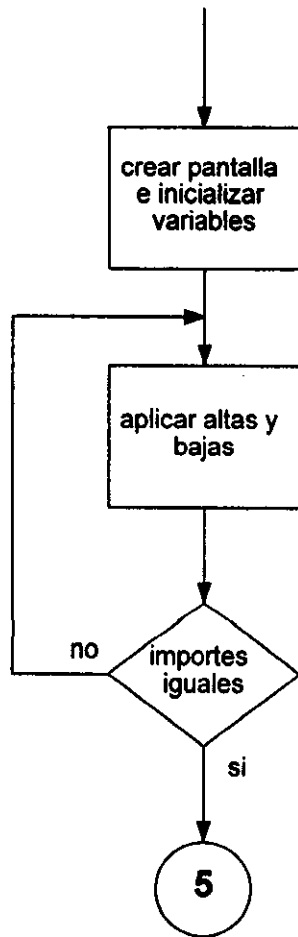
(NotaEnDetalle)



7

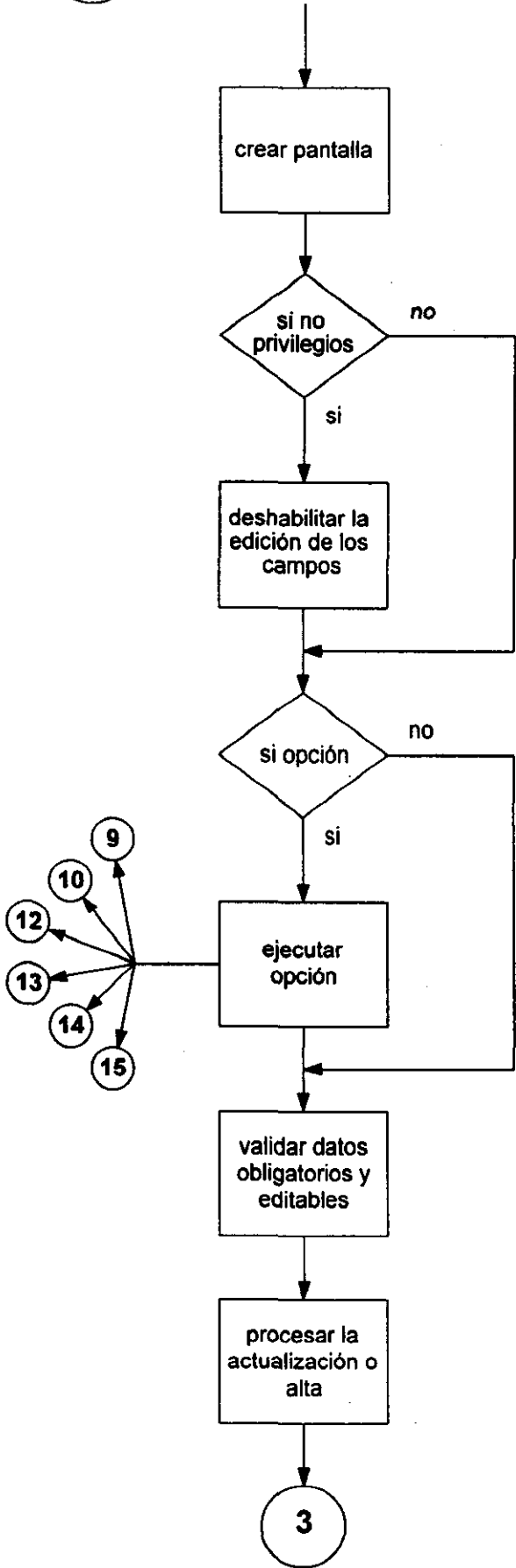
Notas de Entrada al Almacén - Desglose por Proyecto

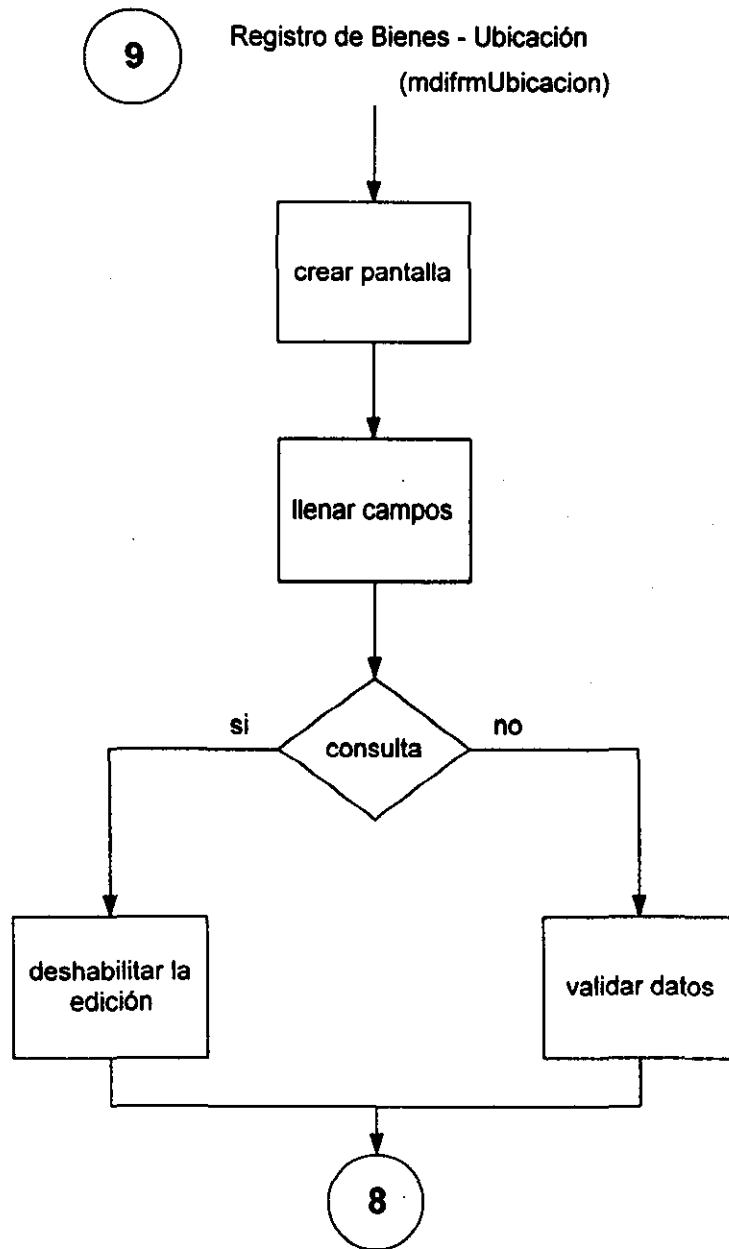
(mdifmProyectonota)

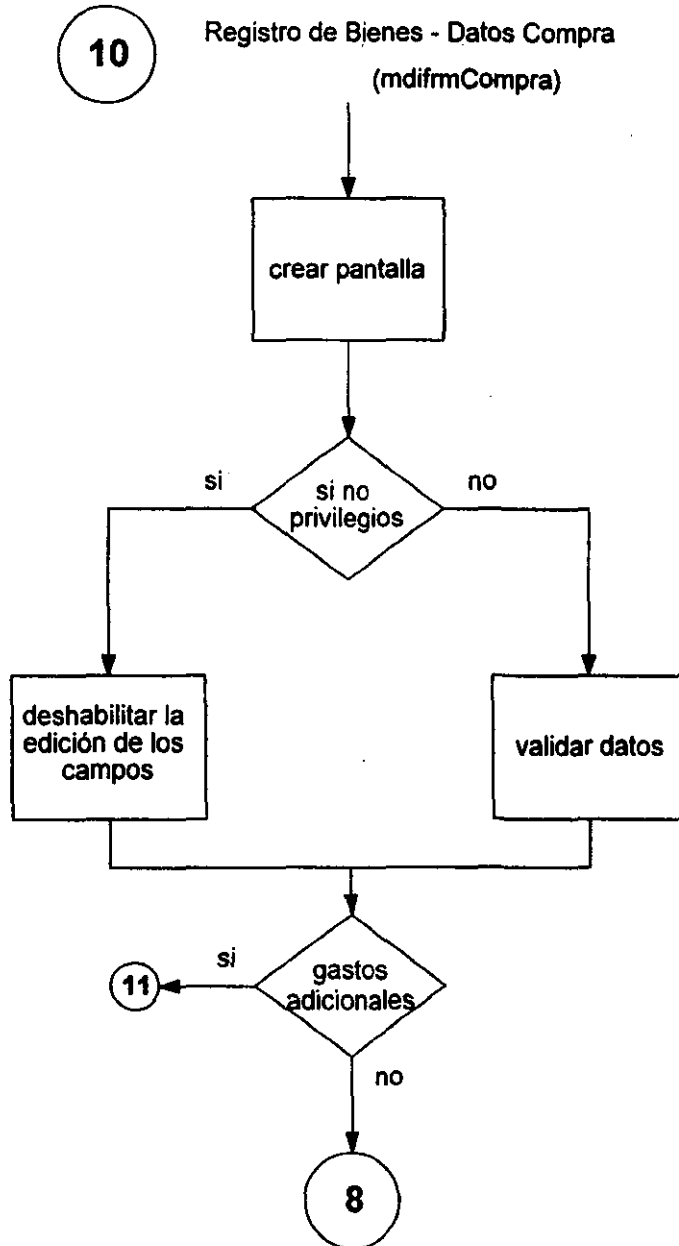


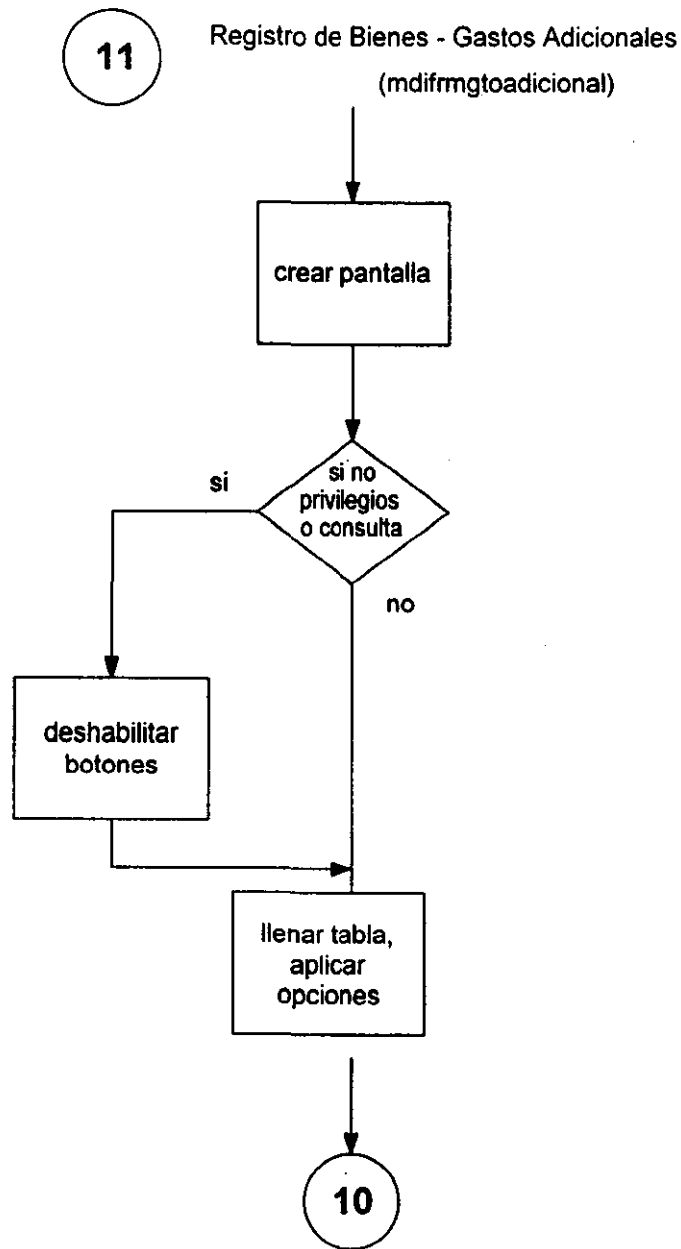
8

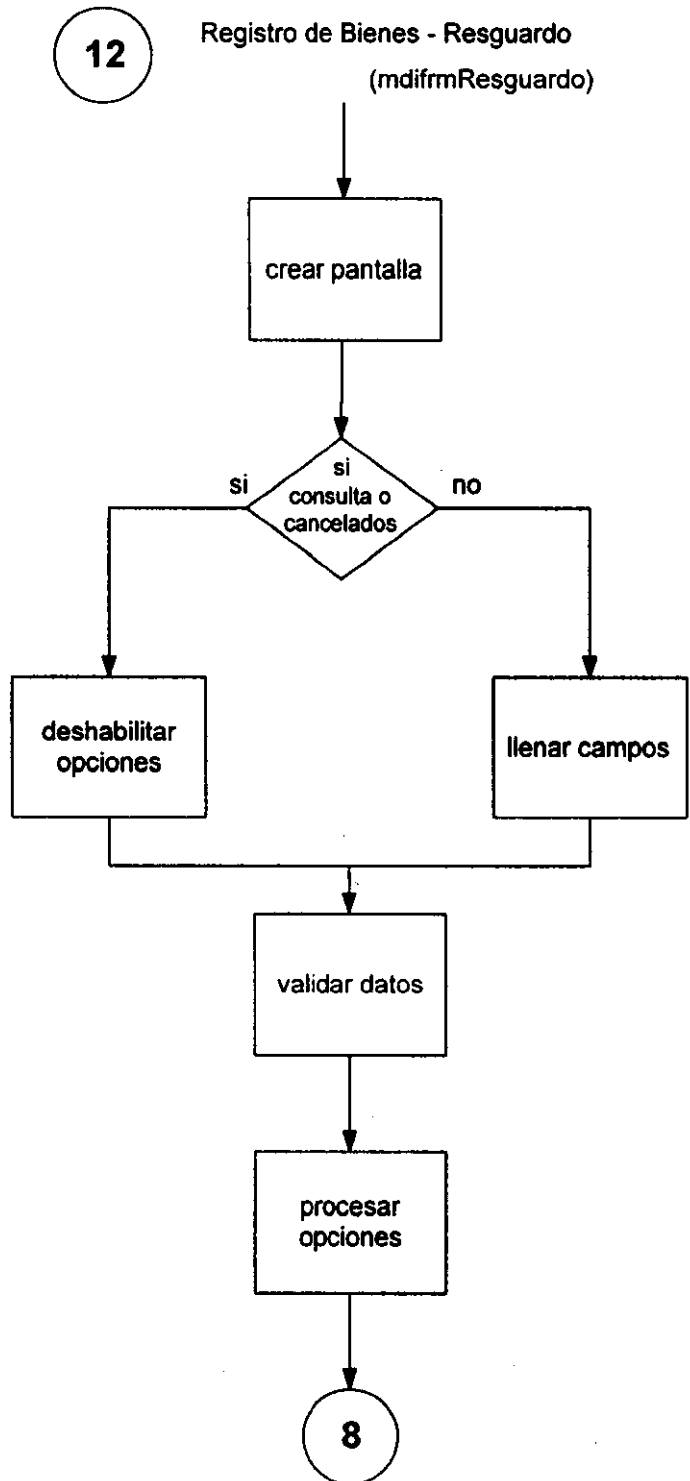
Registro de Bienes - Principal de Captura
(mdifrmGeneral)

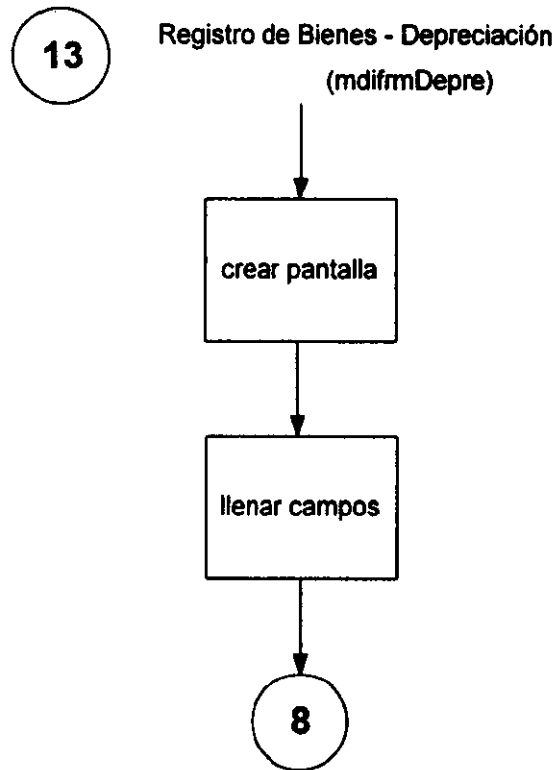


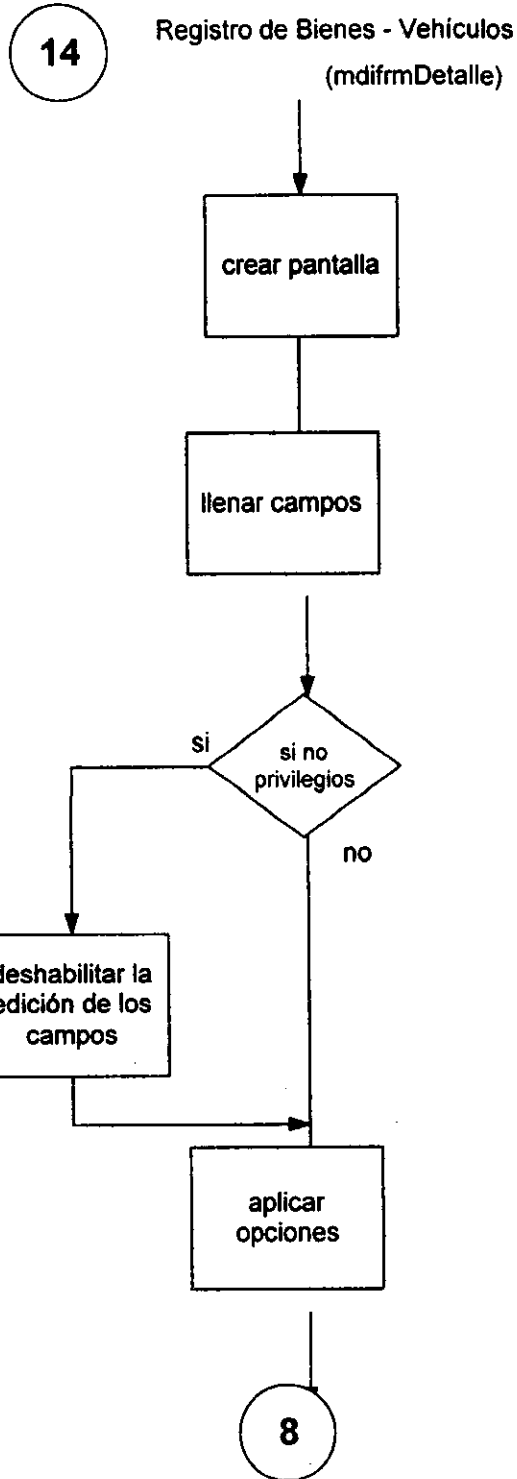












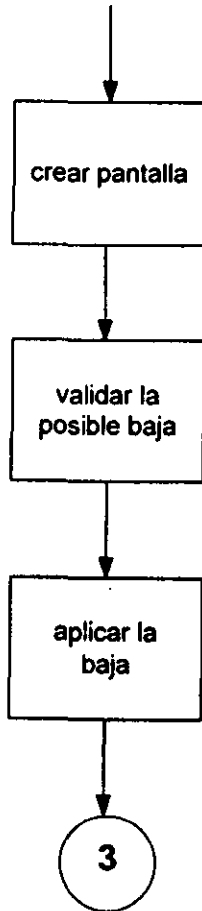
15

Registro de Bienes - Técnicas
(mdifrmCaracteristicas)



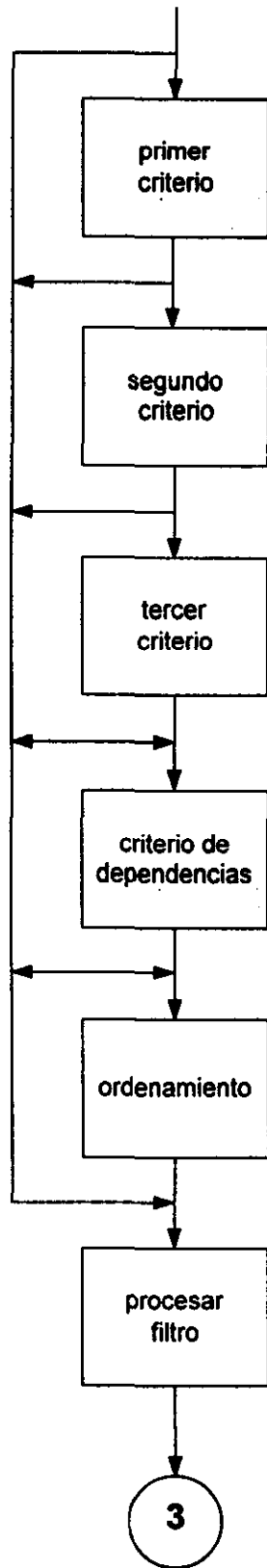
16

Registro de Bienes - Bajas
(mdifrmBajas)



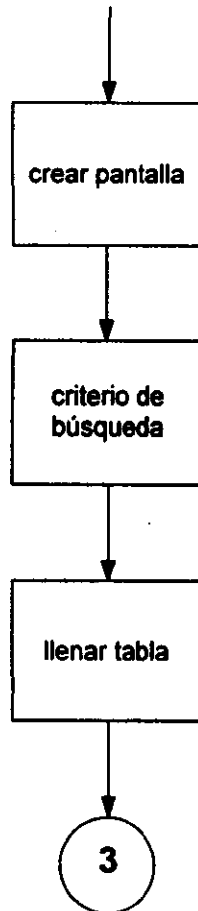
17

Registro de Bienes - Filtro
(digFiltro)



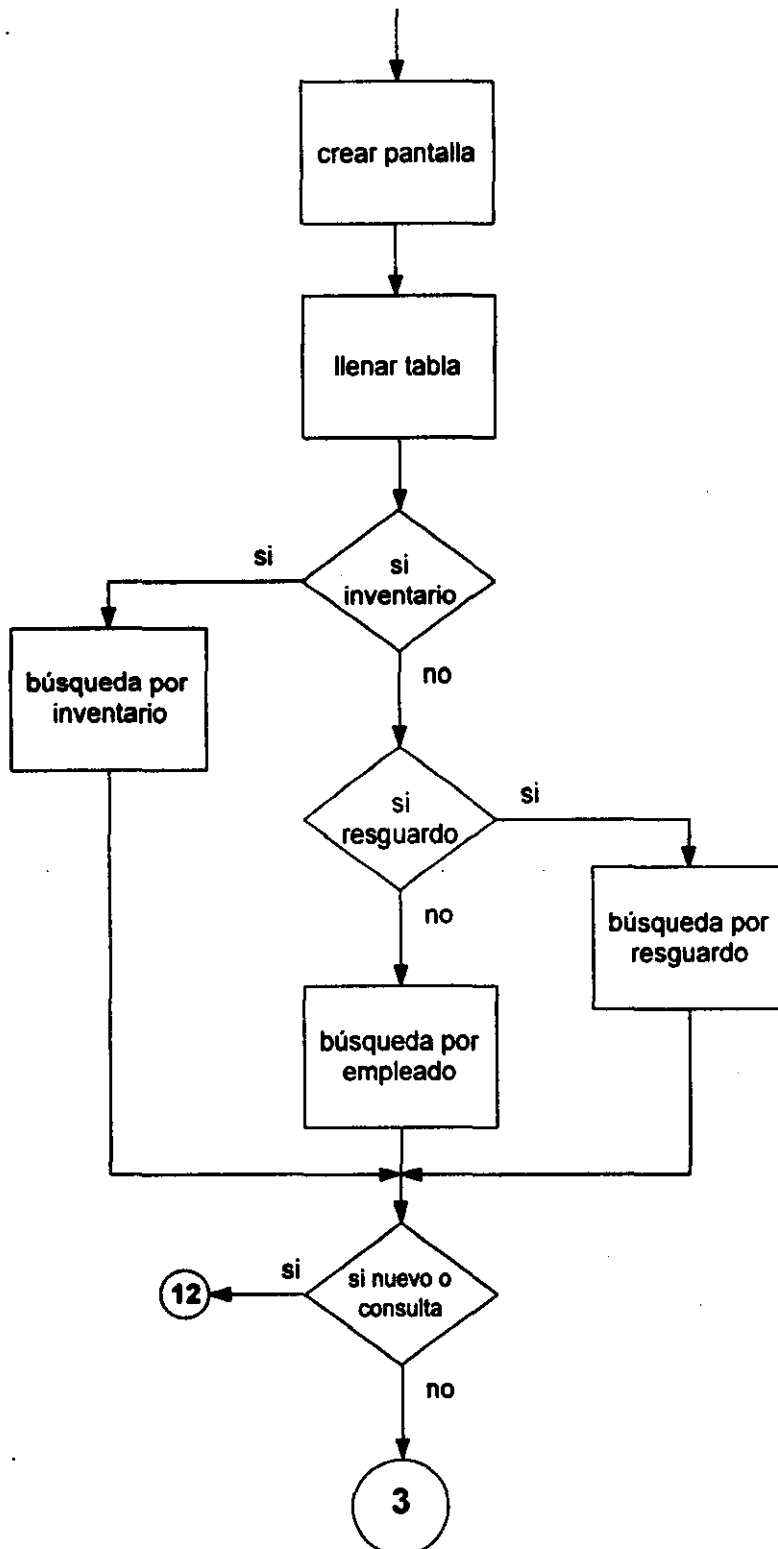
18

Registro de Bienes - Consulta Bajas
(ConsBiBaja)



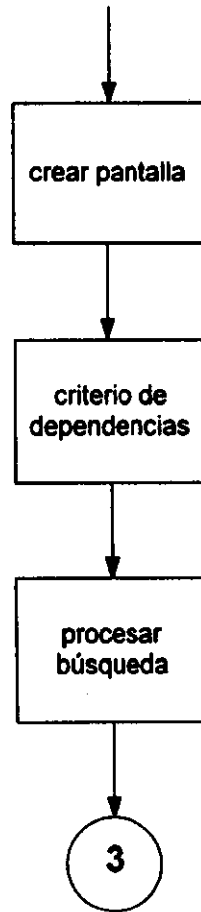
19

Resguardos - Principal de Resguardo
(frmResguardo)



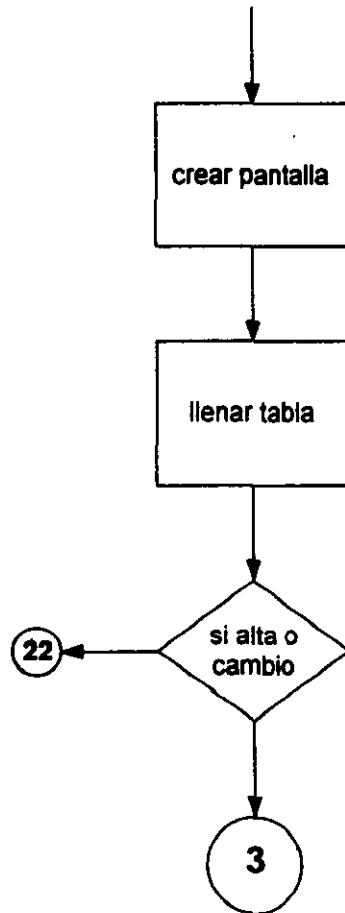
20

Resguardos - Resguardo Dependencia
(dlgResguDepto)



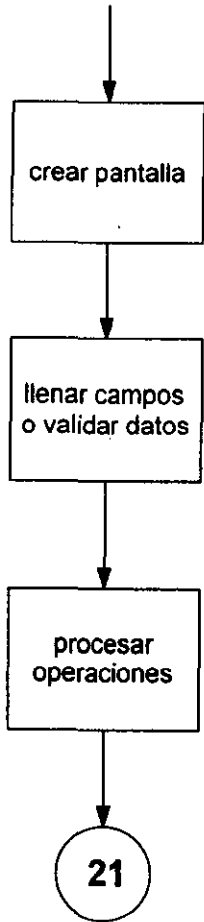
21

Gestión - Principal de Traspasos
(mdifrmGralTraspaso)



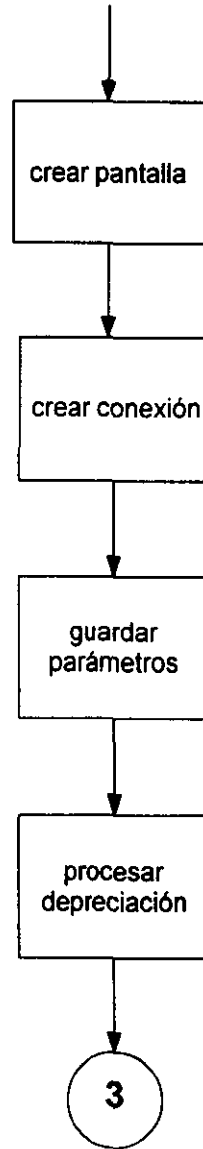
22

Gestión - Traspaso Detalle
(mdifmTraspaso)



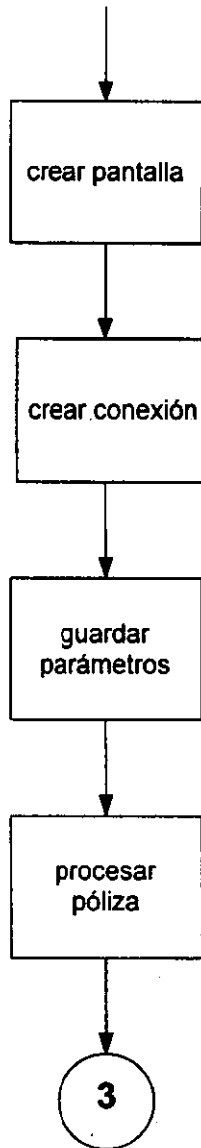
23

Procesos - Depreciación
(dlgDepreciacion)



24

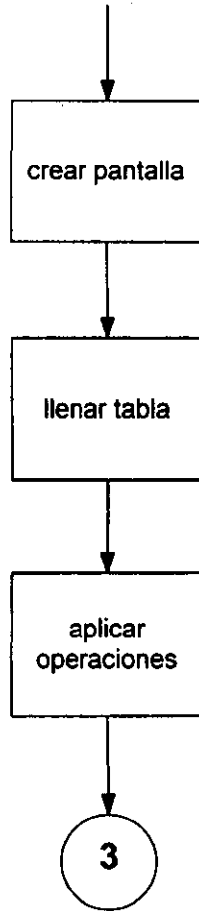
Procesos - Póliza
(dlgPoliza)



25

Utilerías - Usuarios

(frmUsuarios)



CAPITULO 4

4 Implementación

- 4.1 Lenguaje
- 4.2 Algoritmos
- 4.3 Pruebas y Correcciones

La implementación es una etapa dentro de la ingeniería de software que debe tener el soporte previo de un diseño del sistema. Es una etapa en la que los resultados son más tangibles para el usuario final, debido a que la implementación involucra la codificación, las pruebas y las correcciones del sistema.

La forma en que se desarrollará el sistema será en base a los módulos, esto quiere decir que la programación será cumpliendo módulo por módulo y no todos al mismo tiempo, con esto lograremos aislar las pruebas y los errores que puedan surgir en cada módulo, teniendo al final un mejor resultado global.

4.1 Lenguaje

El lenguaje utilizado para la realización de este sistema es SQLWindows clasificado como un lenguaje para generar aplicaciones de proceso de datos. Este lenguaje soporta el fácil manejo de la creación, modificación y borrado de datos. También considera que la integridad de los datos es esencial.

Una introducción de SQLWindows fue dada en el capítulo de fundamentos. Este es un lenguaje de programación orientado a eventos en el cual puede estar incluido todo el soporte de la programación orientada a objetos. Se le conoce como lenguaje orientado a eventos porque el código de la aplicación se ejecuta en base a una secuencia de acciones que serán proporcionadas desde un ambiente gráfico.

4.2 Algoritmos

Los algoritmos que definen la funcionalidad del sistema son los presentados a continuación. Estos algoritmos tienen su fundamento en el diseño realizado para el sistema, contemplan los módulos requeridos y las actividades realizadas en cada uno de ellos, así como la seguridad que es aplicada en el sistema.

INICIO

Pantalla Inicio (dlgINICIO)

Al crearse la pantalla

- Encender CATBOTONES TRUE
 - Leer "mcaf.ini" para traer la base de dato, la cuenta y contraseña del sistema
 - Conectarse a la base de datos
 - Si se conecta
 - Agregar dos Handle a la base de datos
 - Deshabilitar el autocommit de los Handle
 - Sino marcar el Error y Salir
 - Destruir la pantalla de inicio y crear la pantalla del Password
-

Pantalla Password (dlgPassword)

Campo Usuario

- Al desplegarse el combo
 - Asignar la selección a la variable del combo
- Al clic
 - Asignar la selección a la variable del combo
- Al perder el focus
 - Si el usuario fue escrito
 - Traer su clave
 - Sino fue escrito
 - Terminar el Programa

Botón Continuar

- Al clic
 - Traer la contraseña de la base de datos
 - Si son diferentes
 - Marcar Error
 - Salir
 - Sino
 - Encender constantes relacionadas con la dependencia de la cuenta
 - Si cuenta es administrados
 - Encender constantes relacionadas con la dependencia
 - Enciende la constante de seguridad de accesos en el sistema
 - Destruir la pantalla de password
-

Pantalla Principal (mdiMAF)

Encender variables del sistema
Borrar los resguardos de los bienes dados de baja
Desplegar el Menú Principal
Si se realizó una selección
Ejecutarla

NOTAS DE ENTRADA

Pantalla Notas de Entrada al Almacén General (mdifrmNotas)

Al crearse la pantalla

 Si no tiene privilegios

 Deshabilita las bajas y las altas

 Activar las actualizaciones

 Llena la tabla de notas de entrada

Tabla de Notas de Entrada

 Al doble clic sobre un renglón

 Crear la pantalla de Nota de Entrada con parámetro de actualización

Botón Buscar

 Al clic

 Llenar la tabla con las ocurrencias solicitadas en el filtro

Botón Borrar

 Al clic

 Borrar los registros correspondientes a la nota seleccionada en todas las

tablas

Botón Nuevo

 Al clic

 Crear la pantalla de Nota de Entrada con parámetro de nuevo

Botón Salir

 Al clic

 Destruir la pantalla actual

Pantalla Notas de Entrada al Almacén•Detalle (NotaEntrada)

Al crearse la pantalla

 Si es actualización

 Llenar los campos

 Deshabilitar la edición en algunos campos no modificables

 Si no tiene privilegios

 Deshabilitar la edición en todos los campos

Campo Nota de Entrada

 Al perder el focus

 Si existe la Nota de Entrada

 Marcar Error

Campo Fecha Nota

 Al tener el focus

 Traer la fecha del día

 Al perder el focus

 Si no valida la Fecha

 Marcar Error

Campo Factura

 Al perder el focus

 Si no valida caracteres

 Marcar Error

Campo Fecha Factura

 Al perder el focus

 Si Campo Factura vacío

 Encender el focus en Campo Factura

 Sino

 Si no valida Alguna Fecha

 Marcar Error

Campo Importe

 Al perder el focus

 Si Importe es cero

 Marcar Error

Botón Aplicar

 Al clic

 Si existen campos vacios

 Marcar Error

 Sino

 Si es Alta

 Insertar

 Sino

 Actualizar

Botón Nota Detalle

 Al clic

 Ocultar las pantallas de notas

 Crear la pantalla de Nota en Detalle

Botón Imprime Nota

Al clic

Crea la pantalla de Firma

Botón Desglose Nota

Al clic

Crea la pantalla de Desglose por Proyecto

Botón Salir

Al clic

Destruir la pantalla actual

Mostrar la pantalla anterior

Pantalla Notas de Entrada al Almacén•Nota en Detalle (NotaEnDetalle)

- Al crearse la pantalla
 - Si existe la nota
 - Ocultar opciones
 - Sino
 - Copiar importe
 - Si no tiene privilegios
 - Deshabilitar la edición y opciones
- Tabla Movimiento Nota
 - A la llamada del usuario
 - Llena la tabla
 - Manda mensaje al campo Acumulado
 - Columna Consecutivo
 - Si no existe valor
 - No continuar
 - Columna Surtidas
 - Si no existe valor
 - Marcar Error
- Campo Acumulado
 - A la llamada del usuario
 - Suma el importe
- Botón Aplicar
 - Al clic
 - Si es insertar
 - Insertar
- Botón Borrar
 - Al clic
 - Si Nota asignada a un Bien
 - Marcar error
 - Sino
 - Borrar
- Botón Agregar
 - Al clic
 - Insertar un renglón en la tabla
- Botón Salir
 - Al clic
 - Si existe la nota
 - Regresar valores a la pantalla anterior
 - Sino
 - Si Importes Diferentes
 - Marcar Error
 - Sino
 - Destruir la pantalla
 - Restaurar la pantalla anterior

Pantalla Nota de Entrada al Almacén•Desglose por Proyecto (mdifrmProyectornota)**Al crearse la pantalla**

Encender variables

Mandar mensaje a la Tabla Proyectos

Campo Importe

Al crearse el Campo

Copiar el importe

Campo Acumulado

A la llamada del usuario

Suma importe

Tabla Proyectos

A la llamada del usuario

Llena la Tabla

Mandar mensaje al Campo Acumula

Botón Aplicar

Al clic

Si Alta

Insertar

Si Baja

Borrar

Actualizar

Mandar mensaje a la Tabla Proyectos

Botón Borrar

Al clic

Marcar renglones para borrado

Botón Nuevo

Al clic

Insertar renglón en la Tabla Proyectos

Botón Salir

Al clic

Si importes diferentes

Marcar error

Sino

Destruir ventana

REGISTRO DE BIENES

 Pantalla Registro de Bienes•Principal de Captura (mdifrmGeneral)

Al crearse la pantalla

Encender variables

Mandar mensaje a la principal de captura (limpiar)

Si Cambio

Cargar datos

Si Consulta

Deshabilitar campos

Sino

Si Tiene Permiso

Habilita campos

Sino

Deshabilita campos

A la llamada del usuario (limpiar)

Limpiar Campos

A la llamada del usuario (validar)

Si (no. Inventario, status, partida, determinante) vacios

Marcar error

Sino

Si Botón Actualización

Return true

Sino

Si Botón Ubicación

Crea la pantalla de Ubicación

Si Botón Compra

Crea la pantalla de Datos Compra

Si Botón Resguardo

Crea la pantalla de Resguardo

Si Botón Depreciación

Crea la pantalla de Depreciación-Revaluación

Si Botón Vehículo

Crea la pantalla de Detalle

Si Botón Características

Crea la pantalla de Características Técnicas

A la llamada del usuario (genérica)

Concatena (nombre determinante, marca, modelo, serie)

Al destruirse la pantalla

Habilita los Botones

Campo Inventario

Al crearse

Si Alta

Habilitar el campo

- Al editarse
 - Encender variable
 - Al perder el focus
 - Si diferente de blancos
 - Rellenar con ceros
 - Si datos no salvados y se edito campo
 - Si existe inventario
 - Marcar error
- Campo Status
 - Al desplegarse el combo
 - Llenar el combo
 - Al perder el focus
 - Si no tiene privilegios sobre el tipo de status
 - Marcar error
- Campo Serie
 - Al perder el focus
 - Mandar mensaje a la principal de captura (genérica)
- Campo Actualización
 - Al crearse
 - Tomar fecha del sistema
- Campo Nota Entrada
 - Al perder el focus
 - Si no existe
 - Marcar error
 - Sino
 - Crea la pantalla de Nota en Detalle
- Campo Partida
 - Al desplegarse el combo
 - Llenar el combo
 - Al perder el focus
 - Si no existe
 - Marcar error
 - Sino
 - Si Partida de transporte
 - Habilita Botón Vehículo
 - Sino
 - Deshabilita Botón Vehículo
- Campo Determinante
 - Al desplegarse el combo
 - Llenar el combo
 - Traer el CADMS
 - Al perder el focus
 - Si no existe
 - Marcar error
 - Sino
 - Mandar mensaje a la principal de captura (genérica)

Campo Marca
 Al perder el focus
 Si Alta
 Mandar mensaje a la principal de captura (genérica)

Campo Modelo
 Al perder el focus
 Si Alta
 Mandar mensaje a la principal de captura (genérica)

Campo Descripción
 Al tener el focus
 Concatena (nombre determinante, marca, modelo, serie)

Botón Agrega
 Al clic
 Habilita Botones
 Mandar mensaje a la principal de captura (limpiar)

Botón Ubicación
 Al clic
 Mandar mensaje a la principal de captura (validar)

Botón Compra
 Al clic
 Mandar mensaje a la principal de captura (validar)

Botón Resguardo
 Al clic
 Mandar mensaje a la principal de captura (validar)

Botón Depreciación
 Al clic
 Mandar mensaje a la principal de captura (validar)

Botón Vehículo
 Al clic
 Mandar mensaje a la principal de captura (validar)

Botón Características
 Al clic
 Mandar mensaje a la principal de captura (validar)

Botón Actualización
 Al clic
 Mandar mensaje a la principal de captura (validar)
 Si datos correctos
 Si no existe dependencia
 Marcar error
 Sino
 Si Alta
 Insertar datos
 Sino
 Actualizar datos

Si actualizar
 Mandar mensaje al Botón Salir
 Sino
 Deshabilita Botones
 Botón Salir
 Al clic
 Si datos no salvados
 Si no Consulta
 Mandar advertencia
 Sino
 Destruir ventana
 Sino
 Destruir ventana
 Si pantalla Varios activa
 Mandar mensaje a pantalla Varios-Botón actualización

Pantalla Registro de Bienes•Ubicación (mdifrmUbicacion)

Al crearse
 Llenar los campos
 Si Consulta
 Deshabilita campos
 Campo Dependencia
 Al desplegase el combo
 Llena combo
 Al perder el focus
 Si no existe
 Marcar error
 Campo Edificio
 Al desplegarse el combo
 Si existe Zona
 Llena combo
 Botón Salir
 Al clic
 Si pantalla Varios activa
 Ocultar pantalla Varios
 Si no existe Dependencia
 Mandar error
 Sino
 Destruir pantalla
 Mostrar pantalla Principal de Captura

Pantalla Registro de Bienes•Datos Compra (mdifrmCompra)

Al crearse

Si no tiene privilegios

Deshabilitar campo Valor Compra

Llena campos

Campo Valor Compra

Al perder el focus

Actualizar Valor Total

Campo Proveedor

Al desplegase el combo

Llena combo

Botón Gasto Adicional

Al clic

Ocultar pantalla Principal de Captura y Datos de Compra

Crear la pantalla de Gastos Adicionales

Botón Salir

Al clic

Destruye ventana

Si pantalla Varios activa

Ocultar pantalla Varios

Mostrar pantalla Principal de Captura

Pantalla Registro de Bienes Gastos Adicionales (mdifrmgtoadicional)

Al crearse

Si Consulta o no tiene privilegios

Ocultar Botones

Tabla Gastos

Al crearse

Mandar mensaje a tabla Gastos

A la llamada del usuario

Llena tabla

Columna Auxiliar

Si no valido

Mandar error

Botón Insertar

Al clic

Insertar renglón en la Tabla Gastos

Botón Borrar

Al clic

Borra renglones marcados

Actualiza campos

Botón Aplicar

Al clic

Insertar datos

Actualizar campos

Mandar mensaje a la tabla Gastos

Botón Salir

Al clic

Destruye ventana

Si pantalla Varios activa

Ocultar pantalla Varios

Mostrar pantalla Datos Compra

 Pantalla Registro de Bienes•Resguardo (mdifrmResguardo)

Al crearse

Encender variables

Llena campos

Si Actualización

Mostrar Botones y Tabla

Mandar mensaje a pantalla Resguardo (llenar)

Si Consulta

Mandar mensaje a pantalla Resguardo (deshabilitar)

Si Cancelados

Ocultar Botones

Sino

Mostrar Botones

A la llamada del usuario (deshabilitar)

Deshabilita campos

Llena datos

A la llamada del usuario (llenar)

Llena tabla Resguardo

Campo Resguardo

Al perder el Focus

Si existe Resguardo

Llena campos

Si tiene Bienes

Si Fecha Fin cumplida

Mandar aviso cancelado

Sino

Mandar aviso registrado

Si respuesta si

Limpiar variables

Sino

Si pantalla varios activa

Mandar mensaje a pantalla Resguardo

(llenar)

Mandar mensaje a pantalla Resguardo

(deshabilitar)

Campo Fecha Fin

Al perder el Focus

Si fecha no valida

Mandar error

Campo Empleado

Al desplegase el combo

Llena combo

Al perder el focus

Si clave del empleado

Trae nombre del empleado

```

Si no existe empleado
    Marcar error
Sino
    Llena campos
    Si existen resguardos vigentes
        Mandar advertencia
    Si respuesta si
        Limpiar variables
    Sino
        Si pantalla varios activa
            Mandar advertencia
        Sino
            Mandar mensaje a pantalla Resguardo (llenar)
    
```

Tabla Resguardo

Columna Inventario

```

Si existe inventario sin resguardo
    Validar inventario
Sino existe
    Marcar error
Si existe resguardo
    Marcar error
    
```

Columna Descripción

```

Al tener el focus
    Si se agrega
        Insertar
    
```

Botón Agregar

```

Al clic
    Dar el focus a Tabla resguardo
    
```

Botón Borrar

```

Al clic
    Borrar
    
```

Botón Salir

```

Al clic
    Destruye ventana
    Si pantalla Varios activa
        Ocultar pantalla Varios
    Si pantalla Principal de Resguardos activa
        Mostrar pantalla
    Else
        Mostrar pantalla Principal de Captura
    
```

Pantalla Registro de Bienes•Depreciación (mdifrmDepre)

Al crearse
 Llenar campos
 A la llamada del usuario
 Llena año de compra
 Botón Salir
 Al clic
 Destruye ventana

Pantalla Registro de Bienes•Vehiculos (mdifrmDetalle)

Al crearse
 Si existe partida
 Llenar campos
 Si no tiene privilegios
 Deshabilitar la edición en todos los campos
 Campo Uso Auto
 Al desplegarse el combo
 Llena combo
 Botón Salir
 Al clic
 Si campos vacíos
 Marcar error
 Sino
 Actualizar variables
 Destruir ventana

Pantalla Registro de Bienes•Técnicas (mdifrmCaracteristicas)**Al crearse****Encender variables****Tabla Componente****Al crearse****Llenar tabla****Al editarse****Llenar columna de descripción****Al desplegarse****Llenar columna de descripción****Al perder el focus****Traer la descripción****Botón Agregar****Al clic****Insertar renglón****Botón Borrar****Al clic****Si confirma****Borrar****Botón Salir****Al clic****Si se inserto****Si no existe inventario****Insertar inventario****Insertar componente****Actualizar componentes****Destruir ventana**

Pantalla Registro de Bienes•Bajas (mdifrmBajas)

- Al crearse
 - Inicializa variables
- Al destruirse
 - Habilita Botones
- Campo Inventario
 - Al perder el focus
 - Si existe inventario escrito
 - Validar inventario
 - Si no existe
 - Marcar error
 - Sino
 - Si existe como baja
 - Marcar error
 - Sino
 - Deshabilitar edición

- Campo Fecha
 - Al tener el focus
 - Si no existe fecha
 - Tomar la del sistema

- Campo Motivo
 - Al clic
 - Traer motivo
 - Al desplegarse
 - Llenar combo
 - Al perder el focus
 - Traer motivo

- Botón Aplicar
 - Al clic
 - Aplicar baja

- Botón Salir
 - Al clic
 - Habilitar Botones
 - Si baja inconclusa
 - Mandar advertencia
 - Sino
 - Destruir ventana
-

Pantalla Registro de Bienes•Filtro (dlgFiltro)**Al crearse** **Inicializa campos** **Inicializa variables****Al destruirse** **Habilita Botones****Campo Opción1** **Al perder el focus** **Actualizar variable de acuerdo a elección1****Campo Operador1** **Al perder el focus** **Actualizar variable de acuerdo a elección1****Campo Valor1** **Al desplegarse** **Llenar campo en base a opción1** **Al perder el focus** **Actualizar variable de acuerdo a elección1****Campo Valor1a** **Al perder el focus** **Actualizar variable****Campo Conjunción1** **Al clic** **Si verdadero** **Actualizar variable** **Sino** **Deshabilitar campos****Campo Disyunción1** **Al clic** **Si verdadero** **Actualizar variable** **Sino** **Deshabilitar campos****Campo Operador2** **Al perder el focus** **Actualizar variable de acuerdo a elección2****Campo Operador2** **Al perder el focus** **Actualizar variable de acuerdo a elección2****Campo Valor2** **Al desplegarse** **Llenar campo en base a opción2** **Al perder el focus** **Actualizar variable de acuerdo a elección2****Campo Valor2a** **Al perder el focus** **Actualizar variable**

Campo Conjunción2

Al clic

Si verdadero

Actualizar variable

Sino

Deshabilitar campos

Campo Disyunción2

Al clic

Si verdadero

Actualizar variable

Sino

Deshabilitar campos

Campo Opción3

Al perder el focus

Actualizar variable de acuerdo a elección3

Campo Operador3

Al perder el focus

Actualizar variable de acuerdo a elección3

Campo Valor3

Al desplegarse

Llenar campo en base a opción3

Al perder el focus

Actualizar variable de acuerdo a elección3

Campo Valor3a

Al perder el focus

Actualizar variable

Campo Dependencia

Al editarse

Validar permisos

Al desplegarse

Validar permisos

Llenar el combo

Al clic

Traer dependencia

Al perder el focus

Actualizar variable

Campo Ordena

Al perder el focus

Actualizar variable de acuerdo a elección de ordenamiento

Campo Subdirección

Al clic

Si verdadero

Actualizar variable

Campo Raiz

Al clic

Si verdadero

Actualiza variable

Botón Filtrar

Concatenar variables

Crear la pantalla de Reporte Filtro

Destruir ventana

Botón Salir

Destruir ventana

Pantalla Registro de Bienes•Consulta Bajas (ConsBiBaja)**Campo Inventario**

Al editarse

Llenar tabla

Tabla Bajas

Al crearse

Llenar tabla

Botón Salir

Al clic

Destruir la ventana

RESGUARDOS

Pantalla Resguardos•Principal de Resguardo (frmResguardo)

Al crearse

 Si Baja

 Deshabilita Botones

 Sino

 Habilita Botones

Al crearse completamente

 Mandar mensaje al Botón Actualizar

Al destruirse

 Habilitar Botones de pantalla Principal

Tabla Resguardos

Al doble clic

 Actualizar variables

 Crear la pantalla de Resguardo

Campo Resguardo

Al editarse

 Limpia campos

Campo Inventario

Al editarse

 Limpia campos

Campo Empleado

Al editarse

 Limpia campos

Botón Buscar

Al clic

 Si campo Resguardo no vacío

 Busca por resguardo

 Si campo inventario no vacío

 Busca por inventario

 Si campo empleado no vacío

 Busca por empleado

Botón Actualizar

Al clic

 Limpia campos

 Mandar mensaje al botón Actualizar

A la llamada del usuario

 Llenar la tabla Resguardos

Botón Imprime

Al clic

 Crear la pantalla de Reporte Resguardo

Botón Nuevo

Al clic

 Limpia variables

 Crear la pantalla de resguardo

Botón Salir
Al clic
Destruye ventana

Pantalla Resguardos• Resguardo Dependencia (dlgResguDepto)

Campo Dependencia
Al desplegarse
Llenar combo
Al perder el focus
Si no existe
Marcar error

Botón Imprime
Al clic
Crea la pantalla de Reporte Resguardo
Destruye ventana

Botón salir
Al clic
Destruye ventana

GESTION

Pantalla Gestión Principal de Traspasos (mdifrmGralTraspaso)

Al crearse

Mandar mensaje a tabla Traspaso

Tabla Traspaso

A la llamada del usuario

Llenar tabla

Al doble clic

Crear la pantalla de Traspaso Detalle

Botón Nuevo

Al clic

Crear la pantalla de Traspaso Detalle

Botón Salir

Al clic

Destruye ventana

Pantalla Gestión* Traspaso Detalle (mdifrmTraspaso)

- Al crearse
 - Inicializar variables
 - Si actualización
 - Llena campos
- Campo Interno
 - Al clic
 - Actualizar variable
- Campo Externo
 - Al clic
 - Actualizar variable
- Campo DepRemite
 - Al desplegarse
 - Llenar combo
 - Al clic
 - Traer dependencia
 - Al perder el focus
 - Si no existe dependencia
 - Marcar error
- Campo DepRecibe
 - Al desplegarse
 - Llenar combo
 - Al clic
 - Traer dependencia
 - Al perder el focus
 - Si DepRemite igual DepRemite
 - Marcar error
 - Sino
 - Si no existe dependencia
 - Marcar error
- Campo Zona
 - Al desplegarse
 - Llenar combo
 - Al perder el focus
 - Traer zona
- Campo Edificio
 - Al desplegarse
 - Si existe zona
 - Llenar combo
 - Al perder el focus
 - Traer edificio

Botón Nuevo
 Al clic
 Si existen datos
 Llenar tabla Inventarios
 Sino
 marcar error

Botón Borrar
 Al clic
 Si no esta autorizado
 Borrar
 Sino
 Mandar error

Tabla Inventarios
 A la llamada de usuario
 Llenar tabla
 Columna Inventario
 Al perder el focus
 Si no existe inventario
 Marcar error

Botón Aplicar
 Al clic
 Si alta
 Genera número de nota
 Insertar
 Si baja
 Borrar
 Actualizar
 Mandar mensaje a la tabla Inventarios

Botón Autoriza
 Al clic
 Si no existe autorización
 Mandar mensaje al Botón Aplicar
 Sino
 Mandar error

Botón Imprime
 Al clic
 Si es dependencias
 Crear la pantalla de reporte Traspaso1
 Si es subdirecciones
 Crear la pantalla de reporte Traspaso2

Botón Salir
 Al clic
 Destruye ventana
 Mandar mensaje a la tabla Traspaso

PROCESOS

Pantalla Procesos• Depreciación (dlgDepreciacion)

Al crearse

 Conexión nueva

Al destruirse

 Terminar conexión

Botón Ejecutar

 Al clic

 Ejecutar proceso

 Destruir ventana

Pantalla Procesos• Póliza (dlgPoliza)

Al crearse

 Conexión nueva

Al destruirse

 Terminar conexión

Botón Ejecutar

 Al clic

 Ejecutar proceso

 Destruir ventana

 Crear pantalla de reporte Póliza

UTILERIAS

Pantalla Utilerías• Usuarios (frmUsuarios)

Al crearse

Mandar mensaje a la forma usuarios

A la llamada del usuario

Llenar la tabla Usuarios

Al destruirse

Habilita Botones

Botón Nuevo

Al clic

Insertar registro

Botón Borrar

Al clic

Marcar registro

Botón Aplicar

Al clic

Borrar

Insertar

Actualizar

Botón Salir

Al clic

Destruye ventana

4.3 Pruebas y Correcciones

La etapa de pruebas y correcciones trabaja en paralelo con la implementación del código, aunque en realidad, nos referimos a esta etapa como las pruebas y correcciones que se realizan cuando se empiezan a trabajar datos reales.

El proceso de probar y corregir un sistema empieza desde el momento de la generación del código. Al irse desarrollando un módulo, se realizan pruebas de funcionamiento con datos irreales, para detectar errores y poder hacer las correcciones necesarias. La práctica de esto es necesaria para eliminar la gran mayoría de los errores, pero cuando se toman datos reales, puede suceder que su comportamiento no sea el adecuado o varíe con relación a lo requerido; cuando sucede esto, se verifica si el error fue una necesidad o un concepto mal considerado, si se trata de un error en el código o si es una nueva requisición, en este último caso la corrección no es realizada.

Otras pruebas importantes son las que realizan los usuarios finales del sistema. Aún cuando en un sistema se crea una interfaz amigable, los usuarios finales, que en su mayoría no tienen mucha cultura informática, son la prueba más importante y fuerte del sistema. Estos usuarios saben como debe ser el flujo de la información, si la presentación de los datos es la adecuada y si en realidad se están optimizando las actividades. En este tipo de pruebas con usuarios finales, cabe mencionar que el hecho de no contar con cultura informática, provoca que surjan casos muy especiales no contemplados en el diseño, estos casos, de existir, se deben integrar en el sistema para evitar problemas posteriores.

Cuando los datos reales han descubierto y corregido errores, entonces se tiene la certeza de que el sistema ya está cumpliendo con parte de los objetivos. Para cumplir con la otra parte de estos objetivos, se aplica una prueba de volumen de datos para ver el comportamiento del sistema; en este volumen se combinan datos reales con datos irreales, con el fin de manejar los datos reales en un volumen cercano al estimado, donde los datos irreales son el complemento. Los errores encontrados con esta prueba van ligados a la

respuesta en tiempo real del sistema, si el sistema en alguna parte disminuye su rendimiento, entonces se tiene que buscar y corregir la causa de este alentamiento.

Otra prueba es con relación a la integridad de los datos, al ocurrir un corte inesperado de energía eléctrica; la prueba consiste en cortar la energía de la PC al momento que se esta trabajando con el sistema y posteriormente verificar la integridad de los datos.

Como última prueba al sistema, se verifica la conexión simultánea de todos los usuarios, para ver el funcionamiento de la base de datos dentro del servidor; los errores buscados en esta prueba se relacionan al manejo de la concurrencia por parte de la base de datos, en caso de existir errores de este tipo el encargado de revisar y corregir el error es el Administrador de la Base de Datos (DBA).

Los resultados del sistema al probarlo con datos reales y usuarios finales, fueron satisfactorios. Hubo problemas en las pantallas que contenían las fechas contables y de inventario, el problema fue resuelto al retomar los conceptos de fecha de compra, fecha de la factura, fecha de la nota de entrada, fecha de recepción y fecha de alta en el sistema. Otro tipo de problemas fue el de ajustar detalles en la presentación de los datos. Y por último, los errores de código, que generaban pérdida de información en pantalla sin dañar la integridad de la base de datos. Existieron casos no contemplados por el sistema, como el de querer abrir dos pantallas idénticas para visualizar dos bienes o revisar los datos de la nota de entrada al estar realizando cambios en los bienes, esto se resolvió con la deshabilitación de Botones en algunas pantallas y en momentos precisos.

Para la prueba de volumen de datos se tuvo el caso de dos consultas muy lentas, en este problema se encontró que los datos solicitados no tenían ningún índice asociado a ellos, por lo que su búsqueda era secuencial, esto se resolvió con la creación de un índice para esos datos.

En la prueba del corte de energía no existieron errores de integridad de los datos. Finalmente, en la prueba de concurrencia, los bloqueos de tablas que realiza el Sistema

Administrador de Bases de Datos, no fueron de consideración para el desempeño del sistema.

CONCLUSIONES

Las conclusiones de este trabajo de tesis, giran alrededor de los puntos tratados en el mismo y de vivencias personales. Al realizar el trabajo se logró analizar, diseñar e implementar el Sistema de Activo Fijo, a la vez, se llegó a unificar los sistemas informáticos de la empresa y dejar una plataforma adecuada, con esto se nota un mejor manejo de la información utilizada en el área Administrativa y Contable del Instituto Mexicano del Petróleo.

La primera conclusión en el sentido de haber analizado, diseñado e implementado el Sistema de Activo Fijo, es que en cada una de estas etapas se han resuelto dificultades. Por ejemplo en el Análisis, organizar la información que se recaba con los usuarios, es muy laborioso, ya que se tiene que ver en donde ir ubicando todo lo que el usuario dijo y por supuesto, no todo lo que comenta el usuario lo piensa con ideas ordenadas. La utilización de formatos para el análisis fue de gran ayuda, pero aún así es mucha información la que se tiene que organizar.

Al diseñar se logró poner en práctica todos esos conocimientos teóricos que se tienen asimilados, conocer conceptos manejados en los DBMS, aprender a usar herramientas especiales (ERWIN) para el diseño y sobre todo haber adquirido la experiencia al lado de gente experimentada en la materia.

En la implementación, la parte de las pruebas, permitió ver la gran variedad de formas en las que puede ser evaluado un software, ya sea por su velocidad, funcionalidad, presentación, uniformidad, etc. También se tuvieron buenos resultados en la optimización de los recursos proporcionados por el lenguaje de programación SQLWindows, al lograr aprovechar la potencia de la programación orientada a objetos aplicada en la programación orientada a eventos.

Una segunda conclusión, refiriéndonos a la unificación de los sistemas informáticos, es que para este objetivo, el sistema cumplió en el sentido de aportar su producción para la integración en la empresa de nuevos sistemas ligados a éste; estamos hablando de sistemas

como el 'Sistema de Control de Direcciones IP', 'Sistema de Control de Equipo de Cómputo' y 'Sistema de Control de Licencias'; estos sistemas contarán con información veraz, confiable e institucional y todo será sobre la misma información, sin necesidad de manejar bases de datos diferentes para los datos que son comunes entre ellos.

Como tercera conclusión, a la finalización de este trabajo se ha dejado una plataforma robusta para los datos, éstos tienen una manera fácil para ser manipulados, una integridad muy confiable, un alto rendimiento en operaciones de entrada y salida, y seguridad para la información.

Ahora que el Instituto Mexicano del Petróleo está actualizado en el ámbito informático, la productividad, para una parte de la administración, mejora; se obtienen resultados en menor tiempo; y es mayor el beneficio en tiempo-costos.

Con la implementación de este sistema en el instituto, la educación informática de los trabajadores, se fomenta más. Al mismo tiempo se fortalece a las herramientas informáticas, como participantes del desarrollo.

Haber realizado este trabajo me ha mostrado que los conocimientos que recibe uno en el transcurso de su formación académica son el comienzo de una formación dentro de un ambiente real de trabajo. El aprender a tratar a las personas de conocimientos en el área y de conocimientos nulos. El manejar problemas reales cuando alguien se resiste al cambio o quiere desacreditar el trabajo realizado. Algo importante es la utilización de una metodología para la realización del trabajo, en la que uno observa que la teoría muestra lo general y la práctica lo particular, es decir, la teoría nos enseña todas las cosas posibles a considerar, y la práctica solo toma lo que se aplica en ese caso.

Como reflexión final y personal me da gran satisfacción el saber que la formación recibida por parte de la UNAM, fue muy amplia, completa y es parte de una base sólida para llegar a un desarrollo pleno como persona y profesionista.

BIBLIOGRAFIA

<http://w3.mor.itesm.mx/~jtorres/Clases.html>

<http://www.hys.com.pe/educanet/informat.html>

<http://galileo.unex.es/apuntes/hti/>

ORACLE Documentation

“Introducción a los sistemas de Bases de Datos”

Date C.

Ed. Addison-Wesley Iberoamerica

1986

“Software Engineering a Practitioner’s Approach”

Pressman Roger

Ed. Mc Graw Hill

1982

“Oracle Programación PL/Sql”

Scott Urman

Ed. Mc Graw Hill

1998

“SQL para Windows”

DESERH

Ed. DESERH

1998

“Programación Orientada a Objetos SQLWindows”

Orientación Académica

Ed. Orientación Académica

1998