

**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE INGENIERIA**



**SISTEMA INTEGRAL DE CONTROL
DE INVENTARIOS PARA ALMACENES**

TESIS PROFESIONAL

QUE PARA OBTENER EL TITULO DE

INGENIERO EN COMPUTACION

PRESENTAN:

**ELIZABETH ARACELI GOMEZ SANSON
ELIZABETH MORALES SOLIS**

200002

**DIRECTOR DE TESIS
ING. SANTIAGO IGOR VALIENTE GOMEZ**





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.

Dedico esta Tesis a:

*En primer lugar a DIOS porque sin el no
hubiera podido ser lo que ahora soy
y tener lo que ahora tengo.*

*Gracias SEÑOR, porque sé que siempre
has estado conmigo en todo momento y
sé que sin tu ayuda no lo hubiera
podido lograr.*

*A mis Padres, porque gracias a su Apoyo,
a su Cariño y a sus Consejos he salido adelante.
En especial a mi Madre porque siempre me
ha sabido guiar en todos los aspectos y porque
sin su cariño y paciencia no hubiera podido
culminar una de mis más grandes Metas.
Gracias Mamita. Te Quiero Mucho.*

*A mis Hermanos que siempre me han
apoyado, tanto moral como económicamente,
(Lourdes, Tere, Mary, Felix, Margarito),
en especial a mi hermano Juan y a mi
hermana Toma.*

*A mi Hermano Juan porque en el ámbito
profesional es un ejemplo a seguir y porque
me ha apoyado y ayudado siempre.*

*A mi Hermana Toma, por su cariño, su apoyo
y sus consejos que desde niña siempre me ha
dado y porque siempre he podido contar con ella
en todo momento.*

*A todos **mis Sobrinos**, en especial a aquellos que siempre han estado conmigo desde niña (Vero, Norma), gracias por compartir tantas cosas juntas.*

*A mi Novio, **Francisco Muñoz**, porque siempre me ha apoyado tanto moral como económicamente, y porque me ha demostrado que puedo contar con Él siempre, y porque sé que sin su apoyo y sin su **AMOR** no lo hubiera conseguido
Gracias, TE AMO.*

*A todas **mis Amigas** por su amistad incondicional por su apoyo y sus consejos. Muchas Gracias.
(Adrianita, Alma, Angélica. Josefina).*

*A **mis Profesores**, por enseñarme sus conocimientos que para mí son tan valiosos y útiles, y en especial al Ing. Santiago Igor Valiente Gómez por ser nuestro Director de Tesis, y por brindarnos su amistad.
Muchas Gracias.*

*A mi Amiga **Elizabeth Morales**, porque gracias a su amistad, a su apoyo y sobre todo a su paciencia, hemos concluido una de nuestras metas más anheladas.
Gracias por tu amistad. Te Quiero Mucho.*

Mil Gracias a todos por su ayuda.

Elizabeth Araceli Gomez Sanson

Dedico esta Tesis a:

***A ti Señor que siempre me has ayudado
y has estado conmigo en los momentos más
difíciles de mi vida.***

***A ti Señor ya que todo lo que tengo
y lo que ahora soy te lo debo a ti.***

***A mis Padres, José Morales y Estela Solís, ya que
sin sus Consejos, Amor
y Paciencia no hubiese podido llegar
a este momento. Los Quiero Mucho***

***A mis Hermanos Raúl, José
Nancy y Brenda que en todo momento
han estado conmigo y me siguen apoyando.***

***A ti Nancy aunque ya no estás físicamente
con nosotros pero me viste iniciar una etapa más
en mi vida, ahora concluida,
sé que me sigues viendo.***

***A todos mis amigos gracias por
su Amistad, Confianza y Apoyo.***

***A todos mis profesores, por compartir
conmigo sus conocimientos.***

***Al Ing. Santiago Igor Valiente Gómez, por
darme su confianza y apoyo.***

A todos y cada uno de ellos ;Gracias!.

Elizabeth Morales Solís

INDICE GENERAL

INTRODUCCION	1
CAPITULO I CONCEPTOS GENERALES	3
1.1 DEFINICIÓN DEL PROBLEMA	1
1.1.1 <i>Meta</i>	1
1.1.2 <i>Justificación</i>	2
1.2 INGENIERIA DE SOFTWARE	3
1.3 CICLO DE VIDA DE LOS SISTEMAS	4
1.3.1 <i>El Modelo en Espiral</i>	5
1.3.2 <i>Técnicas de Cuarta Generación</i>	7
1.3.3 <i>Diseño Modular</i>	8
1.3.3.1 <i>Diagrama de Estructura</i>	9
1.4 SISTEMAS DE BASES DE DATOS	14
1.4.1 <i>Elementos de un Sistema de Bases de Datos</i>	15
1.4.2 <i>Independencia de Datos</i>	17
1.4.2.1 <i>Independencia Lógica</i>	17
1.4.2.2 <i>Independencia Física</i>	18
1.4.3 <i>Integridad de los Datos</i>	19
1.4.3.1 <i>Validación de Datos</i>	19
1.4.3.2 <i>Integridad Referencial</i>	20
1.5 MODELOS DE BASES DE DATOS	21
1.5.1 <i>Modelo Jerárquico</i>	22
1.5.2 <i>Modelo de Red</i>	24
1.5.2.1 <i>Estructura de Red Simple y Compleja</i>	25
1.5.3 <i>Modelo Relacional</i>	25
1.5.3.1 <i>Elementos del Modelo Relacional de Bases de Datos</i>	26
1.5.3.2 <i>La Estructura Relacional</i>	26
1.5.3.3 <i>Diagrama Entidad-Relación</i>	29

CAPITULO II	TEORIA DEL ANALISIS Y DISEÑO	30
2.1	ARQUITECTURA DE UNA BASE DE DATOS	30
2.2	MODELO CONCEPTUAL	30
2.2.1	<i>Diseño del Modelo Conceptual</i>	31
2.2.1.1	<i>Términos del Modelo Conceptual</i>	31
2.2.1.2	<i>Relaciones</i>	32
2.2.1.3	<i>Grados de Relación entre Entidades</i>	33
2.2.1.4	<i>Cardinalidad de Relación entre Entidades</i>	33
2.3	MODELO LOGICO O INTERNO	37
2.3.1	<i>Normalización de Bases de Datos</i>	37
2.3.1.1	<i>Objetivos de la Normalización</i>	38
2.3.1.2	<i>Dependencia Funcional</i>	39
2.3.1.3	<i>Primera Forma Normal (1FN)</i>	41
2.3.1.4	<i>Segunda Forma Normal (2FN)</i>	42
2.3.1.5	<i>Tercera Forma Normal (3FN)</i>	45
2.4	MODELO FISICO O INTERNO	47
2.5	DIAGRAMA DE FLUJO	47
2.5.1	<i>Elementos del Diagrama de Flujo</i>	48
2.5.1.1	<i>Guía para la Construcción de un Diagrama de Flujo</i>	49
2.5.1.2	<i>Niveles de un Diagrama de Flujo de Datos</i>	50
2.5.2	<i>Características de los Diagramas de Flujo de Datos</i>	51
2.5.2.1	<i>Descripción de los Procesos</i>	51
2.6	DICCIONARIO DE DATOS	52
2.6.1	<i>Contenido de un Diccionario de Datos</i>	53
2.6.2	<i>Funciones del Diccionario de Datos</i>	54
CAPITULO III	INFRAESTRUCTURA TECNOLÓGICA	55
3.1	REQUERIMIENTOS DEL SISTEMA	55
3.1.1	<i>Sistema Operativo</i>	57
3.1.2	<i>MIS Windows95 y MS Windows NT</i>	58
3.1.3	<i>Manejadores de Bases de Datos (DBMS)</i>	59
3.2	Lenguaje de Programación	60

3.2.1 Herramientas de Planificación de Sistemas de Gestión	62
3.2.2 Herramientas de Gestión de Proyectos	62
3.2.3 Herramientas de Soporte	64
3.2.4 Herramientas de Análisis y Diseño	64
3.2.5 Herramientas de Programación.....	65
3.2.6 GeneXus.....	67
3.2.7 Una Implementación del Desarrollo Incremental con GeneXus.....	69
3.2.7.1 Diseño	70
3.2.7.2 Transacciones.....	70
3.2.7.3 Reportes	72
3.2.7.4 Procedimientos.....	72
3.2.7.5 Work Panels	73
3.2.7.6 Web Panels.....	73
3.2.7.7 Menús.....	74
3.2.7.8 Data Views	74
3.2.7.9 Prototipo	75
3.3 IMPLEMENTACION	76
3.4 MANTENIMIENTO	79
3.5 DOCUMENTACION	80
3.6 CONSOLIDACION DE VARIAS APLICACIONES Y REUTILIZACION DEL CONOCIMIENTO	81
CAPITULO IV IMPLEMENTACION DE SICIPA	85
4.1 PROPUESTA DE SOLUCION.....	85
4.2 DISEÑO MODULAR	86
4.3 METODOLOGIA.....	86
4.3.1 MODELO EN ESPIRAL.....	87
4.4 DESARROLLO MODULAR	88
4.5 DIAGRAMA MODULAR DE SICIPA	89
4.5.1 Módulos de SICIPA.....	90
4.5.1.1 Módulo de Configuración.....	92
4.5.1.2 Módulo de Movimientos.....	94
4.5.1.3 Módulo de Seguridad.....	97
4.5.1.4 Módulo de Consultas.....	98
4.5.1.5 Módulo de Reportes	99

4.6 DIAGRAMA ENTIDAD-RELACIÓN.....	106
4.7 DIAGRAMA DE FLUJO DE SICIPA.....	110
4.8 DICCIONARIO DE DATOS DE SICIPA.....	122
4.9 DESARROLLO DE SICIPA.....	125
CAPITULO V MANTENIMIENTO Y SEGURIDAD.....	130
5.1 SEGURIDAD DEL SISTEMA.....	130
5.1.1 Métodos de Control de Acceso a Datos.....	131
5.1.2 Objetivos Por Asegurar.....	132
5.2 MANTENIMIENTO DEL SISTEMA.....	133
5.2.1 Tipos de Mantenimiento.....	139
5.2.1.1 Correctivo.....	139
5.2.1.2 Adaptativo.....	139
5.2.1.3 Perfectivo.....	140
5.2.1.4 Preventivo.....	140
5.2.2 Características de Mantenimiento.....	140
5.3 FALLOS GENERALES DEL SISTEMA.....	141
5.4 MEJORAS DEL SISTEMA.....	141
5.5 ASISTENCIA AL USUARIO FINAL.....	143
CONCLUSIONES.....	144
BIBLIOGRAFIA.....	146
APENDICE A: MANUAL DE USUARIO.....	148
APENDICE B: VOCABULARIO DE TERMINOS.....	183
APENDICE C: CODIGO EJEMPLO DE SICIPA.....	190

INTRODUCCION

Debido al crecimiento acelerado de información que se ha presentado en los últimos años, se tiene la necesidad de contar con herramientas de apoyo como son los sistemas computacionales. De esta manera, con el paso del tiempo y gracias a dichos sistemas podemos obtener de una forma más rápida y sencilla, reportes y consultas de un grupo de datos para analizarlos y poder tomar decisiones más acertadas en un tiempo menor al que tomaría hacerlo manualmente.

Actualmente pequeñas o grandes corporaciones cuentan con información que necesita ser analizada más a detalle y no cuenta con un sistema que agilice las operaciones internas y toma de decisiones, o en su defecto, cuentan con sistemas desarrollados con tecnologías obsoletas que administran los datos pero la información que se obtiene de ellos no es la adecuada para cubrir las necesidades requeridas en su momento.

Tal es el caso de la Dirección General de Obras y Servicios Generales de la UNAM. Esta dependencia tiene a su cargo la administración de la información de inventarios generada por los almacenes coordinados por la misma. Cuenta con un sistema obsoleto (desarrollado en Clipper) que no cubre las recientes necesidades para la buena toma de decisiones.

Con el fin de aprovechar el cúmulo de información que dicha dependencia genera, surge la necesidad de crear un sistema de administración de datos por computadora, con la intención de brindar un medio de información ágil y oportuno que facilite el análisis de los datos requeridos en un momento dado.

El presente trabajo está enfocado al desarrollo de dicho sistema, el cual, deberá ofrecer a todos los usuarios la consulta en pantalla de la información actualizada en los rubros presupuestales, inventariales y de gestión administrativa, brindando además, la posibilidad de imprimir en forma total o parcial los datos consultados.

Para fines de explicación, el presente trabajo se dividió en cinco capítulos, cada uno de los cuales está dedicado a explicar a detalle los elementos utilizados para el desarrollo del sistema.

De esta manera, en el primer capítulo se hace referencia a conceptos generales necesarios para el mejor análisis de los requisitos del sistema.

En el segundo capítulo, se desarrollan los temas relacionados con la teoría del análisis y diseño de sistemas, es aquí, donde es recabada la información necesaria para el sistema, dando ejemplos para entender mejor el análisis y diseño del mismo.

En el tercer capítulo, se define la arquitectura de software (programa de cómputo) a utilizar para el desarrollo del sistema, analizando las ventajas y desventajas que presenta, además, se habla de la herramienta de programación necesaria para la aplicación.

En el cuarto capítulo, se describe la metodología a utilizar durante el desarrollo, especificando a detalle cada una de las etapas que conforman la construcción del sistema.

En el quinto capítulo, se explica a detalle los diferentes tipos de mantenimiento de sistemas y seguridad de la información.

Finalmente, se tienen las conclusiones obtenidas a través del desarrollo de la aplicación.

CAPITULO I. CONCEPTOS BASICOS

1.1 DEFINICIÓN DEL PROBLEMA

Actualmente los almacenes de los talleres de conservación que dependen de la Dirección General de Obras y Servicios Generales de la UNAM presentan un manejo deficiente de los bienes, debido a que se realiza de manera semimanual, es decir, apoyados en un sistema obsoleto desarrollado en Clipper, pero que no resuelve todas sus necesidades, por lo que la seguridad y organización en los inventarios se ve afectada y se ha detectado la necesidad de implementar un sistema que cumpla con los siguientes objetivos:

- Simplificar y agilizar el control y operación de los servicios de los almacenes.
- Generar información en forma oportuna para que pueda ser analizada por el personal autorizado.
- Manejo de información confiable que fortalezca la toma de decisiones y apoye la operatividad interna de los almacenes.
- Control de la información generada por los módulos considerados en el diseño del sistema.
- Tener mayor seguridad en el manejo de la información
- Definición de un sistema con los atributos de calidad requeridos:
 - Confiabilidad
 - Portabilidad
 - Eficiencia
 - Solidez
 - Corrección

1.1.1 META

Se pretende obtener un sistema que facilite las operaciones realizadas en los almacenes, que van desde el ingreso de un nuevo artículo hasta su egreso, considerando los servicios de consulta, modificaciones, devoluciones, bajas, etc. Así como tener un mejor control en la administración interna del activo fijo y equipo de cómputo.

1.1.2 JUSTIFICACIÓN

La Dirección General de Obras y Servicios Generales de la UNAM, con el fin de hacer frente eficientemente a las crecientes necesidades en el manejo de información en los almacenes de conservación, tomó la decisión de implementar un sistema más automatizado, que permita apoyar el control de información de los almacenes en forma expedita y racional aprovechando para ello los recursos existentes, es decir, utilizando adecuadamente la capacidad instalada.

1.2 INGENIERIA DE SOFTWARE

La ingeniería del software abarca un conjunto de tres elementos clave: metodologías, herramientas y procedimientos, que facilitan al ingeniero controlar el proceso de desarrollo del software y suministrar a los que practiquen dicha ingeniería las bases para construir software de alta calidad de una forma productiva

Las *metodologías* de la ingeniería del software indican "cómo" construir técnicamente el software. Abarcan un amplio espectro de tareas que incluyen: planificación y estimación de proyectos, análisis de los requisitos del sistema y del software, diseño de estructuras de datos, arquitectura de programas y procedimientos algorítmicos, codificación, prueba y mantenimiento.

Existen diversas metodologías de desarrollo de software, por lo que es necesario analizarlas y evaluarlas, para seleccionar la que mejor se adapte a las necesidades planteadas. Normalmente se selecciona una metodología estándar con las variaciones y modificaciones requeridas. En el presente apartado se analizará la metodología de Ciclo de Vida Clásico de los Sistemas asociado al Modelo en Espiral.

Generalmente se entiende por Metodología, refiriéndose a cualquier ámbito o trabajo, a un sistema ordenado para la obtención de un fin.

Si se considera el entorno informático, es decir, la producción o desarrollo de sistemas informáticos, es evidente que el uso de una metodología en este proceso aporta algunas ventajas que hacen aconsejable su uso.

La modelización de datos y estructuras según diversas técnicas empieza a ser posible sin gran esfuerzo, y el mantenimiento de un diccionario de datos y de la generación de la documentación de una aplicación dejará de ser la cruz con la que siempre han cargado los analistas.

Las *herramientas* de la ingeniería del software suministran un soporte automático o semiautomático para los métodos. Hoy existen herramientas para soportar cada uno de éstas metodologías. Cuando se integran las herramientas de forma que la información creada por una herramienta pueda ser usada por otra, se establece un sistema para el soporte del desarrollo del software, llamado CASE (*Computer-Aided Software Engineering: Ingeniería del Software Asistida por Computadora*). CASE combina software, hardware y bases de datos sobre ingeniería del software.

Los *procedimientos* de la ingeniería del software son el pegamento que junta los métodos y las herramientas y facilita un desarrollo racional y oportuno del software de computadora.

1.3 CICLO DE VIDA DE LOS SISTEMAS

La Figura 1.1 ilustra el ciclo de vida clásico para la ingeniería del software, algunas veces llamado "modelo en cascada", el paradigma del ciclo de vida exige un enfoque sistemático y secuencial del desarrollo del software que comienza en el nivel del sistema y progresa a través del análisis, diseño, codificación, prueba y mantenimiento. Modelizado a partir del ciclo convencional de una ingeniería, el paradigma del ciclo de vida abarca las siguientes características:

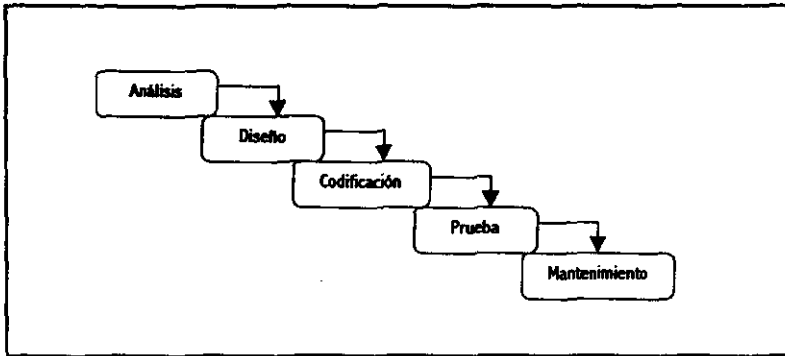


Figura 1.1 Ciclo de vida clásico

Análisis. El Análisis de los requisitos del Software es un proceso de recopilación de los requisitos se centra e intensifica especialmente para el software. Para comprender la naturaleza de los programas que hay que construir, el ingeniero de software debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridos.

Diseño. El diseño del software es realmente un proceso multipaso que se enfoca sobre cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedural y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software que pueda ser establecida de forma que obtenga la calidad requerida antes que comience la codificación. Al igual que los requisitos, el diseño se documenta y forma parte de la configuración del software.

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

Prueba. Una vez que se ha generado el código, comienza la prueba del programa. La prueba se centra en la lógica interna del software, asegurando que todas las sentencias se han probado, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren, utilizando datos o entradas reales o semejantes a estos.

Mantenimiento. La fase de mantenimiento se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas por la evolución del entorno del software y a las modificaciones debidas a los cambios de los requisitos del cliente dirigidos a reforzar o a ampliar el sistema. La fase de mantenimiento vuelve a aplicar los pasos de las fases de definición y de desarrollo, pero en el contexto del software ya existente. Durante la fase de mantenimiento se encuentran tres tipos de cambios:

Corrección. Incluso llevando a cabo las mejores actividades de garantía de calidad, es muy probable que el cliente descubra defectos en el software. El *mantenimiento correctivo* cambia el software para corregir los defectos.

Adaptación. Con el paso del tiempo es probable que cambie el entorno original para el que se desarrolló el software. El *mantenimiento adaptativo* consiste en modificar el software para acomodarlo a los cambios de su entorno externo.

Mejora. Conforme utilice el software, el cliente/usuario puede descubrir funciones adicionales que podrían interesar que estuvieran incorporadas en el software. El *mantenimiento perfectivo* amplía el software más allá de sus requisitos funcionales originales.

El software, indudablemente, sufrirá cambios después de que se libere a la D.G.O. y S.G., esto debido a que debe adaptarse a cambios del entorno externo.

1.3.1. EL MODELO EN ESPIRAL

El modelo en espiral para la ingeniería del software ha sido desarrollado para cubrir las mejores características tanto del ciclo de vida clásico, como de la creación de prototipos, añadiendo al mismo tiempo un nuevo elemento: el análisis de riesgo. El modelo, representado mediante la espiral de la figura 1.2 define cuatro actividades, representadas por los cuatro cuadrantes.

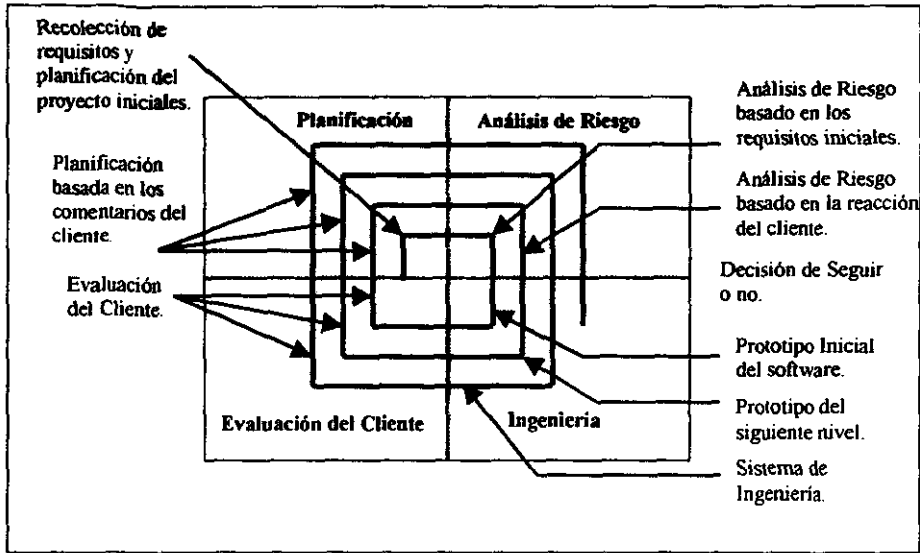


Figura 1.2 El Modelo en Espiral

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

Un aspecto intrigante del modelo en espiral se hace evidente cuando consideramos la dimensión radial representada en la Figura 1.2. Con cada iteración alrededor de la espiral (comenzando en el centro y siguiendo hacia el exterior), se construyen sucesivas versiones del software, cada vez más completas. Durante la primera vuelta alrededor de la espiral se definen los objetivos, las alternativas y las restricciones, y se analizan e identifican riesgos.

Si el análisis de riesgo indica que hay una incertidumbre en los requisitos, se puede usar la creación de prototipos en el cuadrante de ingeniería para dar asistencia tanto al encargado como al cliente. Se pueden usar simulaciones y otros modelos para definir más el problema y refinar los requisitos.

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

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

El modelo en espiral para la ingeniería del software es actualmente el enfoque más realista para el desarrollo de software y de sistemas a gran escala. Utiliza un enfoque "evolutivo" para la ingeniería del software, permitiendo al desarrollador y al cliente entender y reaccionar a los riesgos en cada nivel evolutivo. Utiliza la creación de prototipos como un mecanismo de reducción de riesgo, pero, lo que es más importante, permite a quien lo desarrolla aplicar el enfoque de creación de prototipos en cualquier etapa de la evolución del producto. Mantiene el enfoque sistemático correspondiente a los pasos sugeridos por el ciclo de vida clásico, pero incorporándola dentro de un marco de trabajo interactivo que refleja de forma más realista el mundo real. El modelo en espiral demanda una consideración directa de riesgos técnicos en todas las etapas del proyecto y, si se aplica adecuadamente, debe reducir los riesgos antes de que se conviertan en problemáticos.

1.3.2 TÉCNICAS DE CUARTA GENERACIÓN (T4G)

El término "Técnicas de Cuarta Generación" (T4G) abarca un amplio espectro de herramientas de software que tienen algo en común: todas facilitan, a quien desarrolla el software, la especificación de algunas características del software de alto nivel. Luego, la herramienta genera automáticamente el código fuente basándose en la especificación del técnico.

Actualmente, un entorno para el desarrollo de software que soporte T4G puede incluir todas o algunas de las siguientes herramientas: lenguajes no procedurales para consulta a bases de datos, generación de código, facilidades gráficas de alto nivel y facilidades de hoja de cálculo. Todas estas herramientas están disponibles, pero sólo para ámbitos de aplicación muy específicos.

Con muy pocas excepciones, el ámbito de aplicación actual de las T4G está limitado a las aplicaciones de sistemas de información de gestión, concretamente al análisis de información y la obtención de informes relativos a grandes bases de datos. Sin embargo, las nuevas herramientas CASE (Computer-Aided Software Engineering Ingeniería de Software Asistido por Computadora) soportan ahora el uso de T4G para la generación automática de "esquemas de código" para aplicaciones de ingeniería y de tiempo real, tal es el caso de Genexus, herramienta CASE que se describirá más adelante.

Los datos preliminares recogidos en compañías que usan T4G parecen indicar que el tiempo requerido para producir software se reduce mucho para aplicaciones pequeñas y de tamaño medio, y que la calidad de análisis y diseño para las aplicaciones pequeñas, también se reduce.

1.3.3. DISEÑO MODULAR

El Diseño Modular recomendado para el diseño de un sistema modular se denomina diagrama estructural. Un diagrama estructural simplemente es un diagrama que consta de rectángulos, los cuales representan a los módulos y se conectan por medio de flechas, como se muestra en la figura 1.3.

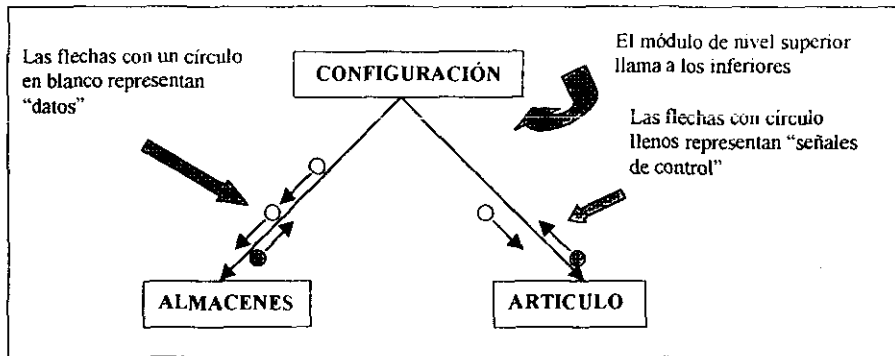


Figura 1.3 Diagrama de Estructura A

Los módulos que se encuentran abajo, es decir, los de Almacenes, Artículos son submódulos del módulo Configuración.

Para recalcar que el diagrama estructural es una técnica del diseño descendente, las flechas de conexión se dibujan con una dirección de arriba hacia abajo. A los lados de las flechas de conexión se dibujan dos flechas más pequeñas. Tales flechas con círculos vacíos se denominan "datos", mientras que las flechas con círculos llenos se denominan indicadores de control. Estas flechas indican que algo pasa, ya sea por abajo del módulo superior o de regreso del módulo inferior.

1.3.3.2 DIAGRAMA DE ESTRUCTURA

A través de los diagramas de estructuras se puede modelar el control del sistema, así como la descomposición de las funciones en forma jerárquica.

En un diagrama de estructura, como se muestra en la figura 1.4 los módulos son representados por rectángulos. Se representa la dependencia (jerarquía) entre módulos, las instancias de repetición y decisión así como el flujo de los datos de control y otros a través de las funciones. Los módulos del diagrama de estructura son los mismos que los que aparecen en los distintos niveles del DFD.

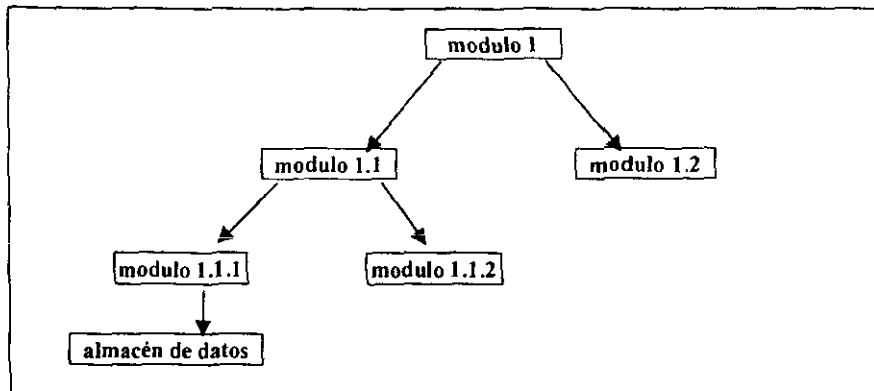


Figura 1.4 Diagrama de Estructura B

Aunque el módulo padre de un diagrama de estructura o módulo raíz puede tener dos o n hijos en su segundo nivel de descomposición, se recomienda descomponer este módulo en 3 hijos, cada uno de ellos dará origen a una rama en el diagrama de estructura, es decir, cada uno de ellos a su vez podrá tener otros módulos hijos.

Estas ramas son:

- Rama aferente: su objetivo es capturar u obtener la información proveniente generalmente del usuario.

- Rama de proceso: transforma la información capturada, es decir las entradas, en las salidas del sistema.

- Rama eferente: su objetivo es entregar las salidas del sistema al usuario o al terminador que corresponda.

El Diseño Modular propone descomponer la programación en fracciones lógicas y manejables. Este tipo de programación se apega bien al diseño descendente (el cual consiste en observar el sistema y luego explosionarlo en partes más pequeñas o subsistemas) como se muestra en la figura 1.5, porque enfatiza las interfaces entre los módulos, más que mantenerlas ignoradas hasta el final del desarrollo del sistema. De manera ideal, cada módulo del sistema debe ser funcionalmente cohesivo, de tal manera que satisfaga sólo una función.

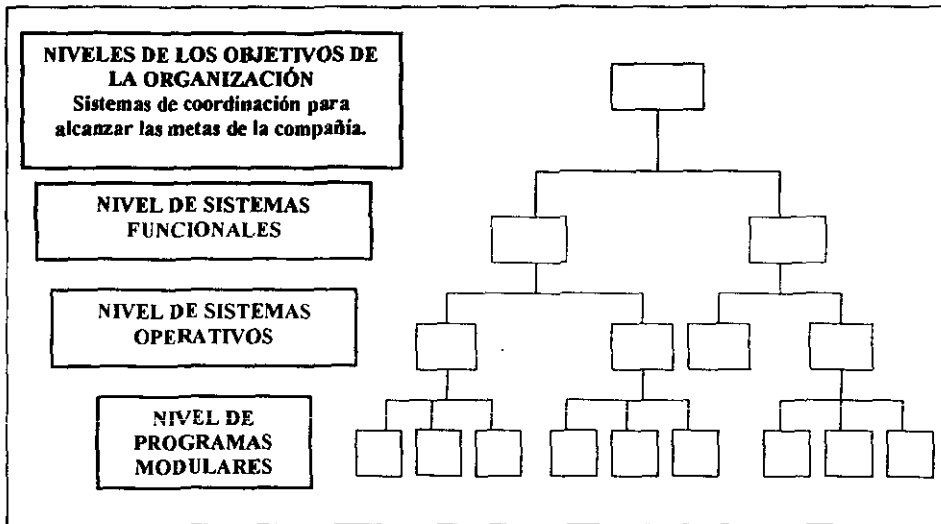


Figura 1.5 Diseño Modular

El diseño de programación modular tiene algunas ventajas básicas como:

1. Los módulos son más fáciles de escribir y de revisar, ya que están virtualmente autocontenidos.
2. La detección de un error dentro del un módulo es menos complicada.
3. El mantenimiento de módulos es más fácil.
4. La problemática de los módulos es más fácil de entender.

Algunos lineamientos para la metodología de la programación modular incluyen:

- Mantener el módulo de un tamaño manejable
- Prestar atención particular a las interfaces críticas (esto es, a los datos y a las variables de control que pasan entre los módulos).
- Minimizar el número de módulos que el usuario necesite modificar cuando haya cambios.
- Mantener las relaciones bien establecidas.

La calidad del diseño debe ser una meta para el diseñador. El diseño estructurado ofrece guías para apoyar al diseñador a determinar módulos y sus interconexiones, que mejor realizarán los requerimientos especificados por el analista. Las dos reglas más importantes son las referentes al acoplamiento y la cohesión.

Cohesión

Grado en el cual los componentes de un módulo (típicamente las instrucciones individuales que los conforman) son necesarios y suficientes para llevar a cabo una sola función bien definida. En la práctica, esto significa que el diseñador debe asegurarse de no fragmentar los procesos esenciales en módulos y también debe asegurarse de no juntar procesos no relacionados en módulos sin sentido. Los módulos son aquellos que son funcionalmente cohesivos (es decir, módulos en los cuales cada instrucción es necesaria para poder llevar a cabo una tarea bien definida). Los peores módulos son los que son coincidentalmente cohesivos (es decir, donde sus instrucciones no tienen una relación significativa entre uno y otro).

Los grados de cohesión, de menor a mayor son:

- a) **Cohesión Coincidental.** No existe una relación significativa entre los elementos del módulo.
- b) **Cohesión Lógica.** La relación entre los elementos del módulo está basada en obtener ventajas en el procesamiento, por ejemplo, todos manipulan el mismo dato. Normalmente esto implica tener un código truculento o compartido, que degrada los propósitos de un buen diseño.
- c) **Cohesión Temporal.** Los elementos del módulo constituyen un conjunto que se ejecuta secuencialmente en un punto fijo en el tiempo. Aunque tiende, a veces, a confundirse con la cohesión lógica, la diferencia está en que este tipo de módulos más simple y se ejecuta sin la intervención de otras aplicaciones.
- d) **Cohesión Comunicacional.** Los elementos del módulo hacen referencia al mismo conjunto de datos. Aquí se presenta un grado "aceptable" de cohesión.

e) **Cohesión Secuencial.** Implica que la salida de un elemento es la entrada para el próximo.

f) **Cohesión Funcional.** Aquí, todos los elementos del módulo están orientados a la realización de una función única.

Acoplamiento

El Acoplamiento es el grado en el cuál los módulos se interconectan o se relacionan entre ellos. Entre más fuerte sea el acoplamiento entre módulos en un sistema, más difícil es implantarlo y mantenerlo, pues entonces se necesitará un estudio cuidadoso para la modificación de algún módulo o módulos.

En la práctica, esto significa que cada módulo debe tener interfaces sencillas y limpias con otros, y que se debe compartir un número mínimo de datos entre módulos.

También significa que un módulo dado no debe modificar la lógica interna o los datos de algún otro módulo; lo que se conoce como una conexión patológica.

Tamaño del Módulo

De ser posible, cada módulo debe ser lo suficientemente pequeño como para caber en una sola página (o para que se pueda desplegar en una sola pantalla). Desde luego, a veces no es posible determinar que tan grande va a ser un módulo hasta hacerlo escrito, pero las actividades iniciales de diseño a menudo darán al diseñador una buena pista de que el módulo será grande o complejo. Si es así, debe subdividirse en uno o más niveles de submódulos.

Alcance del control

El número de subordinados inmediatos que un módulo administrador puede llamar se conoce como el alcance del control. Un módulo no debe poder llamar a más de una media docena de módulos de nivel inferior. La razón es evitar la complejidad: si el módulo tuviera, por ejemplo, que llamar a 25 módulos de nivel inferior, entonces seguramente contendrá tanta lógica compleja que nadie lo entenderá (un sin fin de if-then anidados). La solución es introducir un nivel intermedio de módulos administradores, como haría un administrador de una organización humana.

Alcance del efecto / Alcance del control

Esta regla establece que cualquier módulo afectado por el resultado de alguna decisión debe ser subordinado (aunque no necesariamente un subordinado inmediato) del módulo que toma la decisión. Es un tanto análogo a la regla de administración que dice que cualquier empleado afectado por los resultados de la decisión de algún administrador (es decir, dentro del alcance de efecto de la decisión), debe estar dentro del alcance de control del administrador (es decir, trabajando entre la jerarquía de personas que se reportan con el administrador). Violar esta regla en un ambiente de diseño estructurado usualmente lleva a un paso innecesario de banderas y condiciones (lo cual incrementa el acoplamiento entre módulos), la toma redundante de decisiones o (en el peor de los casos) conexiones patológicas entre módulos.

Manejo Autónomo de Errores

Los módulos deben tener la capacidad de manejar sus propias condiciones de error, tanto en la detección como en la corrección de los mismos. De no ser así, el manejo de banderas (flags) de control y la transmisión de datos erróneos a otros módulos aumentara considerablemente el acoplamiento.

Parsimonia

Se refiere a la economía de recursos que se emplean para la obtención de un resultado. Esto es, sólo se debe realizar lo que se pide. Mientras mayor la parsimonia, mejor el diseño.

1.4 SISTEMAS DE BASES DE DATOS

Una Base de Datos puede definirse como una colección de datos interrelacionados entre sí y almacenados en conjunto, sin redundancias innecesarias, la cual tiene como finalidad servir a una o más aplicaciones de la mejor manera posible.

La representación gráfica de una Base de Datos es:

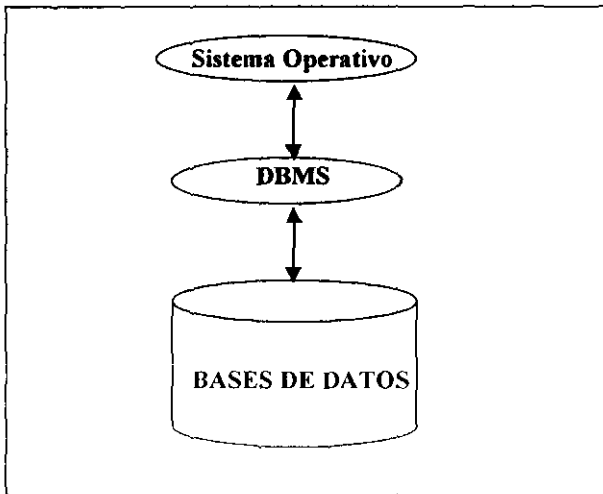


Figura 1.6 Representación de una Base de Datos.

La potencia de cálculo de una computadora se deriva de las características físicas que posee:

- Rapidez
- Precisión
- Memoria

Las características citadas provienen de los componentes electrónicos que conforman una computadora; velocidad de conmutación de circuitos electrónicos, rapidez de transmisión de señales eléctricas, fiabilidad de los circuitos y gran capacidad de almacenamiento de información.

Un papel importante de una Base de Datos es proteger los datos almacenados frente al acceso por parte de un usuario no autorizado.

Los sistemas de gestión de Bases de Datos organizan y estructuran los datos de tal modo que puedan ser recuperados y manipulados por usuarios y programas de aplicación. Las estructuras de los datos y las técnicas de acceso proporcionadas por un DBMS particular se denominan Modelo de Datos. El Modelo de Datos determina la <<personalidad>> de un DBMS y las aplicaciones para las cuales está particularmente bien conformado.

Cuando la gestión de Bases de Datos se popularizó durante los años setenta y los ochenta, emergieron un puñado de modelos de datos populares. Cada uno de estos primeros modelos de datos tenían ventajas y desventajas que jugaron papeles importantes en el desarrollo del modelo de datos relacional. En muchos sentidos el modelo de datos relacional representó un intento de simplificar los modelos de datos anteriores.

Ante la introducción de los sistemas de gestión de la Base de Datos, todos los datos permanentemente almacenados en un sistema informático, tales como la nómina y los registros de contabilidad, se almacenaban en archivos individuales.

Los problemas de mantener grandes sistemas basados en archivos condujo a finales de los años sesenta al desarrollo de los sistemas de gestión de Bases de Datos. La idea detrás de estos sistemas era sencilla, tomar la definición de los contenidos de un archivo y la estructura de los programas individuales, y almacenarla junto con los datos en un Base de Datos.

Utilizando la información de la Base de Datos, el DBMS que la controlaba podría tomar un papel mucho más activo en la gestión de los datos y en los cambios a la estructura de la Base de Datos.

Una de las aplicaciones más importantes de los sistemas de gestión de Base de Datos primitivos era el planteamiento de la producción para empresas de facturación.

1.4.1 ELEMENTOS DE UN SISTEMA DE BASES DE DATOS

Un Sistema de Bases de Datos incluye cuatro elementos los cuales son:

- Datos

Un Dato es una característica de un hecho o ente que por sí solo no es significativo. Sólo cuando están en conjunto adquieren su completo significado, es decir, se convierte en información.

- **Hardware**

El Hardware de una Base de Datos consiste básicamente en el medio en que reside de manera física.

- **Software**

Entre las bases de Datos físicas y los usuarios se encuentra el sistema manejador de Base de Datos (Data Base Management System, DBMS), que es un conjunto de programas que realizan las siguientes funciones:

- Crear la Base de Datos.
- Permite el acceso a los registros.
- Extrae información de acuerdo a los requerimientos de los usuarios.
- Permite la actualización de los registros.

Para realizar las funciones mencionadas, un DBMS tiene como mínimo los siguientes componentes.

1. Un lenguaje de definición de datos (Data Definition Lenguaje, DDL) que permite crear la estructura de cada registro.
2. Un Lenguaje de Manejo de Datos (Data Manipulation Lenguaje, DML) que permite agregar, borrar o modificar registros de la Base de Datos. Los lenguajes de consulta y los generadores de reportes forman parte del DML.
3. Un Diccionario de Datos (DD) que permite mantener la integridad de la Base de Datos, pues contiene datos de los datos, es decir, información referente a los formatos de los registros, derechos de acceso, usuarios, etc. No debe confundirse este concepto con el Diccionario de Datos como herramienta de análisis de sistema.
4. Un Lenguaje de Programación para construir programas o reportes complejos. Actualmente algunos manejadores de Bases de Datos ofrecen generadores de aplicaciones como parte integral del mismo.

- **Usuarios**

Se considera que hay tres grandes tipos de usuarios:

- El Administrador de la Base de Datos.
- El Programador de aplicaciones.
- El Usuario Final

El Administrador de la Base de Datos es quien tiene a su cargo el diseño de la Base de Datos y el control de ellos. Entre otras funciones, el Administrador de la Base de Datos define la estructura y métodos de acceso a la Base de Datos, los tipos de autorización de acceso para los usuarios y los controles de integridad, además de establecer las estrategias de respaldo y recuperación de la información.

El programador de aplicaciones se encarga de escribir los programas que hacen uso de la Base de Datos.

El diseño de aplicaciones ha cambiado drásticamente en los últimos años y consecuentemente los estilos de programación. De una programación secuencial se paso a una programación modular, para llegar a una programación estructurada con muy buenos resultados.

A partir de la programación estructurada los esfuerzos se centraron en potenciar la modularidad y la reutilización del código, lo que condujo a la programación orientada a objetos que se complementa con una programación basada en componentes de software. Pero que los estilos hayan cambiado, no significa que el último elimine a los anteriores; al contrario, se apoya en los anteriores.

1.4.2. INDEPENDENCIA DE DATOS

La independencia de los datos se refiere a la protección contra los programas de aplicación de ocasionar modificaciones ocasionadas cuando se altera la organización física o estructura lógica de la base. Un sistema de base de datos ofrece dos tipos de independencia de datos: física y lógica.

1.4.2.1 INDEPENDENCIA LÓGICA

La independencia lógica se refiere a la protección de algunos programas de aplicación cuando se modifica el esquema debido a cambios hechos en otro programa del mismo tipo. Por ejemplo, los campos en un archivo conceptual en el esquema se pueden insertar o borrar, pero el programa de aplicación que usa ese archivo debe continuar funcionando correctamente sin necesidad de modificaciones, siempre y cuando los campos insertados o eliminados no sean referenciados en los programas.

El DBMS logra esta independencia en dos pasos:

- 1) Las definiciones de datos para cada tipo de registro conceptual se describen en el esquema, el cual está separado físicamente de los programas de aplicación. Así cualquier cambio en la definición de los datos resultará solamente en una modificación y recopilación del esquema, mientras que los programas de aplicación serán inmunes a los cambios.

2) Los registros lógicos externos requeridos por un programa de aplicación se definen en un subesquema. El DBMS extrae de un esquema los elementos de datos necesarios del subesquema. El esquema o subesquema creado por el DBMS actúa como "traductor" entre el usuario (por un lado del proceso) y los datos en cuestión. Si se cambian los datos de alguna forma, sólo se necesita cambiar el traductor intermedio para acomodar el cambio de datos.

1.4.2.2 INDEPENDENCIA FÍSICA

La independencia física se refiere a la protección contra modificaciones de los programas de aplicación debido a cambios en la estructura del archivo o cambios en las características de los campos en los archivos que usan.

A continuación se describen algunas dificultades que enfrentan los programadores de lenguajes de alto nivel cuando ocurren cambios en sus archivos. Como son:

- a) **Cambio de la organización de los archivos:** Suponga que se recrea un archivo mediante una organización directa direccionable usando la distribución en vez de una organización secuencial indexada. Todos los programas que usan el archivo se deben modificar y recompilar para que así, reflejen el cambio en la especificación del archivo.
- b) **Cambio del tipo o del nombre de un campo:** Suponga que para su representación de salida, se cambia el tipo de dato de un elemento de representación binaria a decimal. Se debe cambiar la especificación del tipo del campo en el programa. Análogamente, el cambio de la longitud de un campo significa el recargar nuevamente el archivo así como la modificación y recompilación de los programas afectados.
- c) **Cambio de claves primaria o secundaria:** El cambio de las claves usadas para consultas requiere recrear los archivos de datos para crear nuevas trayectorias de acceso, modificar y recompilar los programas afectados para cambiar las especificaciones de claves y proposiciones de E/S.
- d) **Cambio del tamaño máximo del arreglo:** El archivo de datos debe cargarse nuevamente y la especificación del tamaño del arreglo en el programa se debe modificar.

1.4.3 INTEGRIDAD DE LOS DATOS

El término **Integridad de los Datos** se refiere a medidas de seguridad usadas para mantener correctos los datos en la Base de Datos. Tal vez la mejor manera para explicar la integridad de los datos sea entendiendo la manera en que se introducen los errores en la Base de Datos. Algunas fuentes potenciales de error son:

1. Falla del equipo (hardware)
2. Actualización incompleta
3. Defectos en el software
4. Inserción de datos no válidos
5. Errores humanos.

Una falla del sistema, tal como un golpe en la cabeza (head crash), el choque de la cabeza de lectura/escritura sobre la superficie del disco puede destruir físicamente los datos en el mismo. Otra causa de errores menos dramática son las fallas del sistema. Si se cae el sistema cuando se lleva a cabo una operación WRITE, esto puede resultar en una actualización incompleta causando inconsistencia en los datos. Los errores del software pueden deberse a "errores de lógica" (bugs), que son bastante familiares.

Para asegurar la integridad de los datos, no sólo son necesarios la confiabilidad física y lógica (hardware y software), sino también de que los datos sean correctos. La validación de los datos de entrada es una operación común en el procesamiento de datos. Los datos pueden validarse por medio de un programa de aplicación al momento de leer cada registro, o puede validarse todo un archivo ya escrito mediante algún programa separado.

La tecnología de Bases de Datos proporciona recursos que refuerzan la integridad de los datos sin necesidad de demasiada programación extra por parte del usuario. Existen distintas maneras de asegurar la integridad de los datos como por ejemplo:

- Validación de Datos
- Integridad Referencial

1.4.3.1 VALIDACIÓN DE DATOS

Validación del tipo de los datos

Así como ocurre en la programación con lenguajes de alto nivel, los elementos de los datos en una Base de Datos se pueden especificar como carácter, decimal, fijo, punto flotante, etc. El contenido de cada elemento de entrada debe coincidir con sus características; de otra manera el DBMS mandará el mensaje apropiado de error.

Validación del valor de los datos

El contenido de un campo de entrada puede validarse para cierto rango de valores numéricos o un número finito de valores alfa-numéricos. Por ejemplo, el campo ALMACEN puede contener sólo cuatro valores distintos: "Zoquepa", "Talleres", "Zona Cultural", "Viveros".

Validación sobre los valores de la claves primaria y secundaria

Cada registro en una Base de Datos se puede representar únicamente por un valor de clave simbólico, que es distinto de su dirección física. En la especificación de un esquema, a cada tipo de registro se le designa como la clave principal un campo o una combinación de dos o más campos; esa clave se puede usar para identificar de manera única a cada registro en el archivo.

El DBMS asegura que el valor clave-primaria sea único y no nulo. (Un valor nulo significa que el campo está vacío o no está definido). Mientras se introduce cada registro en la Base de Datos, el DBMS valida su valor clave-primaria. Si un registro de entrada contiene un valor duplicado o un valor nulo, el DBMS generará un mensaje de error.

Cuando se declaran en el esquema los valores clave-secundaria, el usuario puede especificar si se permiten los valores duplicados de tal clave. Si no se permiten, el DBMS efectuará el mismo tipo de comprobación de la consistencia para la unicidad de la clave secundaria como lo hace para la clave principal.

1.4.3.2 INTEGRIDAD REFERENCIAL

La idea básica de la Integridad Referencial es asegurar que no existan registros hijo si es que no existe el registro padre correspondiente. Análogamente, la integridad referencial implica que la eliminación de registros causará la eliminación de todas las transacciones hechas por él. Sin embargo, lo contrario no se cumple. Puede existir un registro padre sin necesidad de algún hijo asociado a él.

La Integridad Referencial es un sistema de reglas que se utilizan para garantizar que las relaciones entre los registros de tablas relacionadas son válidas y que no se eliminan ni modifican accidentalmente datos relacionados.

Se puede establecer la integridad referencial cuando se cumplen todas las condiciones siguientes:

- El campo coincidente de la *tabla principal* es una clave principal o tiene un índice único.

- Los campos relacionados tienen el mismo tipo de datos.
- Ambas tablas pertenecen a la misma base de datos. Si las tablas están vinculadas, deben ser tablas en el mismo formato. Y se debe abrir la base de datos en la que están almacenadas para poder establecer la integridad referencial. La integridad referencial no puede exigirse para tablas vinculadas procedentes de bases de datos en otros formatos.

Cuando se exige la integridad referencial, deben observarse las reglas siguientes:

- No se puede introducir un valor en el campo de Clave externa de la tabla relacionada que no exista en la clave principal. No obstante, puede introducir un valor Nulo en la clave externa, especificando que los registros no están relacionados. Por ejemplo, no puede tener un pedido asignado a un cliente que no existe pero puede tener un pedido asignado a nadie mediante la instrucción de un valor Nulo en el campo Id de cliente.
- No se puede eliminar un registro de una tabla principal si existen registros coincidentes en una tabla relacionada. Por ejemplo, no puede eliminarse un registro de empleados de la tabla Empleados si existen pedidos asignados al empleado en la tabla Pedidos.
- No se puede cambiar un valor de clave principal en la tabla principal si ese registro tiene registros relacionados. Por ejemplo, no puede cambiarse el Identificador (Id) de un empleado en la tabla Empleados si existen pedidos asignados a ese empleado en la tabla Pedidos.

1.5 MODELOS DE BASES DE DATOS

Los modelos de Bases de Datos son herramientas conceptuales para describir los datos, sus relaciones, su semántica y sus limitantes, permiten interpretar de una manera abstracta fenómenos del mundo real.

Existen cuatro modelos de Bases de Datos que son:

- Modelo Jerárquico
- Modelo de Red
- Modelo Relacional

1.5.1 MODELO JERARQUICO

En un Modelo Jerárquico los datos están estructurados en forma arborescente y las relaciones entre los diferentes tipos de registros se resuelven mediante punteros o enlaces entre ellos. Se establece una jerarquía de modo que las relaciones entre un registro y otro relacionado con él, es decir, es una relación "padre-hijo", las cuales tiene como condición que un registro "hijo" no puede existir si no existe el registro "padre" asociado a él. La estructura es como la que se muestra en la figura 1.7.

Con excepción de la raíz, todo nodo está relacionado con otro nodo de nivel más alto llamado padre. Ningún elemento puede tener más de un padre, sin embargo, todo elemento puede tener más elementos de nivel más bajo llamados hijos. Los elementos que se encuentran en las puntas de las ramas, es decir, que no tienen hijos se les llama hojas.

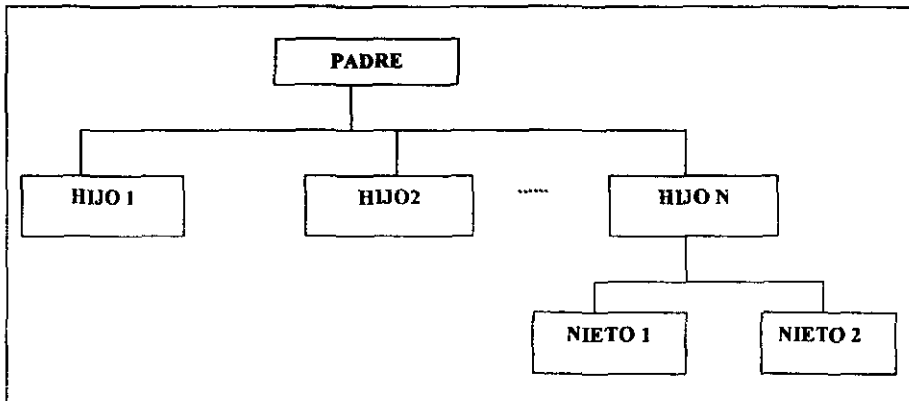


Figura 1.7 Estructura Jerárquica

Los árboles se utilizan tanto para las descripciones lógicas, donde describen las relaciones que existen entre los tipos de segmentos o tipos de registro; como para las descripciones físicas donde se le emplea para describir conjuntos de apuntadores y relaciones entre entradas en los índices. Dentro del Modelo Jerárquico existen diferentes tipos de Jerarquías que son:

- **Jerarquías Balanceadas**

En una jerarquía balanceada todos los nodos deben tener el mismo número de ramas, el árbol empieza por la raíz en forma descendente y va avanzando de izquierda a derecha.

Es fácil implementar una organización física de datos para los árboles que tienen un número fijo de ramas por nodo que para los que tienen números variables.

- **Jerarquías Desbalanceadas**

La mayoría de las organizaciones de datos por supuesto que no encajan en una estructura jerárquica balanceada sino que requieren números de ramas diferentes en los diversos nodos son llamadas Jerarquías Desbalanceadas.

En este tipo de jerarquías el número de nodos puede variar respecto al número de ramas, a diferencia de la jerarquía balanceada, la Jerarquía Desbalanceada es un poco difícil de implementar en una organización física de datos ya que como se mencionó sería un árbol con un número variable de ramas por nodo.

- **Jerarquías Binarias**

La jerarquía binaria es una categoría especial de la estructura jerárquica balanceada, que admite sólo dos nodos por rama. Las jerarquías binarias al igual que otras jerarquías balanceadas son principalmente utilizadas en la representación física de los datos y no en las representaciones lógicas.

- **Jerarquías n-arias**

Este tipo de jerarquía es un poco problemático debido a que los vínculos entre las ramas y nodos son M:N (muchos a muchos). Este tipo de jerarquías utilizan tipos de registros virtuales o también llamados apuntadores, es decir, un tipo de registro virtual es un tipo de registro con la propiedad de que cada uno de sus registros contiene un apuntador a un registro de otro tipo de registros PV. HV siempre desempeña el papel de "hijo virtual" y PV "padre virtual".

La idea es incluir más de un esquema jerárquico en el esquema de la base de datos jerárquica y usar apuntadores de los nodos de un esquema jerárquico al otro para representar los vínculos.

Las estructuras jerárquicas implican por lo general que hay una correspondencia simple de hijo a padre, es decir, cada hijo tiene sólo un padre. Sin embargo, la correspondiente inversa es compleja, de uno a muchos, por que cada padre puede tener varios hijos.

1.5.2 MODELO DE RED

El Modelo de Red se obtiene cuando se permite relaciones entre "hermanos", es decir, entre registros de un mismo padre lógico. En ella se pueden construir esquemas como el que se muestra en la figura 1.8.

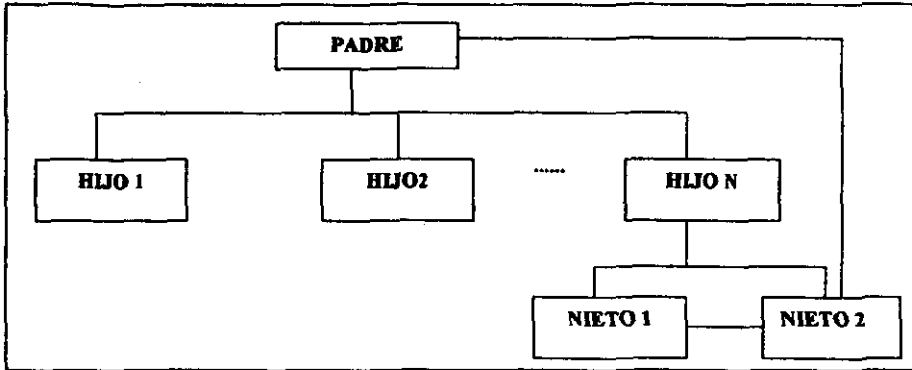


Figura 1.8 Estructura de red

Debemos considerar al Modelo Jerárquico como un tipo de Modelo de datos en Red con ciertas restricciones.

Una desventaja de este modelo es que algunas estructuras de red comienzan a tomar una apariencia de "telaraña", con apuntadores que salen en todas direcciones, por lo cual podría llegar a ser compleja y difícil de modificar.

Si en una relación entre datos un hijo tiene más de un padre, la relación no puede ser descrita por medio de un árbol o estructura jerárquica, se le describe en cambio, como una estructura de red.

En un Modelo de Red cualquier elemento puede vincularse con cualquier otro elemento. Al igual que en la estructura jerárquica, la estructura de red puede ser descrita en términos de padres e hijos, y dibujada de cierta forma que los padres siempre quedan arriba de los hijos. Pero en la estructura en red, un hijo puede tener más de un padre.

En algunas estructuras de red es natural referirse a niveles, como ocurre con las estructuras jerárquicas.

1.5.2.1 ESTRUCTURAS DE RED SIMPLE Y COMPLEJA

En una buena cantidad de las estructuras de red que representan relaciones entre tipos de registros o tipos agregados de datos, la correspondencia entre padres e hijos es similar a la que existe en un árbol, la correspondencia de hijo a padre es simple y la de padre a hijo es compleja.

Sin embargo, en un Modelo de Red es común que la estructura sea compleja, es decir, que los hijos tengan muchos padres, así como los padres tengan muchos nodos hijos.

La razón que justifica el patentizar la diferencia entre estructuras simples y complejas reside en el hecho de que las últimas requieren métodos más elaborados para su representación física.

Las estructuras de red se pueden reducir a una forma más simple introduciéndoles algo de redundancia. En algunos casos la redundancia involucrada en la transformación no es mucha y se tolera sin problemas.

Una estructura de red no es tan fácil de convertir en una estructura de árbol, ya que se requieren dos arboles para ser representada.

En general, cada relación con correspondencia compleja en los dos sentidos tiene que ser remplazada por dos vinculaciones del tipo de estructura jerárquica.

1.5.3 MODELO RELACIONAL

El Modelo Relacional de datos es definido en 1970 por el Dr. F. Codd que lo define por medio de una serie de reglas cuyo objetivo es lograr la independencia de la representación lógica de los datos de su almacenamiento físico.

Esta independencia tanto física como lógica se refiere a tres aspectos:

- Independencia de ordenación

En este tipo de independencia, el resultado obtenido en un acceso no depende de cómo estén ordenados los datos físicamente.

- Independencia de la indexación

Es este tipo de independencia se separan los índices de los datos haciendo que la creación y mantenimiento sean manejados por el sistema.

- Independencia de los caminos de acceso

En este tipo de independencia se hace que la navegación a través de los datos no tenga que estar previamente establecida consiguiendo así unas formas de acceso más flexibles.

El Modelo Relacional es una estructura lógica en vez de física.

1.5.3.1 ELEMENTOS DEL MODELO RELACIONAL DE BASES DE DATOS

El modelo relacional de bases de datos consta de una estructura de datos, unos conjuntos de operadores relacionales y dos reglas de integridad. Estos tres componentes están vinculados con la definición, la manipulación y la integridad de los datos, respectivamente.

La estructura relacional es la denominada relación o tabla. El conjunto de operadores racionales es lo que Codd definió como "Álgebra relacional"; tales operadores producen nuevas relaciones a partir de otras existentes. Las dos reglas de integridad son las de integridad de entidad y de integridad referencial.

1.5.3.2 LA ESTRUCTURA RELACIONAL

Una de las mayores ventajas del modelo relacional es la simplicidad de su estructura, la llamada relación. La manera más simple de definir una relación es como una tabla de datos. Una relación esta formada por tuplas y cada tupla tiene elementos llamados atributos. Una tupla equivale a un renglón dentro de una tabla y un atributo a una columna.

Un sistema de manejo de base de datos (DBMS) utiliza un modelo de datos para definir la estructura fundamental de los mismos. Un modelo de datos expresa las relaciones entre las entidades.

El número de los atributos en una relación es el grado de la relación y el número de tuplas es la cardinalidad. El grado de una relación normalmente no cambia después de su creación, en cambio la cardinalidad varía según se agregan nuevas tuplas o se eliminan las existentes.

Ejemplo:

Nombre	omicilio	Teléfono
Enrique Guzmán	axaca 72	615-12-89
Amparo Torres	epeyac 124	756-30-80
Dolores Mendoza	uahutémoc 185	579-45-90
Isabel Espinosa	iruelos 23	393-78-45

El conjunto de valores de donde un atributo obtiene sus valores es el dominio del atributo. Los dominios son conceptuales en su naturaleza y pueden o no estar explícitamente contenidos en la base de datos.

Ya que en una tabla no esta permitido tener renglones duplicados, cada tabla debe tener una llave primaria, es decir un atributo o una combinación de estos que identifique de manera única cada renglón dentro de la tabla.

Las llaves primarias son de fundamental importancia en el modelo relacional, ya que para localizar un renglón dado en una base de datos, es necesaria la combinación del nombre de la tabla y del valor de la llave primaria.

El concepto de llave foránea es otro aspecto fundamental en las bases de datos relacionales. Una llave foránea es un atributo o una combinación de estos, dentro de una relación R que es la llave primaria de otra relación S.

Una base de datos relacional puede definirse simplemente como un conjunto de relaciones. Para Date, una base de datos relacional es "una base de datos que es percibida por sus usuarios como un conjunto de tablas (y nada mas que tablas)". La expresión "percibida por los usuarios " utilizada por Date es importante, pues los conceptos del modelo relacional son aplicables únicamente a los niveles externo y conceptual de bases de datos. De hecho, el modelo relacional no propone nada a nivel interno, más bien se refiere a la manera en que los usuarios ven a la base de datos.

Toda relación o tabla debe cumplir las siguientes reglas.

- Toda relación o tabla debe tener un nombre único y distinto a otros dentro de la misma base de datos.
- Cada atributo o columna debe tener un nombre propio, único y distinto a otras, dentro de la misma tabla.
- Cada celda o intersección renglon-columna debe aceptar exactamente uno y solo un valor nunca un conjunto o arreglo de datos.

- Los valores en una misma columna deben ser del mismo tipo.
- Cada renglón de la relación debe ser distinto. No debe haber renglones idénticos.
- El orden de los renglones no es significativo.
- El orden de las columnas no es significativo.

Una de las principales ventajas del planteamiento relacional es:

- **Simplicidad** : Es decir, la facilidad de comprensión por parte del usuario final. El usuario final se enfrenta con un modelo de datos. Sus solicitudes se formulan en términos del contenido de la información y no reflejan la complejidad de los aspectos relacionados con el sistema. Un modelo relacional es prácticamente lo que el usuario ve, y no necesariamente lo que se implantara físicamente.
- **Independencia de los datos**: Esto debe constituir uno de los principales objetivos de cualquier sistema de manejo de base de datos. El modelo relacional elimina los detalles relativos a la estructura del almacenamiento y la estrategia del acceso de la interfaz con el usuario. Este modelo proporciona un grado de independencia de los datos relativamente más alta que los dos modelos que se han a discutido. Sin embargo, para poder usar esta propiedad del modelo relacional, el diseño de las relaciones debe ser complejo y exacto.
- **Fundamentos teóricos**: El modelo relacional está basado en la bien desarrollada teoría matemática de las relaciones. El riguroso método del diseño de una base de datos (usando la normalización) da a este modelo un fundamento sólido.
- **Las Búsquedas**: Pueden ser mucho más rápidas que en los sistemas que deben seguir una cadena de apuntadores
- **Las operaciones de proyección**: Son fáciles de implementar, por lo que la creación de nuevas relaciones necesarias para aplicaciones particulares resulta fácil de realizar.
- **Las estructuras relacionales**: Son mucho más fáciles de modificar que las estructuras jerárquicas o de red. En ambientes donde la flexibilidad es importante, esto es primordial.

Este Modelo puede considerarse como una forma de representación universal, ya que es relativamente fácil convertir casi cualquier otro tipo de estructura o modelo de Bases de Datos al esquema relacional.

La representación tabular usada en el esquema relacional es fácil de comprender por los usuarios y fácil de implementar en el sistema físico de Bases de Datos.

1.5.3.3 DIAGRAMA ENTIDAD-RELACION

En 1976, Peter Chen propuso el modelo de datos denominado "Entidad-Relación", en el que combina las ventajas de los modelos de datos relacionales y de red. En el documento denominado "The Entity-Relationship Model: Toward an Unified View of Data", Chen presentó el nuevo modelo, acompañado de una técnica de diagramación como herramienta para el diseño de bases de datos: *El Diagrama Entidad-Relación o E-R*. El Diagrama E-R es actualmente utilizado como una herramienta para presentar los requerimientos de datos durante el diseño conceptual de Bases de Datos el cual se presenta y desarrolla en el capítulo IV..

CAPITULO II. TEORIA DE ANALISIS Y DISEÑO

2.1 ARQUITECTURA DE UNA BASE DE DATOS

De acuerdo con ANSI/X3/SPARC (Standard Planning and Requerimientos Committee of the American National Standards Institute on Computers and Information Processing: Comité de Planeación y Requerimientos del Instituto Nacional Estadounidense de Estándares en Computación y Procesamiento de la Información), el diseño de bases de datos es un proceso de creación de modelos: un Modelo Conceptual, un Modelo Lógico y un Modelo Físico.

2.2 MODELO CONCEPTUAL

El Modelo Conceptual es una herramienta de comunicación entre los diversos usuarios de los datos y como tal se desarrolla sin tomar en cuenta la representación física de los mismos. Este Modelo se usa para organizar, visualizar, planear y comunicar ideas y debe ser independiente del sistema de manejo de Base de Datos.

El Modelo Conceptual es un modelo del mundo real expresado en términos de entidades, relaciones y atributos. El resultado de esta fase es el Diagrama Entidad-Relación (E-R), en el que se describen entidades, relaciones y atributos, de una manera simple y comprensible.

Este modelo representa la visión organizacional de la base que se obtiene al integrar los requerimientos de todos los usuarios en una empresa. Es la representación del contenido de la Base de Datos. Un Modelo Conceptual consta de las siguientes definiciones:

Definición de los Datos: En el Modelo Conceptual se describen el tipo de datos y la longitud del campo de todos los elementos en la Base de Datos. Los elementos por definir incluyen artículos elementales (*Atributos Campos de Datos propiedades que definen o caracterizan a una entidad. Al atributo frecuentemente se le llama elemento de datos. campo de datos, campo ítem de datos o ítem elemental*) y registros conceptuales (*Entidades objetos tangibles que pueden describirse con palabras, código numérico o no numérico. Estos pueden ser una persona, lugar, una cosa, un evento o un concepto acerca del cual se registra información*).

En el Modelo Conceptual la Base de Datos aparece sólo como una colección de registros lógicos, sin descriptores o apuntadores de almacenamiento.

2.2.1 DISEÑO DEL MODELO CONCEPTUAL

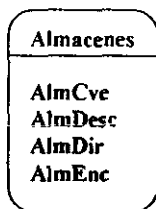
Una de las principales funciones de administración de la Base de Datos es desarrollar un Modelo Conceptual, también llamado "modelo de la empresa", de la organización.

2.2.1.1 TERMINOS DEL MODELO CONCEPTUAL

- **Entidades**

Una Entidad es un objeto o evento cuyo dato es necesario conocer y contener en una Base de Datos. Ejemplos de Entidades son: Artículos, Almacenes, Empleados, etc. Toda entidad tiene características que la hacen diferente a otras, es decir, tienen atributos propios. Una Entidad es única y diferente a otras, así como también sus ocurrencias. Para identificar de modo único una entidad se utiliza una llave de entidad como identificador que puede estar formado por uno o más atributos de la entidad. En el caso de "Artículos", la llave de la entidad será la clave del artículo, pues esta clave es diferente para todos y cada uno de los artículos.

En un Diagrama E-R, una Entidad se representa con un rectángulo, en cuyo interior se escribe el nombre que identifica a la entidad. Para nombrar una entidad se debe elegir un nombre que represente claramente el hecho u objeto que identifica. Regularmente se utilizan sustantivos como nombres de entidades. En el siguiente diagrama se muestra un ejemplo de una entidad del sistema:



- **Atributos**

Un Atributo es una propiedad de una entidad, es cualquier detalle que sirve para calificar, identificar, clasificar, cuantificar o expresar el estado de una entidad. Por ejemplo, para la entidad "Artículos" sus atributos serían la clave de artículo, nombre del artículo, unidad de medida, etc.

- **Relaciones**

Una Relación es una asociación entre entidades. Existen tres tipos de relaciones para las entidades. Esas relaciones son: Relación 1:1, Relación 1:N, Relación N:M. Las cuales se explicaran más adelante.

2.2.1.2 RELACIONES

Una Relación es una asociación entre entidades. Por ejemplo: "Almacén" puede estar relacionado o asociado con la entidad "Artículos" a través de la relación almacena. Una ocurrencia de la relación "Almacén" almacena "Artículos", sería por ejemplo "Almacén 1 almacena arena", donde "Almacén 1" y "arena" son ocurrencias de "Almacén" y "Artículos" respectivamente.

Para nombrar relaciones se usa regularmente un verbo. Por ejemplo:

Almacén – Almacena – Artículos

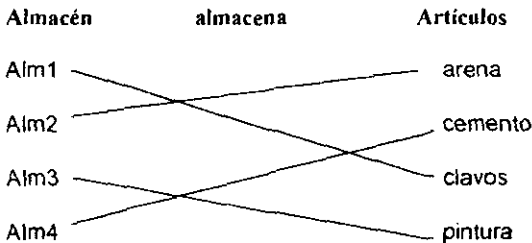
Proveedor – Suministra – Artículos

Empleados – Laboran en – Almacén

El diagrama E-R para el ejemplo de "Almacén almacena Artículos" sería como a continuación se muestra:



Para ver con mayor claridad el concepto de relación, veamos el siguiente diagrama de asociación entre las ocurrencias de las entidades mencionadas. Por un lado se presenta la entidad "Almacén" con las ocurrencias "Alm1", "Alm2", "Alm3" y "Alm4". Por otra parte se presenta la entidad "Artículos" con las ocurrencias "arena", "cemento", "clavos", "pintura". La relación se describe mediante líneas que unen las ocurrencias de ambas partes.



La interpretación del diagrama anterior sería:

- El Alm1 almacena clavos
- El Alm2 almacena arena
- El Alm3 almacena pintura
- El Alm4 almacena cemento


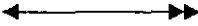

Los diagramas de asociación de ocurrencias son un complemento de los diagramas E-R pues muestran gráficamente y a modo de ejemplo, la asociación entre ocurrencias de entidades, utilizando valores reales o ficticios de alguna propiedad de las entidades representadas.

2.2.1.3 GRADOS DE RELACION ENTRE ENTIDADES

El grado de una relación entre entidades se refiere al número de entidades asociadas. Así, existen relaciones unarias, binarias, ternarias, ..., n -arias. En una relación unaria participa solamente una entidad; este tipo de relación es también conocida como recursiva pues sus ocurrencias se asocian con otras dentro de la misma entidad. En las relaciones binarias participan sólo dos entidades; este tipo de relación se presenta con mayor frecuencia que otros tipos. En las relaciones ternarias participan tres entidades y así, en las relaciones n -arias participan n entidades.

2.2.1.4 CARDINALIDAD DE RELACION ENTRE ENTIDADES

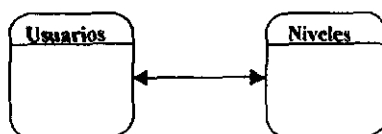
La cardinalidad de una relación se refiere al número de ocurrencias que pueden estar asociadas entre entidades. Por lo tanto, existen relaciones uno a uno (1:1), uno a muchos (1:N) y muchos a muchos (M:N). Para distinguir la cardinalidad en un diagrama E-R se tiene la siguiente nomenclatura:

- * Relación de uno a uno (1:1) 
- * Relación de uno a muchos (1:N) 
- * Relación de muchos a muchos (M:N) 

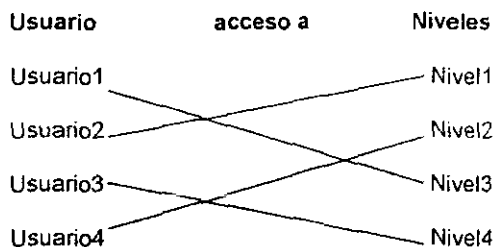
Relación uno a uno (1:1)

Una relación uno a uno existe cuando las ocurrencias de las entidades participantes aceptan como máximo una ocurrencia asociada.

Para ejemplificar una relación 1:1 supóngase que para el acceso a SICIPA(Sistema Integral de Control de Inventarios Para Almacenes), se establece que los usuarios pueden tener sólo un nivel de acceso. Para efectos de registrar el acceso, pueden definirse las entidades "Usuarios", que se refiere a los usuarios del sistema; y la entidad "Niveles" que se refiere a los niveles disponibles para acceso al sistema.

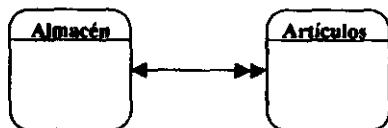


Este ejemplo puede verse con mayor claridad en el siguiente diagrama de ocurrencias, en el que se observa que un usuario tiene uno y sólo un nivel de acceso al sistema, y que un nivel es asignado a sólo un usuario.

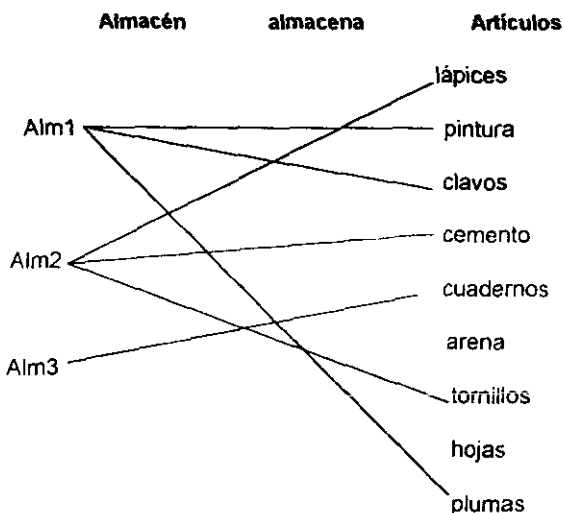


Relación uno a muchos (1:N)

En una Entidad uno a muchos, una ocurrencia de una Entidad A puede asociarse a n ocurrencias de una Entidad B, y las ocurrencias de B se asocian únicamente con una ocurrencia de A

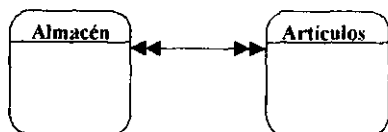


Para ejemplificar este tipo de relación, utilizaremos la relación de "Almacén almacena Artículos". En el siguiente diagrama de ocurrencias se puede observar que un almacén puede tener varios artículos almacenados y que cada artículo es almacenado en sólo un almacén

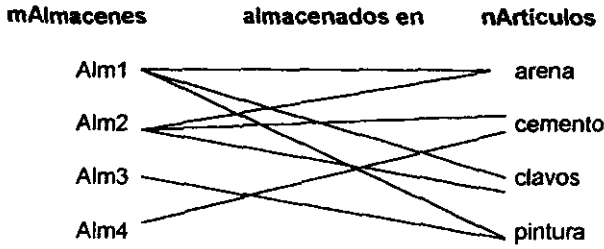


Relación muchos a muchos (N:M)

En una Relación muchos a muchos, varias ocurrencias de una entidad pueden estar asociadas a varias ocurrencias de otra.



Para ejemplificar esta relación veamos lo siguiente: supóngase ahora que cada almacén almacena más de un artículo y que cada artículo está almacenado en varios almacenes. En este ejemplo se definen las entidades de "Almacén" y "Artículos". La relación para estas entidades sería "almacenados en". El diagrama de ocurrencias se presenta a continuación:



2.3 MODELO LOGICO O INTERNO

El Modelo Lógico de Bases de Datos relacionales tiene como objetivo obtener relaciones normalizadas a partir de diagramas E-R. Paul Beynon define el modelo lógico de una base de datos como un modelo del mundo real expresado en términos de relaciones, dominios, llaves primarias y llaves foráneas.

Las relaciones obtenidas deben someterse a un proceso de normalización.

El Modelo Lógico es la representación del nivel inferior de una Base de Datos. Es el nivel más bajo y se refiere a la manera en que los datos están físicamente almacenados. Este modelo mapea la base lógica hacia el almacenamiento físico y establece trayectorias de datos (mediante apuntadores o índices) para el acceso aleatorio a las Bases de Datos. Un Modelo Interno es descrito por el DBMS (*Data Base Management System*: Sistema Manejador de Bases de Datos) como un esquema interno.

El Esquema Interno contiene especificaciones detalladas del almacenamiento de todos los registros almacenados en la base de datos así como los descriptores del sistema: apuntadores, palabras de control y trayectorias de datos necesarias para recuperación sobre claves secundarias.

2.3.1 NORMALIZACION DE BASES DE DATOS

La Teoría de la Normalización es una ayuda que proporciona un procedimiento riguroso para el diseño de Bases de Datos. Una Base de Datos mal diseñada puede funcionar inicialmente, pero puede mostrar anomalías en el almacenamiento debidas al agrupamiento indiscriminado de los campos cuando se efectúen en los archivos las operaciones de inserción, actualización o eliminación.

La teoría de la normalización ayuda a reconocer las cualidades no deseadas en un archivo y la forma de corregirlas. Con este procedimiento, un archivo conceptual se representa como una tabla de dos dimensiones llamada relación; la forma más simple para representar datos es mediante una tabla. El enfoque relacional generalmente se usa en el proceso de normalización para llegar a modelos de Bases de Datos. La meta final del proceso de normalización es la agrupación de todos los atributos (o campos) de una base de datos en relaciones adecuadas para que la base se pueda almacenar con el mínimo de datos redundantes.

2.3.1.1 OBJETIVOS DE LA NORMALIZACION

- Eliminar en lo posible todos los datos que mantengan anomalías
- La estructura debe ser tal que haya lugar para todos los datos requeridos.
- La redundancia que pueda existir deberá ser causada por los elementos que son identificadores o llaves. Por lo que debe elegir aquellos que no estén sujetos a actualizaciones.

Los efectos indeseables son las anomalías que pueden presentarse en las operaciones de actualización, inserción y eliminación.

Anomalía de Inserción. No cabe almacenar nueva información sobre una entidad en particular hasta que se establece su relación con otra entidad.

Anomalía de Eliminación. La eliminación de un solo registro puede ocasionar la eliminación de toda la ocurrencia de una entidad,

Anomalía de Actualización. Si el valor de un atributo cambia, debe cambiar en los múltiples sitios donde se encuentra definido.

- Conservar toda la información
- Maximizar la flexibilidad.

La capacidad de adaptabilidad ante los cambios, maximizan la independencia del uso particular de los datos.

La normalización requiere tres o más acciones sobre un atributo de una entidad (1FN, 2FN, 3FN, BCFN), hasta que el modelo alcance su última etapa de no redundancia, por lo regular la normalización está comprendida hasta la tercera etapa, donde el modelo ya alcanza un estado manejable y entendible para el usuario y el programador.

Esto se establece en las dependencias funcionales que existen entre los atributos y los dominios particulares a esa relación. Las dependencias funcionales las determina directamente el significado o la semántica del contenido de la base de datos según la interpretación del diseñador de ésta.

La normalización después de la tercera forma normal implica que se trata de una dependencia multivalor no trivial o MVD unión binaria. En la realidad es rara vez necesaria la normalización más allá de la tercera forma normal, porque la dependencia multivalor en general se trata en la etapa inicial del proceso de normalización.

2.3.1.2 DEPENDENCIA FUNCIONAL

El concepto de *Dependencia Funcional (DF)*, se tomó de las matemáticas elementales. Se dice que Y es función de X, $Y = f(X)$, si el valor de Y está siempre determinado por el valor de X. Si se aplica la misma terminología a una relación, la dependencia funcional entre los atributos A y B en una relación se define como sigue: El atributo A es funcionalmente dependiente del atributo B si el valor de A está determinado por el valor de B. Tal dependencia se simboliza:



Considere la relación Almacén de la Figura 2.1 que consta de tres atributos AlmCve, AlmDesc y AlmDir. Para un valor dado de AlmCve, digamos A1, sólo puede haber una dirección AlmDir. Así, el atributo AlmDir es funcionalmente dependiente del atributo AlmCve. La dependencia funcional de AlmDir en AlmCve se denota por:



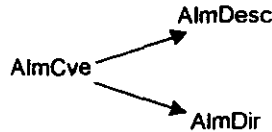
En otras palabras AlmCve determina a AlmDir. Se dice que AlmCve es un determinante de AlmDir.

AlmCve	AlmDesc	AlmDir
A1	Talleres	C.U.
A2	Zoquipa	Zoquipa
A3	Viveros	C.U.
A4	Z.Cultural	C.U.

Figura 2.1 Dependencia Funcional entre atributos de una Relación
 Los atributos AlmDesc y AlmDir son funcionalmente dependientes de AlmCve, porque sólo puede haber un AlmDesc y un AlmDir para un AlmCve dado.

Análogamente, el atributo AlmDesc es funcionalmente dependiente del atributo AlmCve porque para un valor dado de AlmCve puede haber sólo un AlmDesc.

Los atributos AlmDesc y AlmDir de la relación Almacén son funcionalmente dependientes de su clave principal, AlmCve. Esta relación de atributos se puede representar por el siguiente diagrama de dependencia funcional:



Se simplificará, el diagrama colocando atributos con el mismo determinante en la misma línea:



Esta representación más simple es útil cuando el diagrama de dependencia funcional contiene una gran cantidad de atributos.

La clave primaria de una relación puede ser una clave compuesta (o concatenada) que consta de más de un campo. Por lo tanto, un atributo puede ser funcionalmente dependiente de un grupo de atributos en vez de sólo un atributo. Un campo en una clave principal compuesta se llama *atributo principal*. Cualquier campo que no forme parte de la clave principal se llama no-clave.

El término de *Dependencia Funcional Total* se refiere al tipo de dependencia donde un atributo es funcionalmente dependiente de todos los campos de la clave primaria en vez de sólo algún subconjunto de la clave primaria. El propósito de establecer un diagrama de dependencia funcional para una relación es usar un método gráfico para determinar la forma normal de la relación.

2.3.1.3 PRIMERA FORMA NORMAL (1FN)

Una relación está en *Primera Forma Normal (o 1FN)* si todos los campos en cada registro contienen un solo valor tomado de sus dominios respectivos. El dominio de un campo es el rango de valores continuos o discretos permitidos para el campo. Por ejemplo, en la relación de Artículos del sistema que se desarrolló, si los valores de ArtCnt son enteros positivos entre 1 y 99, el dominio ArtCnt es el conjunto de enteros 1, 2, ..., 99.

ArtCve	AlmCve	ArtDesc	ArtCnt	ArtEMin	ArtEMax	ArtEAct	ArtPU	ArtPP
Art1	A1	Tornillo	99	50	200	55	100	120
Art2	A2	Mesa	50	20	100	30	20	25
Art3	A1	Cuadro	12	2	50	6	150	75

Figura 2.2 Primera Forma Normal

Las anomalías de almacenamiento se pueden atribuir a la presencia de uno o más campos no-clave que no son total y funcionalmente dependientes de la clave principal. Afortunadamente, las anomalías de almacenamiento de la relación 1FN se pueden eliminar con el siguiente procedimiento:

- i) Quitar de la relación 1FN todos los campos no-clave que no sean totalmente dependientes de la clave primaria.
- ii) Guardar los campos no-clave que fueron quitados en relaciones nuevas y adecuadas.

El procedimiento de la división de una relación en dos o más relaciones más pequeñas en base a las relaciones de atributos se llama *proceso de normalización*. Los detalles son:

PASO 1. Escoger una clave primaria que pueda representar de manera única cada registro en la relación.

PASO 2. Construir un diagrama de dependencia funcional describiendo las relaciones entre los atributos.

PASO 3. Dividir la relación 1FN de tal manera que todos los campos no clave en cada relación dividida sea total y funcionalmente dependientes de la clave primaria.

2.3.1.4 SEGUNDA FORMA NORMAL (2FN)

Una relación es o pertenece a la *Segunda Forma Normal (o 2FN)* si es 1FN y cada atributo no-clave de la relación es total y funcionalmente dependiente de su clave principal.

Las relaciones derivadas en la figura 2.3, son de la segunda forma normal porque todos sus atributos no-clave ya son total y funcionalmente dependientes de sus claves primarias. De cualquier manera, aun cuando las anomalías de almacenamiento 1FN quedan eliminadas cuando se alcanza la 2FN, pueden surgir otro tipo de anomalías de almacenamiento.

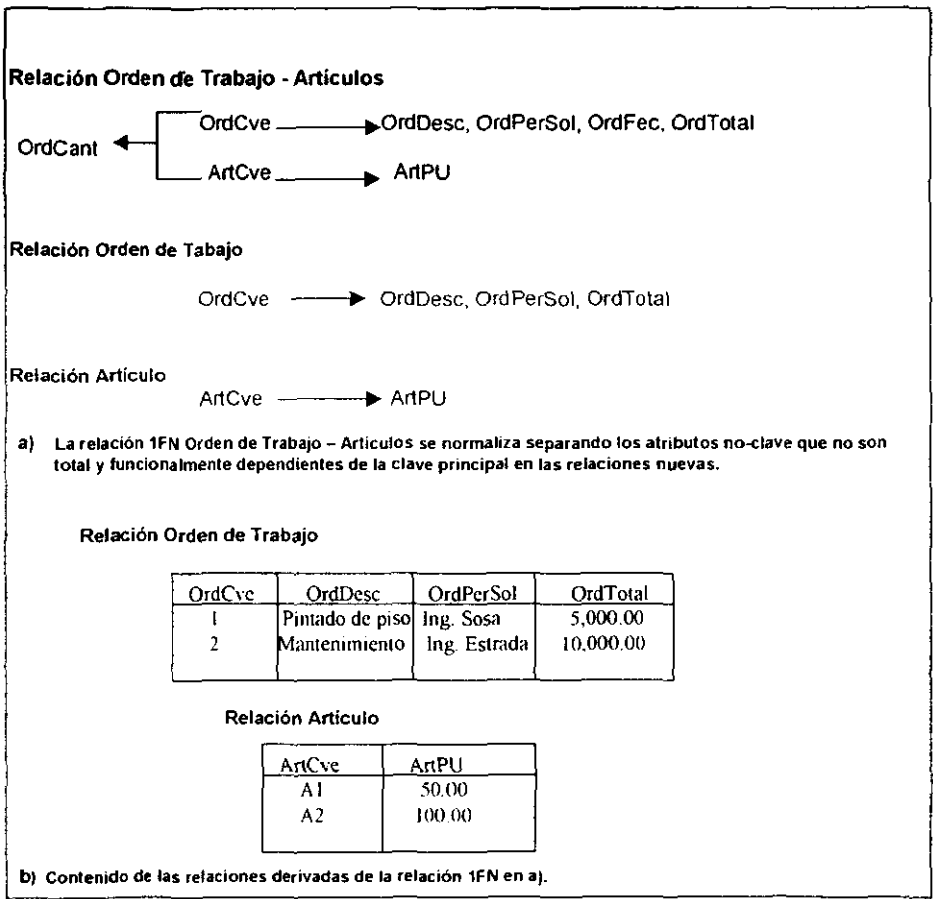


Figura 2.3 Segunda Forma Normal A

Una relación 2FN puede presentar anomalías de almacenamiento si cualquiera de sus no-claves depende transitivamente de la clave primaria si es funcionalmente dependiente de otra no-clave, en otras palabras, depende indirectamente de la clave principal.

Las anomalías de almacenamiento en una relación 2FN son causadas por la dependencia transitiva de no-claves en su clave primaria. Por tanto, una de las soluciones al problema es eliminar la dependencia transitiva de la relación 2FN almacenando las no-claves que son transitivamente dependientes de la clave primaria en una relación nueva y adecuada.

PASO 1. Examinar cada atributo no-clave de la relación Orden de Trabajo - Artículos para determinar si es funcionalmente dependiente de otra no-clave. El diagrama de dependencia funcional mostrado en la figura 2.3 a) indica que ArtPU es directamente dependiente de ArtCve, pero sólo transitivamente dependiente de OrdCve.

PASO 2. Crear una nueva relación para almacenar la no-clave transitivamente dependiente ArtPU y su determinante ArtCve. La relación 2FN se divide en dos relaciones, Orden de Trabajo y Artículos. El atributo ArtCve debe aparecer en ambas relaciones para que sirva como campo de conexión para ligar las relaciones Orden de Trabajo y Artículos. De la figura 2.3 b) se puede determinar que el Precio Unitario para el Artículo A1 es 50.00 al asociar el valor A1 del campo ArtCve, que es común a ambas relaciones.

Una vez que desaparece la dependencia transitiva de una relación, se eliminarán las anomalías de la relación 2FN original. Como se muestra en la figura 2.4.

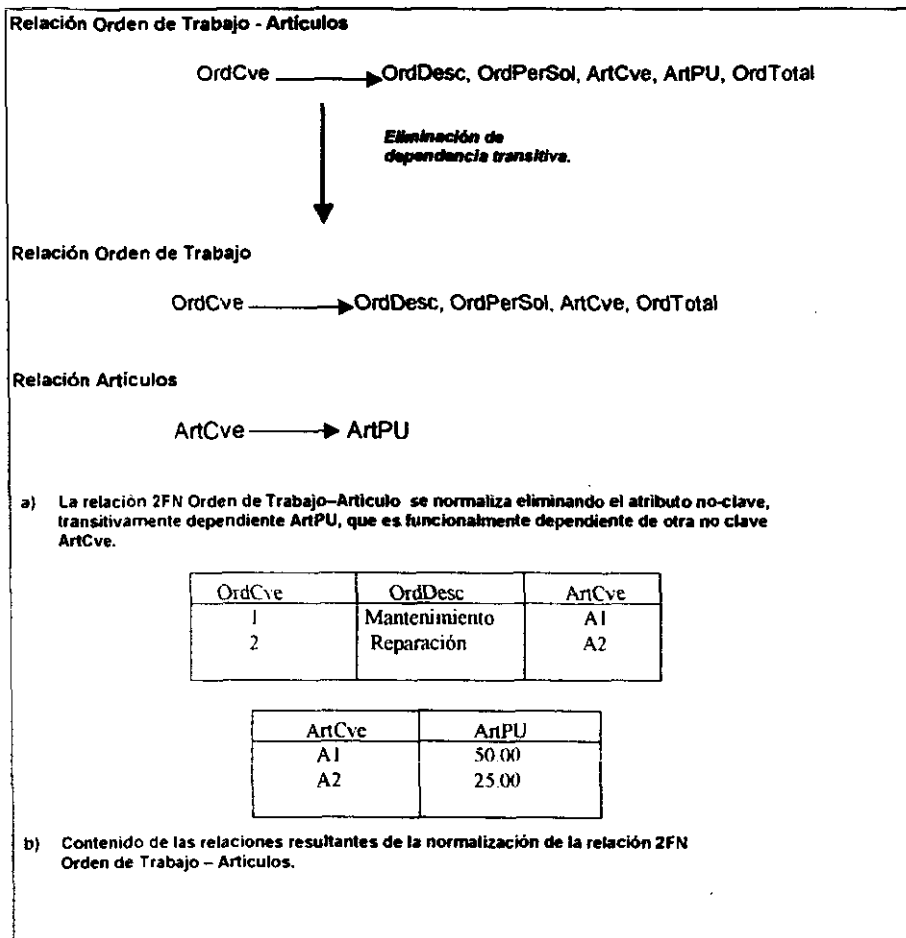


Figura 2.4 Segunda Forma Normal B

2.3.1.5 TERCERA FORMA NORMAL (3FN)

Una relación es *Tercera Forma Normal (3FN)* si es 2FN y ningún atributo no-clave en la relación es funcionalmente dependiente de algún otro atributo no-clave.

Las relaciones derivadas de la figura 2.5 son 3FN. En la mayoría de los casos el proceso de normalización queda completo cuando todas las relaciones derivadas son 3FN.

Codd dio la definición original de 3FN en 1972. Se le corrigió posteriormente y la definición revisada 3FN se conoce como la forma normal de *Boyce, Codd (BCFN Forma Normal de Boyce, Codd)*; una relación es BCFN si cada determinante en la relación es una clave aspirante. Como se mencionó anteriormente, si existe algún atributo que resulte ser total y funcionalmente dependiente de otro, se llama determinante. Una clave aspirante es un atributo o un grupo de atributos cuyo contenido puede representar de manera única a cada registro de una relación.

Cuando en una relación hay más de una clave aspirante, una de las claves aspirantes se designa como la clave primaria.

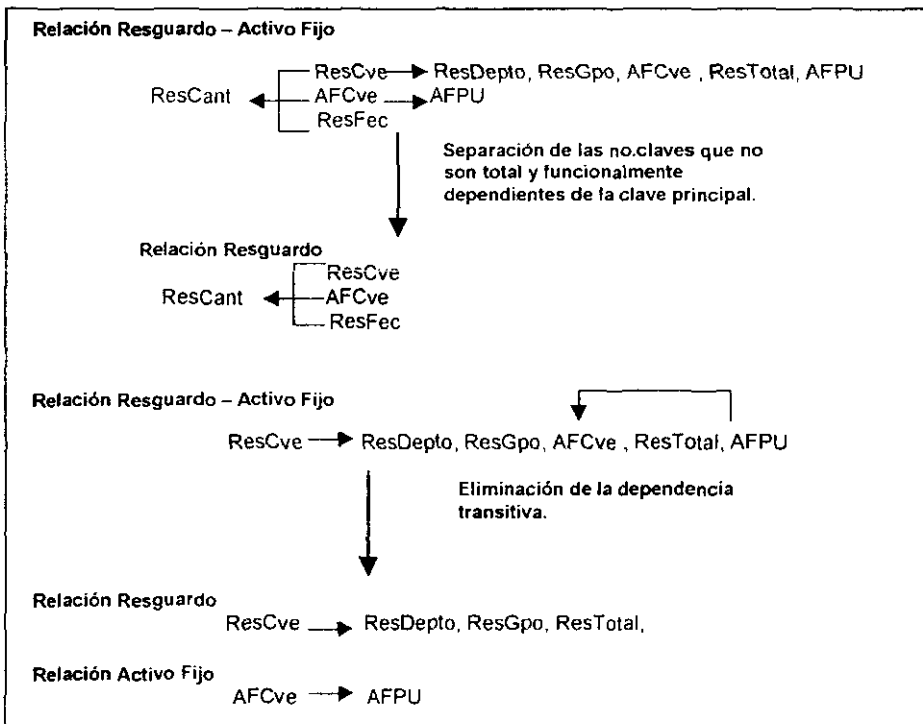
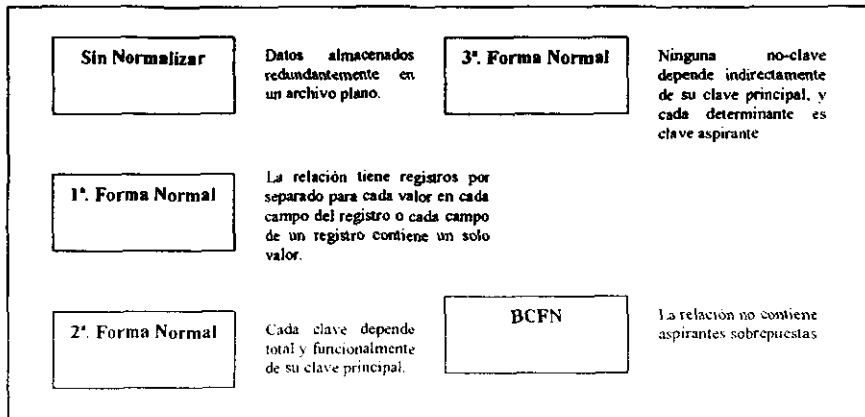


Figura 2.5 Tercera Forma Normal

En la mayoría de los casos, cuando una relación es 3FN también es BCFN. Por ejemplo, las relaciones derivadas de la figura 2.5, son 3FN y también BCFN. Considere la relación Resguardos. Esta relación es 3FN porque ninguna no-clave depende transitivamente de otra clave. También es BCFN ya que cada determinante es clave aspirante. Sin embargo, la definición BCFN es más restrictiva que la 3FN original. En otras palabras, cuando una relación es 3FN, no es necesariamente BCFN. La situación ocurre cuando dos claves aspirantes sobrepuestas están contenidas en una relación.

Mediante distintas interpretaciones de las reglas semánticas, el procedimiento de normalización se repite y se requiere de diversas pruebas con distintas agrupaciones antes de llegar al diseño satisfactorio.



El proceso de normalización descompone una relación inicial universal conteniendo todos los datos del diseño de la base en varias relaciones más pequeñas. Cada una de las relaciones resultantes se examinan para ver si es necesaria una normalización posterior.

Una relación se clasifica como no normalizada, 1FN, 2FN, 3FN, BCFN de la siguiente manera:

- i) Un archivo no plano es una relación sin normalizar donde uno o más atributos contienen valores anidados repetidos.
- ii) Una relación es 1FN si cada uno de sus atributos contiene un solo valor.
- iii) Una relación es 2FN si es 1FN y todos sus atributos no-clave son total y funcionalmente dependientes de la clave principal.
- iv) Una relación es 3FN si es 2FN y ningún atributo no-clave es transitivamente dependiente de la clave principal.
- v) La definición revisada de 3FN se llama BCFN. Es más restrictiva que la definición original de 3FN en que todos los determinantes en una relación BCFN deben ser claves aspirantes.

2.4 MODELO FISICO O EXTERNO

El Modelo Físico o Externo representa la percepción individual de cada programador de la Base de Datos, porque los programadores pueden estar trabajando en distintos subconjuntos de una base integrada.

Este modelo es el más cercano a los usuarios y se refiere al modelo en que cada usuario ve los datos para su propio uso. Así cada usuario accesa a un modelo externo individual. Así como el modelo conceptual, los programadores pueden imaginar que los archivos externos usados por sus programas en la base de datos, existen de la manera que ellos los perciben. Cuando un programa convoca un comando E/S, el DBMS extrae los datos requeridos por los registros lógicos externos de uno o más registros físicos de la base de datos.

Una Base de Datos puede tener varios Modelos Externos, pero sólo un Modelo Conceptual y un Modelo Lógico.

2.5 DIAGRAMA DE FLUJO DE DATOS

Los primeros trabajos sobre modelos de análisis aparecieron a finales de los 60's y principios de los 70's, pero la primera aparición del enfoque de análisis estructurado fue como complemento de otro tema importante "El Diseño Estructurado".

El término "Análisis Estructurado" fue popularizado por DeMarco. En su primer libro sobre esta materia. DeMarco presentó y denominó los símbolos gráficos clave que permitirían a un analista crear modelos de flujo de información; sugirió heurísticas para utilizar esos símbolos; surgió el uso del Diccionario de Datos y narrativas de procesamiento como complementos a los modelos de flujo de información. En los años siguientes, Page-Jones, Gane y Sarson y muchos otros propusieron variaciones del enfoque de análisis estructurado.

El Diagrama de Flujo de Datos es una técnica gráfica que representa el flujo de la información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida.

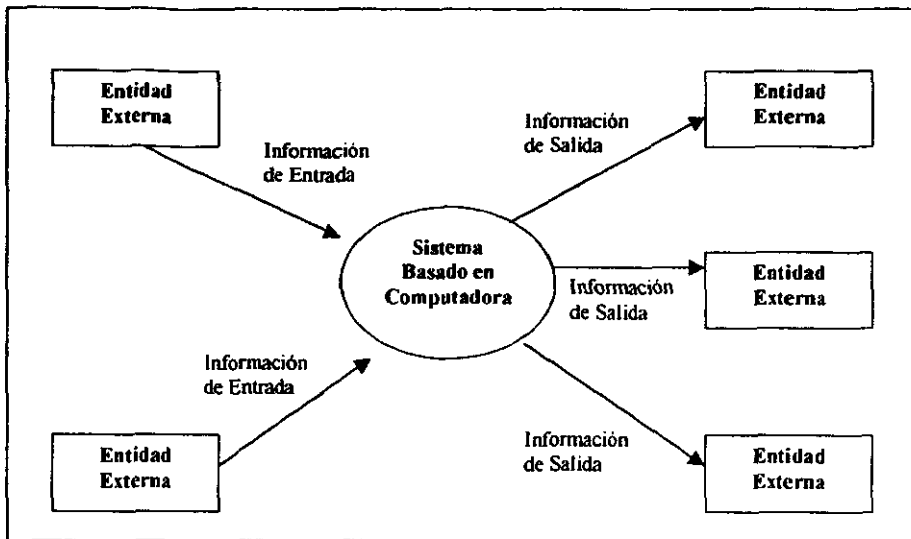
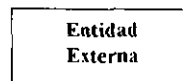


Figura 2.6 Diagrama de Flujo

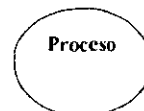
2.5.1 ELEMENTOS DEL DIAGRAMA DE FLUJO DE DATOS

Los diagramas de flujo de datos consisten en procesos, almacenadores, flujos de información y entidades externas.

- Las entidades externas, también denominadas Productores o Consumidores de Información, Son las entidades con las que el sistema se comunica, es por ello que residen fuera de los límites del sistema a ser modelado. Por lo regular se trata de personas, organizaciones, documentos externos u otros objetos que proporcionen o consuman información del sistema.



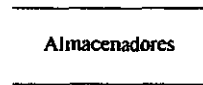
- Los procesos se representan por medio de círculos, o burbujas en el diagrama. Representan las diversas funciones individuales que el sistema lleva a cabo.



- Los Flujos de Datos se representan por medio de flechas y curvas; la cabeza de la flecha indica la dirección del flujo de datos, el cual representa la información que dichos procesos requieren como entrada o la información que genera como salida.



- Los Almacenadores se representan por medio de dos líneas paralelas, y se utilizan para almacenar datos que serán utilizados por uno o más procesos.



2.5.1.1 GUIA PARA LA CONSTRUCCION DE UN DIAGRAMA DE FLUJO DE DATOS

1. Escoger o asignar nombres con significado para los procesos, flujos, almacenes y entidades externas.
2. Numerar los procesos en forma decimal, por ejemplo 1.0, 1.2, etc. Los números sirven para relacionar adecuadamente un proceso con el siguiente nivel del DFD que la describe más a fondo.
3. Redibujar el DFD tantas veces como sea necesario estéticamente.
4. Evitar los DFD excesivamente complejos.
5. Asegurarse de que el DFD sea internamente consistente y que también lo sea con cualquiera DFD relacionado con él.
- 6 Evitar procesos con sólo entradas o salidas, así como flujos y procesos no etiquetados.

2.5.1.2 NIVELES DE UN DIAGRAMA DE FLUJO DE DATOS

Se organiza el DFD global en una serie de niveles de modo que cada uno proporcione sucesivamente más detalles sobre una porción del nivel anterior. Esto es análogo a la organización de mapas en un atlas.

A través del uso de los Diagrama de Flujo de datos se puede representar un sistema o aplicación a cualquier nivel de abstracción, dado que, los DFD's pueden ser refinados en niveles que representan un mayor flujo de información y un mayor detalle funcional. Estos niveles comprenden desde el diagrama de contexto, diagrama de nivel 0 y diagramas de nivel 1, 2, 3,...N.

Diagrama de Contexto

El DFD de primer nivel consta sólo de un proceso, que representa al sistema completo, con datos de entrada y salida respectivamente, los flujos de datos muestran las interfaces entre el sistema y los terminadores externos. El primer nivel de un DFD es el Diagrama de Contexto. El Diagrama de contexto enfatiza las siguientes características importantes del sistema:

- Las personas organizaciones y sistemas con los que se comunica el sistema, se conocen como entidades externas, agentes externos o actores.
- Los datos que el sistema recibe del mundo exterior y que deben procesarse de alguna forma.
- Los datos que el sistema produce y que se envían al mundo exterior.
- Los almacenes de datos que se producen por el sistema y que se envían al mundo exterior.
- La frontera entre el sistema y el mundo exterior.

Diagrama de Nivel 0

El DFD que sigue del Diagrama de Contexto se conoce como Diagrama de Nivel 0. El cual representa la vista de más alto nivel de las principales funciones del sistema, al igual que sus principales interfaces

Diagramas de Nivel 1,2,3,...N

Estos Diagramas de Nivel 1,2,3,...,N son el resultado del refinamiento de cada una de las burbujas o procesos en distintos niveles para mostrar un mayor detalle. Es importante señalar que se debe mantener la continuidad del flujo de información, es decir, que las entradas y las salidas de cada refinamiento deben ser las mismas.

2.5.2 CARACTERISTICAS DE LOS DIAGRAMAS DE FLUJO DE DATOS

En la elaboración de los Diagramas de flujo, se debe observar que cumplan con las siguientes características:

1. Consistencia

La consistencia se obtiene al evitar sumideros infinitos, procesos que tienen entradas pero no salidas, procesos de generación espontánea que tiene salidas sin tener entradas.

2. Balance

El nivel de balance compara las entradas y salidas de un proceso en un diagrama padre con los flujos de un diagrama hijo, es decir, los flujos de información que entran a un DFD hijo, deben estar representados en el padre por los flujos de información.

2.5.2.1 DESCRIPCION DE LOS PROCESOS

La descripción de los procesos, se utiliza para describir los procesos del modelo funcional que aparecen en el final de refinamiento, con el propósito de definir lo que debe hacerse para transformar entradas en salidas, a través de narrativas textuales o descripciones en lenguaje estructurado, que servirán como guía para el diseño del componente de programa que implantará el proceso.

En la especificación de procesos en general se describe lo siguiente:

- La información que entra y la que sale del módulo.
- La información que es retenida en el módulo.
- Una explicación del procedimiento, que indique los principales puntos de decisión y las tareas.
- Un breve tratamiento de las restricciones.

Una descripción de procesos no necesariamente debe realizarse empleando el lenguaje estructurado, también pueden emplearse narrativas textuales que indiquen la función del proceso, las funciones o procedimientos y los parámetros de entrada y salida de tales funciones, sin llegar a detallar.

2.6 DICCIONARIO DE DATOS

La segunda herramienta de modelado importante, aunque no tiene la presencia y atractivo gráfico de los DFD, los diagramas Entidad-Relación o los diagramas de estructuras, es el Diccionario de Datos.

El Diccionario de Datos es un listado organizado de todos los datos pertinentes al sistema, con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento común de todas las entradas, salidas, componentes de los almacenes y cálculos intermedios. El Diccionario de Datos define los datos haciendo lo siguiente:

- Describe el significado de los flujos y almacenes que se muestran en los DFD.
- Describe la Composición de agregados de paquetes que se mueven a lo largo de los flujos, es decir, paquetes complejos que pueden descomponerse en unidades más elementales.
- Describen la composición de los paquetes de datos en los almacenes.
- Especifica los valores y unidades de piezas elementales de información en los flujos de datos y en los almacenes de datos.
- Describe los detalles de las relaciones entre almacenes que se enfatizan en un diagrama de Entidad-Relación y otro modelo de datos.

El diccionario de datos es una especialidad dentro de los diccionarios de referencia que se utiliza en la vida diaria. El diccionario de datos es una referencia de datos acerca de los datos, recopilados por el analista de sistemas para guiarse sobre el análisis y el diseño de un sistema.

Como documento, recopila y confirma lo que un término específico significa para la gente de una organización. Los diagramas de flujos de datos son un buen inicio para la recopilación de los términos del diccionario de datos. El diccionario de datos sirve también como el estándar consistente de los datos elementales.

Los Diccionarios de Datos automatizados (que también forman parte de los instrumentos CASE) son valiosos porque permiten la referencia cruzada de datos sencillos. Además estos diccionarios pueden ser relevantes para los grandes sistemas que producen varios miles de datos elementales que requieren ser catalogados y así contar con referencias cruzadas.

2.6.1 CONTENIDO DE UN DICIONARIO DE DATOS

Una manera de saber lo que debe contener un Diccionario de Datos, es visualizar cómo llegará a utilizarse. Es el elemento básico de referencia para localizar los nombres y atributos de los datos utilizados en todo sistema de organización. Por esto se deben incluir todos los datos elementales.

Con el fin de ser utilizados, los registros del Diccionario de Datos deben contener información referente a las categorías siguientes:

- El nombre y sinónimo del dato.
- Las descripciones del dato.
- Los datos elementales que se relacionan con el término.
- El rango permitido del dato.
- La longitud disponible en caracteres.
- Una adecuada codificación.
- Cualquier otra información pertinente de edición.

Nombres y sinónimos

El Diccionarios de Datos debe contener el nombre de cada dato, esto es, la manera de denominar el dato en la mayoría de los programas, y su sinónimo. Todo esto debe quedar registrado en el Diccionario de Datos para facilitar la comunicación entre los departamentos y sus programas.

Descripción

Debe incluir una descripción textual del dato elemental, la cual debe ser concisa pero informativa para cualquiera que la consulte.

Rango permitido

Debe incluir los distintos rangos y límites que se aplican al elemento. El rango significa el intervalo disponible de datos.

Longitud del Dato

Se refiere a la longitud permitida para el acceso a un dato elemental. La longitud siempre se da en función del número de caracteres impresos y no por la cantidad requerida de memoria.

Codificación adecuada

Cada dato debe incorporarse al Diccionario de Datos junto con su código, si es que lo tiene y el significado de este. Es indispensable que la codificación sea consistente.

La información requerida para asegurar la adecuada edición de los datos debe estar presente en el Diccionario de Datos. Esto incluye a cualquier orden pertinente. Cuando el Diccionario de Datos se integra de manera correcta, es útil para el desarrollo del sistema la codificación del mismo y su mantenimiento.

2.6.2 FUNCIONES DEL DICCIONARIO DE DATOS

Un Diccionario de Datos es una Base de Datos por derecho propio: contiene datos sobre los datos. Algunas veces es llamado *diccionario de meta-datos* porque describe todos los elementos de los datos de la base. El DBA usa el Diccionario de Datos en cada etapa de un ciclo de vida de la base, empezando con la etapa embrión de la acumulación de datos hasta la de diseño, implantación y mantenimiento.

El Diccionario de Datos ideal debe incluir todo lo que el DBA quiera saber acerca de la base:

- 1) Descripciones externa, conceptual e interna de la Base de Datos.
- 2) Descripciones de entidades (tipos de registro), atributos (campos), así como referencias cruzadas, origen y significado de los elementos de los datos.
- 3) Sinónimos, homónimos y código de autorización y seguridad.
- 4) Qué esquemas externos son usados y por qué programas, quiénes son los usuarios y que autorizaciones o derechos tienen.

Los elementos de datos, con el mismo nombre pero distinto significado se llaman homónimos; nombres diferentes que representan los mismos elementos de datos se llaman sinónimos.

CAPITULO III. INFRAESTRUCTURA TECNOLOGICA

3.1 REQUERIMIENTOS DEL SISTEMA

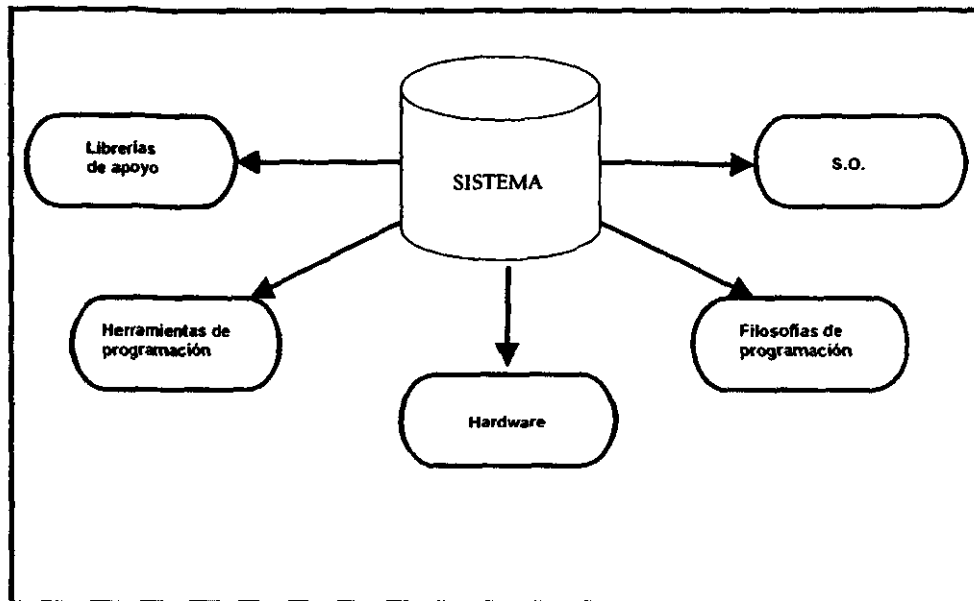
Los alcances y limitaciones de cualquier proyecto se definen cuando se conocen los requerimientos de usuario. Siendo esta la parte más importante en el ciclo de vida del desarrollo de sistemas. La información sobre la factibilidad y alcance del proyecto, los requerimientos de operatividad de datos y los datos mismos, son los elementos que permiten al diseñador conocer a fondo toda la problemática para posteriormente ofrecer una solución.

La responsabilidad del usuario es comunicar todo lo que desea que haga el sistema. Esto es de suma importancia ya que es aquí en donde se va a recopilar toda la información. Pero se debe tener cuidado ya que la duración del desarrollo del sistema se puede ir alargando si el usuario falla al proporcionar las respuestas detalladas durante el análisis.

Este proceso requiere de una intensa comunicación entre el usuario y el diseñador del sistema contando con una definición exacta de lo que va a ser el sistema. Es por ello que se debe de informar a las personas que están ligadas con los procesos a automatizar sobre la importancia del estudio que se efectuará para implantar el nuevo sistema, así como los beneficios que se obtendrán al termino de éste y por consiguiente la necesidad de contar con información verídica proveniente de ellos para poder cumplir con los objetivos planteados al principio.

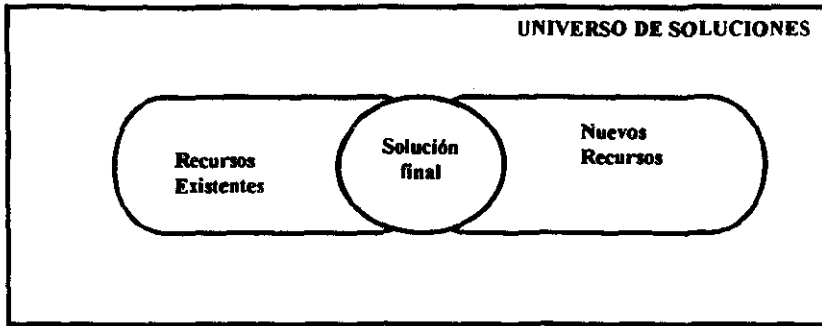
Otro punto también que se consideraba de gran importancia es saber en que plataforma se va a desarrollar el sistema ya que se puede contar con muchas opciones para la solución de problemas relacionados con el proceso de un sistema computacional, desde múltiples opciones de hardware, sistemas operativos, hasta las herramientas de programación.

Estas opciones se multiplican más considerando que dentro de una herramienta de programación hay varias opciones como son: Librerías de apoyo, algoritmos, filosofías de programación, etc. Ver figura 3.1. Opciones de una herramienta de programación.



Para poder encontrar la mejor solución tenemos que centrarnos bien en el problema principal y a partir de él derivar la opción más idónea para el desarrollo previendo todos los requerimientos que debe tener un buen sistema.

El elegir una opción no es del todo fácil, ya que se tiene que considerar que se puede contar con la opción de comprar nuevos recursos o utilizar los que se tienen. En caso de que se elijan los recursos ya existentes, se pueden acortar las opciones de solución, que encaminan a un universo de soluciones, pero que dentro de él se puede encontrar la solución óptima, es decir, dar el mayor rendimiento con el mínimo de recursos. Ver figura 3.2.Opciones de solución.



Tenemos que elegir y elegir bien todos los requerimientos de nuestro sistema como lo es:

- Un Sistema Operativo
- Un Manejador de Base de Datos
- Un Lenguaje de Programación

3.1.1 SISTEMA OPERATIVO

Dada la magnitud del Sistema a desarrollar, se requería de un Sistema Operativo capaz de soportar la operación de un gran número de usuarios en forma simultánea, compatible con diferentes tipos de clientes, como por ejemplo Microsoft, y con capacidad de correr en diferentes plataformas; que maneje adecuadamente la memoria y el multiprocesamiento simétrico, es decir, que el sistema operativo pueda asignar diferentes tareas a un mismo procesador; de esta manera, si se tienen varios trabajos y uno se termina antes que otro, el sistema operativo podrá ocuparse de otra actividad.

Investigando se llegó a la conclusión de que para nuestro sistema se utilizaría Windows 95 ó Windows NT de Microsoft, ya que reunían los requisitos y el potencial para el desarrollo requerido, de tal manera que se procedió a realizar un análisis de ambos sistemas, obteniéndose las siguientes conclusiones.

3.1.2 MS. WINDOWS 95 Y MS. WINDOWS NT

Estos Sistemas Operativos han venido a mejorar el rendimiento general, es decir, la combinación del rendimiento del microprocesador, transferencia de datos y acceso a memoria.

Windows 95 ha sido perfeccionado para que su manejo le resulte más rápido y sencillo. La multitarea es una de sus características primordiales; está diseñado para manejar sin esfuerzo las necesidades múltiples y simultáneas de un usuario, es decir, trabaja también en un entorno multiusuario que soporta las operaciones de más de una persona a la vez, esto es, cuando se tiene una red.

Windows NT es un sistema operativo diseñado para operaciones de 32 bits, al igual que Windows 95. Esto le permite sacar ventaja de los microprocesadores más avanzados, como los Intel 80386, 80486 y Pentium; así como de las computadoras con tecnología RISC. Los microprocesadores de 32 bits pueden trabajar con cantidades numéricas, direcciones de memoria e instrucciones de mayor tamaño sin necesidad de dividir las operaciones en bloques más pequeños para que puedan tener cabida en el microprocesador.

La multitarea, al igual que Windows 95, es una de sus características y esto significa que el sistema operativo puede realizar varias tareas al mismo tiempo. En este esquema, la tarea o usuario que se identifique como preferente, puede interrumpir la secuencia determinada si fuese necesario, sin tener que esperar obligatoriamente a que la fila termine.

En Windows NT existe la posibilidad de concurrencia de diversas tareas, de manera que si, por ejemplo, una de ellas queda parcialmente parada como consecuencia de un lento acceso al disco, el microprocesador puede concentrar su atención en otras tareas. Básicamente, lo que se consigue es que no se malgasten ciclos de procesamiento. Puede verse cada tarea individual como un corredor de velocidad. Si decae el ritmo de uno de ellos, los otros no frenan su marcha. El beneficio de esto es que los usuarios pueden realizar otras tareas mientras tiene lugar un acceso al disco en segundo plano, o bien, se prepara un trabajo de impresión.

La protección de memoria asegura que los diversos programas van a ser ejecutados cada uno en su parte individual de memoria, no pudiendo contaminar las áreas correspondientes a otras aplicaciones. En el caso de que una determinada aplicación quede bloqueada, las aplicaciones restantes, así como el sistema operativo permanecerán inalterados, dando así la oportunidad al usuario de dar fin a su trabajo en la forma correcta.

El microprocesamiento simétrico es una característica exclusiva de Windows NT, que permite sacar todo el partido de los procesadores múltiples. Si bien es cierto que la vida de los sistemas multiproceso es relativamente corta, NT es el primer sistema operativo que está preparado para realizar un uso eficaz de este tipo de tecnología.

Las computadoras donde se ejecuta Windows para Grupos de Trabajo pueden aprovechar las ventajas del trabajo en red. Además, Windows NT viene equipado con el software y los controladores necesarios para establecer las conexiones con otros tipos de sistemas operativos.

3.1.3 MANEJADORES DE BASES DE DATOS (DBMS)

Una vez identificados los posibles Sistemas Operativos a utilizar, se requería también de un manejador de base de datos que se acoplara al sistema operativo a utilizar; que fuera robusto y manejara altos niveles de seguridad, distribución de bases de datos, mecanismos de respaldo y espejeo de información, entre otras características.

Es por ello que decidimos utilizar Microsoft Access versión 7.0 para Windows como Manejador de Base de Datos, ya que por sus características y por ser un recurso que ya se tenía se decidió adoptar este tipo de Manejador de Base de Datos.

Para esto, también se realizó un análisis de lo que era Access, obteniéndose las siguientes características:

Microsoft Access para Windows, versión 7.0, es una potente herramienta de gestión de datos que se puede utilizar para ordenar, organizar o presentar la información que necesitamos cada día. Su fácil manejo hace que sea una excelente herramienta tanto para aquellas personas que son nuevas en el uso de las Bases de Datos como para aquellas personas que tienen mucha experiencia con ellas.

En resumen, la plataforma que se decidió adquirir y utilizar para el desarrollo del sistema debido a que se adapta mejor a los requerimientos técnicos y económicos de la DGO y S.G., fueron los siguientes: como Sistema Operativo: Windows 95 o Windows NT, y como Manejador de Bases de Datos: Access versión 7.0. Como se muestra en la figura 3.3.

Hardware

C.P.U. compatible con IBM
Procesador 486 o superior
12 MB en memoria RAM (mínimo)
12 MB de espacio en disco duro (HD)
Mouse
Monitor VGA

Software

Windows 95 o Windows NT

Figura 3.3 Equipo Seleccionado para el SICIPA

3.2 LENGUAJE DE PROGRAMACION

Una vez seleccionada la infraestructura tecnologica a utilizar para el proyecto, es decir, ya contando con el Sistema Operativo y el Manejador de Base de Datos, se realizo la investigaci3n de lo relacionado al Lenguaje de Programaci3n y las herramientas adicionales que debían utilizarse para el proyecto.

Para el desarrollo del proyecto se selecciono una herramienta "CASE" (Computer Aided Software Engineering Tools), la cual es una herramienta de soporte automatizado, las herramientas CASE son "Tecnologias de ambientes integrados" ya que estas herramientas incrementan la productividad, se comunican con mayor eficacia con los usuarios, y se puede integrar el trabajo que se realiza sobre el sistema, desde el principio hasta el final del ciclo de desarrollo.

Las herramientas CASE facilitan la interacci3n entre los miembros del grupo al permitir que la elaboraci3n de diagramas sea el proceso dinámico e imperativo, más que aquellos en el cual los cambios son tediosos; y en consecuencia tiendan a disminuir la productividad.

La interacci3n de actividades a través del uso de "Tecnología de ambiente integrado" tambien mejora la comprensi3n de los usuarios sobre como se encuentran relacionadas cada una de las partes del ciclo de vida, así como su interdependencia.

Mencionamos algunas ventajas de utilizar Herramientas CASE

- Promueven el cumplimiento de la metodología de desarrollo de sistemas.
- Facilitan la comunicación entre los usuarios que intervienen en el proceso, lo que incrementa la creatividad de los mismos enriqueciendo el producto final.
- Proveen una mejor visión del sistema.
- Facilitan la elaboración de documentación estándar.
- Reducen costos
- Dan un margen de tiempo más amplio para revisiones.
- Permiten liberar un sistema confiable, eficiente y versátil.
- Permiten alcanzar un nivel competitivo sobre otros en la misma línea de trabajo.
- Aportan un vocabulario común para analistas y usuarios.

Las herramientas de la ingeniería de software asistida por computadora abarcan cada etapa del proceso de ingeniería y cada actividad que se desarrolla a lo largo del mismo. El CASE está formado por un conjunto de bloques que comienzan en el nivel de hardware y del sistema operativo y acaban en cada una de las herramientas.

En la figura 3.4 se representan los bloques que componen el CASE. Los buenos entornos de ingeniería del software se construyen sobre una arquitectura de entorno que engloban los correspondientes sistemas de software y de hardware.

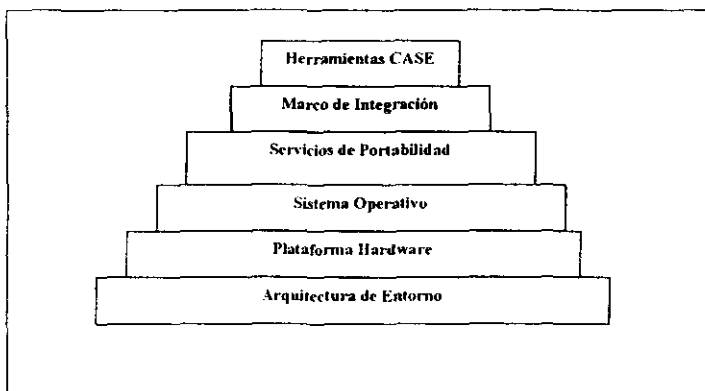


Figura 3.4 Bloques constitutivos del CASE

La arquitectura de entorno, compuesta por la plataforma hardware y el soporte del sistema operativo (incluida la red y la gestión de la Base de Datos) constituye la base del CASE. Pero el entorno CASE, en sí mismo, necesita otros componentes. Un conjunto de servicios de portabilidad constituyen un puente entre las herramientas CASE y su marco de integración y la arquitectura de entorno.

El marco de integración es un conjunto de programas especializados que permiten a cada herramienta CASE comunicarse con las demás; para crear una Base de Datos de proyectos y mostrar una apariencia homogénea al usuario final. Los servicios de portabilidad permiten que las herramientas CASE y su marco de integración puedan migrar a través de diferentes plataformas hardware y sistemas operativos sin grandes esfuerzos de adaptación.

Las herramientas CASE se pueden clasificar por su función, por su papel como instrumentos para el personal técnico o los directivos, por la arquitectura del entorno (hardware y software) que las soportan o incluso por su origen y costo.

3.2.1 HERRAMIENTAS DE PLANIFICACION DE SISTEMAS DE GESTION

Estas herramientas proporcionan un metamodelo del cual se pueden obtener sistemas de información específica, la información de gestión en los requisitos de una aplicación específica, la información de gestión se modeliza según va pasando a través de las distintas entidades organizativas de una compañía. El objetivo principal de las herramientas de esta categoría es ayudar a comprender mejor cómo se mueve la información entre las distintas unidades organizativas.

Estas herramientas proporcionan una ayuda importante cuando se diseñan nuevas estrategias para los sistemas de información y cuando los métodos y sistemas actuales no satisfacen las necesidades de la organización.

3.2.2 HERRAMIENTAS DE GESTION DE PROYECTOS

Utilizando un conjunto seleccionado de herramientas CASE, el director del proyecto puede hacer estimaciones útiles de esfuerzo, costo y duración de un proyecto, definir una estructura de partición del trabajo (EPT) y hacer una planificación realista del mismo y hacer el seguimiento de proyectos de forma continua.

Existen herramientas CASE para hacer un seguimiento que va desde los requisitos de la petición de propuesta inicial del cliente, hasta el trabajo de desarrollo que convierte estos requisitos en un producto final.

Herramientas de planificación de proyectos

Las herramientas que caen dentro de esta categoría se centran en dos áreas principales: el esfuerzo y costo de un proyecto de software; y la planificación del proyecto.

Las herramientas de planificación de proyectos permiten al director de proyecto definir todas las tareas (estructura de partición del proyecto), crear una red de tareas (utilizando normalmente una entrada gráfica), representar las interdependencias entre tareas y modelizar la cantidad de paralelismo posible dentro del proyecto.

Herramientas de seguimiento de requisitos

El objetivo de la herramienta es combinar la evaluación interactiva de texto con un sistema de gestión de base de datos que almacena y categoriza cada registro del sistema extraído de las especificaciones originales. Como se muestra en la figura 3.5.

Herramientas de gestión y medida

Estas herramientas de medida se centran en las características del producto y del proceso. Las herramientas orientadas a la gestión parten de medidas específicas del proyecto que proporcionan una indicación global de la productividad y de la calidad.

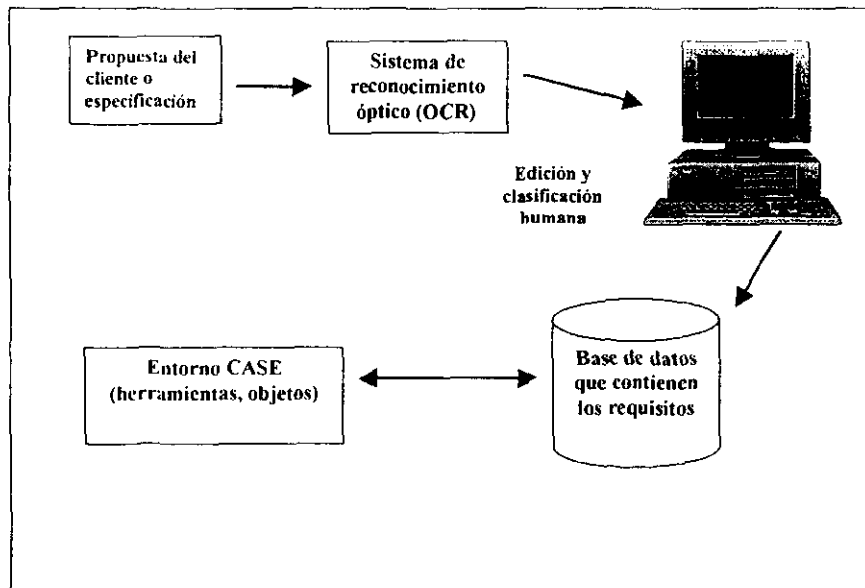


Figura 3.5 Herramientas de Seguimiento de Requisitos

3.2.3 HERRAMIENTAS DE SOPORTE

Estas incluyen herramientas de documentación, herramientas para gestión de redes y software del sistema, herramientas de control de calidad y herramientas de gestión de bases de datos y de configuración de software.

Herramientas de documentación

No es raro que una empresa emplee el 20 ó el 30 por ciento de su esfuerzo de desarrollo en la documentación. Por esta razón, estas herramientas constituyen una opción importante para aumentar la productividad. Las herramientas de documentación suelen estar unidas a otras herramientas CASE por medio de una interfaz de datos suministrada por el vendedor.

Herramientas para software de sistemas

El CASE es una tecnología de estaciones de trabajo. Por esto, el entorno CASE debe soportar software de redes de comunicación de alta calidad, correo electrónico, boletines electrónicos y otras posibilidades de comunicación.

Herramientas de control de calidad

Estas son en realidad herramientas de medida que comprueban el código fuente para determinar su compatibilidad con lenguajes estándar. Otras herramientas extraen técnicas métricas como base para medir la calidad del software que se está desarrollando.

3.2.4 HERRAMIENTAS DE ANALISIS Y DISEÑO

Las herramientas de análisis y diseño permiten al ingeniero de software crear un modelo del sistema que se va a construir. El modelo contiene una representación del sistema que va a construir, contiene una representación de los datos, del flujo del control, del contenido de los datos, representaciones de los procesos, especificaciones de control y otras representaciones del modelo. Las herramientas de análisis y diseño permiten la creación de un modelo y también la evaluación de la calidad del modelo. Mediante la comprobación de la validez y la consistencia del modelo, estas herramientas proporcionan cierto grado de confianza en la representación del análisis y ayudan a eliminar errores antes de que se propaguen al diseño, o lo que es peor, el código mismo.

Herramientas de AE/DE

La mayoría de las herramientas de diseño y análisis se basan en el método de análisis y diseño estructurado (AE/DE). Esta es una técnica de modelización. Permite crear progresivamente modelos más complejos de un sistema, comenzando en el nivel de requisitos y concluyendo con un diseño de la arquitectura.

Herramientas PRO/SIM

Las herramientas de creación de prototipos y de simulación (PRO/SIM) proporcionan la capacidad de predecir el comportamiento de un sistema de tiempo real antes de que sea construido. Proporcionan una visión general de la función, de la operación y de la respuesta, antes de la codificación final.

Herramientas para el diseño y el desarrollo de interfaces

Son un conjunto de componentes de software, tales como menús, botones, estructuras de ventanas, iconos, mecanismos de visualización, controladores de dispositivos y otros elementos de este tipo. Sin embargo, estos conjuntos de herramientas están siendo reemplazados por herramientas para desarrollar prototipos que permiten la creación rápida en pantalla de interfaces sofisticadas ajustadas al estándar elegido para el software.

Máquinas de análisis y diseño

Utilizan una arquitectura basada en reglas que permite que la herramienta sea adaptada a cualquier método de análisis y diseño. Las máquinas de análisis y diseño permiten que el ingeniero de software personalice la herramienta para satisfacer los requisitos.

3.2.5 HERRAMIENTAS DE PROGRAMACION

En esta clasificación se engloban los compiladores, los editores y los depuradores que se utilizan con los lenguajes de programación convencionales. Entran en esta categoría los entornos de programación orientados a los objetos (O-O), los lenguajes de cuarta generación, los generadores de aplicaciones y los lenguajes de consulta a bases de datos.

Herramientas de Codificación Convensionales

Hace tiempo las únicas herramientas de las que disponía un ingeniero de software eran las herramientas de codificación convensionales —compiladores, editores y depuradores. Hoy en día, estas herramientas siguen existiendo en primera línea del desarrollo del software, pero están respaldadas por las otras herramientas CASE.

Herramientas de codificación de cuarta generación

Los sistemas de consulta a Bases de Datos, los generadores de código y los lenguajes de cuarta generación han cambiado la forma en que se desarrollan los sistemas. No hay duda de que el objetivo final del CASE es la generación automática de código esto es, la representación de sistemas a un nivel de abstracción más alto que el de los lenguajes de programación convensionales. Idealmente, estas herramientas de generación de código no sólo traducirán la descripción de un sistema a un programa operativo, sino que también ayudarán a verificar la corrección de la especificación del sistema, de tal forma que la salida resultante satisfaga los requisitos del usuario.

Herramientas de programación orientadas a los objetos

Los entornos de programación orientados a los objetos suelen estar unidos a lenguajes de programación específicos (p. ej.: C++, Visual C++ o Visual Basic). Un entorno O-O típico incorpora características de los interfaces de tercera generación (ratón, ventanas, menús desplegables, operaciones sensibles al contexto, multitarea) con funciones especializadas –una función que permite examinar todos los objetos contenidos en unas bibliotecas de objetos para determinar si pueden ser reutilizados en la aplicación actual.

Lo que se pretende al utilizar una herramienta CASE es llevar una metodología incremental, la cual se construya mediante una serie de aproximaciones sucesivas, acompañando el proceso de aprendizaje natural.

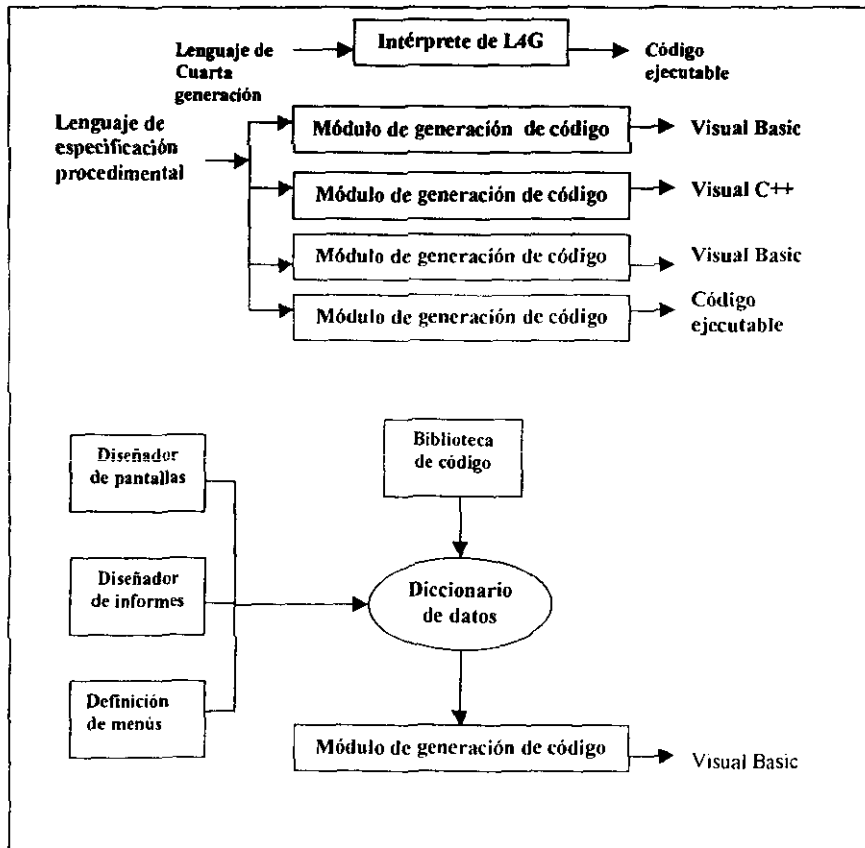


Figura 3.6 Herramientas de Programación

Es por todas estas ventajas que se decidió adoptar como herramienta de desarrollo a **GENEXUS**, la cual es una herramienta CASE inteligente, cuyo objetivo es asistir tanto al analista como a los usuarios en todo el ciclo de vida de las aplicaciones y por que GeneXus trabaja en un ambiente Windows 95 o Windows NT.

Otra de las razones del porque se eligio GeneXus es porque la Base de Datos y los programas de aplicación son generados y/o mantenidos en forma totalmente automática, para el ambiente de producción. GeneXus automatiza todo aquello que pueda ser automatizado como es la normalización, el diseño de los datos, la generación y mantenimientos de la Base de Datos; evitando así dedicarse a tareas rutinarias y tediosas, permitiendo poner la mayor atención en todo aquello que nunca un programa podrá hacer: entender los problemas del usuario.

3.2.6 GENEXUS

GENEXUS es una herramienta inteligente, desarrollada por ARTech, cuyo objetivo es asistir al analista y a los usuarios en todo el ciclo de vida de las aplicaciones.

El diseño y prototipos son realizados y probados en una PC, en un ambiente Windows 95 o Windows NT. Cuando el prototipo es totalmente aprobado por sus usuarios, la base de datos y los programas de aplicación son generados y/o mantenidos en forma totalmente automática, para el ambiente de producción.

GENEXUS está disponible para diferentes ambientes de producción:

- IBM AS/400, arquitectura centralizada
- Microcomputadores y/o Redes
- Cliente/Servidor (Two Tier o Multi Tier Architecture).

Como un subproducto, **GENEXUS** ofrece una documentación rigurosa, autosuficiente y permanentemente actualizada.

A continuación presentamos los componentes de **GENEXUS**:

- ◆ Diseño
- ◆ Prototipo
- ◆ Producción

Dentro de cada uno de los componentes se engloban lo que es:

- ◆ Base de conocimiento
- ◆ Transacciones
- ◆ Paneles de trabajo
- ◆ Reportes
- ◆ Procedimientos
- ◆ Menús
- ◆ File View
- ◆ Web Panels
- ◆ Reglas
- ◆ Subrutinas
- ◆ Propiedades
- ◆ Variables

Para poder utilizar **GENEXUS** se requiere que por lo menos su equipo cumpla con las siguientes características.

- ◆ C.P.U. compatible con IBM
- ◆ Procesador 486 o superior
- ◆ 8 MB en memoria RAM(recomendable 12)
- ◆ 12 MB de espacio en disco duro (HD)
- ◆ Mouse
- ◆ Monitor VGA
- ◆ Windows 95 o Windows NT

Generadores Externos

- ◆ Visual Basic versión 5.0 ó 6.0
- ◆ Visual Fox Pro etc.

3.2.7 UNA IMPLEMENTACION DEL DESARROLLO INCREMENTAL CON GENEXUS

Cuando una aplicación se desarrolla con *GENEXUS* la primera etapa consiste en hacer el Diseño de la misma. Posteriormente se pasa a la etapa de Prototipación en donde *GENEXUS* genera la base de datos y programas para el ambiente de prototipo. Una vez generado el Prototipo debe ser puesto a prueba por el ingeniero de software y el usuario. Si durante la prueba del Prototipo se detectan mejoras o errores se retorna a la fase de Diseño, se realizan las modificaciones correspondientes y se vuelve al Prototipo. Llamaremos a este ciclo de Diseño/Prototipo.

Una vez que el Prototipo está aprobado, se pasa a la etapa de Implementación, en donde *GENEXUS* genera, también automáticamente, la base de datos y programas para el ambiente de Producción.

En resumen, una aplicación comienza con un Diseño, luego se Prototipa, luego se Implementa y en cualquiera de los pasos anteriores se puede regresar al Diseño para realizar modificaciones. Como se muestra en la figura 3.2.4.

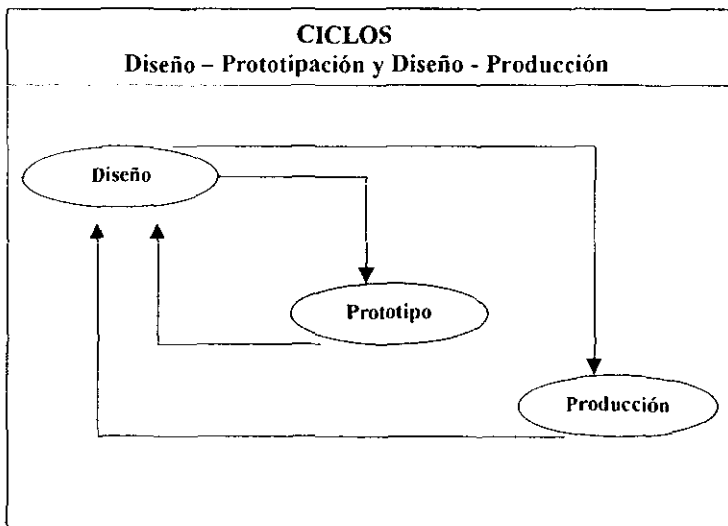


Figura 3.7 Implementación del Desarrollo Incremental con Genexus

3.2.7.1 Diseño

Esta tarea es realizada conjuntamente por el analista y el usuario, y consiste en identificar y describir las visiones de datos de los usuarios. El trabajo se realiza en el ambiente del usuario. Este esquema permite trabajar con un bajo nivel de abstracción, utilizando términos y conceptos que son bien conocidos por el usuario final.

Una consecuencia muy importante, es que la actitud del usuario se transforma en francamente participativa. El sistema pasa a ser una obra conjunta y, como el usuario sigue permanentemente su evolución, su calidad es mucho mejor que la habitual.

De acuerdo al lo visto, *GENEXUS* captura el conocimiento por medio de visiones de objetos de la realidad del usuario.

Los tipos de objetos soportados por *GENEXUS* son: **Transacciones, Reportes, Procedimientos, Work Panels, Web Panels, Menues, Data Views, Styles y Transacciones de Data Warehouse.**

La tarea de diseño consiste, fundamentalmente, en identificar y describir estos objetos. A partir de esto *GENEXUS* sistematiza el conocimiento capturado y va construyendo, en forma incremental, la Base de Conocimiento. Esta Base de Conocimiento es un repositorio único de toda la información del diseño, a partir de la cual *GENEXUS* crea el modelo de datos físico (tablas, atributos, índices, redundancias, reglas de integridad referencial, etc). Así la tarea fundamental en el análisis y diseño de la aplicación se centra en la descripción de los objetos *GENEXUS*.

3.2.7.2 Transacciones

Una transacción es un proceso interactivo que permite a los usuarios crear, modificar o eliminar información de la base de datos. Comúnmente es conocido por los usuarios como una "pantalla".

Ejemplos:

Pantalla para crear, modificar o eliminar los Artículos de un Almacén

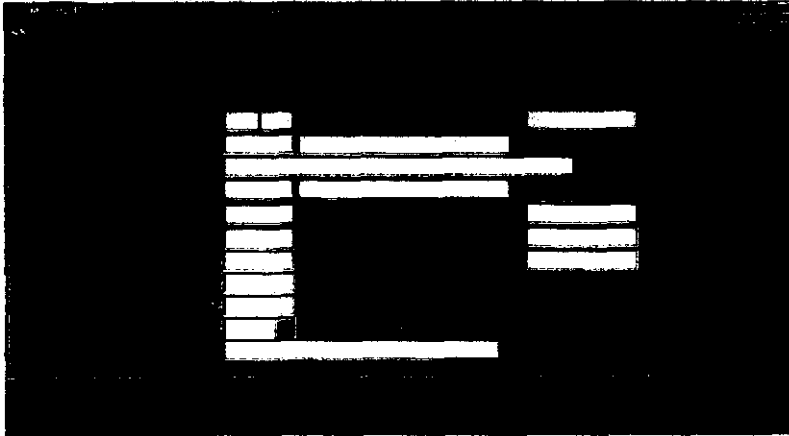


Figura 3.8 Ejemplo del Diseño de una Transacción de Artículos

Pantalla de movimientos: proceso que permite a un usuario crear movimientos e incluso imprimirlos.

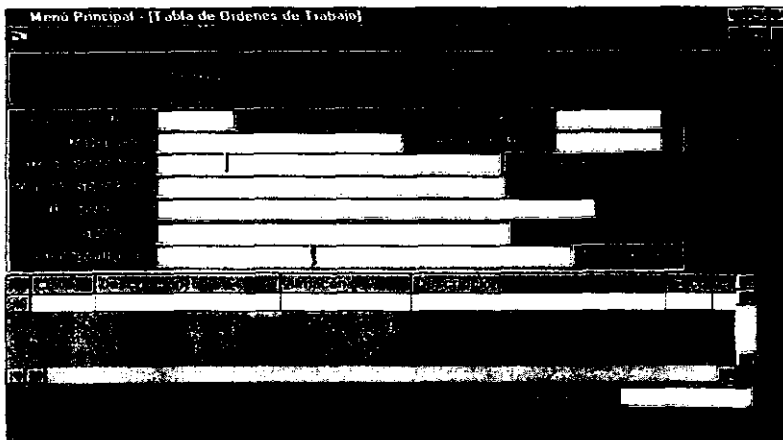


Figura 3.9 Ejemplo del diseño de una Transacción de Ordenes de Trabajo

Una pantalla permite al usuario tomar diferentes acciones como insertar, actualizar, eliminar e imprimir.

3.2.7.3 Reportes

Un informe es un proceso que permite visualizar los datos de la base de datos. La salida del listado puede ser enviada a pantalla o a la impresora (y con ello tenemos un listado convencional).

Artículo	Cantidad	Precio	Existencia	Valor	Costo	Diferencia	Porcentaje
01	10	100	10	1000	1000	0	0.00
02	20	50	20	1000	1000	0	0.00
03	30	30	30	900	900	0	0.00
04	40	20	40	800	800	0	0.00
05	50	10	50	500	500	0	0.00
06	60	5	60	300	300	0	0.00
07	70	2	70	140	140	0	0.00
08	80	1	80	80	80	0	0.00
09	90	0.5	90	45	45	0	0.00
10	100	0.1	100	10	10	0	0.00

Figura 3.10 Diseño de Reportes

Con este objeto se pueden definir desde listados simples (por ejemplo, listar los artículos) hasta muy sofisticados, en donde existan varios cortes de control, múltiples lecturas a la base de datos y parametrización. Sin embargo un informe no actualiza la base de datos.

3.2.7.4 Procedimientos

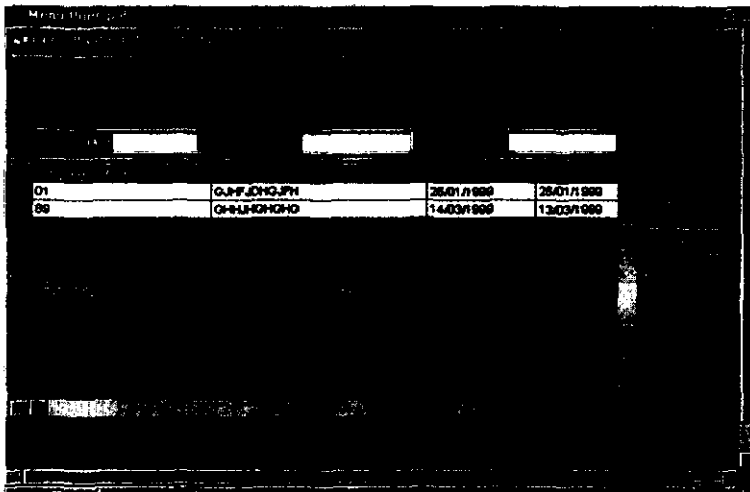
Este objeto tiene todas las características de los Reportes, y además permite actualizar la base de datos. Los procedimientos son comúnmente usados para dos tipos de procesos:

- Procesos batch de actualización. Por ejemplo: eliminar todas las ordenes de trabajo de fecha anterior a una fecha dada.
- Subrutinas de uso general. Por ejemplo: rutina de monto escrito en donde, dado un importe se devuelve un literal con el importe en letras (1010 => 'Mil Diez').

- Procesos a ejecutar en un servidor de aplicaciones o servidor de base de datos: procesos (generalmente escritos en C/SQL o Java) para una Multi Tier Architecture, para ser ejecutados en un servidor de aplicaciones o de Base de Datos.

3.2.7.5 Work Panels

Un Work Panel es una pantalla que permite al usuario realizar consultas interactivas a la base de datos. Por ejemplo: Un work panel que muestra la lista de ordenes de trabajo y que permite (a eleccion del usuario) ver cuales son los importes.



The screenshot shows a window titled 'Work Panel' with a table containing two rows of data. The table has four columns: an ID column, a description column, and two date columns. The first row shows ID '01', description 'OJFJONGJPH', and dates '25/01/1999' and '25/01/1999'. The second row shows ID '02', description 'OJFJONGJPH', and dates '14/03/1999' and '13/03/1999'.

ID	Descripcion	Fecha 1	Fecha 2
01	OJFJONGJPH	25/01/1999	25/01/1999
02	OJFJONGJPH	14/03/1999	13/03/1999

Figura 3.11 Diseño de Work Panels

3.2.7.6 Web Panels

Son similares a los Work Panels pero estos requieren un browser para correr en un ambiente de Internet/Intranet/Extranet.

3.2.7.7 Menús

Un menú muestra una lista de objetos ó items, los cuales el usuario puede elegir para ejecutarlos.

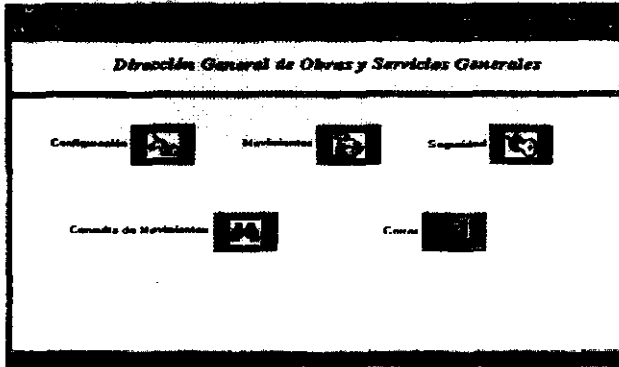


Figura 3.12 Ejemplo del Diseño de Menus

3.2.7.8 Data Views

Son usados para establecer un mapa entre componentes o archivos externos y tablas *GENEXUS*

Styles

Styles, otro tipo de objeto *GENEXUS*, son usados para definir interfaces de usuario y programación estándar. Styles ofrece un ajuste a mecanismos que definen formatos de pantallas, reglas de negocios y eventos que *GENEXUS* incorpora automáticamente dentro del diseño de la aplicación. Esto implica una alta calidad en los sistemas y una notable reducción en tiempo de desarrollo.

Multiple Plataformas

El modelo físico de datos está diseñado de acuerdo a la Teoría de Bases de Datos Relacionales, dirigiéndose a la tercera forma normal. El criterio de normalización de bases de datos se aplica únicamente después de que todos los objetos de las transacciones fueron definidos. Esta normalización es automáticamente ejecutada y mantenida por *GENEXUS*. Sin embargo, el analista puede definir redundancias y es seguro que *GENEXUS* quiera manejarlas también automáticamente.

El almacén de información de **GENEXUS** diseña especificaciones en una forma abstracta, es decir, no depende del ambiente. Esto significa que un solo almacén puede ser generado en donde funcionalmente pueden interactuar con aplicaciones equivalentes para ambientes diferentes.

Esto permite, por ejemplo, usar una aplicación centralizada AS/400 para correr su aplicación en un ambiente cliente/servidor sin tener que modificar el objeto original.

Además, las especificaciones funcionales son totalmente independientes de la Base de Datos, ello significa que siguen permaneciendo igualmente validados después de los cambios hechos en la estructura de datos. Esto es porque **GENEXUS** mantiene automáticamente todos los programas que son generados.

Arquitectura de Múltiples Capas (Multi-Tier)

Hasta hace relativamente poco tiempo, la necesidad de generar aplicaciones multiplataforma se debía a la posibilidad de correr la misma aplicación en varias plataformas, por ejemplo en un sistema de transacciones bancarias se quería tener la aplicación de la casa matriz corriendo en un AS/400 y en una red de PCs en las sucursales pequeñas.

Sin embargo, con el uso progresivo de los ambientes cliente/servidor e Internet/Intranet surge una nueva necesidad: una misma aplicación debe tener ciertas partes corriendo en una plataforma y algunas en otras, y que todas se puedan comunicar correctamente.

La respuesta de **GENEXUS** es permitir varios generadores en un mismo modelo. Ahora se puede indicar para cada ejecutable en que lenguaje se quiere generar. Por ejemplo se puede tener un modelo en donde todos los objetos que interactúan con el usuario se generen en Visual Basic y el proceso de reorganización, todos los procedimientos y WebPanels en C/SQL.

3.2.7.9 Prototipo

En las tareas de diseño están implícitas las dificultades de toda comunicación humana:

- El usuario olvida ciertos detalles.
- El ingeniero no toma nota de algunos elementos.
- El usuario se equivoca en algunas apreciaciones.
- El ingeniero interpreta mal algunas explicaciones del usuario.

Pero, además, la implementación de sistemas es, habitualmente, una tarea que consume bastante tiempo, por lo que:

- Como muchos de estos problemas sólo son detectados en las pruebas finales del sistema, el costo (tiempo y dinero) de solucionarlos es muy grande.
- La realidad cambia, por ello, no es razonable pensar que se pueden congelar las especificaciones mientras se implementa el sistema;
- La consecuencia de la congelación de las especificaciones, es que se acaba implementando una solución relativamente insatisfactoria.

El impacto de estos problemas disminuiría mucho si se consiguiera probar cada especificación, inmediatamente, y saber cual es la repercusión de cada cambio sobre el resto del sistema.

Una primera aproximación a esto, ofrecida por diversos sistemas, es la posibilidad de mostrar al usuario formatos de pantallas, informes, etc. animados por menús. Esto permite ayudar al usuario a tener una idea de qué sistema se le construirá pero, posteriormente, siempre se presentan sorpresas.

Una situación bastante diferente sería la de poner a disposición del usuario para su ejecución, inmediatamente, una aplicación funcionalmente equivalente a la deseada, hasta en los mínimos detalles. Esto es lo que hace *GENEXUS* : Un prototipo es una aplicación pronta, funcionalmente equivalente a la aplicación producción.

La diferencia entre prototipación y producción consiste en que la primera se hace en un ambiente de microcomputador, mientras que la producción se realiza en el ambiente objeto del usuario (IBM AS/400, redes, Cliente/Servidor).

El prototipo permite que la aplicación sea totalmente probada antes de pasar a producción. Durante estas pruebas, el usuario final puede trabajar con datos reales, es decir, prueba de una forma natural, no solamente formatos de pantallas, informes, etc., sino también fórmulas, reglas del negocio, estructuras de datos, etc.

3.3 IMPLEMENTACION

GENEXUS genera automáticamente el código necesario para:

- Crear y mantener la base de datos;
- Generar y mantener los programas para manejar los objetos descritos por el usuario.

Los procesos de generación se dividen en dos fases: **ESPECIFICACION Y GENERACION PROPIAMENTE.**

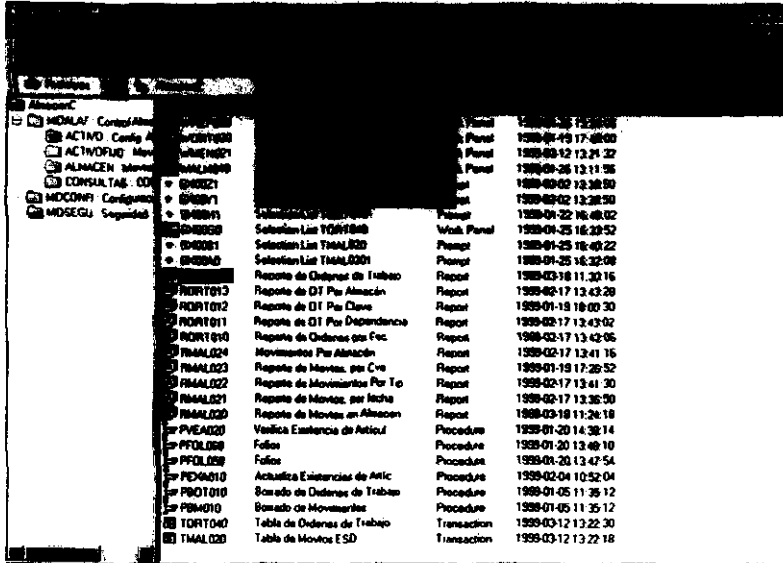


Figura 3.13 Especificación de la Aplicación

La especificación es totalmente independiente del ambiente objeto. La generación, en cada caso, está especializada para el ambiente objeto.

Esto permite que se puedan implementar con facilidad versiones de GENEXUS para diferentes plataformas de ejecución, y que cada una de ellas sea optimizada para su ambiente.

Los ambientes y lenguajes actualmente soportados son:

Ambiente de desarrollo:

El ambiente de desarrollo está constituido por PC's, funcionando con los sistemas operativos Windows 95, Windows 98 y Windows NT.

Ambientes de Producción:

IBM AS/400 (arquitectura centralizada o Servidor), UNIX, Windows NT, PC's, redes.

Lenguajes para el computador IBM AS/400 (arquitectura centralizada):

RPG/400, Java

Lenguajes para microcomputadoras y/o redes de los mismos:

Visual Basic, Visual FoxPro, Java

Cliente/Servidor (DBMSs):

IBM DB2, ORACLE, INFORMIX, MICROSOFT SQL SERVER

Cliente/Servidor (lenguaje cliente):

Visual Basic, Visual FoxPro, Java

Cliente/Servidor(lenguaje servidor de aplicaciones o de base de datos)

C, SQL, RPG, Java.

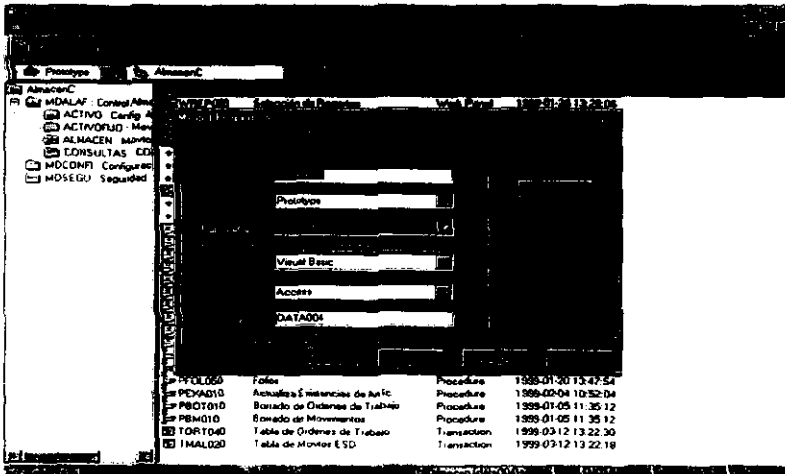


Figura 3.14 Generación de un Nuevo Modelo

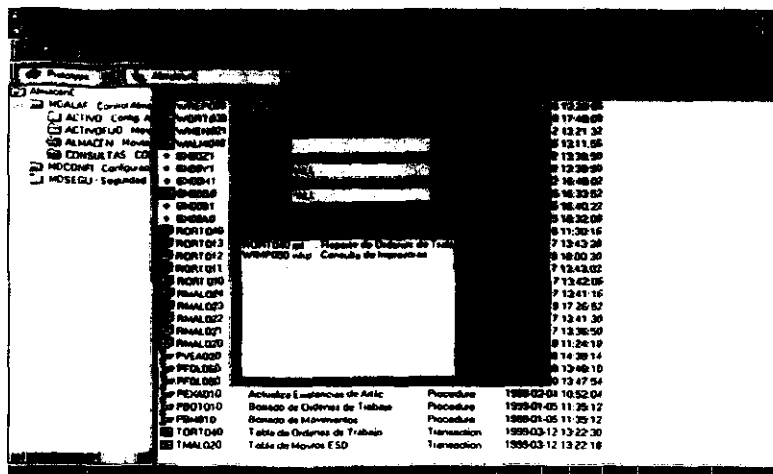


Figura 3.15 Especificación y Generación de Objetos Particulares de la Aplicación

3.4 MANTENIMIENTO

Esta es quizás la característica más importante de *GENEXUS*, y que lo diferencia de manera más clara de sus competidores: el mantenimiento, tanto de la base de datos (estructura y contenido), como de los programas, es totalmente automático.

Impacto de los cambios sobre la Base de Datos.

Análisis de Impacto

Una vez descritos los cambios de las visiones de usuarios, *GENEXUS* analiza automáticamente cuál es el impacto de los mismos sobre la base de datos y produce un informe donde explica como debe hacerse la conversión de los datos y, si cabe, qué problemas potenciales tiene esa conversión (inconsistencias por viejos datos ante nuevas reglas, etc.). El ingeniero decide si acepta el impacto y sigue adelante o no.

Generación de programas de conversión

Una vez que los problemas han sido solucionados o bien se ha aceptado la conversión "default", que *GENEXUS* realiza en forma standard, se generan automáticamente los programas para hacer la conversión.

Impacto de los cambios sobre los programas

Análisis de Impacto

A continuación *GENEXUS* analiza el impacto de los cambios sobre los programas, y produce un diagnóstico informando que programas deben generarse o re-generarse y proporcionando también, para el nuevo programa, o bien el diagrama de navegación bien un pseudocódigo, a elección del ingeniero.

Los programas que serán automáticamente escogidos para generar/regenerar, son los correspondientes a objetos:

- Cuya descripción es nueva.
- Cuya descripción ha sido modificada.
- Tales que los programas anteriores asociados no funcionarían, debido a cambios que se han producido en la base de datos.
- Tales que el sistema esté, ahora, en condiciones de generar programas más eficientes que los anteriores (por ejemplo para aprovechar nuevos índices o redundancias).

3.5 DOCUMENTACION

Toda la información provista por el ingeniero, o inferida por *GENEXUS*, está disponible en un repositorio activo, que constituye una muy completa documentación on-line, permanentemente actualizada.

La documentación incluye, además de la descripción de los objetos especificados, información sobre la base de conocimiento obtenida, y la base de datos diseñada.

La base de conocimiento de *GENEXUS* permite no solamente obtener, en cualquier momento, los conocimientos que en ella fueron almacenados, sino que también cualquier otro que se pueda inferir lógicamente de los almacenados (una regla de integridad referencial, un mapa de navegación, un análisis de impacto de los cambios, un programa, por ejemplo, son inferidos a partir del conocimiento almacenado).

```
Objetos generados : Reporte de Ordenes de Trabajo
-----
(+) No enumeration required

Output device: SCREEN or PRINTER
SourceCode: Visual Basic
Form Class: Graphic

Parameters : @MemoriaC

FOR EACH TORTUDO1 (line 12)
  Order : OrdCve
  Index : TORTUDO3
  Navigation Filter:
    Start from:
      First record
    Loop until:
      Not end of table

---->> TORTUDO1 ( OrdCve, ArtCve, AlmCve )
|
|-----> TORTUDO ( OrdCve )
|
|----->> TERNODO ( EspCVC )
|
|-----> TALBO10 ( AlmCve )
```

Figura 3.16 Análisis de Errores de los Objetos Especificados y Generados

3.6 CONSOLIDACION DE VARIAS APLICACIONES Y REUTILIZACION DEL CONOCIMIENTO

Varias aplicaciones pueden ser diseñadas y prototipadas simultáneamente, por diferentes equipos, utilizando GENEXUS. Estos equipos pueden intercambiar especificaciones de diseño utilizando el módulo KNOWLEDGE MANAGER.

Este módulo permite, automáticamente:

- Comenzar el diseño de una nueva aplicación partiendo de Business Objects, Software Patterns, Domains, Atributos y/o Styles de dominio público o terceras partes.

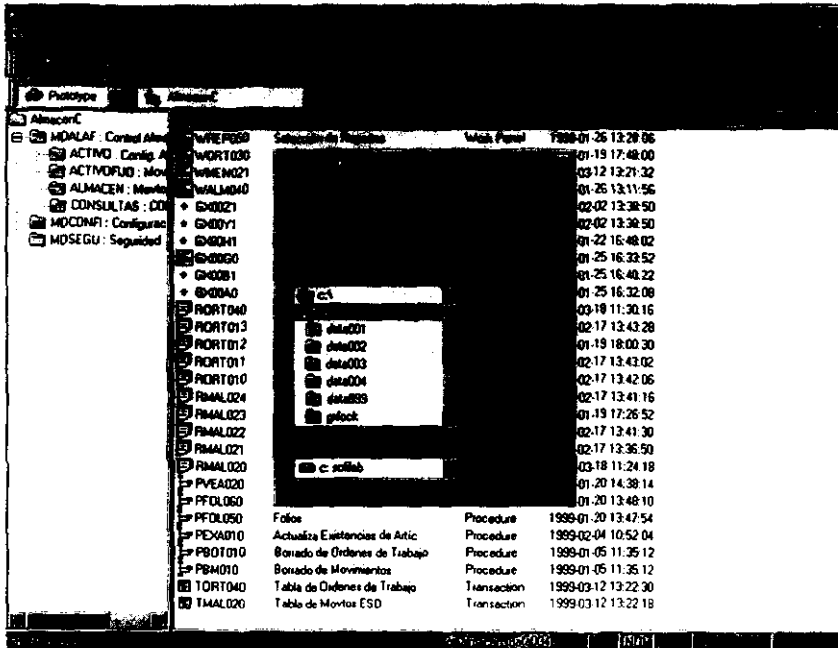


Figura 3.17 Selección de una Aplicación

- Distribuir conocimiento, a partir de la base de conocimiento corporativa, a la de cualquier aplicación.
- Verificar la consistencia entre la base de conocimiento de una aplicación y la corporativa.
- Consolidar dos aplicaciones cualesquiera (es particularmente útil la consolidación del conocimiento)

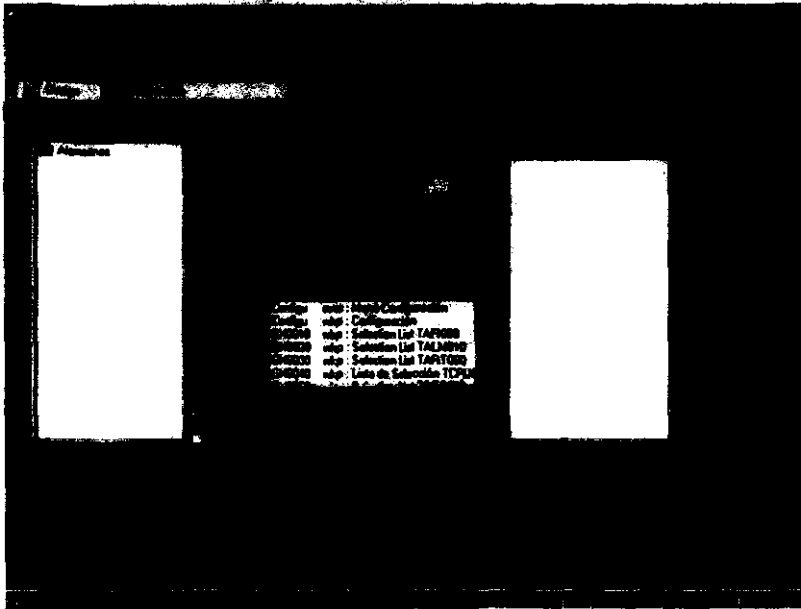


Figura 3.18 Respaldo de una Aplicación

Esto permite una flexibilidad ideal: el ingeniero trabaja con entera libertad en un ambiente de prototipo, con una pequeña base de conocimiento y, sólo cuando su aplicación está pronta desde el punto de vista del usuario, debe tomarse en cuenta la base de conocimiento corporativa, que generalmente será muy grande.

En ese momento, con poderosas ayudas automáticas, se establece el impacto que tendrá la aplicación, o la modificación de la preexistente, sobre el modelo corporativo y, si es el caso, se hacen los cambios para asegurar la consistencia, de una manera muy simple.

GENEXUS tiene algunas características únicas que lo distinguen de sus competidores. Entre ellas pueden destacarse:

- El punto de partida es la descripción de los objetos familiares al usuario (TRANSACCIONES, INFORMES, etc.), sobre los cuales existe el principal conocimiento objetivo que se tiene en la empresa.
- La descripción de cada objeto es totalmente independiente de la de los demás por lo que, en el caso de que se deba modificar la descripción de uno, ello no implicará la necesidad de modificar manualmente la descripción de cualquier otro. Esta característica exclusiva de **GENEXUS** es la que permite un mantenimiento totalmente automático de las aplicaciones.

- La curva de aprendizaje es muy corta.
- Diseño, creación y mantenimiento de la base de datos son totalmente automáticos.
- La aplicación (base de datos y programas) tiene siempre, sean cuales sean las modificaciones que haya sufrido, la mejor calidad:
- La base de datos es siempre la óptima.
- No se modifican los programas: cuando ya no son adecuados, se generan otros nuevos, que los sustituyen.
- Lenguajes poderosos y de muy alto nivel para la definición de PROCESOS, WORK PANELS y WEB PANELS, así como una definición de MENUES muy simple. En estos lenguajes las descripciones de los procesos se hacen sin referirse a los archivos involucrados, los que son inferidos automáticamente en tiempo de generación. Esta característica permite una total independencia entre los datos y dichas especificaciones. Como consecuencia, las especificaciones de alto nivel de *GENEXUS* no necesitan cambio alguno ante modificaciones de la base de datos.
- Utilización simple y eficiente de componentes, archivos o bases de datos externos a *GENEXUS*.
- Mantenimiento 100% automático: El conjunto de estos elementos permite a *GENEXUS* generar y mantener automáticamente el 100 % de los programas en aplicaciones normales de tipo comercial, administrativo, financiero o industrial.
- *GENEXUS* funciona en microcomputadoras, dejando la computadora de producción totalmente libre para el procesamiento de las aplicaciones.
- Fácil distribución del conocimiento corporativo para facilitar el desarrollo de nuevas aplicaciones.
- Simple y poderosa solución para Data Warehousing.
- Verificación automática de consistencia, y consolidación, entre aplicaciones desarrolladas separadamente.
- Independencia de plataforma y arquitectura.
- Simplicidad: *GENEXUS* utiliza los recursos más avanzados de la inteligencia artificial para que el ingeniero en sistemas y los usuarios, pueden usarlo de forma muy simple.

CAPITULO IV IMPLEMENTACION DE SICIPA

4.1 PROPUESTA DE SOLUCION

En este capítulo, se prepara una propuesta del sistema que resume todo lo que se ha encontrado, presenta un análisis de las alternativas y plantea las recomendaciones de lo que deberá realizarse. En el área de sistemas, cada problema es único y en consecuencia nunca habrá sólo una solución correcta. La manera en como se plantea una solución depende de las características individuales de cada analista aunadas a su formación profesional.

Tomando en cuenta cada una de las situaciones descritas en capítulos anteriores, se consideró indispensable crear una solución integral que, por una parte, diera respuesta de manera eficiente y ágil a las necesidades observadas, y por otra, fuera acorde con las nuevas tendencias y tecnologías incorporadas recientemente.

Por tal motivo, surgió el proyecto de crear un sistema para el manejo de información por computadora, a través del cual la Dirección General de Obras y Servicios Generales de la UNAM, pudiera consultar de manera ágil y sencilla información presupuestal y de gestión administrativa, desde la comodidad de su computadora personal, teniendo la capacidad de visualizar los datos en pantalla, guardarlos en archivos de su computadora personal e imprimirlos de manera selectiva; aprovechando en su totalidad la infraestructura con la que cuenta dicha dependencia.

A lo largo de este capítulo, se hace referencia al enfoque sistemático que el desarrollador debe aplicar en el análisis y el diseño de los sistemas de información. Una gran parte de este enfoque, se encuentra en lo que se denomina "ciclo de desarrollo de los sistemas" o "ciclo de vida de los sistemas" (SDLC; Systems Development Life Cycle). El SDLC es un enfoque por etapas de análisis y de diseño, que postula que el desarrollo de los sistemas es más productivo cuando existe un ciclo específico de actividades del analista y de los usuarios.

Actualmente, no existe un acuerdo respecto al número exacto de etapas que conforman el ciclo de desarrollo de los sistemas; sin embargo, por lo general se reconoce la importancia de su enfoque sistemático. Para fines de explicación del presente trabajo, este ciclo se dividirá en seis etapas que inician con el análisis, diseño y desarrollo del sistema; continúan con las pruebas y la implantación del mismo, y finalizan con su mantenimiento.

Cabe mencionar que a pesar de que cada etapa puede identificarse de manera discreta, nunca se lleva a cabo como un evento independiente. En lugar de ello, el ciclo implica la realización simultánea de diversas actividades y, es posible que alguna de ellas se repita en etapas posteriores.

4.2 DISEÑO MODULAR

Se consideró conveniente que el sistema tuviera una estructura modular, a fin de que se dejara abierta la posibilidad de incrementar paulatinamente la aplicación; es decir, el diseño modular permitiría evitar el estancamiento y obsolescencia de la aplicación, favoreciendo la integración de nuevas funciones conforme fueran surgiendo más necesidades de información de la Dirección General de Obras y Servicios Generales, incrementando sus opciones sin importando incluso el hardware y software disponibles y el lugar en donde residan las bases de datos de la información a compartir.

Sin lugar a dudas, el sistema propuesto no era una tarea fácil, pero definitivamente correspondía a lo que se necesitaba para que la dependencia, mejorara la toma de decisiones. Tomando en consideración su relevancia, el desarrollo de la aplicación constituyó el objetivo del presente trabajo.

4.3 METODOLOGIA

Existen diferentes métodos que se pueden aplicar para el desarrollo de software, sin embargo, estos varían dependiendo del modelo de ingeniería que se aplique. Entre los más conocidos se pueden mencionar los siguientes:

- Modelo Lineal Secuencial
- Modelo de Construcción de Prototipos
- Modelo DRA (Desarrollo Rápido de Aplicaciones)
- Modelo Incremental
- Modelo en Espiral

Sin duda alguna, cada uno de ellos tiene sus ventajas y desventajas de acuerdo con los factores que intervienen para el desarrollo de un sistema (los recursos materiales y humanos, tiempos de entrega, dominio de las herramientas, etc.). Para el desarrollo del presente trabajo, se utilizó el Modelo en Espiral, ya que por sus características fue el que mejor se adaptó a las necesidades propias de la Dirección General de Obras y Servicios Generales, principalmente porque permite la creación de nuevas versiones del software en periodos cortos y por su independencia en la creación de nuevos módulos.

4.3.1 MODELO EN ESPIRAL

Retomando una parte descrita en el capítulo I, el Modelo en Espiral, propuesto por Boehm, es un modelo de proceso de software evolutivo que compagina la naturaleza interactiva de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial. A través de dicho modelo se proporciona el potencial para el desarrollo rápido de versiones incrementales del software.

De acuerdo con este modelo, el software se desarrolla en una serie de versiones cada una desarrollada con mayor amplitud de funciones y de capacidades. Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas de ingeniería del sistema.

Cuando empieza este proceso evolutivo, el equipo de ingeniería de software gira alrededor de la espiral en la dirección de las manecillas del reloj, comenzando por el centro. El primer circuito de la espiral produce el desarrollo de una especificación de productos; los pasos siguientes en la espiral se podrían utilizar para desarrollar un prototipo y progresivamente versiones mas sofisticadas del software.

A diferencia del modelo de proceso clásico que termina cuando se entrega el software, el modelo en espiral puede adaptarse y aplicarse a lo largo de la vida del software de computadora. En esencia la espiral permanece operativa hasta que el software se retira. Hay veces que el software esta inactivo, pero siempre que se inicie un cambio, el proceso arranca en el punto de entrada adecuado (por ejemplo, una mejora del producto).

El modelo en espiral es un enfoque realista del desarrollo de sistemas y de software a gran escala. Como el software evoluciona, a medida que progresa el proceso, el desarrollador y el cliente comprenden y reaccionan mejor ante riesgos en cada uno de los niveles evolutivos. Este modelo utiliza la construcción de prototipos como mecanismo de reducción de riesgos, pero lo que es más importante, permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto.

Esta opción mantiene el enfoque sistemático de los pasos sugeridos por el ciclo de vida clásico, pero lo incorpora al marco de trabajo interactivo que refleja de forma más realista el mundo real. El modelo en espiral demanda una consideración directa de los riesgos técnicos en todas las etapas del proyecto y, si se aplica adecuadamente, debe reducir los riesgos antes de que se conviertan en problemáticos.

Pero al igual que otros paradigmas, el modelo en espiral no es la panacea. Puede resultar difícil convencer a grandes clientes (particularmente en situaciones bajo contrato) de que el enfoque evolutivo es controlable. Requiere una considerable habilidad para la evaluación del riesgo, y cuenta con esta habilidad para el éxito. Si un riesgo importante no es descubierto y gestionado, indudablemente surgirán problemas. Finalmente, el modelo en sí mismo es relativamente nuevo y no se ha utilizado tanto como los paradigmas lineales secuenciales o de construcción de prototipos. Todavía tendrán que pasar muchos años antes de que se determine con absoluta certeza la eficacia de este nuevo e importante paradigma.

4.4 DESARROLLO MODULAR

Aunado a la decisión de desarrollar un diseño con enfoque descendente, también se consideró pertinente adoptar para la programación la concepción modular.

El enfoque de programación modular significa descomponer la programación en fracciones lógicas y manejables. Este tipo de programación se apega bien al diseño descendente porque enfatiza las interfaces entre los módulos, en lugar de mantenerlas ignoradas hasta el final del desarrollo del sistema. De manera ideal, cada módulo debe ser funcionalmente cohesivo, de tal manera que satisfaga sólo una función.

El diseño de programas modulares tiene tres ventajas básicas.

Los módulos son más fáciles de escribir y de revisar, ya que están virtualmente autocontenidos. La detección de un error dentro de un módulo es menos complicada, ya que los problemas asociados a un módulo no llegarán a trascender a los otros.

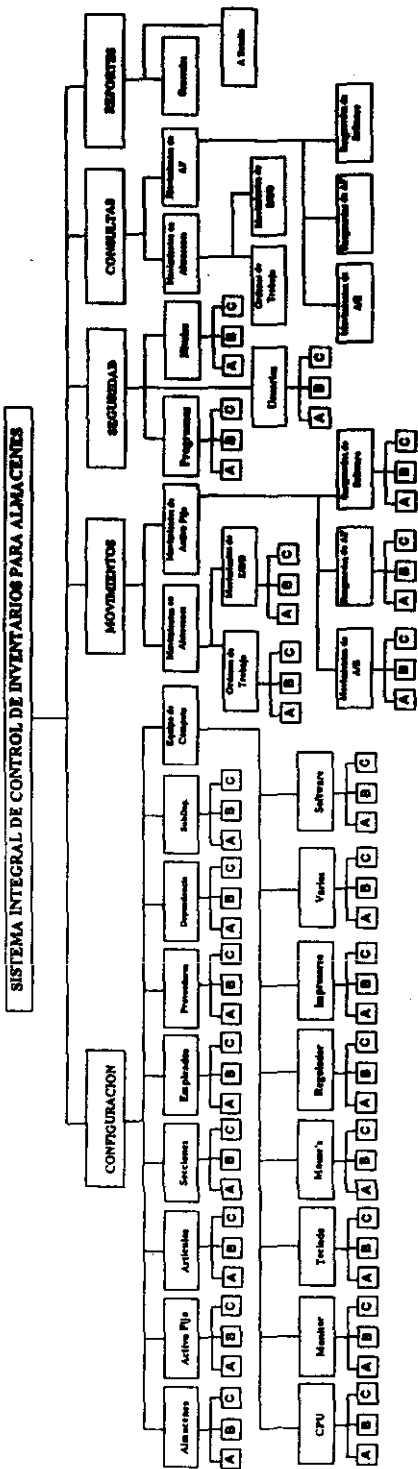
El mantenimiento de los módulos es más fácil. Las modificaciones pueden limitarse a unos cuantos módulos y no al programa completo.

La problemática de los módulos es más fácil de entender, ya que son sistemas autocontenidos. Eso significa que un lector entenderá la función de un módulo específico con sólo tomar su listado de código.

Algunos lineamientos para la programación incluyen:

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

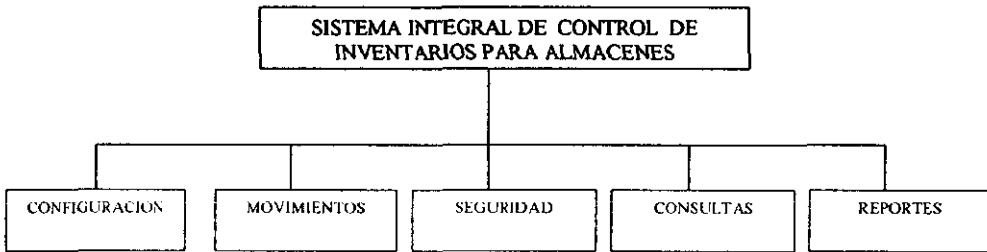
4.5 DIAGRAMA MODULAR DE SICIPA

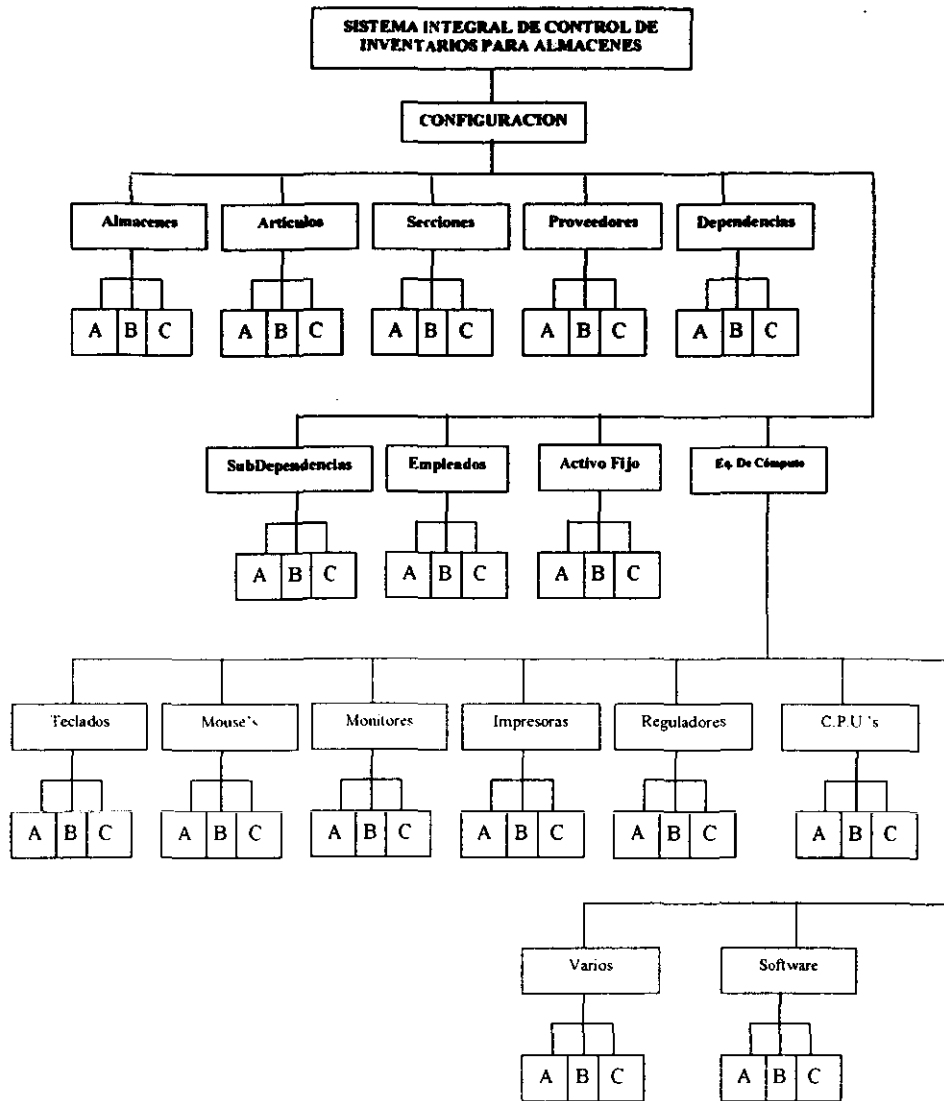


A = Alta
 B = Baja
 C = Cambios
 E/S/D = Entrada/Salida/Devolución
 A/B = Alta/Baja
 AF = Active File

4.5.1 MODULOS DEL SISTEMA INTEGRAL DE CONTROL DE INVENTARIOS PARA ALMACENES

El Sistema Integral de Control de Inventarios para Almacenes está integrado por los siguientes módulos: Codificación, Movimientos, Seguridad, Consultas y Reportes.





A = Alta
B = Baja
C = Cambios

4.5.1.1 El Módulo de Configuración Este módulo tiene como función permitir el registro de los datos de identificación correspondientes a cada uno de los artículos y Bienes de los Almacenes e integrarlos en el inventario de los mismos, así como la modificación de la información correspondiente a aquellos que ya se encuentran registrados, facilitando las tareas de control de la información.

El módulo de configuración se divide en los siguientes submódulos: Almacenes, Artículos, Secciones, Proveedores, Dependencias, Subdependencias, Empleados, Activo Fijo y Equipo de Cómputo.

Almacenes: Este submódulo tiene como función permitir el registro de los datos que identifican los diferentes almacenes de la D.G.O. y S.G., así como la modificación de información ya registrada o la baja total de un registro dado.

Artículos: En este submódulo se dan de alta todos los artículos, con sus correspondientes características, que se tienen en los diferentes almacenes, permitiendo hacer modificaciones y bajas de registros, si en determinado momento se requiere.

Secciones: Este submódulo permite el registro, modificación o baja de cada una de las secciones que se tiene en cada almacén, como son, por ejemplo: carpintería, electricidad, pintura, etc.

Proveedores: En este submódulo se registran los datos de las personas que abastecen a cada uno de los almacenes, permitiendo modificar algún dato o conjunto de datos, o dar de baja todo un registro que sea necesario.

Dependencias: En este submódulo se da de alta la información de lo que respecta a las dependencias que participan en diferentes movimientos que se realizan en los almacenes, y si en algún momento se requiere, se modifica o se da de baja dicha información.

Subdependencias: este submódulo tiene como función registrar toda información necesaria referente a las diferentes subdependencias que integran cada dependencia y que a su vez participan en los movimientos de almacenes. Al igual que los submódulos anteriores también se puede modificar o dar de baja algún registro

Empleados: En este submódulo se registran, modifican o se dan de baja todos los datos de cada empleado de la D.G.O. y S.G. y de los almacenes, para poder asociarlos con los diferentes movimientos que se realizan como son resguardos, movimientos de entradas/salidas/devoluciones, etc.

Activo Fijo: En este submódulo se dan de alta todas las características de cada Activo Fijo que se encuentra en la D.G.O. y S.G. y Almacenes para su manejo en los movimientos de altas/bajas y resguardos.

Equipo de Cómputo: El submódulo de Equipo de Cómputo se divide en: CPU's, Monitores, Teclados, Mouse's, Reguladores, Impresoras, Varios y Software; cada uno de ellos se describe a continuación.

CPU's: En este submódulo se va a registrar todas las características propias de los CPU's como son: capacidad, memoria RAM, velocidad, etc., así como la modificación de uno o varios campos o la baja de todo un registro.

Monitores: En este submódulo se darán de alta, baja o modificación de un registro que tiene la información de las características propias de un monitor.

Teclados: En este submódulo se registran toda la información referente a los teclados como son: marca, modelo, número de serie, etc., si en un momento dado se requiere cambiar algún dato de determinado teclado se puede hacer o en su defecto darlo de baja.

Impresoras: En este submódulo se puede dar de alta las características propias de cada impresora como: descripción, tecnología, marca, modelo, etc., se puede modificar algún dato en específico o borrar todo un registro dado.

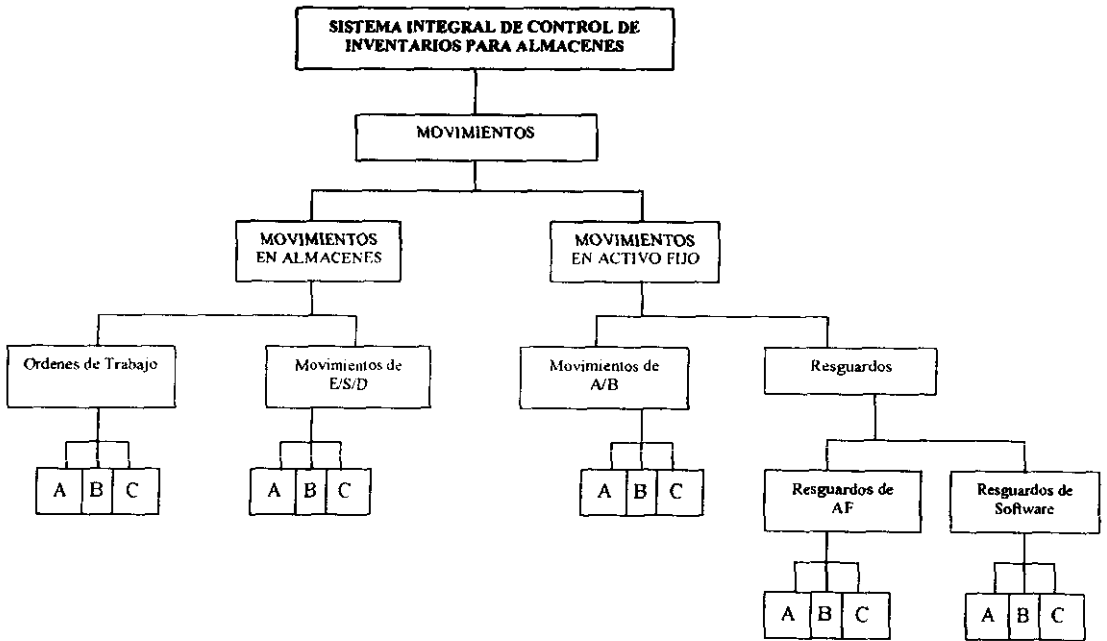
Mouse: en este submódulo se registra cada dato característico de un Mouse, modificar o borrar todos los datos contenidos en un registro.

Reguladores: Aquí se dan de alta, baja, modifican todos los datos necesarios que identifican cada regulador de corriente que se encuentra en la D.G.O y S.G.

Software: En este submódulo se registra todo el software con sus correspondientes características, como: descripción, versión, número de manuales que consta, etc., y al igual que los submódulos anteriores se puede modificar algún dato o dar de baja un registro.

Varios: En este submódulo se registra todo el tipo de hardware que sea diferente de los ya mencionados, como: scanners, unidades DVD, unidades zip, etc.

4.5.1.2 El Módulo de Movimientos Este módulo se divide en Movimientos de Almacén y Movimientos de Activo Fijo, en cada uno de estos módulos se registran las diferentes operaciones que se realizan en los almacenes, al igual que el módulo de configuración en este se puede modificar e incluso eliminar registros dependiendo de la administración interna del almacén.



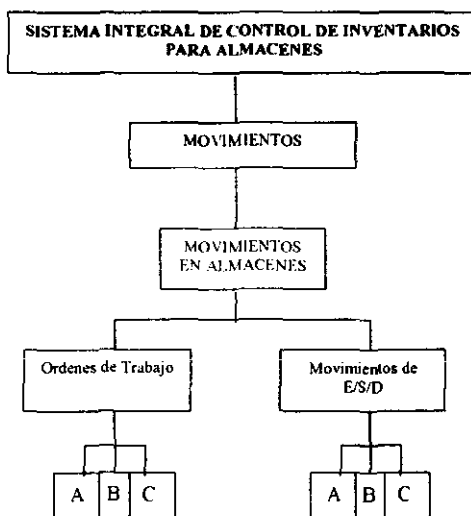
A = Alta
B = Baja
C = Cambios

E/S/D = Entrada / Salida / Devolución.

A/B = Alta / Baja.

El submódulo de **Movimientos en Almacenes** se divide en **Ordenes de Trabajo** y **Movimientos de Entradas/Salidas/Devoluciones**.

Ordenes de Trabajo: En este submódulo se registran todas las ordenes de trabajo utilizadas para pedir un grupo de articulos en especial que justifican la realización de un trabajo específico. También se puede modificar algún campo de una orden de trabajo o borrar el registro correspondiente.

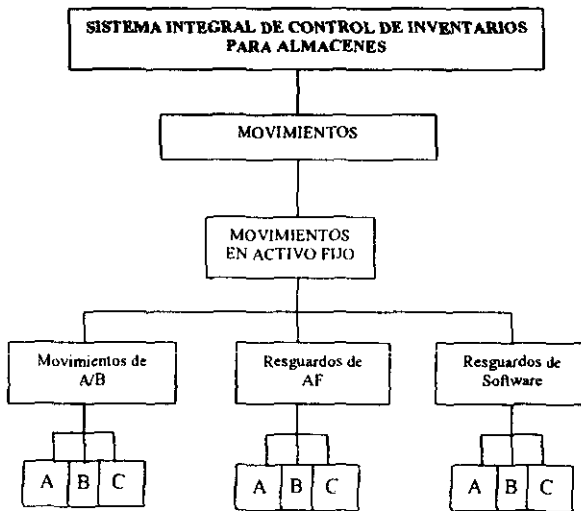


A = Alta
B = Baja
C = Cambios

Movimientos de Entrada/Salida/Devolución: En este submódulo se registran todos los Movimientos de Entradas/Salidas/Devoluciones de uno o más artículos del almacén, así como la modificación de algún dato en especial o la baja completa de un movimiento dado.

Movimientos de Activo Fijo: El submódulo de Movimientos de Activo Fijo se divide en Movimientos de Altas/Bajas, Resguardos de Activo Fijo y Resguardos de Software.

Movimientos de Altas/Bajas: En este submódulo se registran todos los Movimientos de Altas/Bajas de Activo Fijo que se realizan en la D.G.O y S.G. así como las modificaciones de datos de cada movimiento si lo requiere o la baja total de un movimiento determinado.



A = Alta

B = Baja

C = Cambios

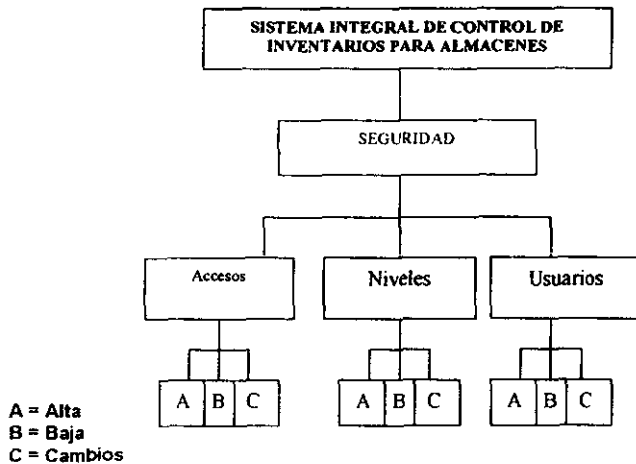
E/S/D = Entrada / Salida / Devolución.

A/B = Alta / Baja.

Resguardos de Activo Fijo: En este submódulo se registran todos los resguardos de Activo Fijo que se realizan en la D.G.O y S.G. y en los Almacenes, al igual que el submódulo de Movimientos de Altas/Bajas de Activo Fijo, también se pueden hacer modificaciones o bajas totales de un registro.

Resguardos de Software: En este submódulo se registran todos los resguardos de software que se efectúen en la D.G.O. y S.G. y en Almacenes, así como modificaciones a los mismos y si se requiere en su momento, se dan de baja.

4.5.1.3 El Módulo de Seguridad Este módulo permite realizar el registro del personal que sea autorizado para efectuar las diferentes operaciones dentro del Sistema. Aquí se registra el nombre y la clave de acceso individual que se le asigna a cada trabajador, así mismo, se pueden dar de baja o cambiar las claves antes registradas.

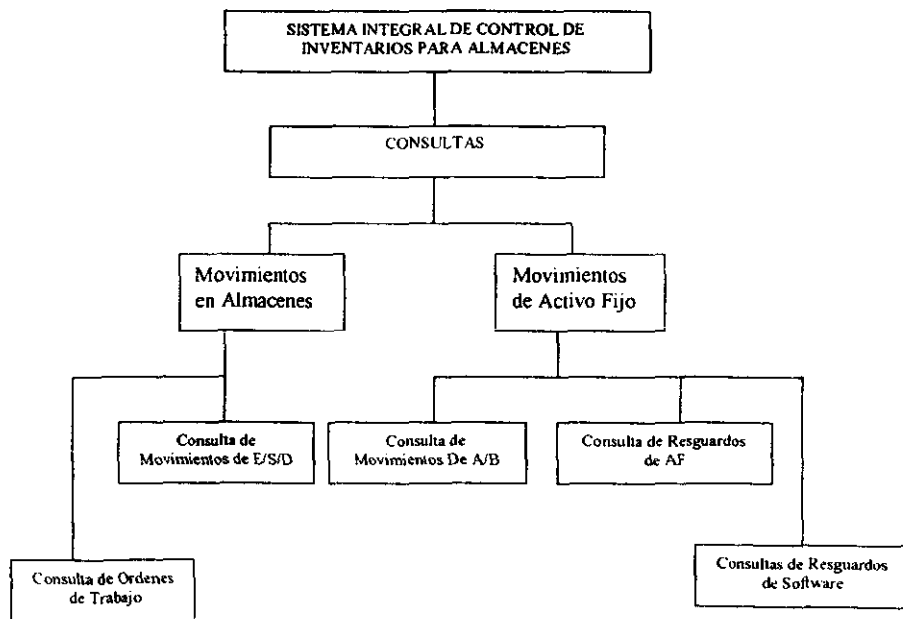


Accesos: Este submódulo tiene como función dar de alta, modificar o borrar todos los accesos a los que se puede entrar en el sistema.

Niveles: En este submódulo se registran todos los niveles que puede haber para el acceso al sistema, dependiendo de la administración interna de la D.G.O. y S.G., así como también la baja o modificación de un registro determinado.

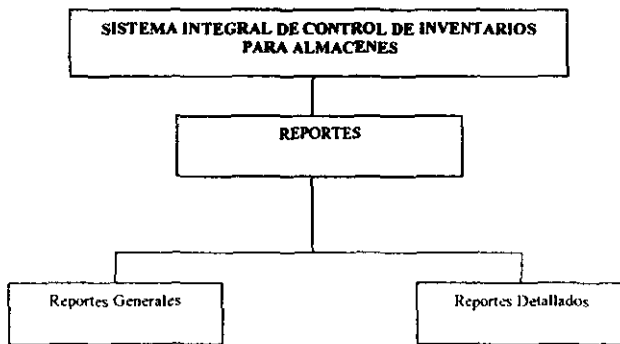
Usuarios: Este submódulo tiene la finalidad de dar de alta a todos aquellos usuarios que trabajarán con el sistema, al igual que los submódulos anteriores, se puede dar de baja o modificar algún registro dado.

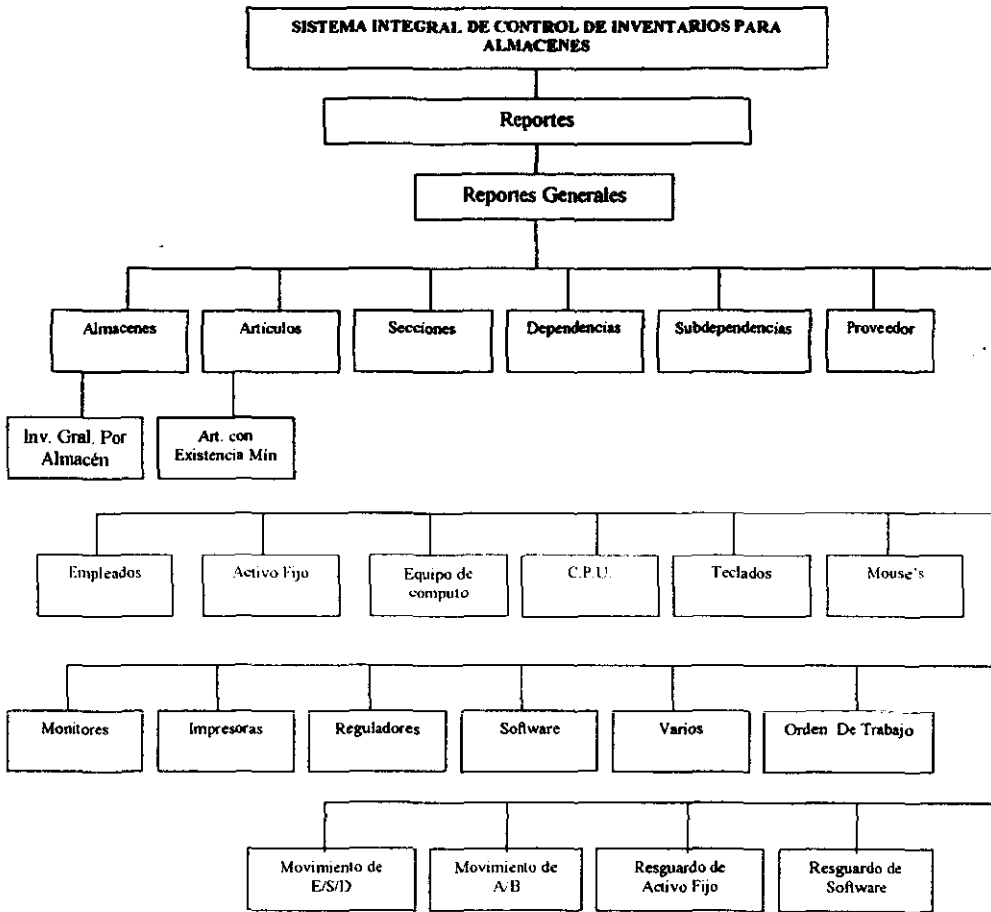
4.5.1.4 El Módulo de Consultas Tiene como función la consulta de los diferentes movimientos que se han realizado dentro de los almacenes y la propia dependencia, así como la persona que realizó determinado movimiento en un periodo de tiempo dado.



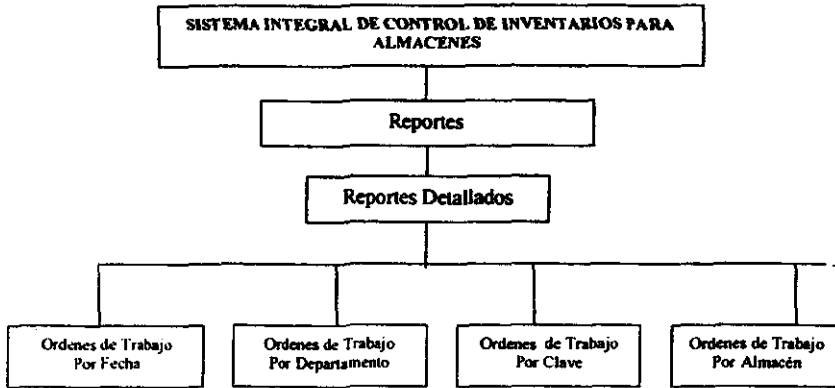
E/S/D = Entrada / Salida / Devolución.
A/B = Alta / Baja.
AF = Activo Fijo.

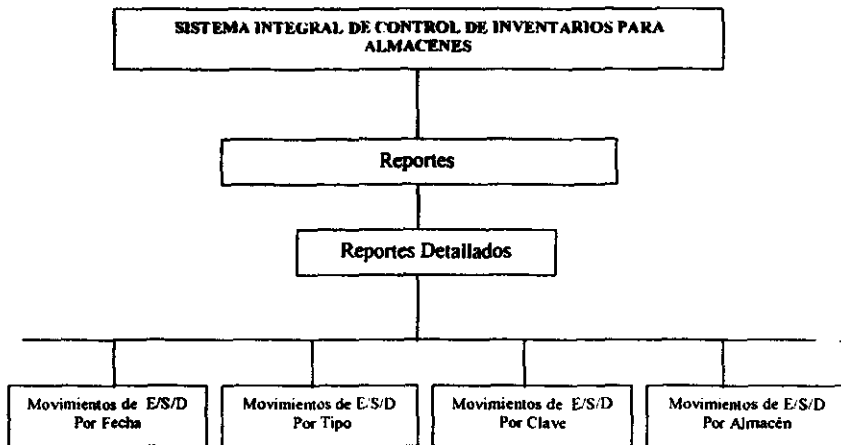
4.5.1.5 El Módulo de Reportes Este módulo permite la obtención de información impresa correspondiente al inventario de los almacenes, brindando dos modalidades de presentación: de detalle y de resumen cada uno de los cuales ofrecen diferentes opciones para la selección de aquellos registros cuya información se requiere imprimir.



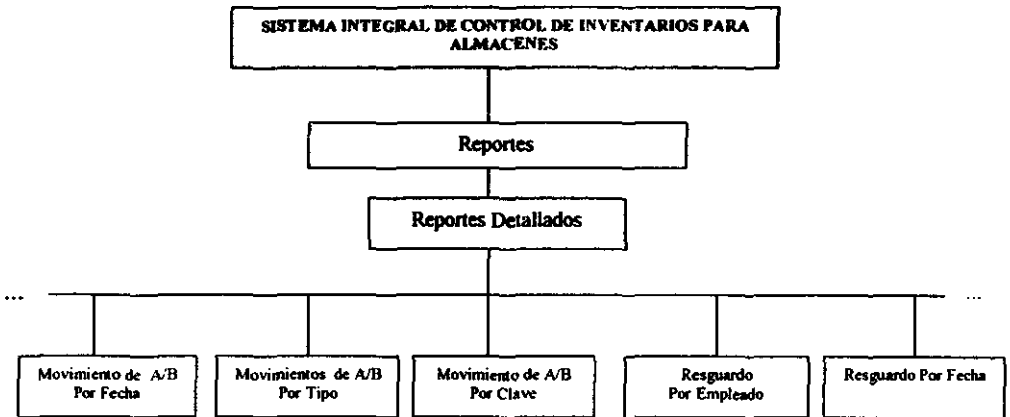


E/S/D = Entrada / Salida / Devolución.
A/B = Alta / Baja.
AF = Activo Fijo.

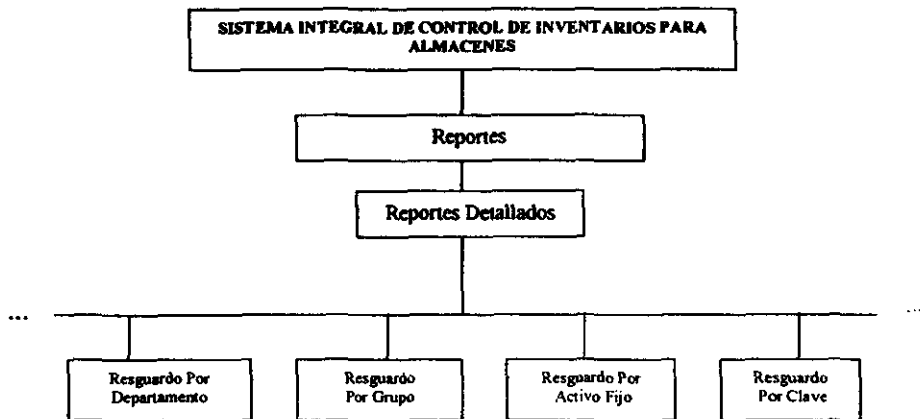




E/S/D = Entrada / Salida / Devolución.



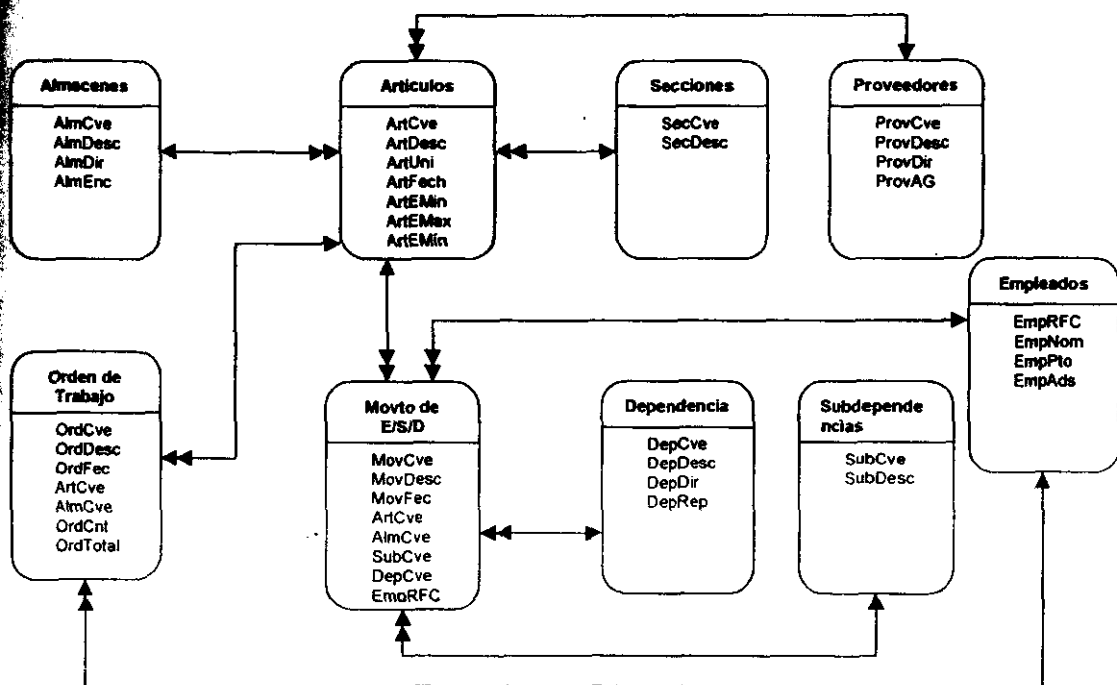
A/B = Alta / Baja



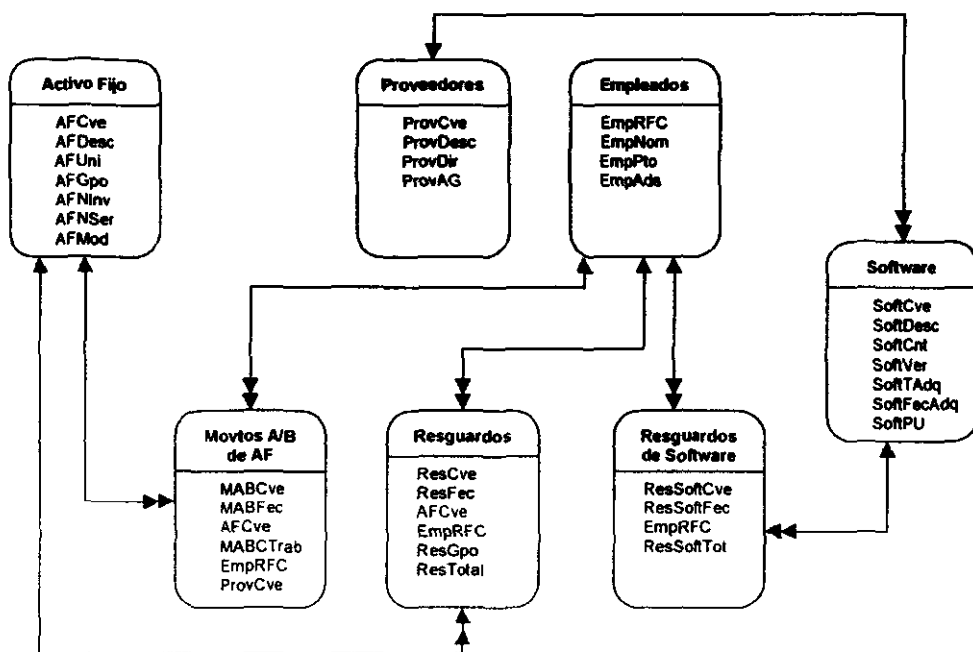
A/B = Alta / Baja



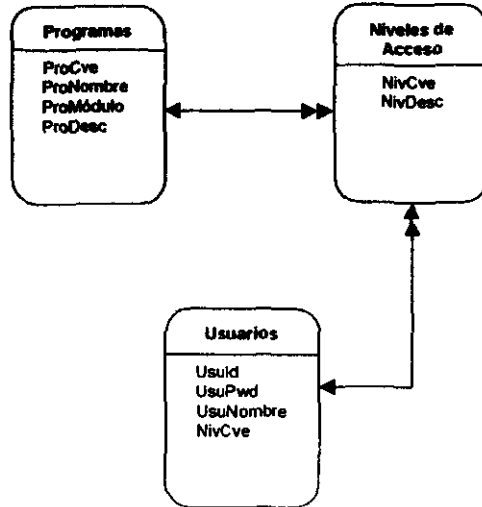
DIAGRAMA ENTIDAD - RELACION de SICIPA (Sistema Integral de Control de Inventarios para Almacenes)

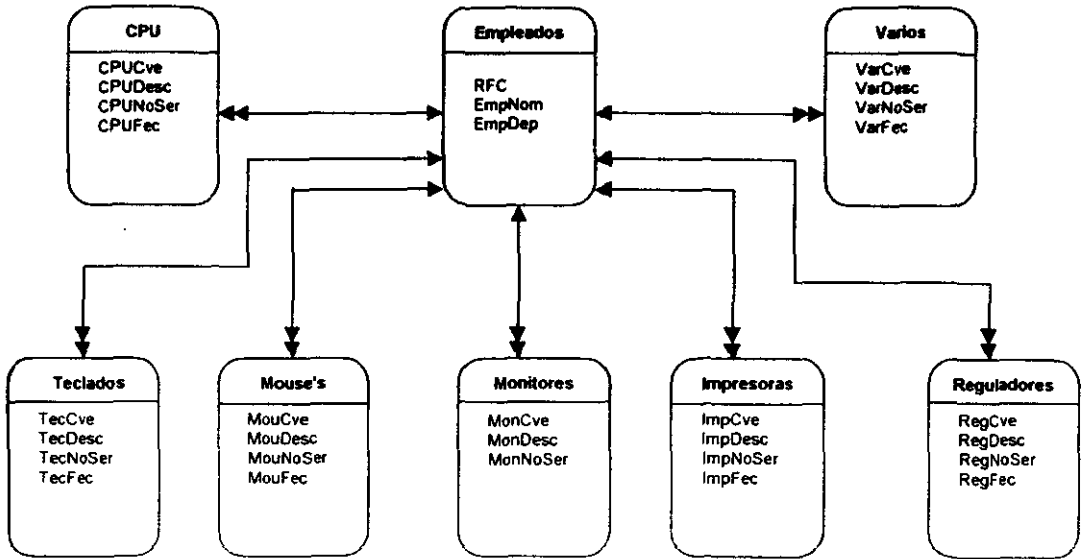


E/S/D = Entrada / Salida / Devolución



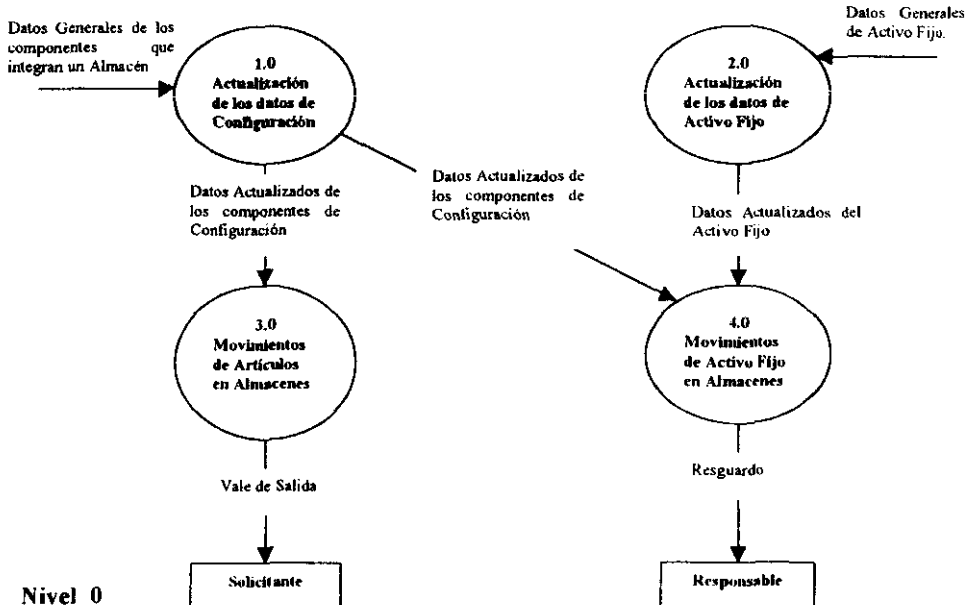
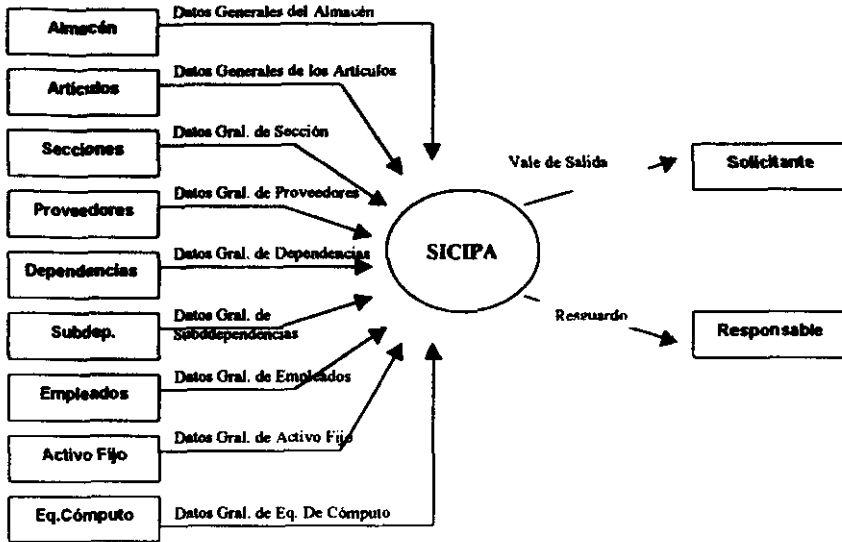
A/B = Alta / Baja
 AF = Activo Fijo





4.7 DIAGRAMA DE FLUJO DEL SISTEMA INTEGRAL DE CONTROL DE INVENTARIOS PARA ALMACENES

DIAGRAMA DE CONTEXTO



Nivel 0

Diagrama de Flujo de Datos "Actualización de los Datos de Configuración"

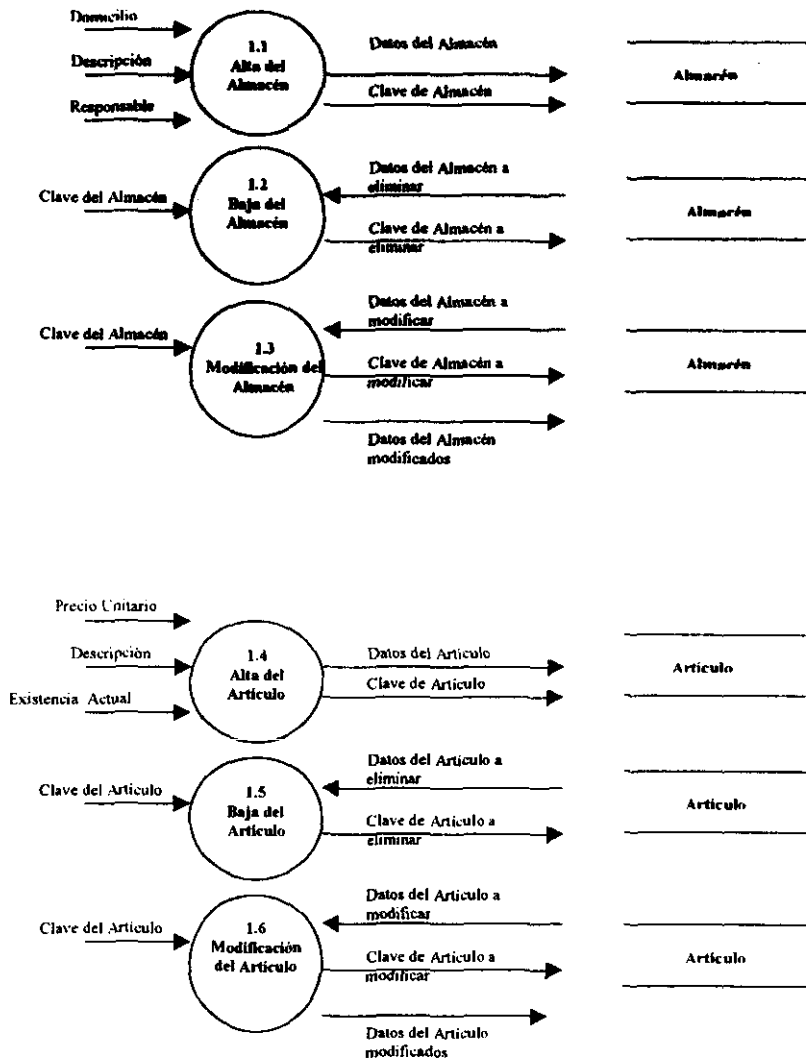


Diagrama de Flujo de Datos "Actualización de los Datos de Configuración"

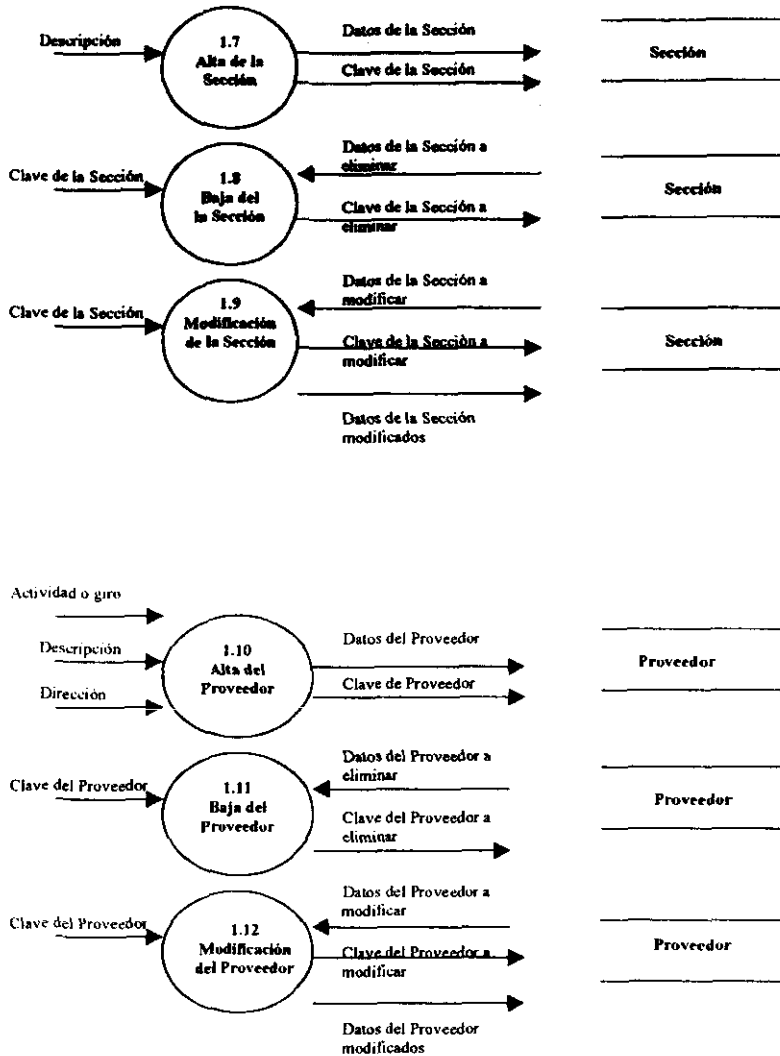
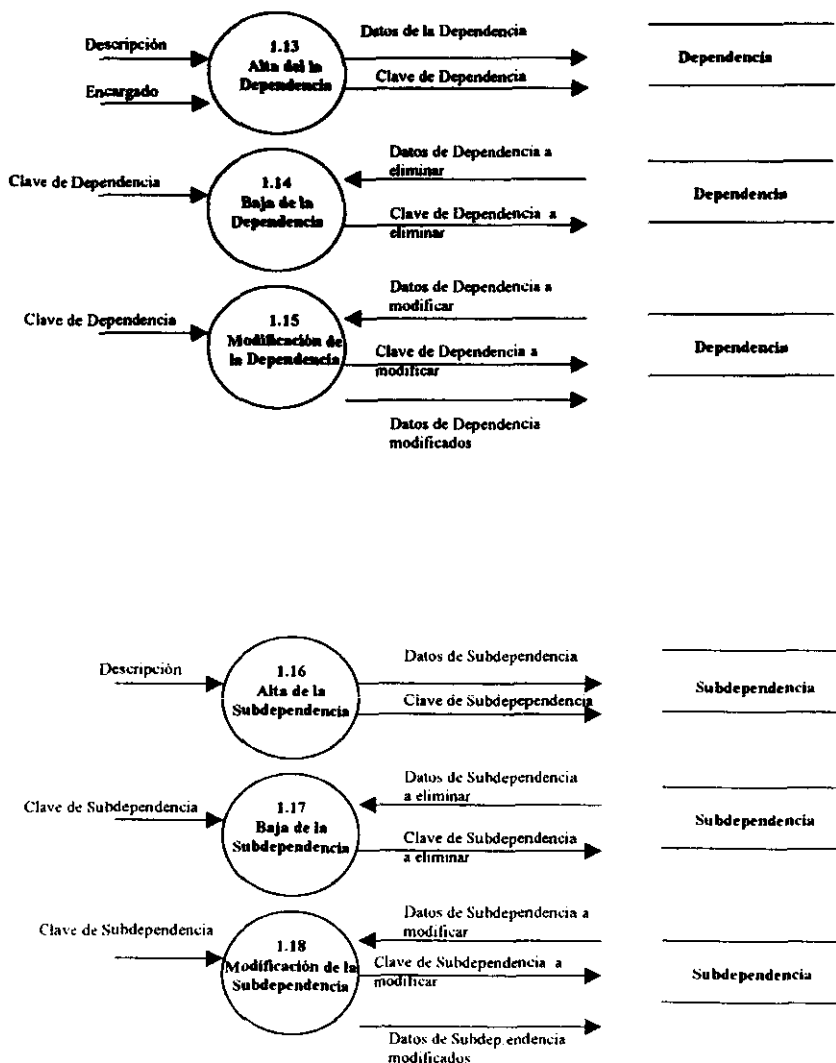
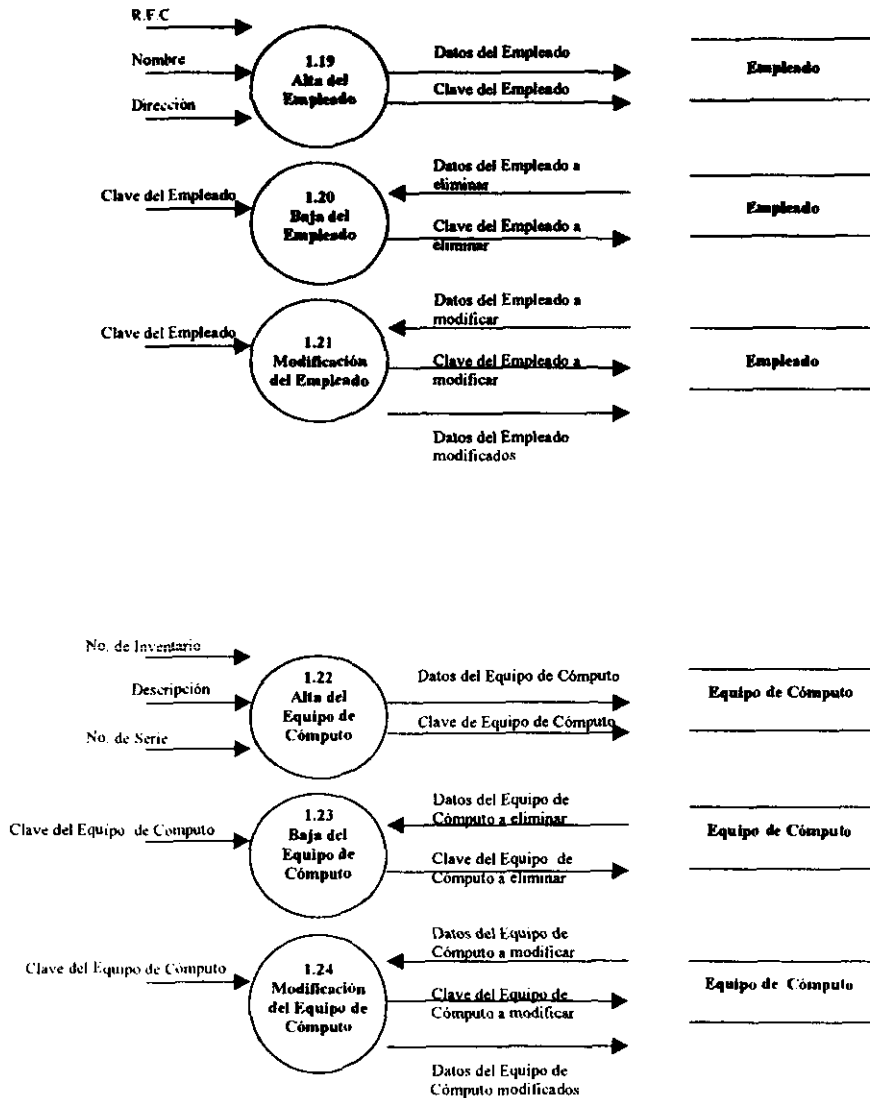


Diagrama de Flujo de Datos "Actualización de los Datos de Configuración"



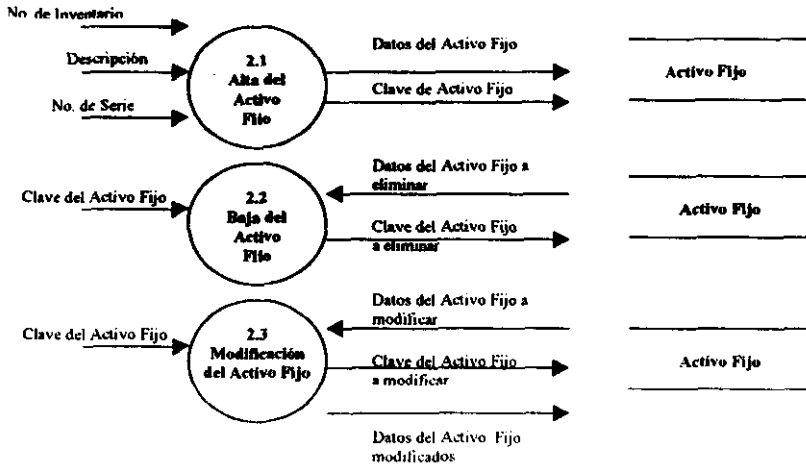
Nivel 1

Diagrama de Flujo de Datos "Actualización de los Datos de Configuración"



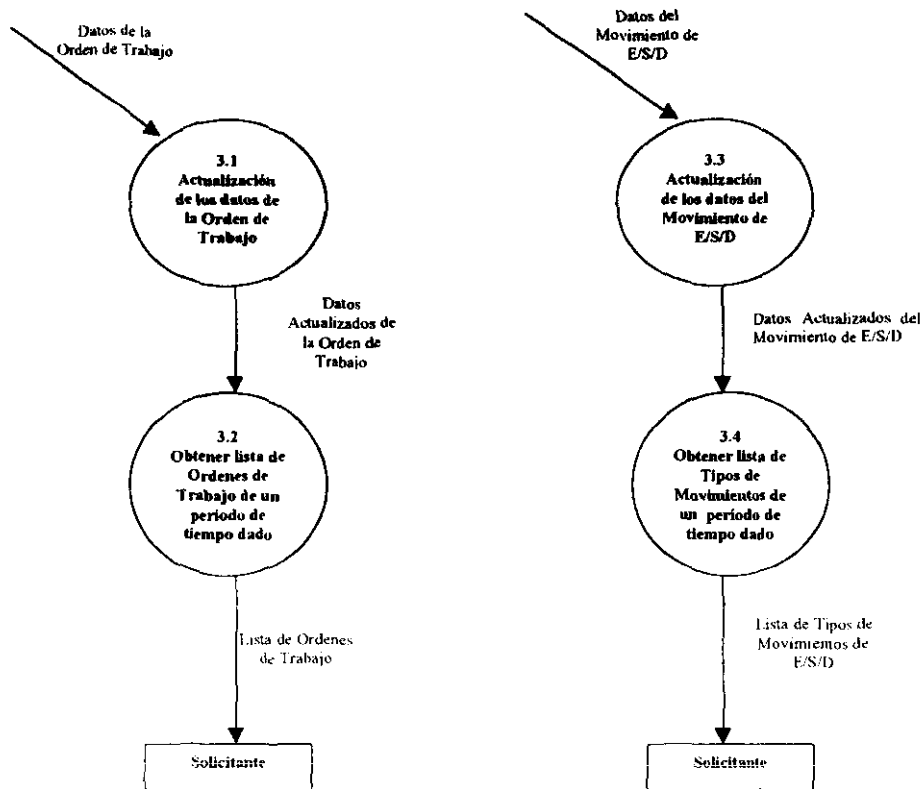
Nivel 1

Diagrama de Flujo de Datos "Actualización de los Datos de Activo Fijo"



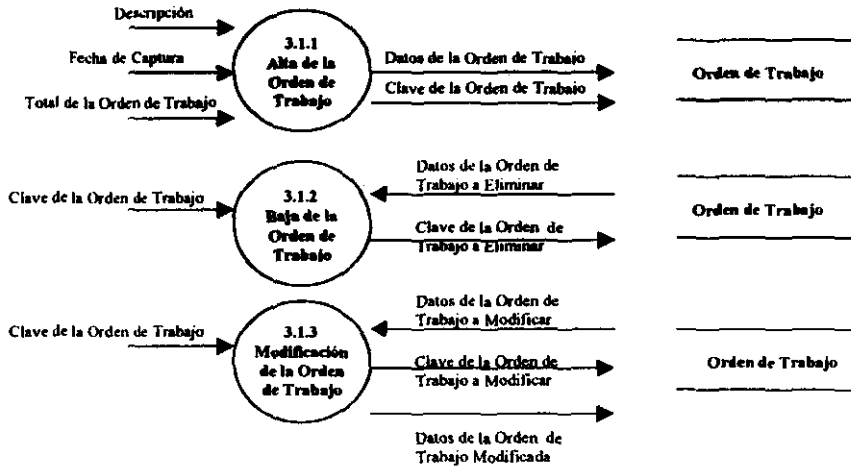
Nivel I

Diagrama de Flujo de Datos "Movimientos de Artículos en Almacenes"



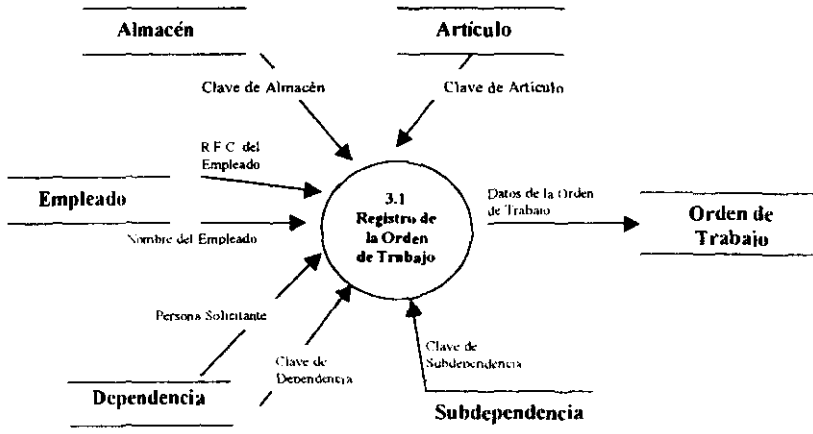
E/S/D = Entrada / Salida / Devolución

Diagrama de Flujo de Datos "Actualización de los Datos de la Orden de Trabajo"



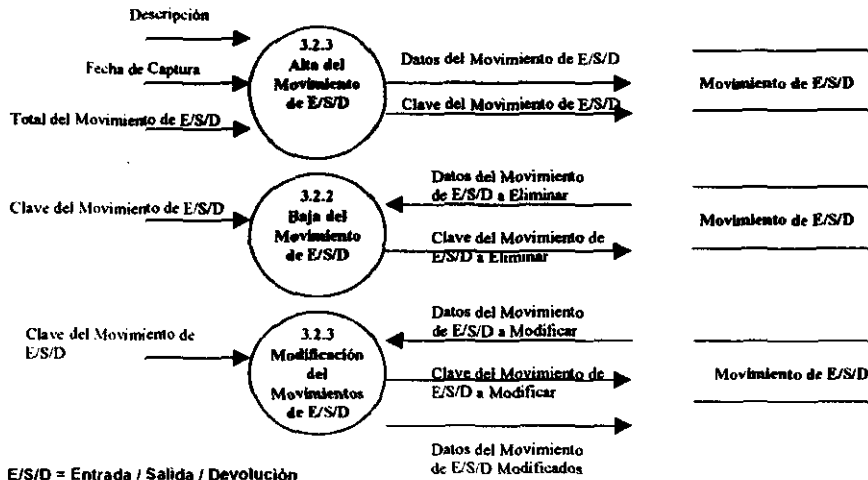
Nivel 1

Diagrama de Flujo de Datos "Registro de la Orden de Trabajo"



Nivel 2

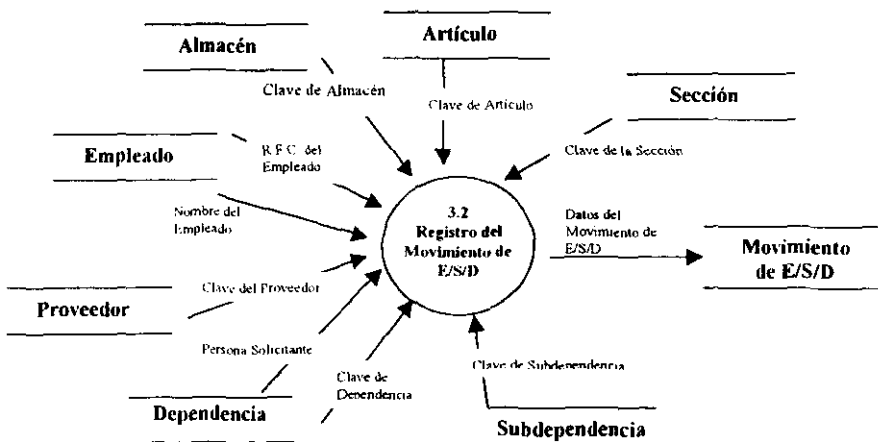
Diagrama de Flujo de Datos "Actualización de los Datos del Movimiento de E/S/D"



E/S/D = Entrada / Salida / Devolución

Nivel 1

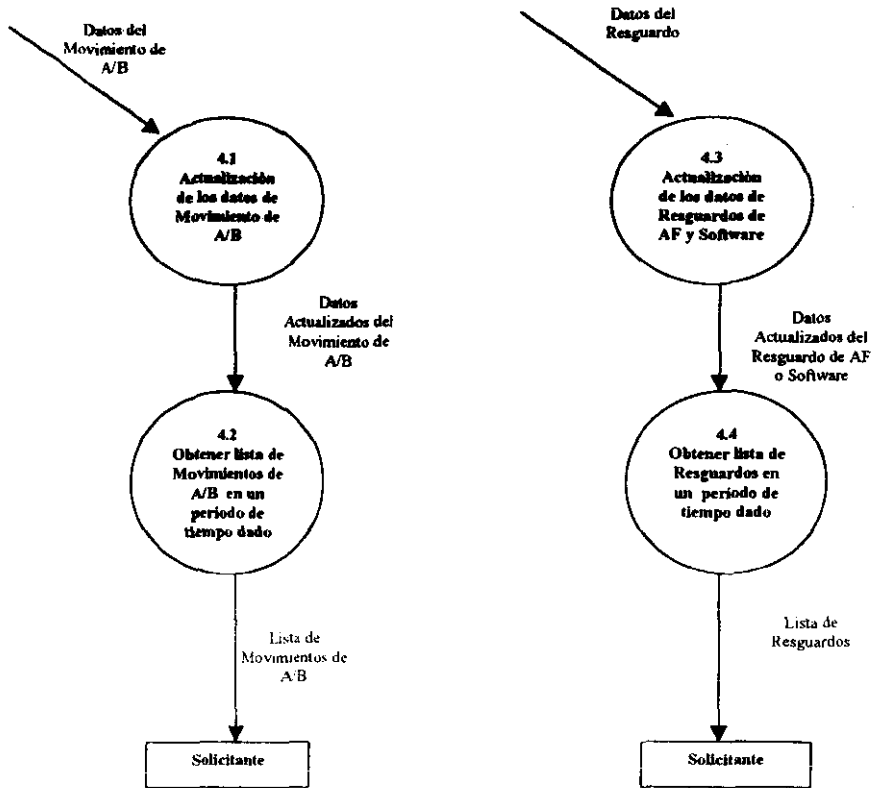
Diagrama de Flujo de Datos "Registro del Movimiento de E/S/D"



E/S/D = Entrada / Salida / Devolución

Nivel 2

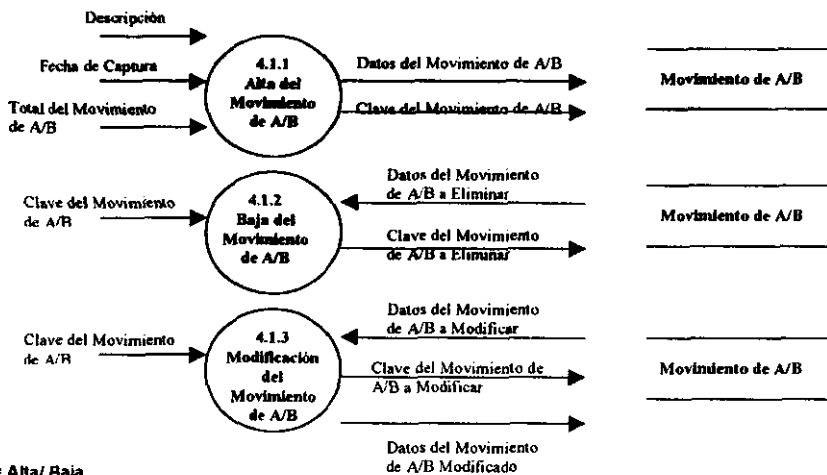
Diagrama de Flujo de Datos "Movimientos de Activo Fijo en Almacenes"



A/B = Altas / Bajas
AF = Activo Fijo

Nivel I

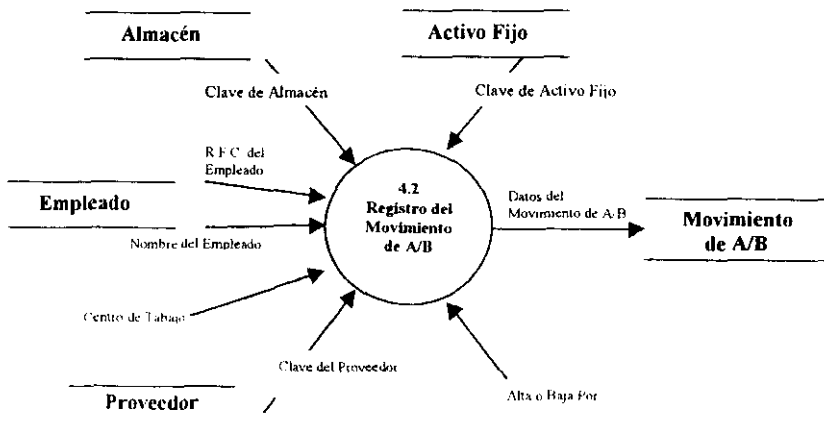
Diagrama de Flujo de Datos "Actualización de los Datos del Movimiento de A/B"



A/B = Alta/ Baja

Nivel 1

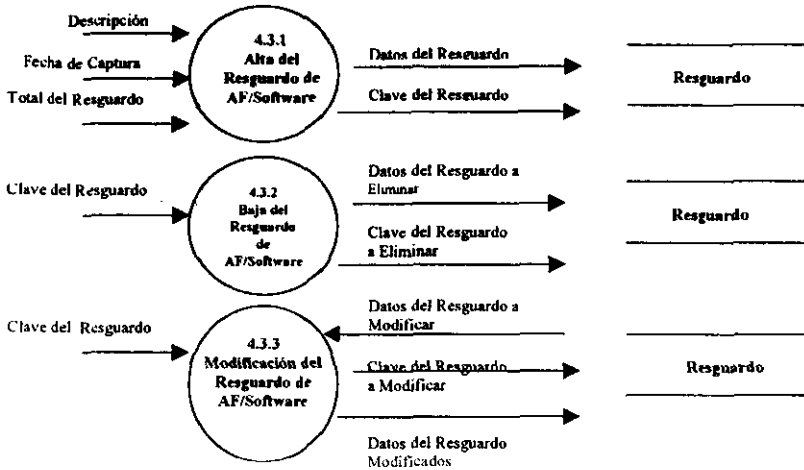
Diagrama de Flujo de Datos "Registro del Movimiento de A/B"



A/B = Alta / Baja

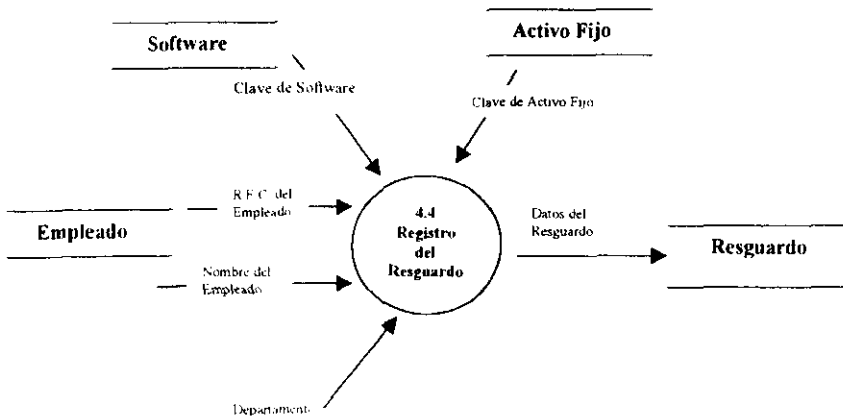
Nivel 2

Diagrama de Flujo de Datos "Actualización de los Datos del Resguardo de AF/Software"



Nivel 1

Diagrama de Flujo de Datos "Registro del Resguardo de AF/Software"



Nivel 2

3 DICCIONARIO DE DATOS

Atributo	Tipo de Dato	Longitud	Descripción
AlcCve	Char	2	Clave de Almacén
AlcDesc	Char	60	Descripción
AlcDir	Char	100	Dirección
AlcEnc	Char	50	Encargado

Atributo	Tipo de Dato	Longitud	Descripción
ProvCve	Char	2	Clave de Proveedor
ProvDesc	Char	60	Descripción
ProvDir	Char	100	Dirección
ProvAG	Char	40	Actividad o Giro

Atributo	Tipo de Dato	Longitud	Descripción
ArtCve	Char	2	Clave de Artículo
ArtDesc	Char	60	Descripción
ArtUni	Char	13	Unidad
ArtFech	Date	8	Fecha de Captura
ArtEMin	N Numérico	9	Existencia Mínima
ArtEMax	N Numérico	9	Existencia Máxima
ArtEAct	N Numérico	9	Existencia Actual
ArtPU	N Numérico	14.2	Precio Unitario
ArtPP	N Numérico	14.2	Precio Promedio
ArtNoFact	Char	9	No. De Factura
ArtPPre	Char	4	Partida Presupuestal

Atributo	Tipo de Dato	Longitud	Descripción
AFcve	Char	8	Clave de Activo Fijo
AFDesc	Char	60	Descripción
AFUni	Char	13	Unidad
AFGpo	N Numérico	3	Grupo al que pertenece
AFNInv	N Numérico	16	No. De Inventario
AFNSer	Char	16	No. De Serie
AFMod	Char	35	Modelo
AFMarc	Char	35	Marca
AFfecAd	Date	8	Fecha de Adquisición
AFTipAdq	Char	35	Tipo de Adquisición
AFPU	N Numérico	14.2	Precio Unitario

Entidad: Dependencias			
Atributo	Tipo de Dato	Longitud	Descripción
DepCve	Char	2	Clave de Dependencia
DepDesc	Char	60	Descripción
DepDir	Char	100	Dirección
DepRep	Char	50	Responsable

Entidad: Programas			
Atributo	Tipo de Dato	Longitud	Descripción
ProCve	Char	8	Clave del Programa
ProNombre	Char	25	Nombre del Programa
ProModulo	Char	25	Modulo al que pertenece
ProDesc	Char	100	Descripción de Módulo

Entidad: Empleados			
Atributo	Tipo de Dato	Longitud	Descripción
EmpRFC	Char	13	R.F.C. Del Empleado
EmpNom	Char	50	Nombre del Empleado
EmpPto	Char	30	Puesto
EmpAds	Char	35	Adscripción

Entidad: Usuarios			
Atributo	Tipo de Dato	Longitud	Descripción
Usuld	Char	8	Identificación del Usuario
UsuPwd	Char	10	Password
UsuNombre	Char	50	Nombre de la Persona

Entidad: Subdependencias			
Atributo	Tipo de Dato	Longitud	Descripción
SubCve	Char	2	Clave de Subdependencia
SubDesc	Char	60	Descripción

Entidad: Niveles de Acceso			
Atributo	Tipo de Dato	Longitud	Descripción
NivCve	Char	8	Clave de Nivel
NivDesc	Char	100	Descripción

Atributo	Tipo de Dato	Longitud	Descripción
CPUCve	Numérico	16	Clave (No. De Inventario)
CPUNSer	Char	16	No. De Serie
CPUDesc	Char	100	Descripción
CPUMarca	Char	25	Marca
CPUModelo	Char	25	Modelo
CPUMemoria	Numérico	3	Memoria
CPUVel	Numérico	4	Velocidad
CPUTRed	Char	35	Tarjeta de Red
CPUTVideo	Char	35	Tarjeta de Video
CPUDA	Char	50	Drive A
CPUDB	Char	50	Drive B
CPUDD	Char	50	Disco Duro

Atributo	Tipo de Dato	Longitud	Descripción
SoftCve	Char	16	Clave de Software
SoftDesc	Char	100	Descripción de Software
SoftCnt	Numérico	3	Cantidad de Software
SoftVer	Char	5	Versión
SoftFecAd	Date	8	Fecha de Adquisición
SoftAdq	Char	35	Tipo de Adquisición
SoftPU	Numérico	14.2	Precio Unitario
SoftMan	Numérico	5	Manuales

Atributo	Tipo de Dato	Longitud	Descripción
SecCve	Char	2	Clave de Sección
SecDesc	Char	60	Descripción

Entidad: Impresoras			
Atributo	Tipo de Dato	Longitud	Descripción
ImpCve	Numérico	16	Clave (No. De Inventario)
ImpNoSer	Char	16	No. De Serie
ImpDesc	Char	100	Descripción
ImpMarca	Char	25	Marca
ImpMod	Char	25	Modelo
ImpTec	Char	30	Tecnología
ImpMult	Char	30	Multiplexor

Entidad: Teclados			
Atributo	Tipo de Dato	Longitud	Descripción
TecCve	Numérico	16	Clave (No. De Inventario)
TecNoSer	Char	16	No. De Serie
TecDesc	Char	100	Descripción
TecMarca	Char	25	Marca
TecMod	Char	25	Modelo

Entidad: Monitores			
Atributo	Tipo de Dato	Longitud	Descripción
MonCve	Numérico	16	Clave (No. De Inventario)
MonNoSer	Char	16	No. De Serie
MonDesc	Char	100	Descripción
MonMarca	Char	25	Marca
MonMod	Char	25	Modelo

Entidad: Varios			
Atributo	Tipo de Dato	Longitud	Descripción
VarCve	Numérico	16	Clave (No. De Inventario)
VarNoSer	Char	16	No. De Serie
VarDesc	Char	100	Descripción
VarMarca	Char	25	Marca
VarMod	Char	25	Modelo

Entidad: Mouse			
Atributo	Tipo de Dato	Longitud	Descripción
MouCve	Numérico	16	Clave (No. De Inventario)
MouNoSer	Char	16	No. De Serie
MouDesc	Char	100	Descripción
MouMarca	Char	25	Marca
MouMod	Char	25	Modelo

Entidad: Reguladores			
Atributo	Tipo de Dato	Longitud	Descripción
RegCve	Numérico	16	Clave (No. De Inventario)
RegNoSer	Char	16	No. De Serie
RegDesc	Char	100	Descripción
RegMarca	Char	25	Marca
RegMod	Char	25	Modelo
RegCap	Char	16	Capacidad

Atributo	Tipo de Dato	Longitud	Descripción
ResCve	Char	8	Clave de Resguardo
ResFec	Date	8	Fecha de Resguardo
ResFecCap	Date	8	Fecha de Captura
ResDep	Char	35	Departamento
ResGpo	Númérico	3	Grupo
ResImp	Númérico	14.2	Importe Por Activo
ResCant	Númérico	5	Cantidad
ResTotal	Númérico	14.2	Total del Resguardo
ResUsu	Char	15	Usuario

Atributo	Tipo de Dato	Longitud	Descripción
MovCve	Char	8	Clave de Movimiento
MovDesc	Char	16	Descripción
MovFec	Date	8	Fecha del Movimiento
MovFecCap	Date	8	Fecha de Captura
MovCant	Númérico	5	Cantidad de "X" Artículo
MovTotal	Númérico	14.2	Total del Movimiento
MovImp	Númérico	14.2	Importe Por Artículo
MovUsu	Char	15	Usuario

Entidad: Movimiento de Artículos			
Atributo	Tipo de Dato	Longitud	Descripción
MABCve	Char	8	Clave de Movimiento
MABFec	Date	8	Fecha del Movimiento
MABFecCap	Date	8	Fecha de Captura
MABCTrab	Char	35	Centro de Trabajo
MABAAlta	Char	20	Movimiento de Alta
MABBaja	Char	20	Movimiento de Baja
MABObser	Char	35	Observaciones
MABCant	Númérico	5	Cantidad de "X" Activo
MABTotal	Númérico	14.2	Total del Movimiento
MABImp	Númérico	14.2	Importe del Movimiento
MABFact	Char	9	Factura
MABUsu	Char	15	Usuario

Entidad: Ordenes de Trabajo			
Atributo	Tipo de Dato	Longitud	Descripción
OrdCve	Char	8	Clave de Orden de Trabajo
OrdDesc	Char	35	Descripción
OrdFec	Date	8	Fecha de la Orden
OrdFecCap	Date	8	Fecha de Captura
OrdPerSol	Char	35	Persona Solicitante
OrdObser	Char	100	Observaciones
OrdSup	Char	35	Persona que Supervisó
OrdCant	Númérico	5	Cantidad de Artículos
OrdImp	Númérico	14.2	Importe Por Artículo
OrdTotal	Númérico	14.2	Total de la Orden
OrdUsu	Char	15	Usuario

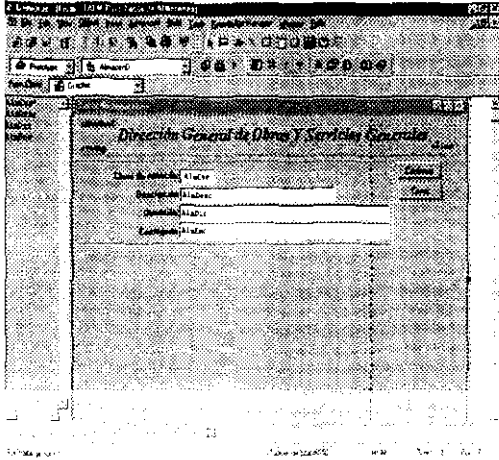
Entidad: Resguardos de Software			
Atributo	Tipo de Dato	Longitud	Descripción
ResSoftCve	Char	8	Clave de Resguardo
ResSoftFec	Date	8	Fecha de Resguardo
ResSoftFCap	Date	8	Fecha de Captura
ResSoftDep	Char	35	Departamento
ResSoftImp	Númérico	14.2	Importe Por Activo
ResSoftCant	Númérico	5	Cantidad de Software
ResSoftTotal	Númérico	14.2	Total del Resguardo
ResSoftUsu	Char	15	Usuario

Entidad: Niveles de Acceso			
Atributo	Tipo de Dato	Longitud	Descripción
NivCve	Char	8	Clave de Nivel
NivDesc	Char	35	Descripción

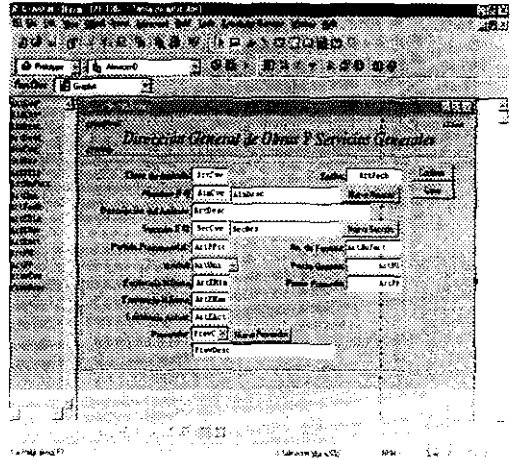
4.9 DESARROLLO DEL SISTEMA

Diseño de algunas de las transacciones: Almacenes, Artículos, Activo Fijo, Empleados, Ordenes de Trabajo, Movimientos de E/S/D, Movimientos de A/B, Resguardos.

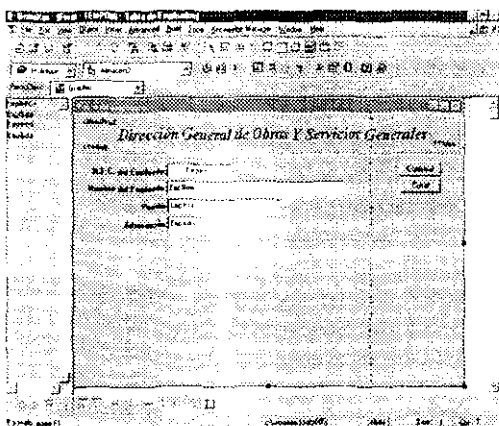
Transacción de Almacenes



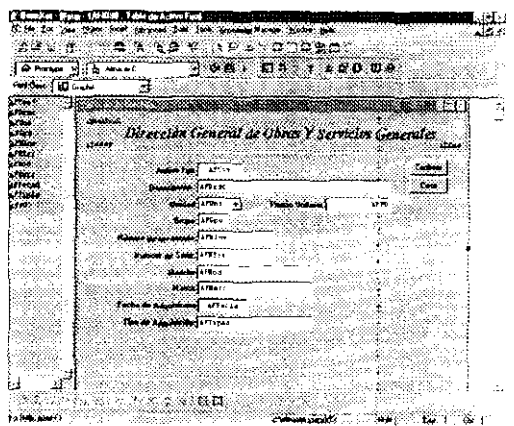
Transacción de Artículos



Transacción de Empleados



Transacción de Activo Fijo



Transacción de Ordenes de Trabajo

Formulario de transacción de Ordenes de Trabajo. Encabezado: Dirección General de Obras Y Servicios Generales. Campos de entrada: Clase de Movimiento, Fecha de Emisión, Fecha de Inicio, Fecha de Fin, Tipo de Orden, Tipo de Servicio, Tipo de Trabajo, Tipo de Materiales, Tipo de Mano de Obra, Tipo de Equipo, Tipo de Herramientas, Tipo de Maquinaria, Tipo de Vehículos, Tipo de Otros. Sección de Tabla de Ordenes de Trabajo con columnas: No. de Orden, Descripción, Cantidad, Unidad, Valor, Estado, Fecha de Emisión, Fecha de Inicio, Fecha de Fin, Tipo de Orden, Tipo de Servicio, Tipo de Trabajo, Tipo de Materiales, Tipo de Mano de Obra, Tipo de Equipo, Tipo de Herramientas, Tipo de Maquinaria, Tipo de Vehículos, Tipo de Otros.

Transacción de Movimientos E/S/D

Formulario de transacción de Movimientos E/S/D. Encabezado: Dirección General de Obras Y Servicios Generales. Campos de entrada: Clase de Movimiento, Fecha de Emisión, Fecha de Inicio, Fecha de Fin, Tipo de Orden, Tipo de Servicio, Tipo de Trabajo, Tipo de Materiales, Tipo de Mano de Obra, Tipo de Equipo, Tipo de Herramientas, Tipo de Maquinaria, Tipo de Vehículos, Tipo de Otros. Sección de Tabla de Movimientos E/S/D con columnas: No. de Orden, Descripción, Cantidad, Unidad, Valor, Estado, Fecha de Emisión, Fecha de Inicio, Fecha de Fin, Tipo de Orden, Tipo de Servicio, Tipo de Trabajo, Tipo de Materiales, Tipo de Mano de Obra, Tipo de Equipo, Tipo de Herramientas, Tipo de Maquinaria, Tipo de Vehículos, Tipo de Otros.

Transacción de Movimientos de A/B

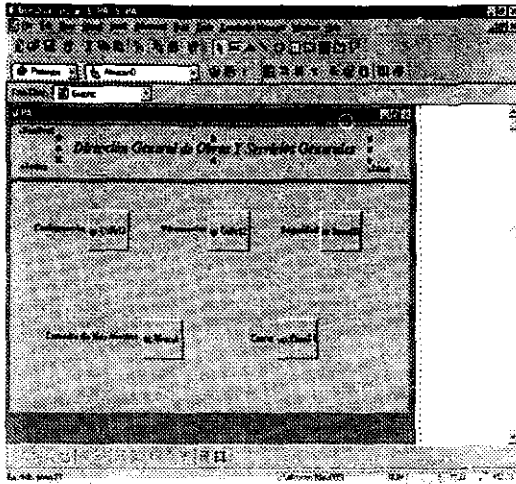
Formulario de transacción de Movimientos de A/B. Encabezado: Dirección General de Obras Y Servicios Generales. Campos de entrada: Clase de Movimiento, Fecha de Emisión, Fecha de Inicio, Fecha de Fin, Tipo de Orden, Tipo de Servicio, Tipo de Trabajo, Tipo de Materiales, Tipo de Mano de Obra, Tipo de Equipo, Tipo de Herramientas, Tipo de Maquinaria, Tipo de Vehículos, Tipo de Otros. Sección de Tabla de Movimientos de A/B con columnas: No. de Orden, Descripción, Cantidad, Unidad, Valor, Estado, Fecha de Emisión, Fecha de Inicio, Fecha de Fin, Tipo de Orden, Tipo de Servicio, Tipo de Trabajo, Tipo de Materiales, Tipo de Mano de Obra, Tipo de Equipo, Tipo de Herramientas, Tipo de Maquinaria, Tipo de Vehículos, Tipo de Otros.

Transacción de Resguardos

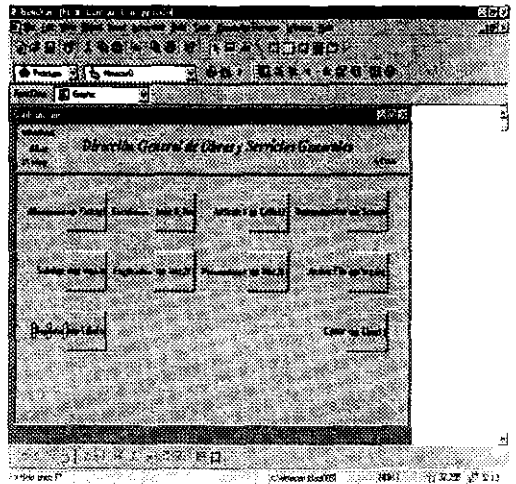
Formulario de transacción de Resguardos. Encabezado: Dirección General de Obras Y Servicios Generales. Campos de entrada: Clase de Movimiento, Fecha de Emisión, Fecha de Inicio, Fecha de Fin, Tipo de Orden, Tipo de Servicio, Tipo de Trabajo, Tipo de Materiales, Tipo de Mano de Obra, Tipo de Equipo, Tipo de Herramientas, Tipo de Maquinaria, Tipo de Vehículos, Tipo de Otros. Sección de Tabla de Resguardos con columnas: No. de Orden, Descripción, Cantidad, Unidad, Valor, Estado, Fecha de Emisión, Fecha de Inicio, Fecha de Fin, Tipo de Orden, Tipo de Servicio, Tipo de Trabajo, Tipo de Materiales, Tipo de Mano de Obra, Tipo de Equipo, Tipo de Herramientas, Tipo de Maquinaria, Tipo de Vehículos, Tipo de Otros.

Diseño de algunas de algunos de los Menús del Sistema: Menú Principal, Menú de Codificación, Menú de Movimientos en Almacenes o Activo Fijo, Menú de Consultas.

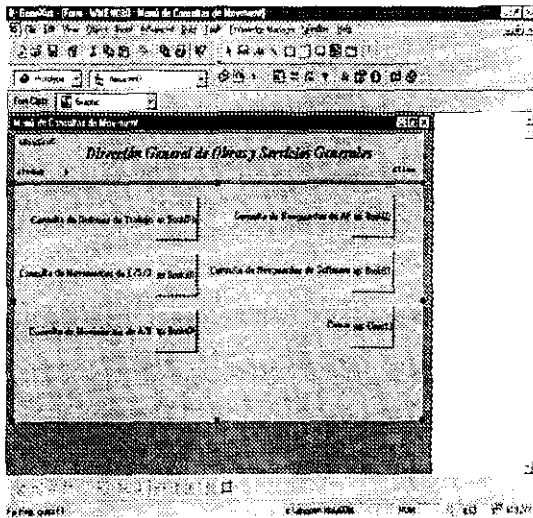
Menú Principal



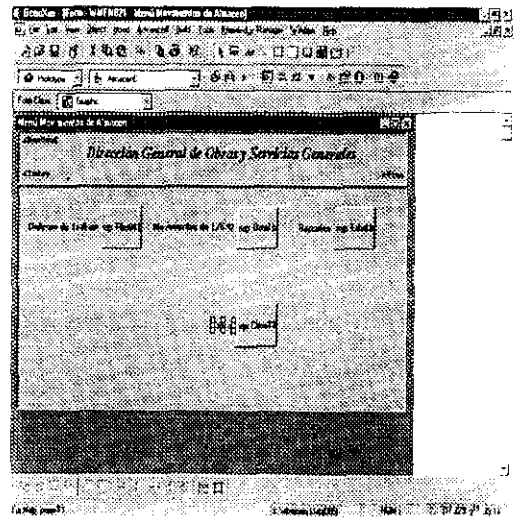
Menú de Configuración



Menú de Consultas



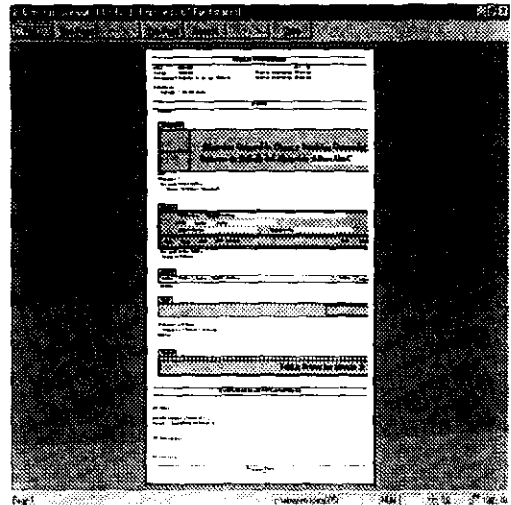
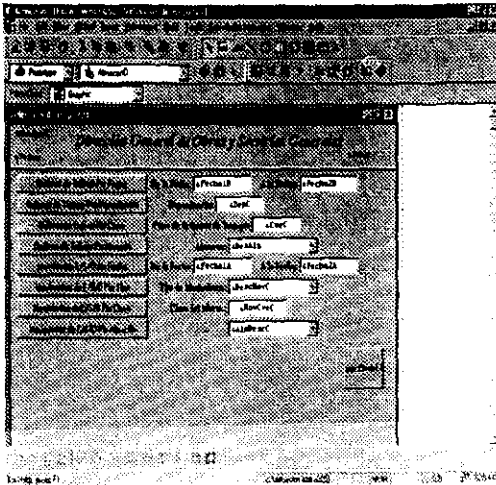
Menu de Movimientos en Almacenes



Diseño de algunas de algunas de las pantallas que mandan llamar los Reportes que genera el Sistema.

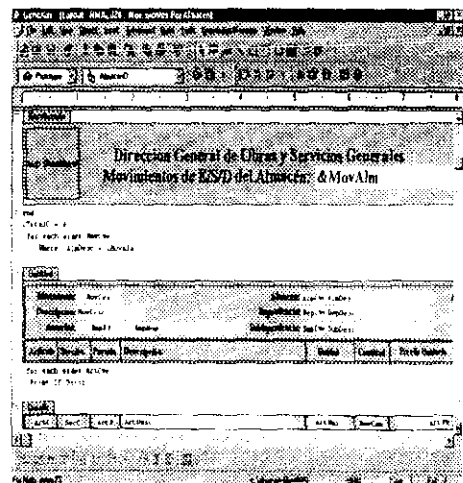
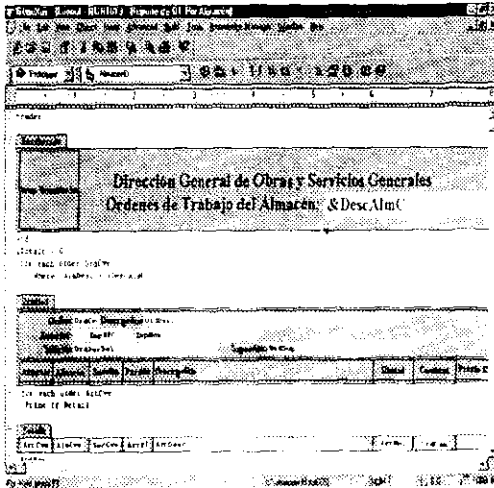
Menú de Reportes Generales

Reporte de Ordenes de trabajo por Dep.



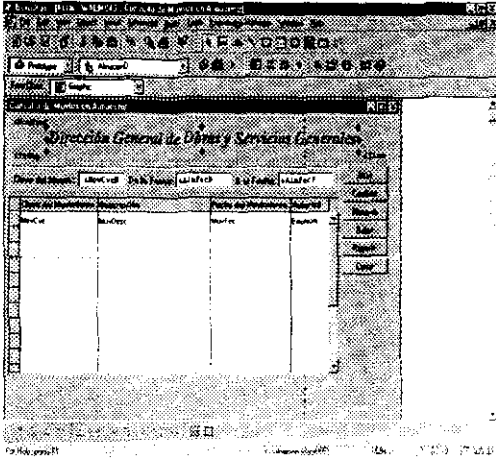
Reporte de Ordenes de trabajo por Dep

Reporte de Movtos. E/S/D Por Almacen

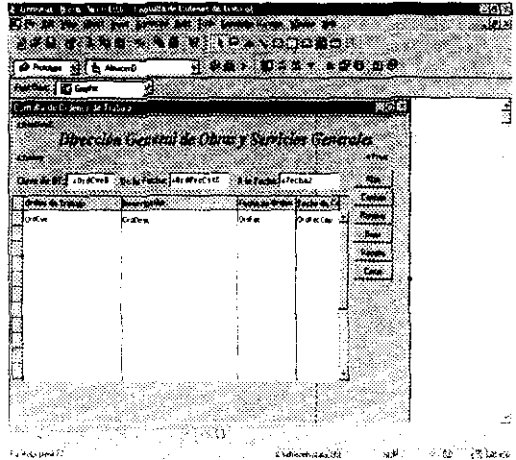


Diseño de algunos de los Wok Panels donde se visualizarán la información de determinado rango de valores de búsqueda.

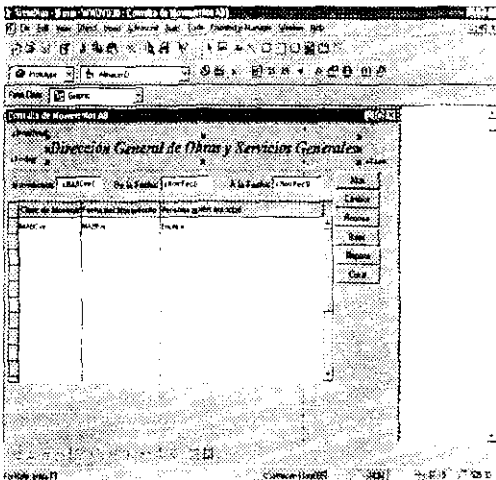
Consulta de Movimientos de Almacenes



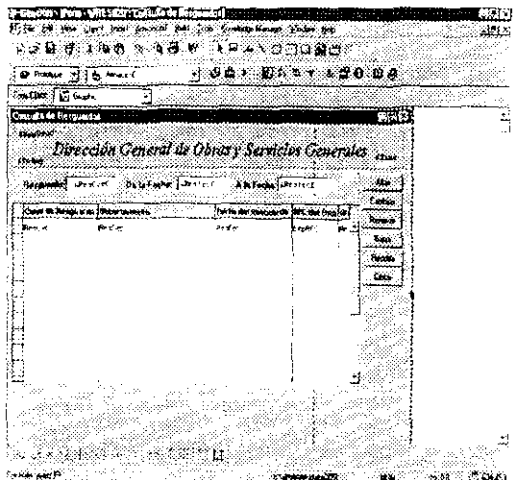
Consulta de Ordenes de trabajo



Consulta de Movimientos de AF.



Consulta de Resguardos



CAPITULO V. SEGURIDAD Y MANTENIMIENTO

5.1 SEGURIDAD DEL SISTEMA

Para analizar los mecanismos de seguridad en el control de concurrencia primero se introduce la noción de transacción. Una petición en línea a la Base de Datos se considera como una transacción que generalmente involucra llamadas a rutinas del DBMS para operaciones. El procesamiento de la transacción empieza tan pronto como se propone la petición del usuario, termina cuando se contesta la petición y en ese momento el sistema libera todos los recursos utilizados para la transacción.

Algunos DBMS tienen mecanismos de seguridad para prevenir actualizaciones concurrentes sin control. Cuando se invocan simultáneamente varias transacciones para acceder los mismos datos, se fuerza a cada transacción a comprobar la existencia de un seguro asociado con los datos.

La figura 5.1.1.1 muestra un mecanismo de seguridad que acomoda en serie dos operaciones de actualización concurrentes. Una transacción puede acceder los datos sólo si el seguro asociado con los datos esta abierto. Una vez que el programa tiene acceso a los datos, pone el seguro para así bloquear el acceso a los datos a las demás transacciones. El seguro se restablece (reset) cuando la transacción completa su operación de actualización.

La seguridad de los datos está fuertemente relacionada con la integridad de los mismos. La seguridad se refiere a la protección de la base de datos contra accesos o modificaciones no autorizadas. Como los datos de una DB se comparten ampliamente, la información confidencial en la base es vulnerable a las intromisiones. Un acceso ilegal puede tener como resultado la destrucción accidental o maliciosa de la DB. Sin control de seguridad, los usuarios no tendrán la privacidad requerida en sus datos confidenciales y el sistema no podrá mantener la integridad de los mismos.

Por estas razones, se proporciona un mecanismo en el DBMS para reforzar la seguridad de los datos. Algunos mecanismos para el control de seguridad son el subesquema, los seguros de control de acceso y el poner los datos en claves secretas.

5.1.1 METODOS DE CONTROL DE ACCESO A DATOS

El Administrador de la Base de Datos (DBA) controla los derechos de acceso escogiendo cierto subconjunto de la base (representado por un subesquema) para el usuario. En otras palabras, los datos ocultos fuera del subesquema serán inaccesibles para las aplicaciones que usen dicho subesquema. Aún más, el DBA puede limitar al usuario sólo a ciertos privilegios (lectura, inserción, borrado y/o actualización) sobre algunos datos.

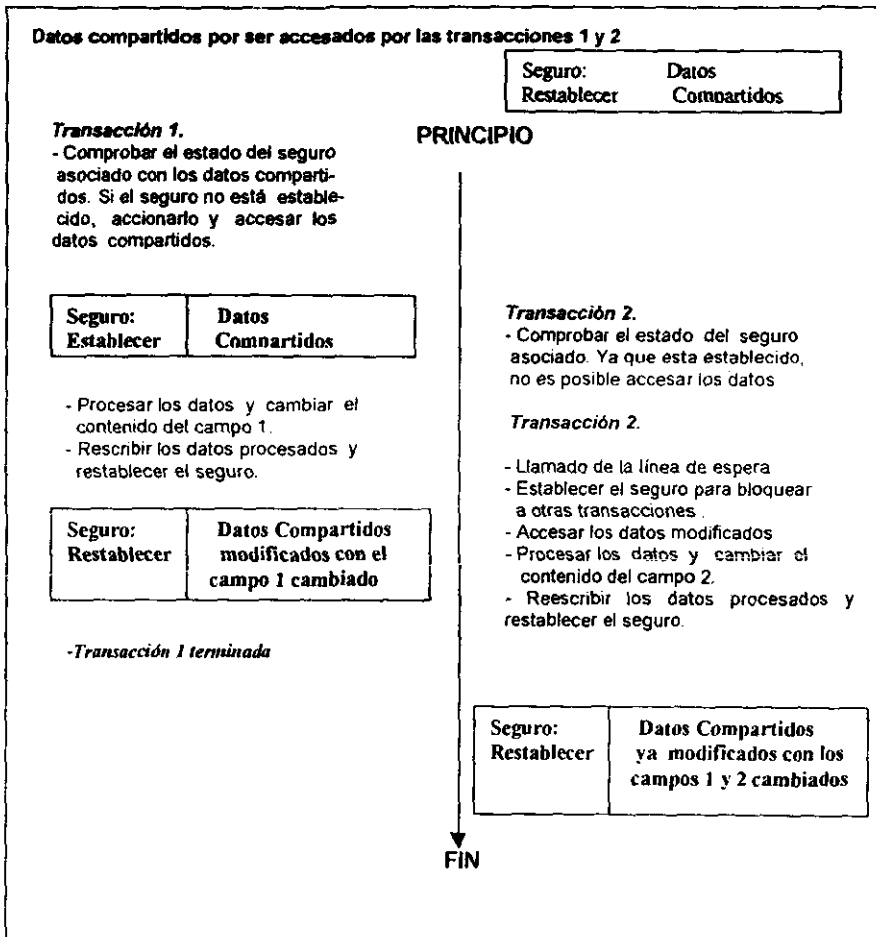


Figura 5.1 Mecanismos de seguridad para actualizaciones simultáneas

5.1.2 OBJETIVOS POR ASEGURAR

Los objetivos por asegurar pueden ser datos, esquemas, subesquemas o programas.

1) Datos

Los seguros de control de acceso sobre campos o registros se pueden especificar en la declaración del esquema o subesquema.

2) Esquemas

A todos los usuarios autorizados se les asigna una prerrogativa de lectura para acceder el esquema. Sin embargo, sólo algunos tienen la prerrogativa de actualizar o escribir para modificar el esquema, ya que si se introduce algún error en la actualización del mismo, toda la base sería inaccesible.

3) Subesquema

El DBA también puede definir distintos niveles de seguridad para el uso de subesquemas. Nuevamente, los seguros para manejar el subesquema se declaran en el mismo subesquema.

4) Programas

Además de los elementos de los datos, esquemas o subesquemas, también puede asegurarse un programa de aplicación para evitar que los usuarios no autorizados corran el programa. El método más sencillo para restringir un programa a ciertos usuarios elegidos es solicitar una contraseña en pantalla antes de entrar en él. En caso de que no sea válida, el programa simplemente pasará por alto el algoritmo y terminará.

5.2. MANTENIMIENTO DEL SISTEMA

El Mantenimiento del sistema ha sido caracterizado como un "iceberg": esperamos que lo que está inmediatamente visible sea todo lo que hay. Realmente, sabemos que bajo la superficie se esconde una gran cantidad de problemas potenciales y de costos. El mantenimiento del software existente puede llevarse hasta el 70 % de todo el esfuerzo gastado por una organización de desarrollo. El porcentaje sigue subiendo a medida que se produce más software. Los lectores que no conozcan el tema se pueden preguntar por qué se requiere tanto mantenimiento y por qué se desperdicia tanto esfuerzo. Una respuesta parcial nos la da Osborne y Chikofsky:

Una gran cantidad del software del que dependemos actualmente tiene una media de edad de entre 10 y 15 años. Incluso aunque esos programas se hubieran creado con las mejores técnicas de diseño y codificación existentes en su momento, fueron creados teniendo como principales requisitos el tamaño del programa y el espacio de almacenamiento. Luego migraron a plataformas nuevas, fueron ajustados a cambios en las máquinas y en los sistemas operativos y mejorados para satisfacer las necesidades de sus nuevos usuarios -todo ello sin tener en cuenta su arquitectura general. El resultado ha sido la existencia de sistemas de software con unas estructuras de datos pobremente diseñadas, una pobre codificación, una logica pobre y una documentación pobre, teniendo que seguir funcionando en la actualidad.

La naturaleza omnipresente del cambio mediatiza todo el trabajo en software. El cambio es inevitable en la construcción de sistemas basados en computadora, por ello, debemos desarrollar mecanismos de evaluación, control e implementación de modificaciones.

El Mantenimiento del Sistema o también llamado Soporte del Sistema es el que se le da a un sistema después de que se pone en operación. Esto incluye desde luego las mejoras que se le pueda hacer al sistema. El soporte es realizado por las personas que se encargan de desarrollar el sistema. Existen algunos objetivos y bloques elementales para el mantenimiento del sistema como son:

- Preservar aquellos aspectos de los programas que fueron ya corregidos. Se debe evitar la posibilidad de que los arreglos en dichos programas originen que otros aspectos de los mismos funcionen de modo diferente.

- Hacer cambios predecibles en los programas existentes para corregir errores que se cometieron durante el diseño y la implantación del sistema. En consecuencia, excluimos de esta actividad las mejoras y las nuevas necesidades.

Para alcanzar estos objetivos, ha de tenerse un conocimiento apropiado de los programas que se están arreglando y de las aplicaciones en que intervienen dichos programas. Ya que si no se tiene ese conocimiento no podríamos llevar acabo de manera adecuada el mantenimiento del sistema y se llevaría a un rotundo fracaso.

En la figura 5.2 se muestran los elementos de información de los sistemas y a continuación se da una descripción de lo que son cada uno de estos elementos:

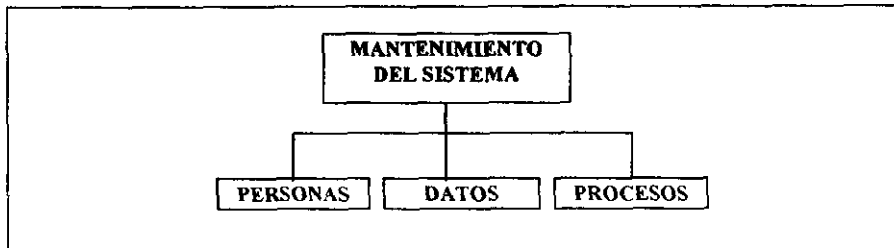


Figura 5.2. Bloques elementales de los sistemas

PERSONAS: El mantenimiento de los sistemas es normalmente iniciado por programadores de sistema, es decir por las personas que lo elaboraron.

DATOS: El mantenimiento de sistemas rara vez influye sobre los datos, salvo por la posibilidad de que se mejore la edición de los mismos.

PROCESOS: Se refiere a los procesos de los sistemas de empresa y de información que se implantan finalmente como programas de aplicación. El mantenimiento de sistemas consiste en arreglar los errores cometidos durante la implantación de dichos programas.

Existen diferentes pasos que conforman el Mantenimiento de Sistemas. Mostramos el diagrama en la figura 5.3 y damos una descripción de ellos para su mejor comprensión.

1. Definir y validar los problemas

Este es el primer paso a seguir en el Mantenimiento de Sistemas en el cual se debe de validar él o los programas consiguiendo reproducirlos. Si el problema no puede reproducirse, debería suspender el proyecto hasta que se reprodujera el problema y el usuario pudiera explicar las circunstancias en las cuales tuvo lugar. Una posible salida serían las solicitudes de cambio validadas. Estas solicitudes de cambio deberían definir las expectativas de solución. Como entrada se tiene el conjunto de errores encontrados al usar el sistema, a los cuales también se les conoce como bugs.

En el caso de que volviera a reproducirse el error, los usuarios deberán documentar del mejor modo posible las circunstancias que llevaron a la aparición del error y de los síntomas del problema.

Se podría decir, que el error se debe a una mala comprensión o a un mal uso del sistema, y las instrucciones de corrección podrían llevar al cierre de todo el proyecto.

Además si el error ha sido validado, se pasarán los problemas y programas validados a la siguiente tarea. Todo el mantenimiento posterior se llevará a cabo sobre una copia del o de los programas. El programa original permanecerá en la biblioteca de programas y podrá usarse en los sistemas en producción hasta que se subsane el error.

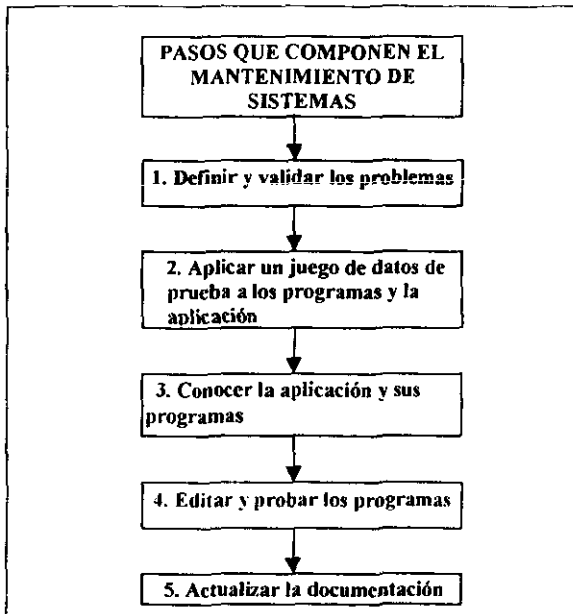


Figura 5.3 Pasos que componen el Mantenimiento del Sistemas

2. Aplicar un juego de datos de prueba a los programas y a la aplicación.

El mantenimiento de sistemas puede descubrir efectos impredecibles y no deseables que influirán sobre el funcionamiento y el rendimiento global de los programas o la aplicación. Por este motivo, recomendamos encarecidamente que, antes de realizar algún cambio en los programas, se ejecuten y se prueben para definir una línea de partida con respecto a la cual puedan compararse los programas y aplicaciones modificadas. Este paso es llevado a cabo por el analista y/o el programador de sistemas.

Los casos del juego de datos de prueba pueden definirse de dos maneras posibles. La primera consistirá en buscar datos de prueba antiguos en el diccionario. Si existieran, deberían de servir como juego de datos de prueba para la ejecución del programa.

La segunda también deberían analizarse si son suficientemente complejos y si fuera necesario, habría que revisarlos. Las respuestas correctas, incluidos los errores, deberían grabarse en el diccionario.

3. Conocer la aplicación y sus programas

Frecuentemente el mantenimiento de sistemas no es realizado por las mismas personas que escribieron el sistema. De hecho, tal vez varias personas hayan escrito partes de un programa o aplicación y ya no estén disponibles para posibles aclaraciones. Por esta razón, es preciso obtener un conocimiento de la aplicación y de los programas problemáticos. Tal vez sorprenda saber, que en su mayoría, los programadores y los analistas invierten en esta tarea más tiempo que en ninguna otra.

En términos ideales, el conocimiento de la aplicación se obtiene del diccionario. Ello supone, claro está, que el conocimiento de la aplicación ha sido mantenido a lo largo de toda la vida de la aplicación. Pero con mucha frecuencia esto no es verdad, en especial en los sistemas más antiguos.

En centros de sistemas de información no basados en diccionarios, puede conseguirse el conocimiento de la aplicación de los programadores y analistas anteriores, pero dicho conocimiento puede no estar al día. Sin embargo, puede ser de utilidad para alcanzar un nivel suficiente de comprensión de la aplicación y de donde encajan los programas problemáticos dentro de dicha aplicación.

El conocimiento de la aplicación y los programas proviene por lo general del estudio del código fuente de los programas probados con juegos de datos de prueba. Por desgracia, la comprensión de los programas puede llevar un tiempo considerable. Esta actividad se realiza por la combinación de una serie de limitaciones como son:

Una deficiente estructura modular.

Uso de lógica no estructurada (procedente de la era anterior a la estructuración).

Mantenimientos previos

Arreglos rápidos y aplicaciones deficientemente diseñadas.

Código muerto, es decir, instrucciones que nunca se ejecutan o a las que nunca se llega, a menudo olvidadas durante fases anteriores de prueba y depuración.

Documentación deficiente o inadecuada.

El propósito de conocer la aplicación es llegar a tener una visión general del asunto, es decir, saber como encajan los programas en una aplicación total y cómo interaccionan con otros programas. También tiene como objetivo la comprensión de los programas conseguir suficiente información sobre como funciona el programa, y sobre lo que no funciona. Conocer los programas puede llevar también a mejores estimaciones del tiempo y los recursos que se requerirán para arreglar los errores. No suele haber atajos posibles en la comprensión de los programas. En la actualidad la tecnología CASE puede servir de ayuda para el mantenimiento.

4. Editar y probar los programas

Dado el conocimiento de la aplicación, los programas y los cambios validos, pueden entonces realizarse los cambios en los programas que han de ser modificados.

Existe una gran diferencia entre editar un nuevo programa y editar un programa existente. Como diseñador de un nuevo programa probablemente se estará muy familiarizado con la estructura y la lógica del programa. Por el contrario, como editor del programa existente, la familiaridad no será tan acusada con dicho programa. Los cambios que se introducen pueden tener un efecto de bucle no deseado que afecte a otras partes del programa o, lo que es aún peor, a otros programas de la aplicación.

Es de esperar que los cambios en el código sean mejores si se conoce la aplicación y el programa. Pero más importantes aún en el mantenimiento de sistemas son las pruebas.

Se consideran esenciales las siguientes pruebas:

Pruebas de Unidades: Este tipo de prueba es esencial, ya que asegura que el programa considerado como solitario arregla el error sin efectos colaterales. En este punto se usarán los datos de prueba y rendimiento actual que se recuperaron, crearon, editaron o generaron.

Prueba del Sistema: Esta prueba también es esencial, y asegura que la aplicación en conjunto, de la que forma parte el programa modificado, aún funciona. Se usarán también los datos de prueba y rendimiento actual.

En términos generales, puede decirse que cuando se programa se regresa a la biblioteca de programas y este queda sujeto al control de versiones. En donde el control de versiones es un proceso por el cual el bibliotecario (normalmente un software) hace un seguimiento de los cambios realizados en los programas. Ello permite recuperar versiones anteriores de los mismos en casos de que en las nuevas versiones surjan problemas inesperados. En otras palabras, el control de versiones permite a los usuarios recuperar una versión del sistema aceptada con anterioridad.

5. Actualizar la documentación

El alto costo del mantenimiento de sistemas se debe, en gran parte, a fallos en la actualización de la documentación de una aplicación, debe modificarse en el diccionario y en la biblioteca de programas. La documentación de la aplicación es, por lo general, responsabilidad del analista de sistemas que da soporte a dicha aplicación. La documentación de los programas suele ser responsabilidad del programador que realiza los cambios en los programas, el programador es responsable de esta actividad.

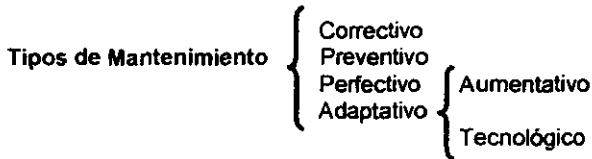
Los cambios en la aplicación se guardan en el diccionario. Los nuevos programas y cambios en los programas se guardan en la biblioteca de programas. Una vez devueltos a la biblioteca, quedan disponibles para producción.

Grabar los cambios de las aplicaciones y los programas en el diccionario y la biblioteca de programas ayudara a los futuros programadores y analistas a reducir el tiempo dedicado al aprendizaje de la aplicación durante futuras tareas de mantenimiento. Los cambios realizados se olvidarán, por pequeños que sean, a menos que se guarde un registro apropiado de ellos.

5.2.1. TIPOS DE MANTENIMIENTO

El mantenimiento del sistema es por supuesto mucho más que una "corrección de errores". Podemos describir el mantenimiento describiendo las cuatro actividades que se llevan a cabo tras distribuir un programa.

Dentro de estas actividades se encuentran los cuatro tipos de mantenimiento, los cuales son:



5.2.1.1. Correctivo

El proceso que incluye el diagnóstico y la corrección de uno o más errores se denomina mantenimiento correctivo. Es la primera actividad del mantenimiento la cual es debida a que no es razonable asumir que la prueba del software haya descubierto todos los errores latentes de un sistema. Durante el uso del sistema se encontrarán errores, los cuales deben ser informados al equipo de desarrollo.

5.2.1.2. Adaptativo

En este tipo de mantenimiento se encuentran implícitos el aumentativo y el tecnológico. El mantenimiento adaptativo se debe a cambios en el ambiente del programa y a la adaptación de nuevas unidades o módulos. Es la segunda actividad que contribuye a la definición de mantenimiento.

La vida útil del software de aplicación puede fácilmente sobrepasar los diez años, pero considerando la evolución del ambiente, en la práctica éste puede volverse obsoleto. Por lo tanto el mantenimiento adaptativo es una actividad que modifica al software para que las iteraciones sean adecuadas con su entorno cambiante.

Aumentativo

Este mantenimiento se da cuando se incluyen nuevas funciones que no se contemplaron al inicio del desarrollo del sistema y surgen como una necesidad del usuario.

5.3 FALLOS GENERALES DEL SISTEMA

De vez en cuando es inevitable que un sistema falle. Este fallo se traduce generalmente en lo que se llama un programa abortado (también llamado Abend o Crash) y en posible pérdida de datos. Es a menudo que el analista de sistemas sea el encargado de arreglar el sistema o de actuar como intermediario entre los usuarios y quienes deben recuperar el sistema. El propósito de esta sección es resumir brevemente el papel del analista en la recuperación de los sistemas.

A veces puede ser sencillo recuperar el sistema, como pulsar una tecla específica o volver a arrancar la computadora. Pero es posible también que se requieran instrucciones de corrección para evitar que vuelva a producirse el fallo generalizado.

El analista debe ponerse en contacto con el servicio de explotación de los sistemas para corregir el problema. Las acciones realizadas por el servicio de explotación suelen consistir en dar fin a la sesión on-line y reinicializar la aplicación y sus programas.

En algunos casos el analista puede tener que recurrir a la administración de datos para recuperar archivos o bases de datos perdidos o deteriorados. Puede también tener que recurrir a la administración de redes para resolver un problema de redes locales o extendidas, también puede recurrir a los técnicos o los representantes de los vendedores para arreglar un problema de hardware.

5.4 MEJORAS DEL SISTEMA

La mayor parte del mantenimiento de adaptaciones se hace como respuesta a la aparición de nuevos problemas de empresa, nuevas necesidades de información o nuevas ideas de mejoras. Por naturaleza, actúa como reacción, cuando algo se estropea o cuando los usuarios transmiten una solicitud. Estas actividades reciben el nombre de mejoras del sistema.

El objetivo de las mejoras al sistema es modificar o ampliar el sistema de aplicaciones como respuestas a las necesidades cambiantes de empresa. Este objetivo puede relacionarse con los bloques elementales de los sistemas de información del modo siguiente.

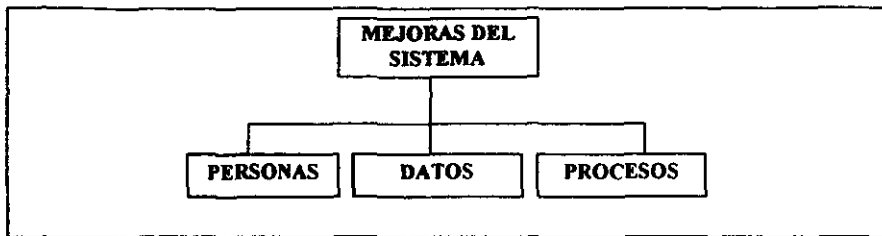


Figura 5.4 Diagramas de Mejoras del Sistema

PERSONAS: En su mayoría, las mejoras a los sistemas son propuestas por los usuarios de los sistemas, si bien los analistas, diseñadores y constructores de sistemas también pueden detectar posibles problemas técnicos relativos al rendimiento, la seguridad y los controles internos.

DATOS: Muchas mejoras de los sistemas son demandados de nueva información que puede derivarse de datos almacenados existentes. Algunas mejoras de datos pueden requerir la ampliación del almacenamiento de datos.

PROCESOS: En su mayoría, las mejoras a los sistemas requieren la modificación de programas existentes o la creación de nuevos programas para ampliar al ámbito general del sistema de aplicaciones.

TECNOLOGIA: Las mejoras a los sistemas se basan en la tecnología.

Los equipos de los sistemas de información se resisten cada vez más a esperar a que un sistema falle. En su lugar, optan por analizar sus bibliotecas de programas para determinar que aplicaciones y programas son los más costosos de mantener o en los cuales el mantenimiento resulta más difícil. Estos sistemas pueden ser adaptados para reducir los costos de mantenimiento.

5.5 ASISTENCIA AL USUARIO FINAL

Otra actividad permanente y relativamente rutinaria en el soporte de sistemas es la asistencia al usuario final, independientemente de como haya sido la formación de usuarios o de la calidad de la documentación, los usuarios requerirán asistencia adicional, por lo general, a disposición de los usuarios para ofrecerles ayuda en el uso diario de aplicaciones específicas.

Las tareas más características comprenden:

Observación rutinaria del uso del sistema.

Realización de estudios o reuniones para conocer el grado de satisfacción del usuario.

Cambiar los procedimientos de empresa para que sean más claros.

Ofrecer formación adicional.

Anotar en el diccionario las ideas y las solicitudes sobre posibles mejoras.

CONCLUSIONES

Este trabajo fue realizado con la intención de resolver un problema cotidiano para el área de inventarios, empleando conocimientos adquiridos durante la carrera de Ingeniería en Computación, además de ponerlos en práctica y emplear tecnología de vanguardia en el desarrollo de sistemas.

Para poder recolectar los requisitos de información referente a la administración de los artículos y activo fijo dentro de los almacenes y la propia Dirección General de Obras y Servicios Generales, se realizaron con gran frecuencia, entrevistas con personal que labora cotidianamente en estas tareas (ingenieros, arquitectos, licenciados, supervisores, empleados). Además se hizo la consulta de diferentes documentos generados por las personas antes mencionadas y se examinó un sistema que en ese tiempo utilizaban.

El desarrollo de SICIPA (Sistema Integral de Control de Inventarios Para Almacenes) ha permitido a las autoras del mismo obtener una serie de experiencias tanto a nivel personal como profesional, mismas que generan las siguientes conclusiones:

La puesta en marcha de SICIPA mejoró significativamente los tiempos requeridos para la captura, proceso y entrega de resultados a los usuarios.

La operación sistematizada de los datos favoreció la confiabilidad de la información reportada, disminuyendo considerablemente los errores de cálculo y comparación debidos a fatiga personal y al proceso de operaciones en forma manual.

Técnicamente, el desarrollo del Sistema permitió comprobar que el uso de tecnología de vanguardia como son las herramientas CASE (en nuestro caso Genexus) en conjunto con un lenguaje de cuarta generación (Visual Basic), en la solución de este problema en específico, se logró lo siguiente:

- * Un producto acorde a las necesidades del usuario
- * Reducción de costos y tiempos de desarrollo.
- * Las pruebas y depuración del sistema se hacen junto con el usuario.
- * El mantenimiento es más sencillo de realizar.
- * El usuario tiene una participación interactiva durante el desarrollo.

El banco de datos que va generando el sistema es lo suficientemente flexible y dinámico para poder tener la información al día y para ampliarla de acuerdo a las necesidades que aparezcan.

La recuperación de la totalidad o parte de la información se realiza en forma expedita, mediante listados impresos o directamente en el monitor del equipo, presentada a detalle o resumida de acuerdo a los requerimientos especificados por el usuario.

El diseño modular con el que fue creado, permite la incorporación gradual de nuevas aplicaciones que habiliten operaciones relacionadas, lo cual asegura que el rango de operación del sistema no sólo permanecerá vigente, sino que se tendrá la capacidad de ampliar progresivamente sus alcances.

Por lo anterior, se concluye que el Sistema cumple con los objetivos planteados al inicio del desarrollo, cubriendo los requerimientos de información de los usuarios y ofreciendo la sistematización adecuada para la reducción del tiempo de respuesta en la entrega de resultados con un alto grado de confiabilidad en la información.

En el aspecto de desarrollo profesional, el Sistema nos permitió obtener experiencia en el manejo de conocimientos adquiridos durante la formación educativa en el área de Ingeniería en Computación aplicados a la solución de problemas reales con requerimientos específicos.

Además nos permitió constatar que la formación profesional en Ingeniería, amplía la perspectiva del individuo hacia otras áreas de conocimiento que no están directamente relacionadas con ella, pero que, sin embargo, requieren de Sistemas como herramienta para la automatización y agilización de funciones y procesos.

BIBLIOGRAFIA

- **Introducción a los Sistemas de Bases de Datos**
Date, C. J. 1988
- **Relation Database: *Seleted Writings***
Date, C.J. 1989
- **An Introduction to Database System**
The System Programming Series
Date, C. J. 1988
- **Ingeniería de software**
Richard Fairley
- **Fundamentos de los sistemas de bases de datos**
S.M. Deen
- **Organización de las Bases de Datos**
James Martin
- **Metodologías de desarrollo**
Antonio López
- **Ingeniería de software**
Roger S. Pressman
- **Metodologías de Desarrollo**
López Fuensaila, Antonio 1991
- **Análisis Estructurado Moderno**
Yourdon, Edward 1993

- **Análisis y Diseño de Sistemas**
Kendall y Kendall

- **Sistemas de Bases de Datos**
Alice Y., H. Tsal

- **Diccionario de Computación**
Alan Freedman

- **Manual de GENEXUS 6.0**
Diseño de Aplicaciones
Artech Consultores 1998

- **Automatic Generation and Maintenance of Programs**
- **Automatic Database Desing, Creation and Meintenance**
<http://www.artech.com.uy>

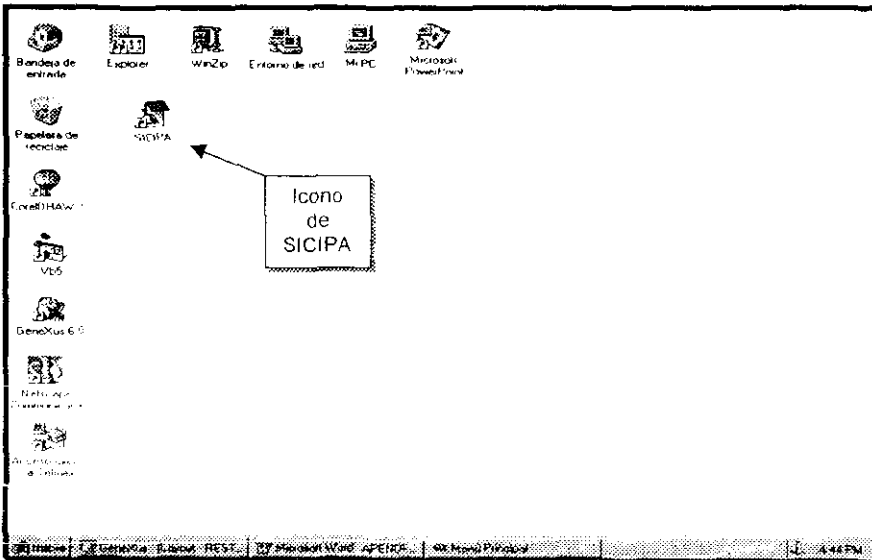
- **Tips & Techniques**
 - * Visual Basic
 - * Develpment Environment<http://www.genexus.com/ctarea.htm>

APENDICE A

MANUAL DE USUARIO

REGISTRO DE USUARIOS

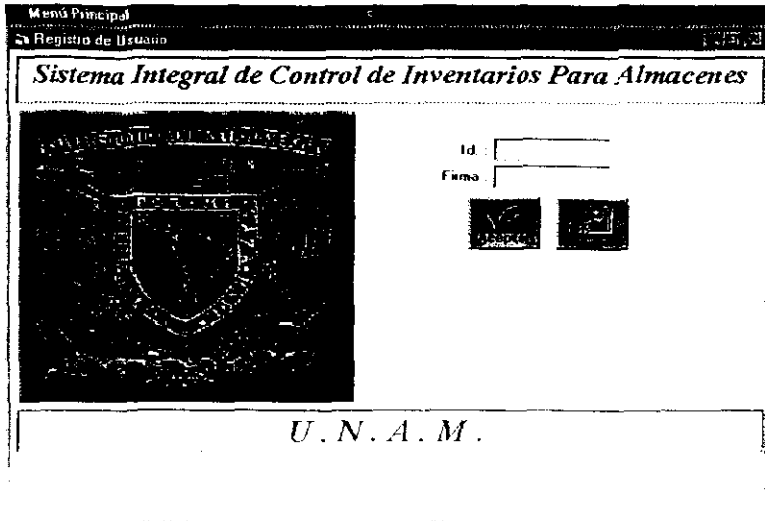
Localice el icono de SICIPA en el escritorio de su computadora, de doble clic sobre ese icono .



Escritorio de su Computadora

La siguiente pantalla aparecerá y deberá hacer lo siguiente

Escriba su identificación y password en el campo correspondiente activando cada campo con el ratón o presionando la tecla [TAB]. Una vez digitado su clave de acceso de clic en el botón de confirmación de clave (este botón lo localizará por la figura de una paloma de Aceptar) con el ratón o bien dando [ENTER].

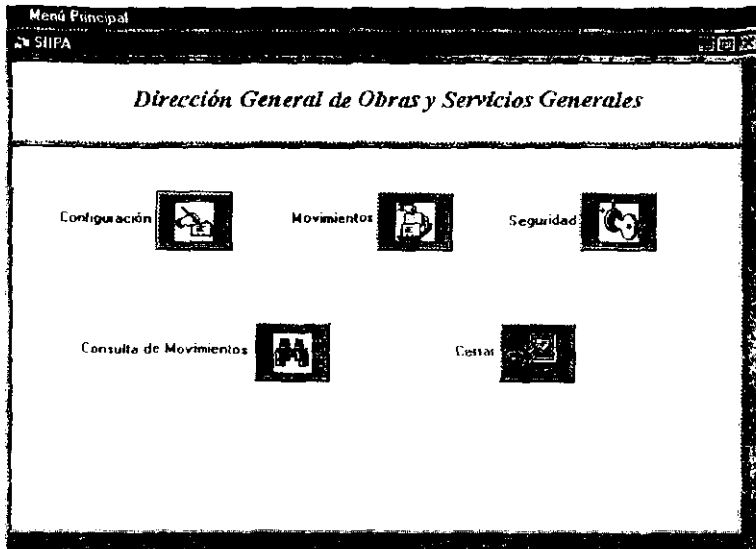


Registro de Usuario

Si su clave es la correcta SICIPA dará acceso al Menú principal del sistema

MENÚ PRINCIPAL

En este menú se visualizarán los diferentes módulos que conforma SICIPA de los cuales usted tendrá acceso si su clave de usuario que le asignó el administrador del sistema, se lo permite



Menú Principal

Como se puede observar, SIIPA (Sistema Integral de Inventarios para Almacenes) está conformado por los módulos de Configuración, Movimientos, Seguridad y Consultas, los cuales se describen a continuación:

CONFIGURACION

En este menú, usted podrá visualizar varios botones a los que tendrá acceso sólo el administrador del sistema. Aquí es donde el administrador podrá configurar la base de datos con todos los parámetros sobre los cuales se trabajará en el almacén, tales como: Artículos, Almacenes, Activo Fijo, Proveedores, etc., así como los datos de los empleados y secciones de los almacenes correspondientes.

MOVIMIENTOS

Este módulo se constituye de dos submódulos más, movimientos en almacenes y movimientos de activo fijo, en los cuales se controla todos los procesos administrativos dentro del almacén o la propia Dirección General de Obras y Servicios Generales, tales como Ordenes de Trabajo, Movimientos de E/S/D (Entrada/Salida/Devolución), Movimientos de A/B (Altas/Bajas), Resguardos, Reportes de los Movimientos, etc.

SEGURIDAD

En este módulo se dan de alta los diferentes usuarios al sistema, se otorgan permisos de acceso a los diferentes módulos de acuerdo a las necesidades de usuario, se puede modificar, así, también, el grupo de permisos de un usuario, se puede eliminar un usuario, etc.

CONSULTAS

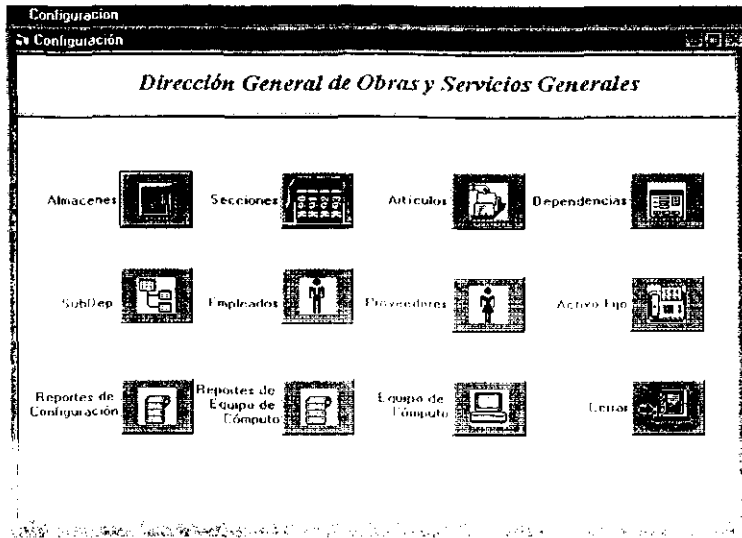
Este módulo presenta un menú en el que se podrá "consultar" los diferentes movimientos realizados en SICIPA en un período de tiempo dado así como la persona quien realizó dicho movimiento. Al igual que en el módulo de seguridad, solo tendrá acceso el administrador del sistema o personal autorizado.

REPORTES

Este módulo está compuesto por otros dos submódulos más, Reportes Generales y Reportes Detallados, en el primero usted podrá visualizar diferentes reportes, a nivel general, en pantalla o impresos, de Artículos, Almacenes, Secciones, Proveedores, Activo Fijo, etc. y en los Reportes Detallados se pueden elegir diferentes reportes de acuerdo a ciertos criterios de impresión como por ejemplo, Reporte de Artículos con Existencia Mínima, Movimientos de E/S/D (Entrada/Salida/Devolucion) por Fecha, Resguardos Por Empleado, etc.

MENU DE CONFIGURACION

En la pantalla del Menú Principal haga clic con el ratón en el botón de configuración. Si usted tiene acceso al menú de configuración podrá visualizar el siguiente menú en el cual se visualizan las siguientes opciones y le permitirán llevar el control y orden en la administración de los recursos.

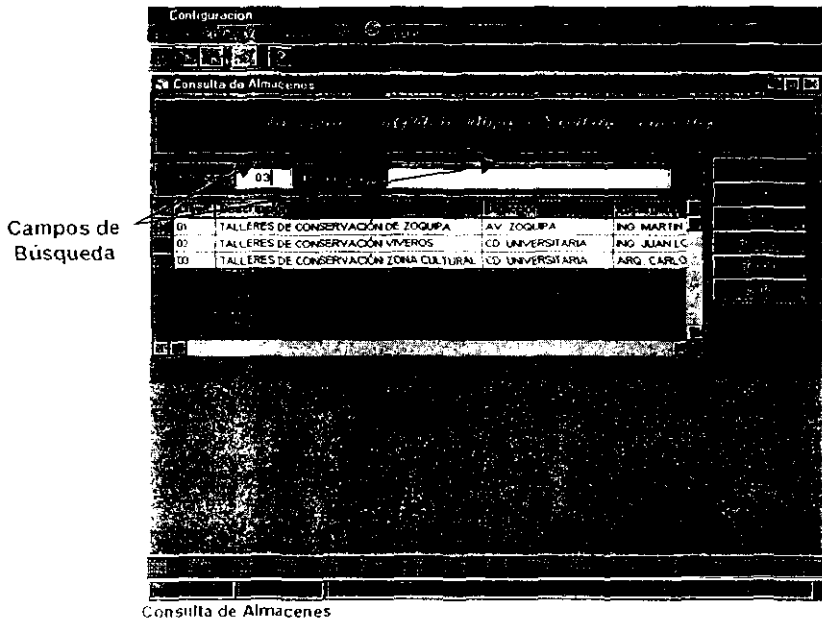


Empecemos a configurar los datos de los almacenes, como son la clave que identificará a un almacén determinado, nombre, dirección y encargado. Para ello dé clic con el mouse en el botón de almacenes y la siguiente pantalla de consulta se activará

Almacenes

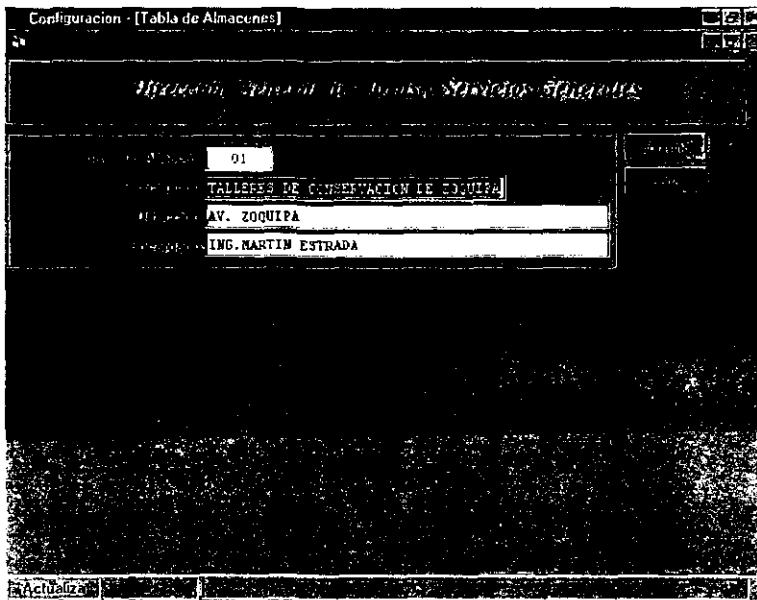
Una vez activado este botón, podrá registrar oprimiendo el botón de “ALTA” que se muestra en la pantalla, lo correspondiente a los diferentes almacenes y sus respectivos responsables o cualquier otra información relevante para la Dirección General de Obras y Servicios Generales.

Para dar de baja de la base de datos un registro determinado, primero selecciónelo y luego oprima el botón de “BAJAS”, es así como lo eliminará de la base de datos.



Si usted busca los datos de un almacén determinado y en la parte inferior tiene una lista muy grande de los almacenes que tiene registrados, solo tiene que digitar la clave del almacén que busca en el campo correspondiente y oprimir el botón de renovar, este le hará la búsqueda que necesita y aparecerá en pantalla el registro buscado. Si usted no se sabe la clave, pero si sabe el nombre del almacén, este podrá digitalizarlo en el siguiente campo, oprimir el botón de renovar y el sistema le dará el o los registros que tiene la base de datos con el nombre digitado.

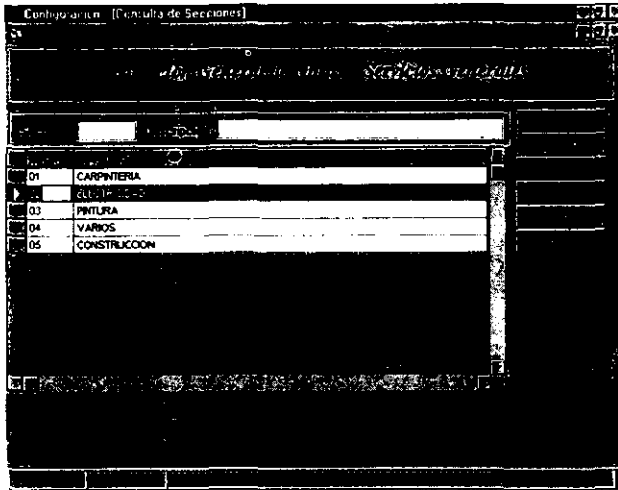
Ahora que si quiere modificar algún dato de un registro determinado, primero seleccione el registro a modificar y luego oprima el botón de "CAMBIOS", este lo llevará a la pantalla de captura donde podrá hacer los cambios que requiere a excepción de la clave que identifica el almacén, ya que esta es única.



Pantalla de Captura de Almacenes

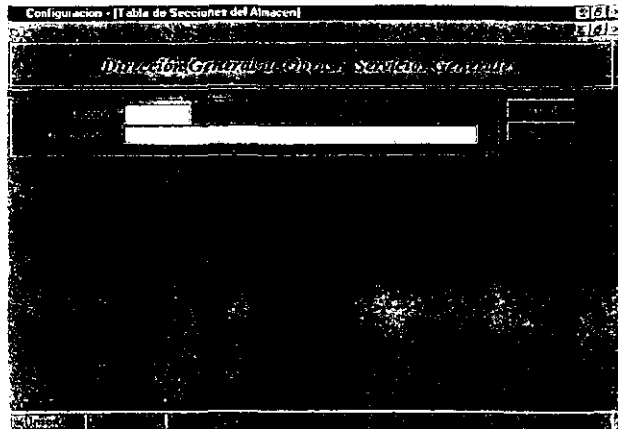
Secciones

Esta opción permite definir las diferentes secciones en que se divide un almacén, como son: carpintería, pintura, electricidad, etc. Para dar de alta un registro, solo tiene que oprimir el botón de secciones del menú de configuración y lo llevará a la siguiente pantalla:



Pantalla de Consulta de Secciones de Almacenes

Para dar de alta un registro sólo oprima el botón de "ALTA", esto lo llevará a la siguiente pantalla en la que podrá dar de alta todos los registros que desee.

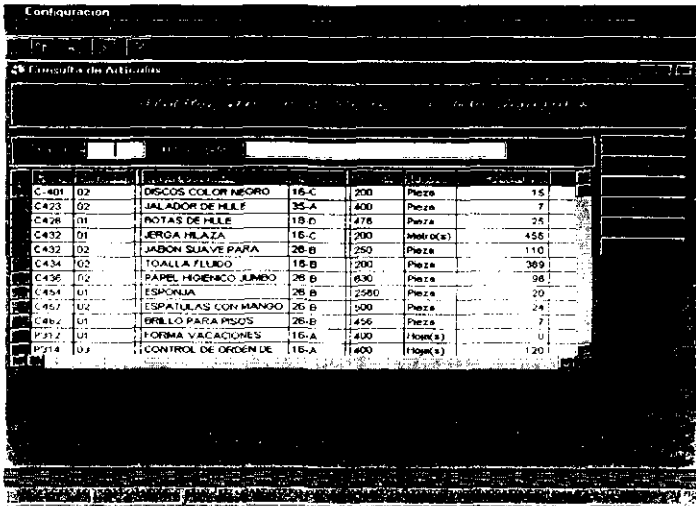


Pantalla de Captura de Secciones de Almacenes

Si requiere eliminar un registro, solo selecciónelo de la lista y oprima el botón de "BAJAS". Si quiere modificar un dato de un registro dado, selecciónelo y oprima el botón de "CAMBIOS"

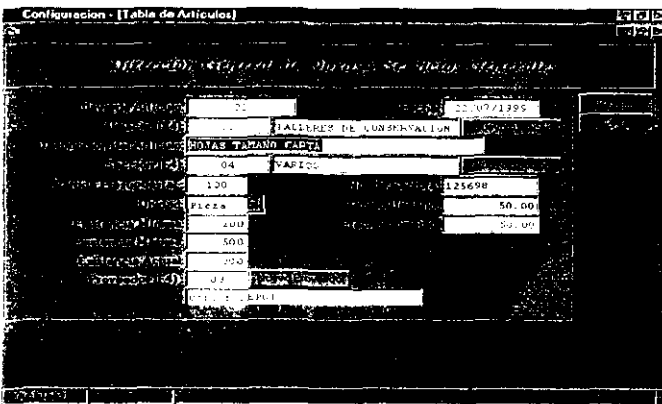
Artículos

Para dar de alta un artículo o conjunto de artículos, oprima el botón de Artículos del menú de configuración y la siguiente pantalla se activará:



Pantalla de Consulta de Artículos

Para dar de alta las características propias de un artículo determinado, solo oprima el botón de "ALTAS", y lo llevará a la siguiente pantalla de captura.



Pantalla de Captura de Artículos

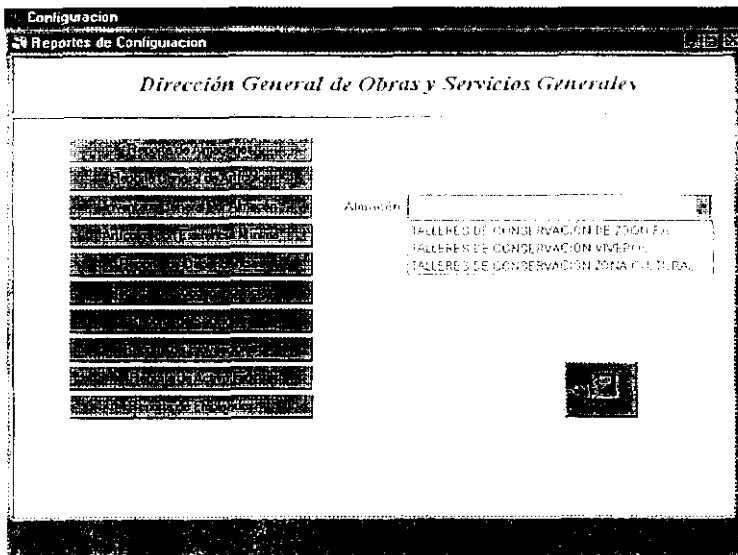
Para modificar algún registro, regrese a la pantalla de consulta de artículos, selecciónelo y oprima el botón de "CAMBIOS", éste lo llevará a la pantalla de captura y ahí se podrán modificar los campos deseados. Ahora que si quiere eliminar uno o varios registros, sólo tiene que seleccionarlos y oprimir el botón de "BAJAS", de esta manera los registros seleccionados se eliminarán de la base de datos.

Si usted está en la pantalla de consulta de artículos y quiere uno en especial, para buscarlo sólo tiene que digitar su clave de identificación en el campo correspondiente y oprimir el botón de "RENOVAR" de esta manera usted podrá visualizar en la lista de artículos, el buscado. Pero si usted no se sabe la clave sólo tiene que teclear el nombre del artículo y oprimir nuevamente el botón de "RENOVAR" para encontrar el artículo deseado.

Los pasos descritos anteriormente se realizan para todos y cada uno de los elementos restantes, *del Menú de Configuración*, que son: Proveedores, Empleados, Dependencias, Subdependencias y Activo Fijo.

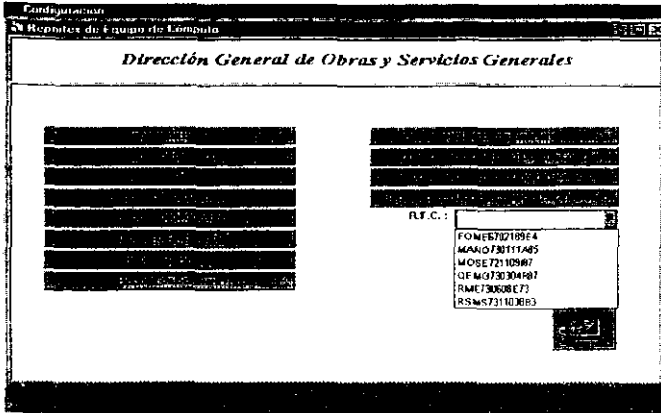
Como puede observar, existen otros tres botones más que son Reportes, Reportes de Equipo de Cómputo y Equipo de Cómputo.

Si oprime el botón de "REPORTES" este lo llevara a un menú de selección en el que podrá elegir el reporte que en su momento necesite, como lo muestra la siguiente pantalla:



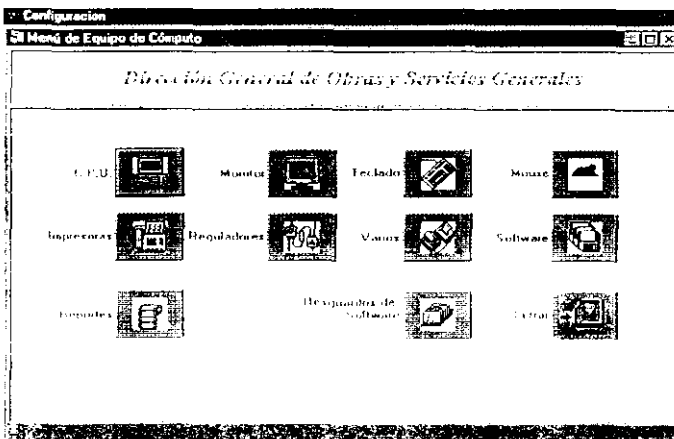
Pantalla de Selección de Reportes del Menú de Configuración

Si oprime el botón de “REPORTES DE EQUIPO DE CÓMPUTO” éste lo llevará a la siguiente pantalla, en la que tendrá acceso a los reportes que necesite de Equipo de Cómputo.



Pantalla de Selección de Reportes de Equipo de Cómputo

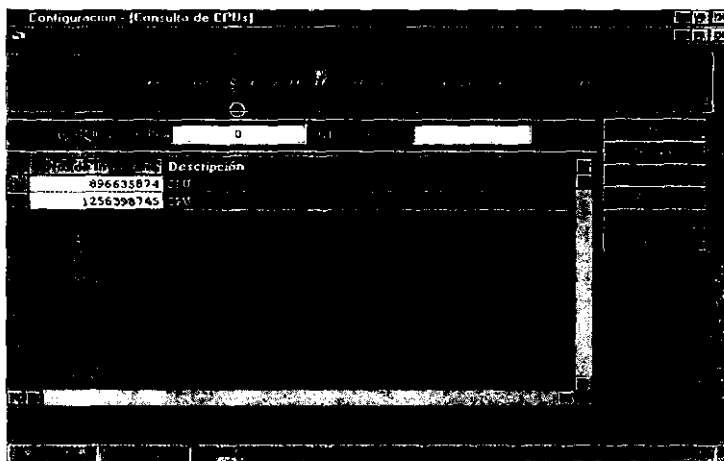
Como se muestra en la siguiente pantalla



Menú de Equipo de Cómputo

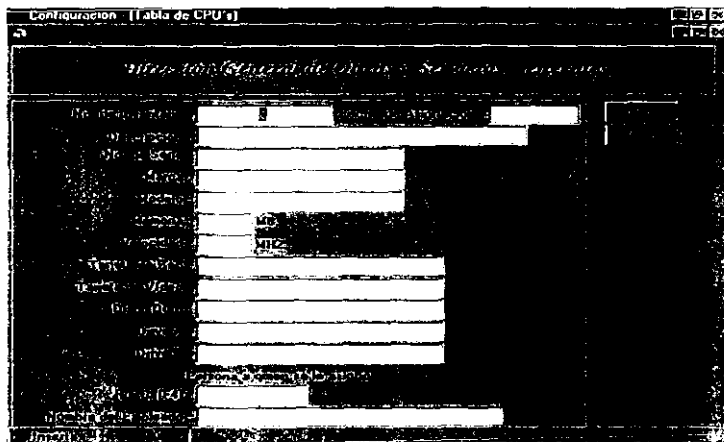
Como puede observar se presentan varios botones con los cuales usted podrá acceder para registrar todo el equipo de cómputo del que necesite llevar una administración adecuada.

El botón de "CPU", lo llevará a la Pantalla de Consulta de CPU's y en ésta usted podrá dar de alta, baja o modificar un registro que requiera en su momento.



Pantalla de Consulta de CPU's

Para buscar un CPU determinado, solo digite el No. De inventario del mismo en el campo correspondiente y oprima el botón de "RENOVAR", si quiere dar de alta un CPU, oprima el botón de "ALTAS" y la siguiente pantalla de captura visualizara y en ella podrá registrar todos los CPU's que requiera en su momento.

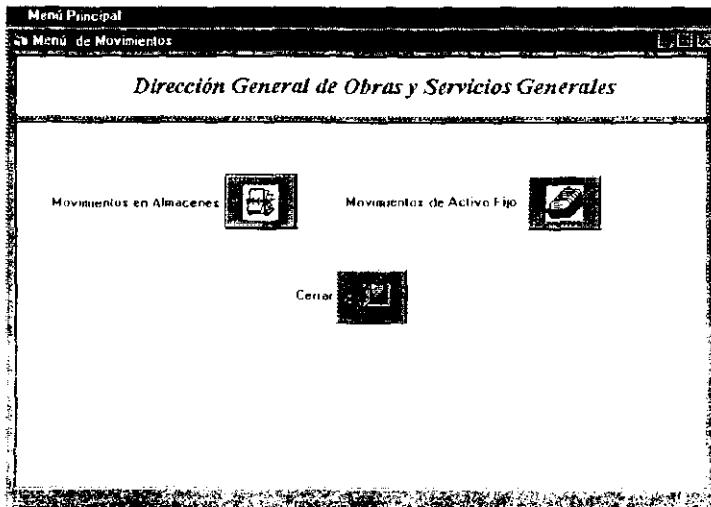


Pantalla de Captura de CPU's

Como puede observar, el menú de Equipo de Cómputo funciona de la misma manera que el menú de Configuración. Aquí se podrán dar de alta, modificar o eliminar las diferentes características propias de algún componente seleccionado, así como el responsable de cada uno.

MENU DE MOVIMIENTOS

Si regresamos al Menú Principal, usted podrá activar el botón de Movimientos se visualizará la siguiente pantalla:

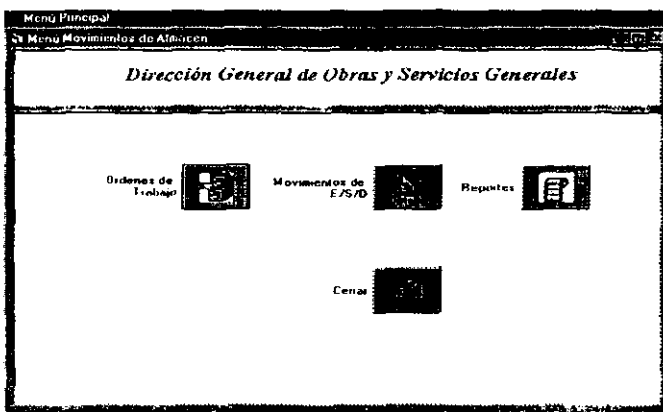


Menú de Movimientos

En este menú, tendrá dos opciones y de acuerdo a su clave de usuario podrá acceder a cada uno de ellos o sólo a uno, estos botones son: Movimientos en Almacenes y Movimientos de Activo Fijo.

MENU DE MOVIMIENTOS EN ALMACENES

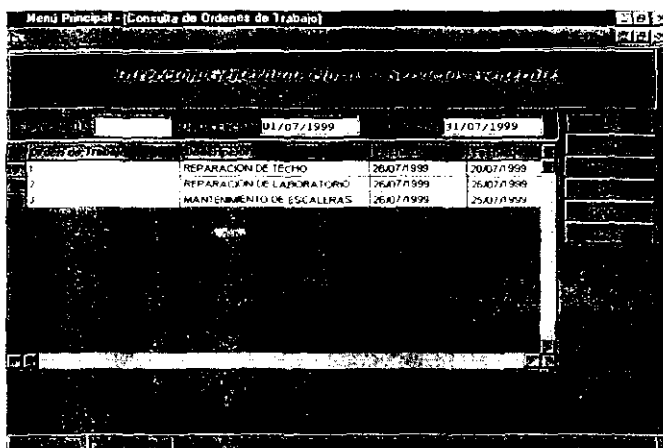
Al oprimir este botón se tendrá acceso a la siguiente pantalla:



Menú de Movimientos en Almacenes

Una vez en este menú usted podrá elegir ya sea registrar una orden de trabajo o un movimiento de Entrada/Salida/Devolucion de uno o varios Articulos o bien consultar la pantalla o impreso de los reportes que genere este módulo.

Si necesita registrar una o varias ordenes de trabajo, solo oprima en el botón correspondiente y visualizará la siguiente pantalla.



Pantalla de Consulta de Ordenes de Trabajo

La pantalla anterior es de consulta de Ordenes de Trabajo, si quiere dar de alta una orden, oprima el botón de "ALTAS" y lo llevará a la siguiente pantalla:

Menú Principal

Tabla de Ordenes de Trabajo

122 04/11/1999

MANTENIMIENTO DE PISOS 15/10/1999

FCA FACULTA DE CONTADURIA Y

SR. ARDELIO LOPEZ JUAREZ

MAN0730111A85 MARTINEZ ALARCON NORMA

C423	JALADOR DE HULE	02	35-A	3	Pieza	10.83	32.49
C428	BOTAS DE HULE	01	18 D	2	Pieza	78.77	157.54

190.03

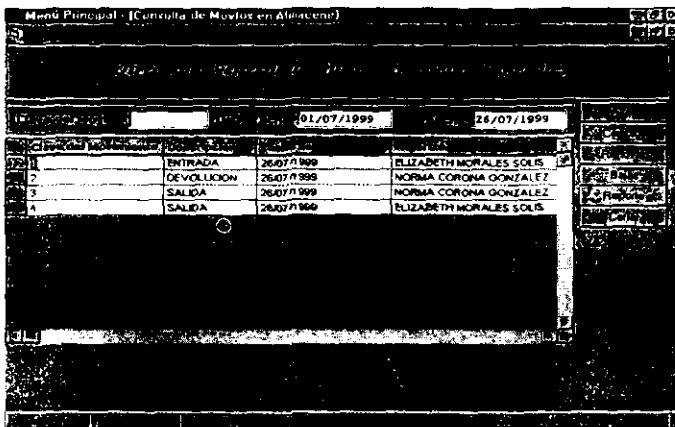
Pantalla de Captura de Ordenes de Trabajo

En esta pantalla usted podrá capturar todos los datos necesarios de un Orden de Trabajo. Al terminar de capturar una Orden de Trabajo, sólo oprima "Confirmar" y el registro se adicionará a la base de datos.

Ahora si quiere modificar un dato de un registro dado, en la pantalla de Consulta de Ordenes de Trabajo, selecciónelo y oprima el botón de "CAMBIOS" y lo llevará a la pantalla anterior, en la que podrá hacer los cambios que necesite.

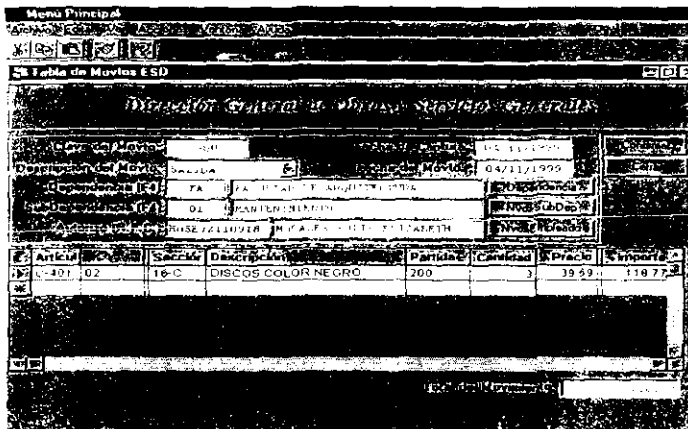
Si necesita eliminar un registro de la base de datos, sólo elijalo de la pantalla de Consulta y oprima el botón de "BAJA" de esta manera se dará de baja el registro seleccionado de la base de datos.

Para registrar un Movimiento de Entrada/Salida/Devolución haga clic con el ratón en el icono correspondiente y aparecerá la pantalla de Consulta de Movimientos de Entrada/Salida/Devolución en la cual encontrará la lista general de los movimientos registrados previamente.



Pantalla de Consulta de Movimientos de Entrada/Salida/Devolución.

Si desea dar de alta un nuevo Movimiento de Entrada/Salida/Devolución, active el botón de "ALTA" que lo llevará a la pantalla de captura de la misma.



Pantalla de Captura de Movimientos de Entrada/Salida/Devolución

Si desea modificar algún dato de un Movimiento de Entrada/Salida/Devolución, sólo tiene que seleccionarlo de la Pantalla de Consulta de Movimientos de Entrada/Salida/Devolución y oprimir el botón de "CAMBIOS" y lo llevará a la pantalla anterior en la cual podrá modificar los campos que se requieran

Ahora si desea dar de baja algún registro, sólo selecciónelo de la pantalla de Consulta y oprima el botón de "BAJA" y de esta manera eliminará el registro seleccionado.

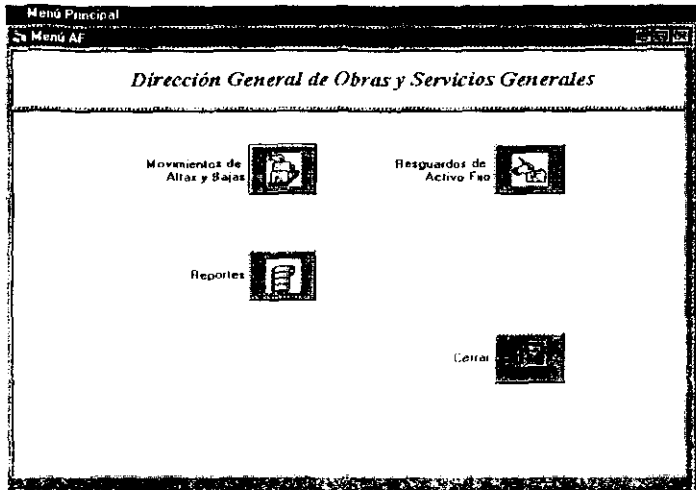
Si requiere algún tipo de reporte ya sea general o particular solo tiene que regresar al menú de Movimientos en Almacenes y oprimir el botón de reportes, usted podrá elegir los reportes que desee y visualizarlos en pantalla o mandarlos imprimir. La pantalla es la siguiente:

The screenshot shows a software interface titled "Menú Principal" and "Selección de Reportes". The main heading is "Dirección General de Obras y Servicios Generales". On the left, there is a vertical list of report options, with "Movimientos de Entrada/Salida/Devolución" selected. On the right, there are several search filters: "De la Fecha" and "A la Fecha" (date range), "Dependencia" (dependency), "Clave de la Orden de Trabajo" (work order key), "Almacén" (warehouse), another "De la Fecha" and "A la Fecha" (date range), "Tipo de Movimiento" (movement type), and "Clave del Movimiento" (movement key). There are also some small icons and a printer icon at the bottom right.

Menú de Selección de Reportes de Movimientos en Almacenes

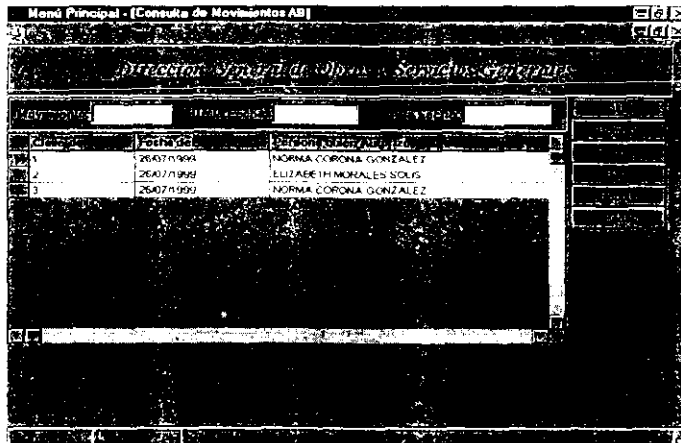
MENU DE MOVIMIENTOS DE ACTIVO FIJO

Vallamos ahora al menú de Movimientos de Activo Fijo:



Menú de Movimientos de Activo Fijo

Para registrar un Movimiento de A/B (Alta/Baja) haga clic con el ratón en el botón correspondiente y aparecerá la pantalla de Consulta de Movimientos de Altas/Bajas en la cual encontrará la lista general de los movimientos registrados previamente.



Pantalla de Consulta de Movimientos de Altas/Bajas

Si requiere registrar un nuevo Movimiento de Alta/Baja sólo oprima el botón de "ALTA" y lo llevará a la siguiente pantalla.

The screenshot shows a software interface titled "Menu Principal - (Tabla de Movimientos AB)". It contains a form with the following fields and values:

- 12566
- 26/07/1995
- 10/07/1999
- EMP ACATLAN
- B. Ingresos Pro:
- 01 MADEPAS DEL SUR
- NOPR14087069L BORHA CORONA GONZALEZ

Below the form is a table with the following data:

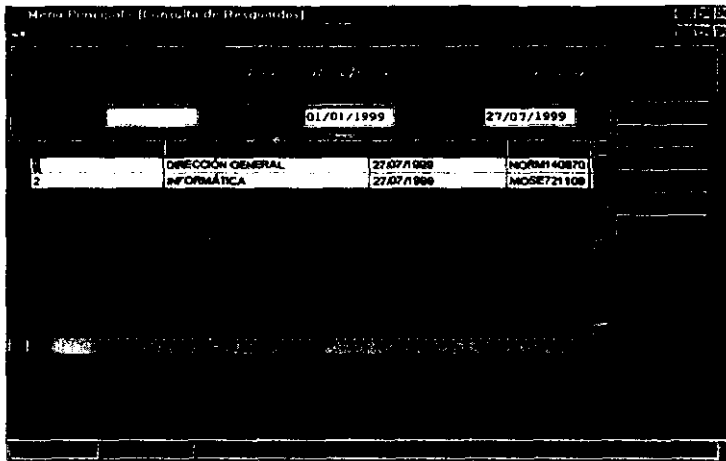
2	12563986	2 TELEFONO CELULAR	1,000.00	2,000.00
---	----------	--------------------	----------	----------

At the bottom right of the screen, there is a summary value: 2,000.00.

Pantalla de Captura de Movimientos de Altas/Bajas

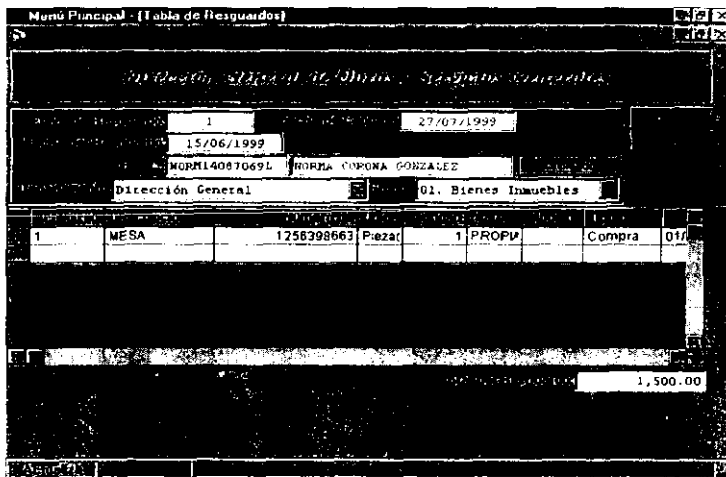
Una vez terminado de capturar todos los campos necesarios de un Movimiento de este tipo, oprima el botón de "Confirmar". Si quiere modificar algún dato determinado de un Movimiento de Alta/Baja, seleccione el registro de la pantalla de Consulta de Movimientos y oprima el botón de "CAMBIOS". Ahora que si desea eliminar un registro de la base de datos, seleccíonelo y oprima el botón de "BAJAS".

Si usted requiere dar de alta algún resguardo de un Activo Fijo solo debe oprimir el botón correspondiente del Menú de Movimientos de Activo Fijo y lo llevará a la siguiente pantalla de Consulta de Resguardos:



Pantalla de Consulta de Resguardos

Oprima el botón de "ALTAS" para registrar un resguardo, este lo llevara a la pantalla de captura correspondiente, como se muestra a continuación:



Pantalla de Captura de Resguardos

Al terminar de capturar todos los datos necesarios para registrar un Resguardo, oprima el botón de "Confirmar". Para modificar uno o varios datos de un Resguardo, selecciónelo de la pantalla de consulta y oprima el botón de "CAMBIOS". Si desea eliminar un registro determinado, selecciónelo de la pantalla de captura y oprima el botón de "BAJAS".

Para consultar un Resguardo determinado, sólo teclee la clave de dicho Resguardo en el campo correspondiente de la pantalla de Consulta, y oprima el botón de "RENOVAR", de esta manera el sistema hará la búsqueda de dicho Resguardo mediante la clave digitada. Otra forma de hacer la búsqueda es tecleando un rango de fechas de cuando fue realizado dicho resguardo y oprimir el botón de "RENOVAR", con esto se hará la búsqueda y podrá visualizar todos los resguardos que se realizaron en ese rango de fechas digitados.

Regresando al Menú de Movimientos de Activo Fijo, se puede observar que también se cuenta con una opción para la consulta en pantalla o impreso, de los diferentes reportes de Movimientos de Activo Fijo que se pueden seleccionar, como se muestra en la siguiente pantalla:

Menú Principal
Selección de Reporte

Dirección General de Obras y Servicios Generales

De la Fecha: A la Fecha:

Tipo de Movimiento Alta/Baja:

Clave del Movimiento:

RFC del Empleado:

De la Fecha:

Clave del Resguardo:

Departamento:

Grupo:

Clave del Activo:

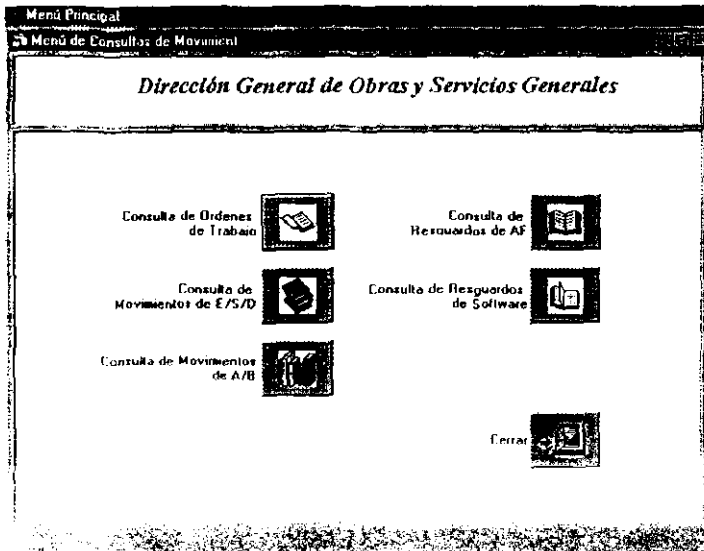
De la Fecha: A la Fecha:

RENOVAR

Pantalla de Selección de Reportes de Movimientos de Activo Fijo

MENU DE CONSULTA DE MOVIMIENTOS

En este menú se muestran cinco opciones, en las que sólo tendrá acceso el administrador del sistema. Estas opciones son Ordenes de Trabajo, Movimientos de Entrada/Salida/Devolución, Movimientos de Alta/Bajas, Resguardos de Activo Fijo y Resguardos de Software. Este sistema fue diseñado para mostrar al usuario todos los movimientos que se han realizado, de acuerdo a un lapso de tiempo determinado, mostrando al administrador del sistema las personas que han realizado dichos movimientos, en modo general y a detalle.



Menú de Consultas de Movimientos

CONSULTA DE ORDENES DE TRABAJO

Si usted quiere consultar quien realizó una o varias ordenes de trabajo durante un periodo de tiempo, sólo oprima el botón 'CONSULTA DE ORDENES DE TRABAJO' este lo llevará a la siguiente pantalla

Menú Principal - [Consulta de: 01]

01/01/1999 27/07/1999

BMS	1	REPARACION DE TECHO	26/07/1999
BMS	2	REPARACION DE LABORATORIO	26/07/1999
BMS	3	MANTENIMIENTO DE ESCALERAS	26/07/1999

Pantalla de Consulta de Movimientos Por Usuario

Como puede observar, en la pantalla anterior se muestra una lista de ordenes de trabajo realizadas en un periodo de tiempo dado asociado al usuario que realizó dicho movimiento, puede seleccionar la orden de trabajo que le interesa y luego oprimir el botón de "DETALLE", éste lo llevará a la siguiente pantalla en la que podrá consultar a detalle el registro que seleccionó

Menú Principal

Archivos Editar Ventas Ordenes Ventas Almacenes

3/8 [F5] [F6] [F7] [F8] [F9] [F10] [F11] [F12]

Tabla de Ordenes de Trabajo

Dirección General de Obras y Servicios Generales

04/11/1999
15/10/1999

MANTENIMIENTO DE PISOS

FEA PAQUETA DE TONDA PETA

OP: AMELIO LOPEZ JUAREZ

ALMACEN: CANO'S DEL TIARU

No	Invento	Descripción	Almacén	Secc	Cantidad	Unidad	Preço	Importe
0423	MALACON DE YV. LI	32	25-A	3.5020	30.03		31.40	
0424	DOTAS DE HUI.F	01	10.11	2.0024	76.771		177.54	

AGUARDA [F1] [F2] [F3] [F4] [F5] [F6] [F7] [F8] [F9] [F10] [F11] [F12]

Pantalla de Consulta a Detalle de Movimientos

Ahora si usted quiere consultar quien realizó determinados Movimientos de Entrada/Salida/Devolución en los almacenes, sólo oprima el botón de "CONSULTA DE MOVIMIENTOS DE ENTRADA/SALIDA/DDEVOLUCION" y lo llevará a la siguiente pantalla:

Menú Principal - [Consulta de Movimientos]

01/01/1999 27/07/1999

	1	ENTRADA	28/07/1999	25.00
EMS	2	DEVOLUCION	28/07/1999	3.00
EMS	3	SALIDA	28/07/1999	
EMS	4	SALIDA	28/07/1999	3.00

Pantalla de Consulta de Movimientos de Entrada/Salida/Devolución

Usted puede seleccionar un movimiento determinado y oprimir el botón de "DETALLE" para chequear todos los elementos que compone el registro seleccionado

Menú Principal

Tabla de Movtos ESD

Movimiento 04/11/1999

Operación de Movto: SALIDA

Fecha de Movto: 04/11/1999

Operación de Movto: SA DEPART DE APUNTICTURA

Operación de Movto: MANEJO DE

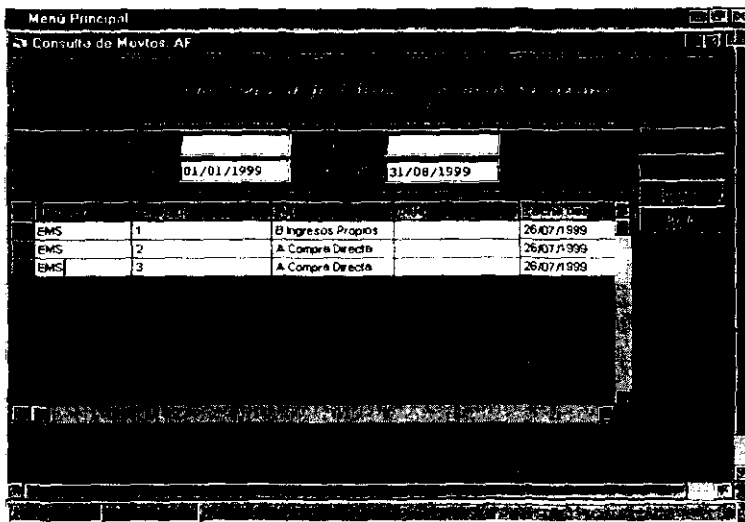
Operación de Movto: RUSIA 11/09/98

Artículo	Sector	Descripción	Parámetros	Cantidad	aprox	IMPORTE
040102	1160	DISCO COLOR NEGRO	1200	3	39.58	118.74

118.74

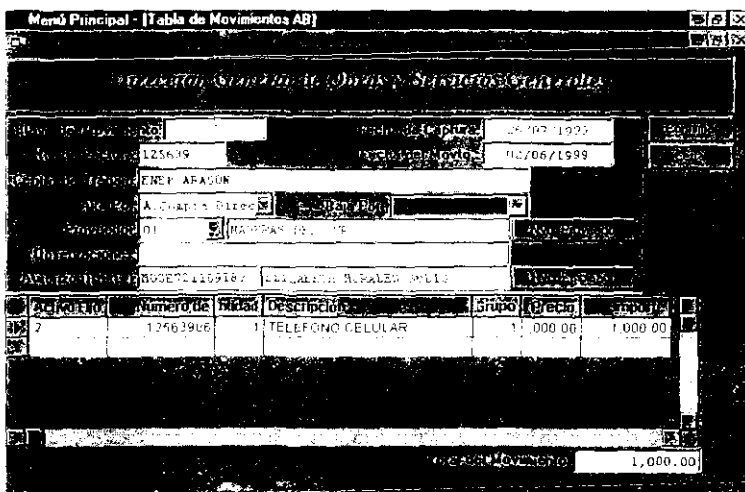
Pantalla de Consulta a Detalle del Movimiento seleccionado en la pantalla Anterior

Para consultar quien realizó determinados Movimientos de Alta/Baja de Activo Fijo, sólo oprima el botón de "CONSULTA DE MOVIMIENTOS DE ALTA/BAJA" y lo llevará a la siguiente pantalla:



Pantalla de Consulta de Movimientos de Alta/Baja por Usuario

Usted puede seleccionar un movimiento determinado y oprimir el botón de "DETALLE" para ver todos los elementos que compone el registro seleccionado.



Pantalla de Consulta a Detalle del Movimiento seleccionado en la pantalla Anterior

Si desea consultar uno o varios Resguardos de Activo Fijo sólo oprima el botón de Movimientos de Resguardos de Activo Fijo, del Menú de Consulta de Movimientos, y se podrá visualizar la siguiente pantalla:

Menú Principal - [Consulta de Resguardos]

USUARIO	FECHA INICIO	FECHA FIN	VALOR
2	01/01/1999	31/06/1999	
2	27/07/1999		1,000.00
2	27/07/1999		1,000.00

Pantalla de Consulta de Movimientos de Resguardos de Activo Fijo Por Usuario

Si quiere consultar a detalle algún resguardo solo selecciónelo de la pantalla anterior y oprima el botón de "DETALLE", y en la siguiente pantalla usted podrá ver todas las características de ese resguardo seleccionado.

Menú Principal - [Tabla de Resguardos]

USUARIO: 2 FECHA: 27/07/1999

FECHA INICIO: 06/05/1999

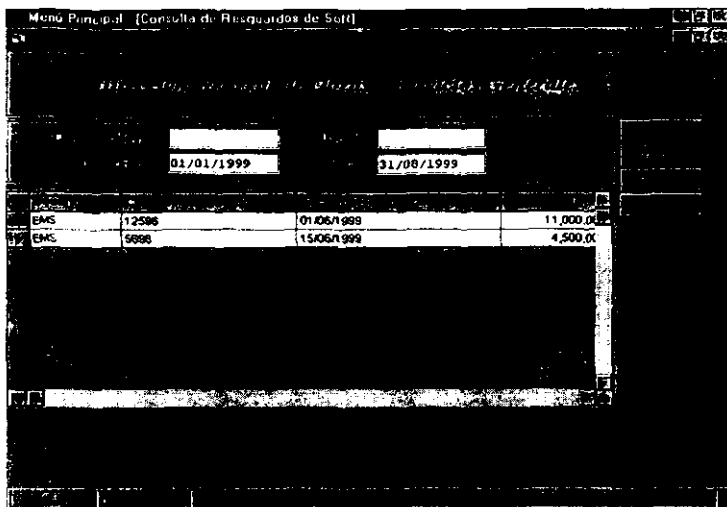
IDENTIFICACION: BOSE721109187 NOMBRE: ELIZABETH MORALES SOLIS

CATEGORIA: Informatica TIPO: 01. Bienes Inmuebles

Activo Fijo	Descripción	Cantidad	Valor	Moneda	Fecha	Operación
2	TELEFONO	1.756.1986	Piezas	1	MOTOR	STARTA: Compra
						TOTAL: 1,000.00

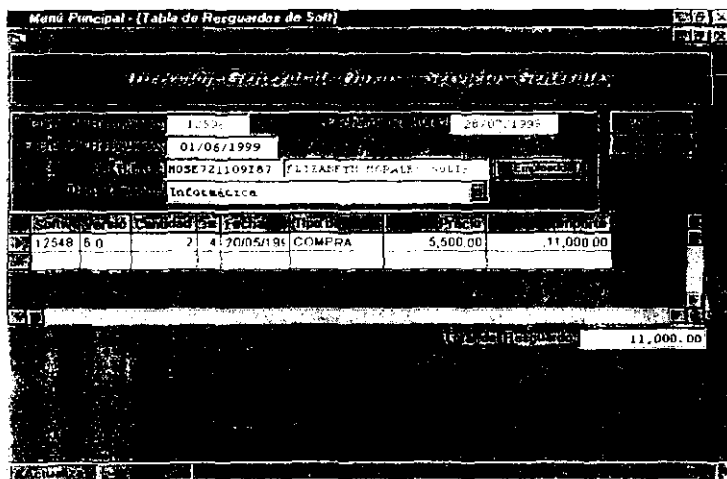
Pantalla de Consulta a Detalle de Movimientos de Resguardos de Activo Fijo

Para consultar quien o quienes realizaron los movimientos de resguardos de software, sólo oprima el botón, del Menú de Consulta de Movimientos, de "CONSULTA DE RESGUARDOS DE SOFTWARE" y podrá visualizar la siguiente pantalla:



Pantalla de Consulta de Resguardos de Software

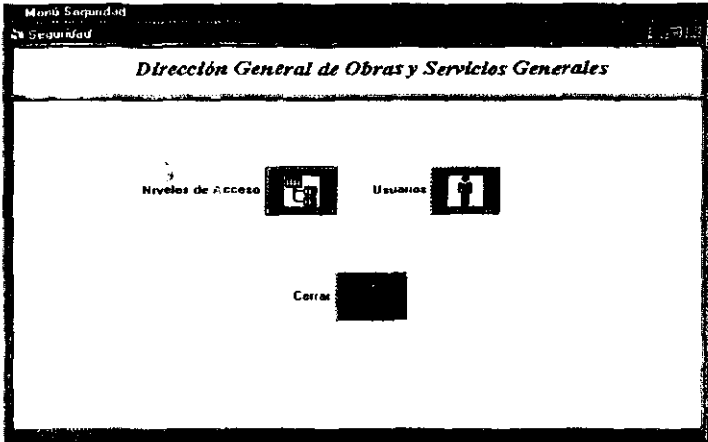
Para consultar a detalle un Resguardo, selecciónelo y oprima el botón de "DETALLE" y podrá ver todas las características que componen el resguardo seleccionado



Pantalla de Captura de Resguardos de software

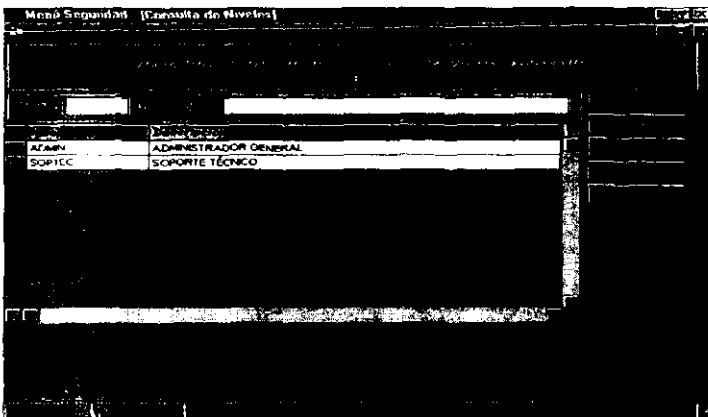
MENU DE SEGURIDAD

En el menú principal oprima el botón de "SEGURIDAD" y el siguiente menú se visualizará:



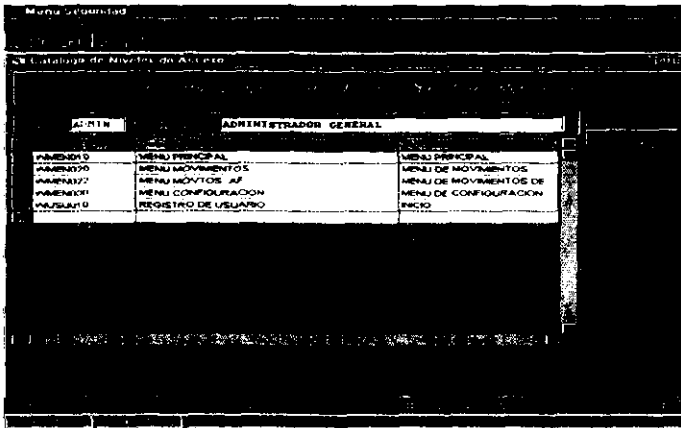
Pantalla de Menú de Seguridad

Para consultar y dar de alta los diferentes niveles de acceso que se tienen en la Dirección General de Obras, para la buena administración del sistema, oprima el botón de "NIVELES" y podrá visualizar la siguiente pantalla:



Pantalla de Consulta de Niveles de Acceso

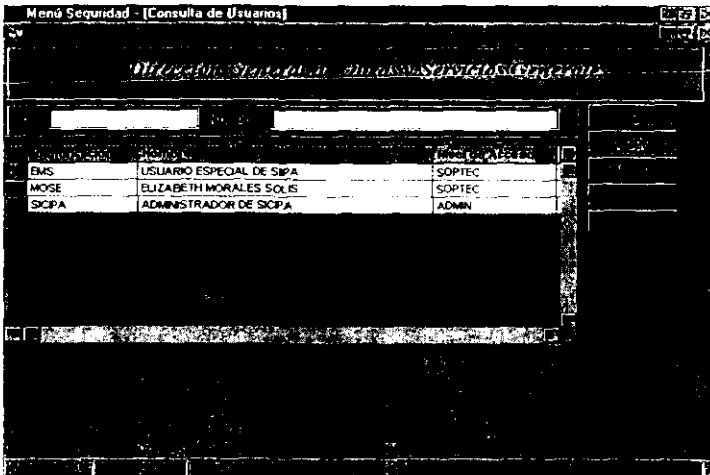
Para dar de alta un nuevo nivel de acceso, oprima el botón de “ALTAS” y capture la clave del nivel, su nombre que lo identificará y oprimiendo la tecla de “F4” se visualizarán los programas a los que tendrá acceso el nivel que se esté dando de alta.



Pantalla de Captura de los Niveles de Acceso

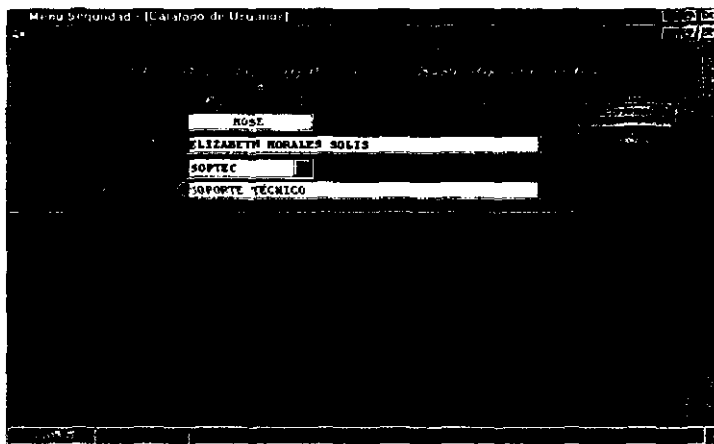
Una vez terminado de capturar el nivel oprima el botón de “CONFIRMA” registre el nivel en la base de datos.

Ahora para dar de alta los usuarios que tendrán acceso al sistema, dentro de acceso que se le asigne, en el Menú de Seguridad, oprima el botón de “USUARIOS” y la siguiente pantalla se podrá visualizar:



Pantalla de Consulta de Usuarios

Para dar de alta un nuevo usuario oprima el botón de "ALTAS" y la siguiente pantalla de captura presentará los campos necesarios para registrar los usuarios que tendrán acceso al sistema.



Pantalla de Captura de Usuarios

Una vez terminado de capturar la clave, el nombre y el nivel de acceso que tendrá el usuario capturado, oprima el botón de "CONFIRMAR", de esta manera la clave del usuario estará registrada en la base de datos.

Ahora si quiere eliminar el usuario, selecciónelo de la pantalla de consulta de usuarios y oprima "BAJAS", para eliminarlo de la base de datos, si quiere modificar algún campo, también selecciónelo de la pantalla de consulta y oprima "CAMBIOS".

Para eliminar un nivel de acceso, primero tiene que eliminar los usuarios que tienen asignado ese nivel y después en la pantalla de consulta de niveles de acceso, seleccionar el nivel y oprimir el botón de "BAJAS" de esta manera eliminará el nivel de acceso de la base de datos.

GUIA DE SOLUCION A PROBLEMAS QUE PUEDEN PRESENTARSE EN EL USO DE SICIPA

PROBLEMA	SOLUCION
I) ARRANQUE DE WINDOWS	
<p>1.- Modo a Prueba de Fallos En este modo no es óptimo trabajar son SICIPA. Cuando Windows arranca en dicho modo, aparece la leyenda "Modo a Prueba de Fallos" en las cuatro esquinas de la pantalla de inicio de Windows.</p>	<p>Reinicie el Sistema</p> <ul style="list-style-type: none"> a) Seleccione el botón de Inicio de la barra de Windows. b) Seleccione Apagar el Sistema. c) Haga Click en la opción Reiniciar el sistema. d) Haga Click en Aceptar. <p>Si el problema persiste, comuníquese a su proveedor de hardware.</p>
<p>2.- No hay imagen en el Monitor</p>	<p>Revise el Monitor</p> <ul style="list-style-type: none"> a) Revise que el monitor se encuentre encendido b) Revise que o haya conexiones sueltas (del monitor a la línea de corriente y del monitor al CPU) c) Revise que el CPU esté encendido d) Mueva los controles de brillo y contraste que se encuentran en la parte inferior o lateral del monitor <p>Si el problema persiste, comuníquelo a su proveedor de hardware</p>

PROBLEMA	SOLUCION
II. INICIO DE SICIPA	
<p>1.- Clave Incorrecta o Sin Autorización</p> <p>Quando este mensaje es desplegado en la pantalla de inicio de SICIPA, significa que su clave no se tecleó bien.</p>	<p>a) Haga Click con el ratón en el campo de "Id."</p> <p>b) Borre todos los caracteres del campo.</p> <p>c) Haga Click con el ratón en el campo de "Firma."</p> <p>d) Borre todos los caracteres del campo.</p> <p>e) Escriba de nuevo su identificación y firma.</p> <p>f) Si el problema persiste, repita los pasos del 1 al 5.</p> <p>Si ya repitió los pasos arriba señalados, varias veces, comuníquese con el administrador del sistema para que verifique su clave o en su defecto le asigne una nueva.</p>
<p>2.- Run Time Error ...</p> <p>Este tipo de mensaje se despliega cuando existe un problema de ejecución de SICIPA.</p>	<p>a) Revise que el sistema SICIPA no se encuentre abierto más de una vez. Esto se verifica asegurándose que en la parte inferior de la pantalla no exista más de un pequeño rectángulo con el título de SICIPA.</p> <p>b) Si se da lo anterior, haga Click en el rectángulo antes señalado y vaya cerrando todas las ventanas que preceden a cada una de ellas, hasta que haya cerrado totalmente todo el sistema de SICIPA y no encuentre más rectángulos abiertos.</p> <p>c) Haga de nuevo doble Click en el icono de SICIPA.</p> <p>d) Si el problema persiste, Reinicie el Sistema, siguiendo los pasos 1 a 4 en la Sección I de esta tabla e intente iniciar SICIPA nuevamente.</p> <p>Si se sigue presentando el problema, después de haber realizado los pasos anteriores, contacte al administrador del sistema.</p>

PROBLEMA	SOLUCION
<p>3.- No Permite Capturar Nada en algún Campo Determinado.</p>	<p>a) Cierre todas las ventanas, de una en una, ya sea mediante el botón señalado por una cruz ■ o en la barra superior, se encuentra la palabra "Archivo" de un Click allí y elija salir.</p> <p>b) Una vez cerrado todas las ventanas abiertas, vuelva a dar doble Click en el icono de SICIPA.</p> <p>Si el problema persiste comuníquese con el administrador del sistema.</p>
<p>III. PROBLEMAS DE IMPRESION</p>	
<p>1.- En la mayoría de los casos será necesario parar y cancelar la impresión</p>	<p>a) Pulse el botón en su impresora que la ponga en pausa. En la mayoría de las impresoras tienen un botón que dice "Pausa-Online".</p> <p>b) Una vez que se detenga la impresora localice en su pantalla el botón de inicio de Windows, por lo regular aparece en la parte inferior izquierda de su pantalla, haga Click con el ratón en este botón.</p> <p>c) Seleccione la opción "Configuración" (quedará sombreada).</p> <p>d) Arrastre el ratón hacia la derecha y luego hacia abajo, esto para seleccionar la opción de "Impresoras" y haga Click sobre esa opción.</p> <p>e) Seleccione el icono de la impresora que tiene instalada y haga un doble Click sobre ella.</p> <p>f) En la ventana de la impresora selecciones la opción "Documento" y arrastre el ratón hacia abajo para elegir la opción "Cancelar Impresión".</p>

PROBLEMA	SOLUCION
<p>2.- La impresión no se encuentra centrada en la hoja</p>	<ul style="list-style-type: none"> a) Cancele la impresión. b) Solicite en SICIPA, nuevamente la impresión del reporte deseado. c) Asegúrese de que en la ventana de la impresora, en la opción "Propiedades" se encuentre seleccionado el Tamaño del Papel "Carta", de no ser así selecciónela mediante un Click del ratón. d) Asegúrese que el papel se encuentre bien colocado en la impresora y sea del tamaño adecuado. e) Presione de nuevo la opción de "Aceptar" en la ventana de la impresora.
<p>3.- Las hojas se atascaron en la impresora.</p>	<ul style="list-style-type: none"> a) Cancele la impresión. b) Desatore el papel y colóquelo nuevamente en la impresora. c) Solicite en SICIPA, nuevamente la impresión del reporte deseado. d) Verifique a partir de que la hoja comenzó a tener problemas de impresión y solicite en la ventana de la impresora que la impresión se haga desde la página en que comenzó con el problema hasta la última página que se desea imprimir. e) Haga Click en el botón de "Aceptar" en la ventana de la impresora.
<p>4.- Se solicitó la impresión de un reporte y el reporte que se despliega en pantalla no corresponde a lo solicitado.</p>	<ul style="list-style-type: none"> a) Cierre la ventana de Reportes. b) Revise que los criterios de impresión solicitados corresponden a lo que desea imprimir. c) Escriba nuevamente los criterios y asegúrese de que son los que desea en la opción adecuada. d) Solicite la impresión nuevamente.

PROBLEMA	SOLUCION
<p>5.- Se solicita ver un reporte en pantalla antes de mandarlo a imprimir, pero este no aparece después de un tiempo considerable.</p>	<p>a) Espere un lapso de tiempo aproximadamente de 2 a 3 minutos si el reporte solicitado es por ejemplo de inventarios en un lapso de tiempo dado o de articulos con existencia minima.</p> <p>b) En caso de que el reporte siga sin responder en ese lapso de tiempo esperado, comuníquese con el administrador del sistema.</p>

APENDICE B

VOCABULARIO DE TERMINOS

Para el desarrollo del Sistema Integral de Control de Inventarios Para Almacenes requerimos de los siguientes términos:

A

Almacén: Lugar fisico donde se concentran los materiales y herramientas para su control.

Archivo de Bases de Datos: Lugar donde se almacenan los datos obtenidos de la Base de Datos.

Atributos (Campo de Datos): Son las propiedades que definen o caracterizan a una entidad. Al atributo frecuentemente se le llama elemento de datos, campo de datos, campo item de datos o item elemental.

B

Base de Datos: Es una colección de datos relacionados acerca de una empresa con múltiples usos.

Base de Datos de Red: Método de organización de bases de datos que permiten relacionar los datos en forma de red.

Boletínaje: Es el intercambio de información con fines de intercambio de artículos, que se realizan entre almacenes.

C

Captura: La captura es la etapa del proceso informático de la cual se pueden obtener grandes logros en productividad. Es importante capturar con efectividad, con el fin de asegurar la calidad de los datos que entran al sistema. La decisión de capturar precede de la interacción misma del usuario con el sistema. Existen dos tipos de datos a capturar: Los datos que cambian o que varían en cada una de las transacciones y los datos que distinguen de otros artículos, de manera concisa, el artículo particular que se procesa.

Cardinalidad: La cardinalidad de una relación se refiere al número de ocurrencias que pueden estar asociadas entre entidades. Por lo tanto, existen relaciones uno a uno (1 : 1) uno a muchos (1 : N), y muchos a muchos (M : N). Para distinguir la cardinalidad en un diagrama E-R se anota un "1", "N" o "M" sobre las líneas de asociación y a un lado de las entidades. Cardinalidad de una relación es el número de tuplas que la componen.

Consulta: También conocida como Query Es una búsqueda de información específica utilizando código para las llamadas. Es una interrogación a una Base de Datos que permite al usuario contar, sumar y listar registros seleccionados contenidos en ella.

Campo: Es la unidad direccionable más pequeña que puede ser referida por un programa.

D

Diccionario de Datos (Glosario de Datos) : Es un depósito central de información acerca de entidades, los campos de datos que representan a las entidades, las relaciones entre éstas, sus orígenes, significados, usos y formatos de representación.

E

Entidad: Es una persona, lugar, una cosa, un evento o un concepto acerca del cual se registra información. Es un objeto tangible que puede describirse con palabras, código numérico o no numérico.

Existencia Mínima: Es la cantidad tope inferior que debe existir de un artículo para solicitar la adquisición del mismo y se evite que este llegue a cero.

Existencia Máxima: Es la cantidad tope superior que puede existir en un momento dado y evitar que el almacén se sature del mismo, ya que el consumo del artículo no justifica su existencia.

Existencia Actual: Es la cantidad real que existe de determinado artículo en el almacén.

G

Grado: Es el número de atributos o columnas de la relación.

I

Inventario: Lista en que se relacionan y detallan los artículos en existencia a una fecha determinada en cada almacén.

Inventario Físico: Es la relación física de la totalidad de artículos, materiales y herramientas con que cuenta un almacén determinado, comparado con los que deben existir según los registros en las tarjetas del cardex, ordenes de entrada y/o salida, facturas y devoluciones

Integridad de los Datos: Mantenimiento de los datos correctos en la base de datos, durante el tiempo de vida del sistema.

Integridad Referencial: Característica de una base de datos donde no hay registros hijo si no existen los padres correspondientes.

K

Kardex: Conjunto de tarjetas en las cuales se registran los movimientos de entradas y salidas diarias de los artículos en los almacenes.

L

Lenguaje de manejo de Datos: Es el lenguaje especial proporcionado por DBMS para permitir a los usuarios efectuar operaciones de Entrada/Salida en la Base de Datos.

LL

Llave: Una llave es un dato elemental en un registro que se utiliza como criterio de identificación para éste.

Llave Primaria: También es conocida como *Clave Primaria*. Es el campo cuyo contenido puede identificar de manera única a cada registro del archivo. Sólo un campo del registro puede ser distinguido como principal.

Llave Foránea: También conocida como *Clave Secundaria*. Es cualquier campo a excepción de la clave principal puede designarse como clave secundaria y se utilizan como clave de búsqueda en consultas.

Llave Compuesta: Es la combinación de una llave primaria con una llave secundaria, formando así una sola clave.

M

Marbete: Documento que identifica el artículo y/o material.

Metodología: Sistema ordenado de proceder par la obtención de un fin.

Modelo de Datos: Es una colección de conceptos matemáticamente definidos, que ayuda a considerar y expresar los objetos, atributos y relaciones entre objetos y sus correspondientes operaciones, es decir, proveen una descripción conceptual de los datos. Consiste de dos elementos: La notación matemática (expresa los datos y las reacciones) y las operaciones sobre los datos (nos sirve para expresar preguntas y otras manipulaciones sobre los datos).

Modelo Entidad-Relación: Método empleado en el diseño de Bases de Datos basado en el análisis de tres modelos semánticos claves: entidades, relaciones y atributos.

Modelo jerárquico: En este modelo la estructura jerárquica de árbol se construye con nodos y ramas. Un nodo es una colección de atributos de datos que describen a la entidad de ese nodo, el nodo más alto es una estructura jerárquica de árbol que se conoce como raíz.

Modelo Relacional: En este modelo, las entidades y sus relaciones se representan con tablas bidimensionales, cada tabla representa una entidad y está compuesta de renglones y columnas.

Modelo Reticular o Distribuido: Este modelo interconecta las entidades de una empresa en una "red", la notación gráfica de este modelo utiliza nodos y ramas. Un bloque representa una entidad o un tipo de registro, cada tipo de registro está compuesto de cero, uno o más atributos.

N

Nodo: Entidad que ocupa una posición en una estructura de datos.

No. de Orden o Factura: Es el número de la factura o del documento con el cual Ingresas el material al almacén, que por lo regular viene impreso en la esquina superior derecha.

Normalización: Es el proceso de agrupar a los campos de datos en tablas que representan a las entidades y sus relaciones. Es una técnica de Bases de Datos que empieza agrupando todos los atributos en una relación universal, la cual después es descompuesta en relaciones más pequeñas hasta que las relaciones divididas pertenezcan a la tercera forma normal.

O

Operación: (Procedimiento) se define como una serie de acciones ejecutables sin Interrupción y que se ejecutan sin interrupción y bajo las mismas condiciones en cuanto a factores descendentes.

Optimización: Tarea de seleccionar la ruta más eficiente para cumplir con un propósito.

Orden de Trabajo: Documento por medio del cual se solicita la realización de un trabajo y la salida de material necesario para llevarlo a cabo.

Orden de Servicio: Documento por medio del cual se solicita la realización de un servicio.

P

Periodo: Es el lapso de tiempo entre fecha y fecha donde se realizan o se han realizado determinados movimientos entre los almacenes.

(P.U) Precio Unitario: Es el precio de los artículos por una sola unidad en el almacén, en un tiempo determinado, es decir, va variando en precio.

Precio Promedio: Es el precio que se obtiene a partir de promediar un precio ya existente con uno de un artículo dado.

Proceso: Es toda actividad de la empresa con sus entradas y salidas establecidas.

Proveedor: Persona o entidad empresarial que abastece o suministra pedidos de determinados artículos a uno o varios almacenes.

R

Redundancia de Datos: Son datos duplicados o guardados más de una vez.

Registro: Colección de valores tomados por un campo de datos relacionados.

Relación: Es una unión o enlace entre dos conjuntos de datos.

Reporte o (Informe): Colección de objetos y figuras impresas en papel, con número y encabezados de pagina. En un programa de Bases de Datos, la opción de informe se usa típicamente para preparar cualquier tipo de salida impresa, el informe puede ser también una lista selectiva de elementos. Es un informe impreso donde se detallan los movimientos que han tenido los artículos de cada almacén en un período determinado.

Resguardo: Documento en el cual se registran los datos de una persona a la cual se hace responsable de determinado equipo o bien, que tenga en su poder.

S

Sección: Lugar físico dentro del almacén, donde se localiza un cierto tipo de material o artículo en especial.

Sistema: Se le llama así a todos los datos interrelacionados en conjunto, que persiguen un mismo objetivo.

T

Tabla Base: Tabla en un modelo relacional de datos. Se realiza físicamente como un archivo.

Tabla de Consulta: Tabla que contiene el valor de la clave buscada y las direcciones de los registros objetivo.

Tarjeta de Movimiento de Almacén: Documento en el cual se registran los movimientos que tenga un material.

Tupia: Es un renglón de la tabla.

U

Unidad: Es la unidad de medida de los artículos (por Ejemplo: metro, pieza, caja, kilos litros, gramos, etc.).

V

Vale de Salida: Documento en el cual se registran los artículos que salen del almacén para su uso determinado.

Vale de Devolución: Documento en el cual se registra el reingreso los artículos nuevamente al almacén, notificando el estado y el porqué de su devolución.

Validación de datos: La validación (tipo, tamaño o valor del campo) de los datos proporcionados (por un programa del sistema o del usuario), antes de que sean cargados en la Base de Datos.

APENDICE C

En el presente Apéndice, se muestra como ejemplo una parte del Código Fuente de SICIPA (Sistema Integral de Control de Inventarios Para Almacenes). Es código de Visual Basic generado por Genexus. Se presentan algunas de las rutinas necesarias para llamar a los menús que integran el sistema.

```
Dim UserCancel As Integer
Dim Response As Integer
Dim gx_valid%
Dim RfK0gs As Integer
Dim GXInitz As Integer
Dim GXNoExit As Integer
Dim GXToReturn As Integer
Dim Srch0gs As Integer
Dim Rfr0gs As Integer
Dim Gx_execsa As Integer
Dim Gx_time As String
Dim Gx_date As Variant
Dim AV6Passc As String
Dim AV5Flgn As Integer
Dim AV4Nomusuc As String
Dim AV3Razsocc As String
Dim Gx_emsg As String
Dim Gx_err As Integer
Dim SLvMd02 As String
Dim GXEnCnt As Integer
Dim next_to_vld As Integer
Dim In_field As Integer
Dim Gx_cursor As String
Dim GX_ERRD%
Dim curr_fld As Integer
Dim ControlDown As Integer
Dim ShiftDown As Integer
Dim LastKey As Integer
Dim GxTopForm As Form
Dim GxTopCtrl As Control
Dim GxOldIndex As Integer
```



```
Private Sub cVNomusuc_KeyPress(KeyAscii As Integer)
```

```
    Select Case KeyAscii
```

```
        Case Is = 13
```

```
            KeyAscii = 0
```

```
            SendKeys "{TAB}"
```

```
    End Select
```

```
End Sub ' cVNomusuc_KeyPress
```

```
Private Sub cVNomusuc_KeyDown(KeyCode As Integer, Shift As Integer)
```

```
    Select Case KeyCode
```

```
        Case Is = 38
```

```
            KeyCode = 0
```

```
            SendKeys "+{TAB}"
```

```
        Case Is = 40
```

```
            KeyCode = 0
```

```
            SendKeys "{TAB}"
```

```
    End Select
```

```
    If KeyCode = vbKeyF1 Then
```

```
        GXTemp& = WinHelp(Me.hwnd, "APPHLP.HLP", HELP_KEY, "NomUsuC")
```

```
        KeyCode = 0
```

```
    End If
```

```
End Sub ' cVNomusuc_KeyDown
```

```
Private Sub cVToday_KeyPress(KeyAscii As Integer)
```

```
    Select Case KeyAscii
```

```
        Case Is = 13
```

```
            KeyAscii = 0
```

```
            SendKeys "{TAB}"
```

```
    End Select
```

```
End Sub ' cVToday_KeyPress
```

```
Private Sub cVToday_KeyDown(KeyCode As Integer, Shift As Integer)
```

```
    Select Case KeyCode
```

```
        Case Is = 38
```

```
            KeyCode = 0
```

```
            SendKeys "+{TAB}"
```

```
        Case Is = 40
```

```
            KeyCode = 0
```

```
            SendKeys "{TAB}"
```

```
    End Select
```

```
If KeyCode = vbKeyF1 Then
    GXTemp& = WinHelp(Me.hwnd, "APPHELP.HLP", HELP_KEY, "Today")
    KeyCode = 0
End If
End Sub ' cVToday_KeyDown
```

```
Private Sub cVToday_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
    If Button = 2 Then
        cVToday.Enabled = False
        T$ = cVToday.text
        cVToday.text = GXCalendar(T$, Me, cVToday, "dd/mm/yy")
        cVToday.Enabled = True
        Gx_date = GxToDate((cVToday.text), "dd/mm/yy", Gx_date, GX_ERRD%)
        cVToday.SetFocus
    End If
End Sub ' cVToday_MouseDown
```

```
Private Sub cVTime_KeyPress(KeyAscii As Integer)
```

```
    Select Case KeyAscii
        Case Is = 13
            KeyAscii = 0
            SendKeys "{TAB}"
    End Select
End Sub ' cVTime_KeyPress
```

```
Private Sub cVTime_KeyDown(KeyCode As Integer, Shift As Integer)
```

```
    Select Case KeyCode
        Case Is = 38
            KeyCode = 0
            SendKeys "+{TAB}"
        Case Is = 40
            KeyCode = 0
            SendKeys "{TAB}"
    End Select
    If KeyCode = vbKeyF1 Then
        GXTemp& = WinHelp(Me.hwnd, "APPHELP.HLP", HELP_KEY, "Time")
        KeyCode = 0
    End If
End Sub ' cVTime_KeyDown
```

```
Private Sub gxBtn6_Click()
```

```
    Call U11V0V2
    If UserCancel Then Unload Me
End Sub ' 'Configuración'_Click
```

```
Private Sub gxBtn7_Click()
```

```
    Call U12V0V2
    If UserCancel Then Unload Me
End Sub ' 'Movimientos'_Click
```

```
Private Sub gxBtn5_Click()
```

```
    Call U13V0V2
    If UserCancel Then Unload Me
End Sub ' 'Seguridad'_Click
```

```
Private Sub gxBtn16_Click()
```

```
    Call U14V0V2
    If UserCancel Then Unload Me
End Sub ' 'Consulta'_Click
```

```
Private Sub gxBtn4_Click()
```

```
    Unload Me
    If UserCancel Then Unload Me
End Sub ' Cancel_Click
```

```
Sub gxrbnU15V0V0(ValCurrCell As Integer)
```

```
    Call RFV0V2
    If UserCancel Then Unload Me
End Sub ' gxrbnU15V0V0
```

```
Sub SAV0V2()
```

```
    Gx_time = Format$(Time(), "hh:mm:ss")
    cvTime.text = RTrim$(Gx_time)
    Gx_date = GxToDate((Format$(Now, "dd/mm/yy")), "dd/mm/yy", GX_NULDATE,
GX_ERRD%)
    cvToday.text = GXFrDate(Gx_date, "dd/mm/yy")
    Gx_err = 0
End Sub ' SAV0V2
```

Sub UPV0V2()

```
Gx_time = Format$(Time(), "hh:mm:ss")
Gx_date = GxToDate((Format$(Now, "dd/mm/yy")), "dd/mm/yy", GX_NULDATE,
GX_ERRD%)
AV5Flgn = 0
Gx_emsg = ""
Gx_err = 0
Gx_execsa = False
gxTabPress = 0
GX_retp = 0
UserCancel = False
Call ICV0V0
next_to_vid = 1
Call SAV0V2
If UserCancel Then
    GXToReturn = True

    Unload Me
    Exit Sub
End If
RfrOgs = True
Call RFV0V2
End Sub ' UPV0V2
```

```
Function VFV0V2(ToValid As Integer) As Integer
    VFV0V2 = True
End Function ' VFV0V2
```

```
Function VLV0V2(next_to_vid As Integer, curr_fid As Integer) As Integer
    Static ILV0V2 As Integer
    Call RDV0V2
    In_field = 0
    If Not ILV0V2 Then
        ILV0V2 = True
        VLV0V2 = True
        For I% = next_to_vid To (curr_fid - 1)
            If Not VFV0V2(I%) Then
                VLV0V2 = False
                Exit For
            End If
        Next I%
        ILV0V2 = False
    End If
End Function ' VLV0V2
```

Private Sub RDV0V2()

 Select Case In_field
 End Select
End Sub ' RDV0V2

Private Sub VVV0V2()

End Sub ' VVV0V2

Sub DSV0V2()

 ' Redisplay Vars
 On Error Resume Next
 cVNomusuc.text = RTrim\$(AV4Nomusuc)
 cVToday.text = GXFrDate(Gx_date, "dd/mm/yy")
 cVTime.text = RTrim\$(Gx_time)
End Sub ' DSV0V2

Private Sub RFV0V2()

 Static gxlnRfr As Integer

 If Not gxlnRfr Then
 gxlnRfr = True
 GXNoExit = True
 Call RDV0V2
 If Rfr0gs = True Then
 Rfr0gs = False
 If Gx_exeoca Then
 SAV0V2
 Else
 Gx_exeoca = True
 End If
 If UserCancel Then GoTo lexRFV0V2
 Call U16V0V2
 If UserCancel Then GoTo lexRFV0V2
 Call DSV0V2
 End If
lexRFV0V2:
 GXNoExit = False
 gxlnRfr = False
 End If
End Sub ' RFV0V2

Sub U12V0V2()

```
' Movimientos' Procedure
Call RDV0V2
Me.AutoRedraw = True
Call PPUSU100(AV4Nomusuc, AV6Passc, "WMEN020", AV5Flgn)
Me.AutoRedraw = False
If Me.Enabled And Me.Name <> MainForm.ActiveForm.Name Then Me.SetFocus
If AV5Flgn = -1 Then
    Call DSV0V2
    MsgBox "Usted No Tiene Autorización Para Entrar a Este Módulo...", vbOKOnly
Else
    GxIsValid% = WWMEN020.GXObjDsp(AV3Razsocc, AV4Nomusuc, AV6Passc)
End If
Call DSV0V2
End Sub ' Event 'Movimientos'
```

Sub U11V0V2()

```
' Configuración' Procedure
Call RDV0V2
Me.AutoRedraw = True
Call PPUSU100(AV4Nomusuc, AV6Passc, "WMEN030", AV5Flgn)
Me.AutoRedraw = False
If Me.Enabled And Me.Name <> MainForm.ActiveForm.Name Then Me.SetFocus
If AV5Flgn = -1 Then
    Call DSV0V2
    MsgBox "Usted No Tiene Autorización Para Entrar a Este Módulo...", vbOKOnly
Else
    Me.AutoRedraw = True
    Call PPUSU170(AV4Nomusuc, AV6Passc)
    Me.AutoRedraw = False
    If Me.Enabled And Me.Name <> MainForm.ActiveForm.Name Then Me.SetFocus
    X = Shell("LConfigu.EXE", 1)
End If
Call DSV0V2
End Sub ' Event 'Configuración'
```

Sub U13V0V2()

```
'Seguridad' Procedure
Call RDV0V2
Me.AutoRedraw = True
Call PPU100(AV4Nomusuc, AV6Passc, "WMEN040", AV5Flgn)
Me.AutoRedraw = False
If Me.Enabled And Me.Name <> MainForm.ActiveForm.Name Then Me.SetFocus
If AV5Flgn = -1 Then
    Call DSV0V2
    MsgBox "Usted No Tiene Autorización Para Entrar a Este Módulo...", vbOKOnly
Else
    Me.AutoRedraw = True
    Call PPU170(AV4Nomusuc, AV6Passc)
    Me.AutoRedraw = False
    If Me.Enabled And Me.Name <> MainForm.ActiveForm.Name Then Me.SetFocus
    X = Shell("LSegund.EXE", 1)
End If
Call DSV0V2
End Sub ' Event 'Seguridad'
```

Sub U14V0V2()

```
'Consulta' Procedure
Call RDV0V2
Me.AutoRedraw = True
Call PPU100(AV4Nomusuc, AV6Passc, "WMEN080", AV5Flgn)
Me.AutoRedraw = False
If Me.Enabled And Me.Name <> MainForm.ActiveForm.Name Then Me.SetFocus
If AV5Flgn = -1 Then
    Call DSV0V2
    MsgBox "Usted No Tiene Autorización Para Entrar a Este Módulo...", vbOKOnly
Else
    GxIsValid% = WWWEN080.GXObjDsp(AV3Razsocc, AV4Nomusuc, AV6Passc)
End If
Call DSV0V2
End Sub ' Event 'Consulta'
```

Sub U16V0V2()

```
' Load Procedure
Call DSV0V2
End Sub ' Event Load
```

Sub GXDspMnu(gxForm As Form, gxMbr As GXMenuBar, Old As String)

Select Case Old

Case Is = "Movimientos"

Call U12V0V2

Case Is = "Configuración"

Call U11V0V2

Case Is = "Seguridad"

Call U13V0V2

Case Is = "Consulta"

Call U14V0V2

Case Else

BGXM_MOD.GXDspMnu gxForm, gxMbr, Old

End Select

End Sub ' GXDspMnu

Private Sub Form_Load()

Call GX_ResizeForm(Me, 45, 300, 9570, 6885)

Move 0, 0

X% = MoveWindow(cVNomusuc.hwnd, 13, 2, 11, 13, 1)

X% = MoveWindow(cVToday.hwnd, 15, 48, 11, 12, 1)

X% = MoveWindow(cVTime.hwnd, 584, 50, 11, 12, 1)

GXFrmMgr 1

End Sub ' Form_Load

Private Sub Form_Activate()

MainForm.mnuGX.Visible = False

GXMb.ClearOptions

GxIsValid% = GXMb.GetLevel("Gx_actions")

If GxIsValid% > 0 Then

GXMb.AddMnu GxIsValid%, GXMb.ItemTotal + 1, "Movimientos", "Movimientos", False, True, True, False, False, "F7"

GXMb.AddMnu GxIsValid%, GXMb.ItemTotal + 2, "Configuración", "Configuración", False, True, True, False, False, "F8"

GXMb.AddMnu GxIsValid%, GXMb.ItemTotal + 3, "Seguridad", "Seguridad", False, True, True, False, False, "F9"

GXMb.AddMnu GxIsValid%, GXMb.ItemTotal + 4, "Consulta", "Consulta", False, True, True, False, False, "F10"

End If

LockWindowUpdate MainForm.hwnd

GXDefTool GXTBObj, MainForm.gxtbr, MainForm.gxImL

LockWindowUpdate 0


```
MainForm.PanMode.Caption = ""
MainForm.PanInfo.Caption = ""
MainForm.PanMessage.Caption = ""
Screen.MousePointer = OMouse
End Sub ' Form_Activate
```

```
Private Sub Form_Deactivate()
```

```
    MainForm.PanMode.Caption = ""
    MainForm.PanInfo.Caption = ""
    MainForm.PanMessage.Caption = ""
End Sub ' Form_Deactivate
```

```
Sub GXMoveMnu(Index As Integer)
```

```
    Select Case Index
        Case Is = 4
            RfrOgs = True
            Call RFV0V2
    End Select
End Sub ' GXMoveMnu
```

```
Private Sub Form_Paint()
```

```
    Dim DC As Long

    DC = Me.hDC
End Sub ' Form_Paint
```

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
```

```
    Select Case KeyCode
        Case Is = 16, 17, 18
            Exit Sub
        Case Is = vbKeyF7
            Call U12V0V2
        Case Is = vbKeyF8
            Call U11V0V2
        Case Is = vbKeyF9
            Call U13V0V2
        Case Is = vbKeyF10
            Call U14V0V2
        Case Else
            BGXM_MOD.GXMnuKey KeyCode, Shift, MainForm, GXMb
    End Select
```

```
If KeyCode = 13 Then
    Call gxrbtnU15V0V0(True)
End If
If KeyCode = 27 Then
    UserCancel = True
End If
If UserCancel Then Unload Me
End Sub ' Form_KeyDown
```

```
Public Function GXObjDsp(frAV3Razsocc As String, frAV4Nomusuc As String, frAV6Passc
As String) As Integer
If Not GXToReturn Then
    On Error Resume Next
    Set GxTopForm = MainForm.ActiveForm
    ' Save caller Active Control info. VB3 and VB5 only.
    Set GxTopCtrl = GxTopForm.ActiveControl
    GxOldIndex = GxTopCtrl.TabIndex
    GxTopCtrl.TabIndex = 0
    If UCase(GxTopForm.LinkTopic) <> "MAINFORM" And UCase(GxTopForm.LinkTopic) <>
"GX_MENU" And GxTopForm.Name <> "WSIIPA" Then GxTopForm.Enabled = False '
Disable calling program
    On Error GoTo 0
    GXInitz = True
    In_field = 0
    GXEnCnt = 0
    GX_ERRD% = 0
    Load Me
    gxBtn6.Picture = LoadPicture(gxFFP("Crdfle13.bmp"))
    gxBtn7.Picture = LoadPicture(gxFFP("Crdfle12.bmp"))
    gxBtn5.Picture = LoadPicture(gxFFP("Secur05.bmp"))
    gxBtn16.Picture = LoadPicture(gxFFP("Binoculr.bmp"))
    gxBtn4.Picture = LoadPicture(gxFFP("Close1.bmp"))
    GXInitz = False
    AV3Razsocc = frAV3Razsocc
    AV4Nomusuc = frAV4Nomusuc
    AV6Passc = frAV6Passc
    Me.Show
    Me.Refresh
    Call UPV0V2
```

```
If Not UserCancel Then GXToReturn = True
Do While GXToReturn
  DoEvents
  Loop
  frAV3Razsocc = AV3Razsocc
  frAV4Nomusuc = AV4Nomusuc
  frAV6Passc = AV6Passc
End If
End Function ' GXObjDsp
```

```
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
```

```
  If GXToReturn And Not GXNoExit Then
    If UnloadMode = vbFormMDIForm Then
      Set GxTopForm = Nothing
      Set GxTopCtrl = Nothing
    End If
  Else
    UserCancel = False
    Cancel = True
  End If
End Sub ' Form_QueryUnload
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
  GXToReturn = False
  MainForm.PanMode.Caption = ""
  MainForm.PanInfo.Caption = ""
  MainForm.PanMessage.Caption = ""
  DBEngine.Idle dbFreeLocks
  On Error Resume Next
  GxTopForm.Enabled = True ' Enable calling program
  GxTopForm.SetFocus
  Set GxTopForm = Nothing
  ' Restore saved control tab index. VB3 and Vb5 only.
  GxTopCtrl.TabIndex = GxOldIndex
  GXFrmMgr -1
End Sub ' Form_Unload
```

```
Sub ICV0V0()
```

```
End Sub ' ICV0V0
```