

20
2ej.



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

DESARROLLO DE UN SISTEMA PARA EL CONTROL
DE LOS ARTICULOS DE CONSUMO EN
DEPENDENCIAS DE LA UNAM UTILIZANDO UNA
MICROCOMPUTADORA

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A :
MA. DEL ROCIO CARRILLO VELAZQUEZ

DIRECTOR DE TESIS: ING. SEBASTIAN POBLANO ORDONEZ

MEXICO, D. F.

JUNIO 1992



TESIS CON
FALLA DE ORIGEN



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

I N D I C E

INTRODUCCION

Capítulo 1

ANALISIS DEL SISTEMA ACTUAL

1.1. Análisis del sistema	4
1.2. Descripción del sistema	5
1.2.1. Procedimiento.	5
1.2.2. Representación gráfica del sistema	7
1.3. Estudio de la organización y el medio ambiente.	12
1.4. Evaluación del sistema.	13

Capítulo 2

INGENIERIA DE SOFTWARE

2.1. Evolución del software.	15
2.2. Crisis del software	17
2.3. Ingeniería de software.	17
2.3.1. Métodos	18
2.3.2. Herramientas	18
2.3.3. Procedimientos	18
2.4. Paradigmas de la ingeniería de software	19
2.4.1. Ciclo de vida de un sistema.	19
2.4.2. Construcción de prototipos	19
2.4.3. Técnicas de la cuarta generación	20
2.5. Fases del desarrollo del software	21
2.5.1. Definición	21
2.5.2. Desarrollo	22
2.5.3. Mantenimiento.	22
2.6. Ciclo de vida de un sistema	23
2.6.1. Análisis del sistema	24
2.6.2. Análisis de requerimientos	24
2.6.3. Diseño del sistema	25
2.6.4. Codificación	25
2.6.5. Integración y pruebas.	25
2.6.6. Mantenimiento.	26

Capítulo 3

ANALISIS DEL SISTEMA

3.1. Planificación del proyecto	27
3.2. Definición del problema.	28
3.2.1. Beneficios que debe brindar la solución óptima.	29
3.2.2. Alternativas de solución.	30

3.3.	Análisis de requerimientos	32
3.3.1.	Requerimientos del sistema.	33
3.3.2.	Requerimientos de equipo.	36
3.4.	Herramientas del análisis estructurado	36
3.4.1.	Diagramas de flujo de datos	37
3.3.1.	Diccionario de datos.	110
3.5.	Miniespecificaciones	112

Capítulo 4

DISEÑO DEL SISTEMA

4.1.	Calidad del software	134
4.2.	Acoplamiento y cohesión.	136
4.3.	Arquitectura del software	137
4.3.1.	Diagramas de estructura del sistema propuesto	148
4.3.2.	Diseño de las bases de datos.	146
4.3.2.1.	Normalización.	148
4.3.2.2.	Diseño conceptual.	150
4.3.2.3.	Diseño lógico.	151
4.3.2.4.	Diseño físico.	157
4.4.	Software para el desarrollo del sistema.	157
4.4.1.	Selección del lenguaje de programación.	157
4.4.2.	Características de Dbase III Plus y de Clipper.	160
4.5.	Diseño de entradas y salidas	162

Capítulo 5

IMPLEMENTACION

5.1.	Codificación	164
5.2.	Documentación del código	165
5.2.1.	Declaración de datos.	166
5.2.2.	Construcción de sentencias.	166
5.2.3.	Entrada/Salida.	166
5.1.4.	Eficiencia.	168

Capítulo 6

INTEGRACION Y PRUEBAS

6.1.	Integración.	177
6.2.	Pruebas.	179
6.3.	Documentación del sistema.	181

Capítulo 7

MANTENIMIENTO

7.1	Mantenimiento	183
CONCLUSIONES.		186

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

Dentro de la administración universitaria existe gran demanda en cuanto a los recursos y servicios que ésta brinda. Debido a esta gran dinámica y cambiante actividad, se han tenido que automatizar los diversos procesos administrativos, y para ello se encuentra la Dirección General de Servicios de Cómputo para la Administración siendo ésta una dependencia que proporciona apoyo y asesoría a otras dependencias universitarias en este ámbito.

Dentro de esta gran variedad de procesos administrativos se encuentra la contabilidad, el ejercicio presupuestal, el control de personal, así como el control de artículos de consumo, al cual se le denomina "Almacén" tema de esta tesis.

Considerando lo anterior, se hace necesaria la utilización de herramientas de cómputo, con el fin de soportar el proceso administrativo y hacerlo más fácil y eficiente.

El aumento en cuanto a los requerimientos de información oportuna y confiable con un mecanismo sistemático, que apoye estos procesos, crea la necesidad de elaborar un sistema para cada uno de ellos.

Los propósitos de este sistema de "Almacén" han sido, contar con los mecanismos sistemáticos que presenten más expedito el manejo de los artículos que forman el almacén interno de cada dependencia universitaria, eliminando así procesos manuales, integrando la información necesaria y aplicando las normas y criterios establecidos para ello.

El objetivo de este trabajo, es desarrollar un sistema de almacén para un microcomputador, que sirva para solucionar y apoyar las necesidades actuales que tienen las unidades administrativas de las diferentes dependencias universitarias.

El contenido de este trabajo esta organizado en siete capítulos, y muestran la forma en que ha sido desarrollado el sistema, desde la concepción hasta su liberación, todo bajo la metodología de "El ciclo de vida de un sistema".

El primer capítulo muestra el panorama general del sistema de almacén actual, la estructura, características y el medio ambiente que lo rodea, así como los problemas que presenta y la solución propuesta, que es el desarrollo de este nuevo sistema.

El segundo capítulo muestra un panorama general sobre la Ingeniería de Software, qué es, sus objetivos y técnicas para desarrollar sistemas. Además se explica brevemente cada una de las etapas o fases que forman el ciclo de vida de un sistema, que es la metodología de la Ingeniería de Software.

El tercer capítulo define el problema, las necesidades por las cuales se requiere el desarrollo de este sistema, así como el análisis de requerimientos.

El capítulo cuatro es referente a las especificaciones del sistema, considerando todas las herramientas para obtener un buen diseño; Se estructuran los archivos y se realizan pruebas con el problema real, además en esta fase se selecciona el lenguaje de programación que será utilizado, en este caso se utiliza el

administrador de bases de datos, Dbase III Plus.

En el capítulo cinco se elabora la codificación, implementación y documentación del sistema de Almacén.

El capítulo seis se refiere a las pruebas que se le realizan a la codificación, es aquí donde se compilan todos los programas, y se verifica integración de elementos de un módulo y la dependencia entre módulos. Además nos explica la forma en que deberá ser operado el sistema por el usuario.

Por último en el capítulo siete se realizan una serie de actividades que se dan una vez que el software se ha puesto en marcha, como es el mantenimiento correctivo, adaptivo y perfectivo.

Posteriormente se presenta la bibliografía empleada en este trabajo.

ANALISIS DEL SISTEMA ACTUAL**1.1. ANALISIS DEL SISTEMA**

La Dirección General de Servicios de Cómputo Para La Administración, diseñó y desarrolló para manejo de información por computadora un sistema de Almacén, el cual tiene como objetivo facilitar el control de los recursos materiales que se manejan en las diversas dependencias universitarias.

La Dirección General de Proveeduría, es la encargada de proporcionar los artículos de consumo, mismos que son controlados por medio de un catálogo de artículos, éste se relaciona directamente a las partidas presupuestales involucradas en el control del almacén.

CONTENIDO DEL CATALOGO

- a) **Clave de artículo:** es una secuencia de siete números, en donde los tres primeros indican la partida presupuestal y los cuatro restantes son un número consecutivo.
- b) **Descripción del artículo:** indica las características del artículo.
- c) **Unidad:** indica la unidad en que se maneja el artículo.

Los siete dígitos de la clave del artículo sirven para generar lo que se llama "Dígito de Control", el cual se utiliza como un número adicional a la clave del artículo para evitar repeticiones en el catálogo particular de las dependencias.

1.2. DESCRIPCIÓN DEL SISTEMA**1.2.1. PROCEDIMIENTO ACTUAL****1. Crear y actualizar los catálogos utilizados en el sistema:**

- a) Catálogo de partidas presupuestales
- b) Catálogo de artículos
- c) Catálogo de departamentos
- d) Catálogo de empleados

2. Realizar el registro de transacciones.

En primer lugar se elabora el registro del inventario inicial, en él se especifican las existencias del almacén al momento de utilizar por primera vez el sistema.

Este registro incluye:

- a) Clave del artículo
- b) Dígito de control
- c) Precio por unidad
- d) Cantidad existente del artículo en el almacén
- e) Existencia máxima
- f) Existencia mínima

3. En segundo lugar se llevan a cabo todas las transacciones referentes a entradas y/o salidas del almacén; así como, actualizaciones de existencias máximas y mínimas.

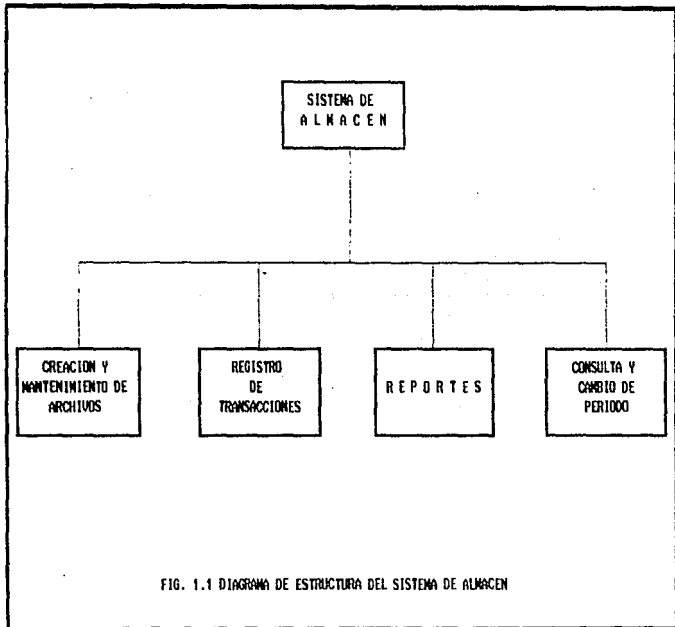
En la Nota de entrada se proporciona:

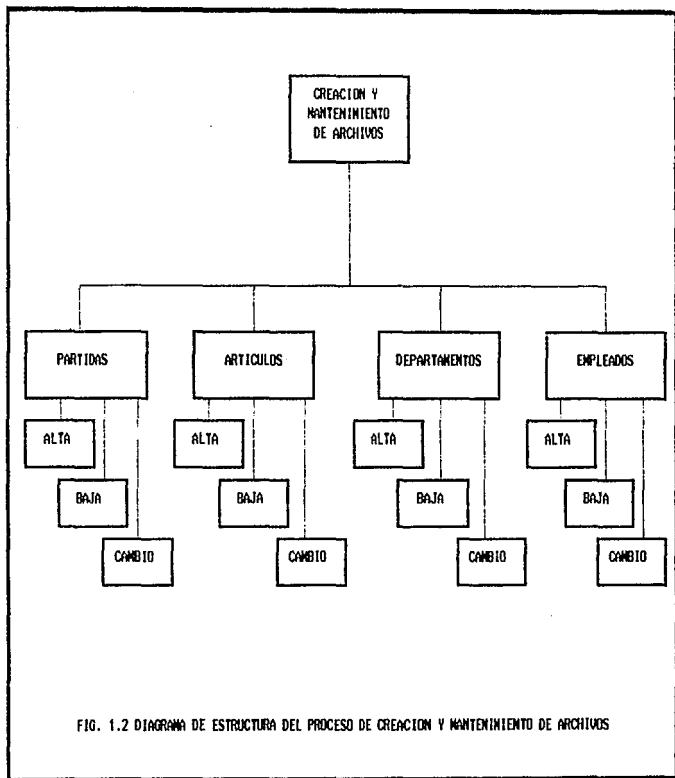
- a) Clave del artículo
- b) Dígito de control correspondiente
- c) Precio unitario

- d) Cantidad de artículos que entran al almacén
- En el Vale de salida se proporciona:
- a) Clave del artículo
 - b) Dígito de control
 - c) Registro federal de causante del empleado autorizado para sacar artículos del almacén
 - d) Cantidad de artículos que saca
4. Obtener una serie de reportes en papel que muestran el estado actual del almacén.
 5. Consulta por pantalla de los movimientos.

1.2.2. REPRESENTACION GRAFICA DEL SISTEMA ACTUAL**Diagramas de estructura**

Muestran la organización jerárquica de los módulos componentes del sistema:





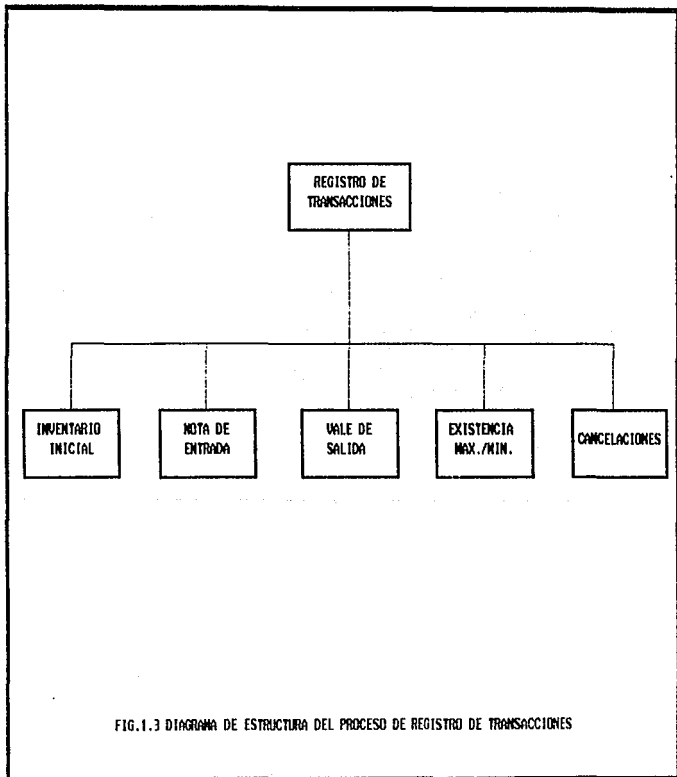


FIG.1.3 DIAGRAMA DE ESTRUCTURA DEL PROCESO DE REGISTRO DE TRANSACCIONES

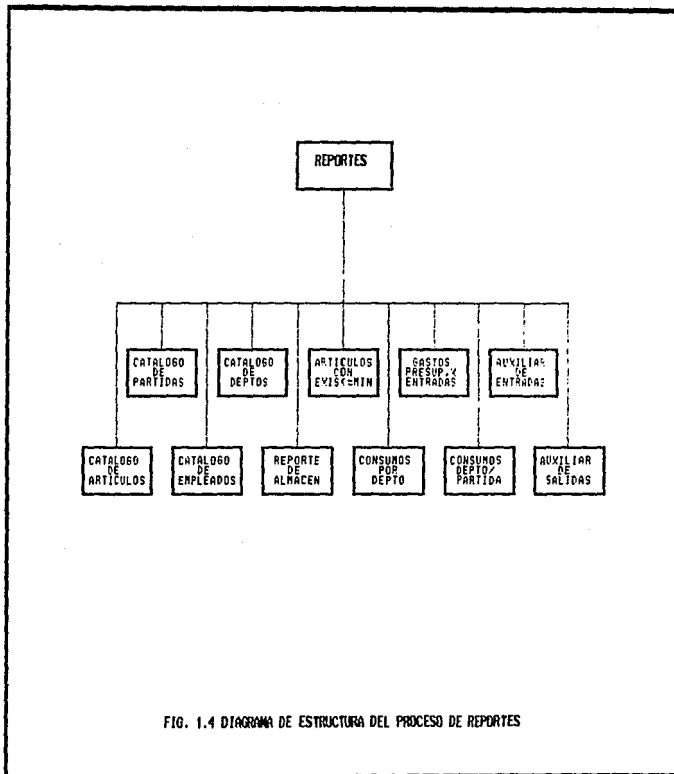


FIG. 1.4 DIAGRAMA DE ESTRUCTURA DEL PROCESO DE REPORTES

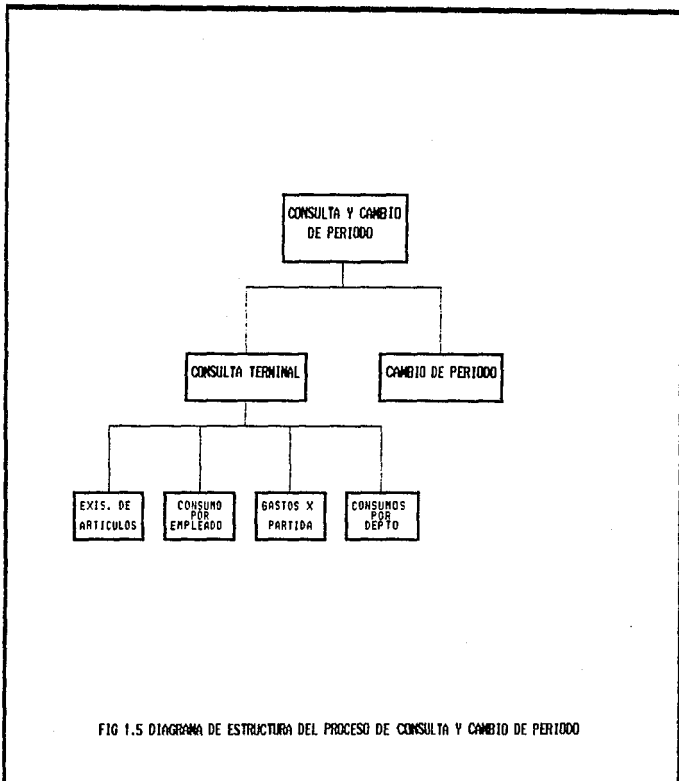


FIG 1.5 DIAGRAMA DE ESTRUCTURA DEL PROCESO DE CONSULTA Y CAMBIO DE PERIODO

1.3. ESTUDIO DE LA ORGANIZACIÓN Y EL MEDIO AMBIENTE

El sistema de almacén se conceptualizó y desarrolló inicialmente como una solución para automatizar el control de los artículos de consumo en la Universidad Nacional Autónoma de México (U.N.A.M.), entonces se contaba con equipo de cómputo marca Cromemco, y el lenguaje de programación utilizado era el Cobol.

Posteriormente la U.N.A.M. adquirió nuevo equipo de cómputo formado por Computadoras PC compatible, que se distribuyó en las diferentes dependencias.

De ahí se generó la necesidad de actualizar el sistema y se procedió a trasladarlo a Cobol para PC, desde luego el Cobol era compatible.

Como es sabido, en el mercado existen avances y surgen versiones mejoradas para el software, el lenguaje Cobol no fué la excepción y se creó una nueva versión de su compilador, que con respecto a la versión anterior contenía grandes mejoras, sin embargo, el problema que se presentó es que no eran compatibles como se suponía, las mejoras exigían nuevas estructuras de los programas, así como de la sintáxis de las instrucciones.

En la U.N.A.M. se contempló la posibilidad de trasladar el sistema haciendo uso de este software, pero presenta gran desventaja de volver a elaborar programas y reestructurarlos en la sintáxis, por lo cual se decide proponer posibles alternativas de solución las cuales se analizarán en el siguiente capítulo.

1.4. EVALUACION DEL SISTEMA

Después de haber analizado el sistema actual, se pueden detectar los siguientes problemas y requerimientos:

- a) La información referente a los artículos es obsoleta con respecto al nuevo catálogo de artículos proporcionado por la Dirección General de Proveeduría.
- b) La información presentada en la consulta por pantalla se ve limitada, por lo que se requieren nuevas formas de consulta de la información.
- c) Se requieren nuevos reportes que presenten el estado del almacén en forma más detallada y concisa.
- d) Se presentan problemas al acceder a los registros de los archivos después de llegar a mil quinientos aproximadamente.
- e) Se presentan problemas en la recuperación de archivos relativos dañados, provocando la pérdida de información.
- f) Se requiere crear un módulo de mantenimiento de los archivos, como lo es la realización de una protección y restauración de información automática desde el sistema sin tener que salir de esté.
- g) Es necesario que el sistema sea amigable y proporcione ayuda durante los procesos, de tal forma que permita al usuario resolver algunas dudas.
- h) Es necesario elaborar una versión especial por cada dependencia usuaria, lo cual aumenta el tiempo en la entrega del sistema.

- 1) Es necesario en un futuro integrar los sistemas de presupuesto, almacén, resguardos, contabilidad y personal en un solo paquete administrativo por lo que se requiere homogeneizarlos.

Actualmente los sistemas de contabilidad, presupuesto y personal administrativo ya se encuentran elaborados con Dbase III Plus. Debido a las limitaciones antes mencionadas que presenta el sistema actualmente, podemos considerar como la primer alternativa de solución, crear una nueva versión y reestructurar un nuevo sistema que cubra los requerimientos actuales, utilizando las ventajas que Dbase III Plus presenta en el manejo de grandes volúmenes de información y la posibilidad de generar índices de una forma rápida y segura, así como el inmediato acceso a los archivos.

Durante las primeras décadas en la era de la computación, uno de los principales desafíos era el desarrollar hardware, que ayudará a reducir los costos de procesamiento y almacenamiento de información.

A mediados de la década de los 80's, los adelantos en microelectrónica fueron notables, lo que permitió tener fuertes resultados en la computación y solucionar los desafíos.

Hoy los problemas y desafíos se ven orientados a la necesidad que ha surgido de reducir el costo y mejorar la calidad de las soluciones basadas en computadoras, soluciones que se obtienen con el software.

2.1. LA EVOLUCION DEL SOFTWARE

El contexto en el que se ha desarrollado el software está fuertemente ligado a las cuatro décadas de la evolución de los sistemas informáticos. Un mejor rendimiento del hardware, un tamaño más pequeño y un costo más bajo, han dado lugar a sistemas más sofisticados y eficientes.

En los primeros años, se implementaban sistemas informáticos pero se sabía poco acerca de la ingeniería de las computadoras, debido a que la mayoría del software era diseñado y utilizado por la misma persona u organización.

La segunda era se extiende desde la mitad de la década de los años 60's hasta finales de los años 70's, la multiprogramación y los sistemas multiusuario, introdujeron nuevos conceptos de interacción hombre-máquina. Con las técnicas interactivas surgió un nuevo mundo de aplicaciones y nuevos niveles de sofisticación del hardware y software.

La segunda era se caracterizó también por el uso de software como producto y la llegada de las "casas de software", quienes desarrollaban gran cantidad de software y se encargaban de comercializarlo y venderlo a miles de usuarios. Pero cuando se detectaba alguna falla en ellos, tenían que corregirlas, así fue como a estas actividades se les denominó "Mantenimiento del Software", en muchas ocasiones este mantenimiento era difícil por lo que surgió la llamada "Crisis del Software".

La tercera era de la evolución de los sistemas de computadoras comenzó a mediados de los años 70's y aún continúa. Los sistemas distribuidos y las computadoras múltiples (cada una ejecutando funciones concurrentemente y comunicándose con alguna otra), son quienes incrementaron notablemente la complejidad de los sistemas informáticos; esta era también se caracterizó por la llegada y amplio uso de microprocesadores y computadoras personales.

Las computadoras personales han sido el principal factor por el que muchas compañías del software tienen tanta demanda en sus productos, además de que estos productos son estándar.

La cuarta era de la evolución de los sistemas está comenzando, actualmente las "técnicas de la cuarta generación para el desarrollo de software" están cambiando la forma en que la comunidad informática elabora los programas de computadora. Por otro lado los sistemas expertos y la inteligencia artificial se empiezan a introducir a un amplio rango de problemas del mundo real. Pero conforme continúa la cuarta era, los problemas continúan por lo que la crisis del software se ha intensificado.

2.2. LA CRISIS DEL SOFTWARE

La crisis del software se refiere al conjunto de problemas encontrados en el desarrollo del software de computadoras; abarca los problemas asociados con: cómo desarrollar el software, cómo mantener un volumen creciente de software y cómo satisfacer la creciente demanda de éste.

En respuesta a la crisis del software muchas industrias están adoptando prácticas de ingeniería de software.

2.3. INGENIERIA DE SOFTWARE

Existen varias definiciones sobre el significado de la ingeniería de software. Sin embargo, la definición propuesta por Fritz Bauer en la primera conferencia dedicada al tema de la crisis del software, en la que define: "La ingeniería de software es el establecimiento y uso de principios de ingeniería robustos, orientados a obtener económicamente software que sea fiable y

funciona eficientemente sobre máquinas reales", se considera la más acertada por que abarca tres elementos claves: métodos, herramientas y procedimientos, que facilitan al gestor controlar el proceso del desarrollo del software y suministrar a los que practiquen dicha ingeniería las bases para construir software de alta calidad de una forma productiva.

2.3.1. METODOS

Los métodos de la ingeniería de software constituyen el "COMO" construir técnicamente el software. Incluyen planificación y estimación de proyectos; análisis de los requerimientos; diseño de estructuras de datos; arquitectura de programas ; y procedimientos algorítmicos, codificación, pruebas y mantenimiento.

2.3.2. HERRAMIENTAS

Las herramientas de la ingeniería de software suministran un soporte automático o semiautomático para los métodos, actualmente existen diferentes herramientas para soportar cada uno de estos métodos. Por ejemplo: la ingeniería de software asistido por computadora.

2.3.3. PROCEDIMIENTOS

Los procedimientos de la ingeniería del software afectan los métodos y las herramientas, facilitando un desarrollo racional y oportuno del software de la computadora. También definen la

secuencia en la que se requieren los controles que ayudan a asegurar la calidad; coordinan los cambios y las guías que facilitan a los gestores del software establecer su desarrollo.

2.4. PARADIGMAS DE LA INGENIERIA DE SOFTWARE

La ingeniería de software está compuesta de pasos que abarcan los métodos, las herramientas y los procedimientos, estos pasos se denominan paradigmas de la ingeniería de software, existen básicamente tres paradigmas:

2.4.1. EL CICLO DE VIDA DE UN SISTEMA

Este paradigma exige un enfoque sistemático y secuencial del desarrollo del software que inicia con en el análisis del sistema y progresa a través del diseño, codificación, pruebas y mantenimiento.

2.4.2. CONSTRUCCION DE PROTOTIPOS

Es un proceso que ayuda al programador, permitiendo la creación de un modelo del software a construir, donde el modelo toma una de las tres formas siguientes: un prototipo en papel, un prototipo que funcione o un programa existente que ejecute parte o toda la función deseada.

2.4.3. TECNICAS DE LA CUARTA GENERACION

El término técnicas de la cuarta generación abarca un amplio espectro de herramientas de software que tienen en común el facilitar el trabajo a quien desarrolla el software y especificar algunas características de éste.

Actualmente, un entorno para el desarrollo del software que soporte este paradigma incluye algunas o todas de las siguientes herramientas: lenguajes no procedimentales para consultar bases de datos, generación de informes, manipulación de datos, interacción y definición de pantallas y generación de código. Además, capacidades gráficas de alto nivel y capacidad de hoja de cálculo, cada una de estas herramientas existen actualmente, pero sólo para dominios de aplicación muy específicos.

Frecuentemente se describe a los paradigmas de la ingeniería del software como métodos alternativos, en vez de métodos complementarios; En muchos casos, los paradigmas pueden y deben combinarse de forma que se utilicen todas las ventajas de cada uno en un solo proyecto. Cada uno, aunque distinto, tiene tres fases en común que son las que forman el desarrollo del software.

Ahora bien, la naturaleza del proyecto es quien dicta el paradigma a utilizar.

2.5. FASES DEL DESARROLLO DEL SOFTWARE

El proceso de desarrollo del software contiene tres fases genéricas independientemente del paradigma de ingeniería elegido, del área de aplicación, tamaño del proyecto y/o complejidad. Estos son: definición, desarrollo y mantenimiento.

2.5.1. DEFINICION

Se enfoca sobre el QUE. Esto es, durante la definición quien desarrolla el software intenta identificar qué información ha de ser procesada, qué función y rendimiento se desea, qué interfaces han de establecerse, qué ligaduras de diseño existen y qué criterios de validación se necesitan para definir un sistema correcto. Por tanto, han de identificarse los requerimientos claves del sistema y del software.

Aunque los métodos aplicados durante la fase de definición variarán dependiendo del paradigma de ingeniería de software aplicado (o combinación de estos), de alguna forma se producen tres pasos específicos:

- a) Análisis del sistema
- b) Planificación del proyecto
- c) Análisis de requerimientos

2.5.2. DESARROLLO

La fase de desarrollo se enfoca sobre el COMO. Esto es, durante el desarrollo, quien elabora el software intenta descubrir cómo han de diseñarse las estructuras de datos y la arquitectura del software, cómo han de implementarse los detalles de procedimiento, cómo ha de trasladarse el diseño a un lenguaje de programación y cómo ha de realizarse la prueba.

Los métodos aplicados durante la fase de definición variarán dependiendo del paradigma de ingeniería del software aplicado, sin embargo, de alguna forma se producirán tres pasos concretos:

- a) Diseño del software.
- b) Codificación.
- c) Prueba del software.

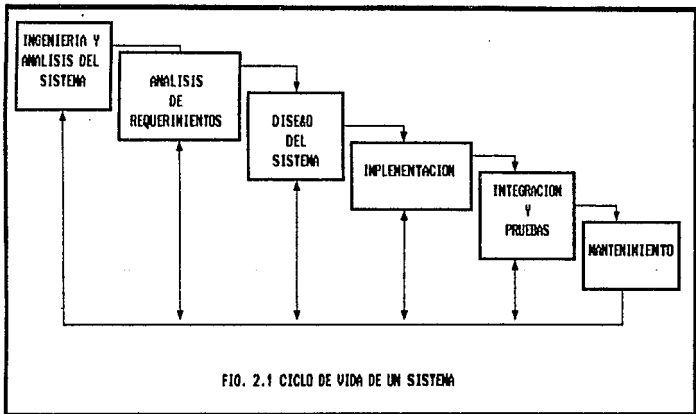
2.5.3. MANTENIMIENTO

Esta fase se enfoca sobre el cambio que va asociado con la corrección de errores, adaptaciones requeridas por la evolución del entorno del software y modificaciones debidas a los cambios de los requerimientos del cliente para reforzar o aumentar el sistema. Durante la fase de mantenimiento se encuentran tres tipos de cambios:

- a) Correctivo
- b) Adaptivo
- c) Perfectivo

2.6. CICLO DE VIDA DE UN SISTEMA

La figura 2.1 muestra el paradigma del ciclo de vida clásico para la ingeniería del software. También llamado algunas veces el "modelo en cascada", paradigma que ha sido utilizado en el desarrollo del sistema de Almacén para un microcomputador, objetivo de esta tesis.



En el proceso de selección de este modelo, se tomaron en cuenta dos factores : la naturaleza del problema y las ventajas que proporciona en la clasificación y control de las diferentes actividades para el desarrollo y mantenimiento del sistema. Factores que permiten una mejor administración y asignación de recursos, así como una mayor calidad y eficiencia en el producto.

2.6.1. ANALISIS DEL SISTEMA

Durante esta etapa se identifica claramente la necesidad del sistema, es decir, cuál es el problema, el POR QUE de éste y se formulan conceptos tales como las restricciones, el objetivo, los recursos disponibles, los recursos necesarios y se establece, si estos son factibles.

El resultado de esta etapa es la formulación del sistema y una serie de requerimientos para su diseño e implementación.

2.6.2. ANALISIS DE REQUERIMIENTOS

Una vez que el problema ha sido completamente entendido, los requerimientos y factibilidad en general de la solución deben ser determinados. En esta fase se define el QUE se tiene, se documenta el plan a seguir, los objetivos, los reportes, las consultas, los tipos de ayuda y las utilerías.

Además permite conocer más a fondo el problema para darle una solución óptima, haciendo uso de lo que se conoce como "Análisis Estructurado".

2.6.3. DISEÑO DEL SISTEMA

El diseño del software es realmente un proceso multipaso que se enfoca sobre tres atributos distintos del programa: estructura de datos, arquitectura del software y detalle procedimental. Este proceso traduce los requerimientos en una representación del software que pueda ser establecida de forma que obtenga la calidad requerida antes de que comience la codificación.

2.6.4. IMPLEMENTACION

El objetivo de la fase de implementación es codificar el diseño y documentar el código; es decir trasladar el diseño a una forma legible para la máquina y elaborar el código, así como compilarlo. Esta fase se concluye cuando el código se encuentra libre de errores.

2.6.5. INTEGRACION Y PRUEBAS

La escritura del código necesita ser integrado y probado rigurosamente, basado en los requerimientos de calidad; este es el objetivo de la fase de integración y pruebas.

Las pruebas normalmente se realizan por pasos, primero los módulos son probados en forma independiente, esta es la "prueba unitaria" después, los módulos son probados en grupos para ver la interacción propia de cada módulo, esta es la "prueba de integración". Posteriormente se procede a la prueba de instalación del sistema en el lugar en donde será operado, y se prueba durante un tiempo.

Será aceptado cuando el usuario quede convencido de que cumple los requerimientos.

2.6.6. MANTENIMIENTO

Después de instalado el software, éste debe operarse y asegurarse de un buen mantenimiento en forma correctiva, perfectiva y adaptiva; en esta fase también se realiza la documentación del sistema, es decir los manuales que sirvan para realizar el mantenimiento adecuadamente, siendo estos entendibles para cualquier usuario; Con esta fase se concluye el ciclo de vida de un sistema.

De esta forma se ha mostrado un panorama general acerca de lo que es la Ingeniería de Software, lo que pretende y la metodología seleccionada para el desarrollo del sistema de "Almacén para un microcomputador ", con el fin de obtener un sistema de alta calidad y alto rendimiento para los usuarios.

El "Ciclo de Vida" de un sistema, como se menciona en el capítulo dos, se inicia con la fase correspondiente al "análisis del sistema", del resultado de ésta, se obtiene la decisión de continuar con la siguiente fase del proyecto denominada "análisis de requerimientos" o abandonarla por incoesteable.

Esta etapa permite describir y realizar un estudio sobre la necesidad y factibilidad del sistema a desarrollar, es decir, se elabora una investigación del medio ambiente actual y se proponen diferentes alternativas de solución.

3.1. PLANIFICACION DEL PROYECTO

El estado que presentan las dependencias de la UNAM, es un ambiente propiamente administrativo en donde las personas encargadas del proceso de control de artículos de consumo carecen de experiencia en el manejo de una computadora. De ahí que, es necesario trazar una capacitación continua que permita la ejecución del proceso tal y como se debe llevar a cabo en la Universidad Nacional.

Recientemente, la UNAM adquirió equipo de cómputo (PC's), asignándose una para cada unidad administrativa de cada dependencia que la conforman, a fin de que éstas lleven el control de sus procesos administrativos en forma automatizada. También, se les asignó software el cual consiste en un procesador de palabras y en el sistema operativo MS-DOS.

3.2. DEFINICIÓN DEL PROBLEMA

En esta actividad se elabora una descripción del problema que permita definir posteriormente los requerimientos.

Una vez definido el problema, se proponen diferentes alternativas de solución, de las cuales se analizan las ventajas y desventajas con el propósito de contar con criterios sólidos de evaluación; de las alternativas se escoge la que brinda mejores ventajas y se desarrolla.

La tarea en sí, es proporcionar a las diversas dependencias universitarias un sistema que brinde mayor facilidad en el manejo y control de los artículos de consumo interno.

Actualmente el control de artículos se realiza con ayuda de un sistema elaborado en lenguaje Cobol, el cual debido a los problemas que presenta este sistema, referentes a el manejo de grandes volúmenes de información, y nuevas opciones del sistema, se plantea la necesidad de elaborar uno nuevo que proporcione mayor ventaja al existente. Además que automatice algunas actividades que se realizan en forma manual como algunos reportes que muestren los gastos y consumos por artículo, empleado, etc; agilice el control de materiales y sea adaptable y manejable para los usuarios.

Por otra parte, es necesario apuntar que a través del proceso de almacén se realiza el control de los artículos, de ahí que en dicho proceso se registren el precio unitario de éstos, existencias máximas y mínimas, y características en general de cada artículo mediante las notas de entrada y vales de salida.

En estos documentos se registran los movimientos de los artículos, obteniéndose a partir de ellos una serie de listados por existencias en el almacén, reportes que sirven de enlace con la contabilidad y el presupuesto.

A través de las notas de entrada y vales de salida, se controla el almacén y a partir de ellos se sabe el estado de éste en cualquier momento.

Es necesario hacer notar que el proceso de almacén, es particular en las dependencias de la universidad, ya que éstas controlan su propio almacén.

3.2.1. BENEFICIOS QUE DEBE BRINDAR LA SOLUCION OPTIMA.

Dentro de los beneficios que debe brindar la solución al problema, se encuentran básicamente dos.

1. Los que se pueden palpar; puede ser el ahorro económico, o el aumento en la productividad.
2. Los beneficios o factores externos que no podemos palpar pero que son visibles para el usuario; es decir, requisitos de complejidad, nivel de complejidad del usuario, previas capacidades para preservar o mejorar, nuevas capacidades por proporcionarse, requisitos de confiabilidad, modificaciones factibles, requisitos de portatibilidad y asuntos de seguridad entre otras.

3.2.2. ALTERNATIVAS DE SOLUCION

Una vez mencionado el problema y los beneficios que debe brindar la solución óptima, se plantean las siguientes alternativas de solución, analizando cada una de ellas para seleccionar a la mejor:

1) Utilizar el sistema de Almacén ya existente, elaborado en lenguaje Cobol.

ventajas:

- Facilidad en la realización del proceso.
- Rapidez en la obtención de información concisa y confiable.
- Aumento en la productividad.

desventajas:

- Presenta problemas en el acceso a registros de los archivos.
- Presenta pérdida de información al realizar la recuperación de archivos relativos.
- Existen limitantes en reportes y consultas.
- No cubre las necesidades actuales.

Por las desventajas expuestas y por que definitivamente esta propuesta no cubra las necesidades actuales, se descarta esta posibilidad.

2) Adquirir un paquete ya elaborado y comercializado.

ventajas:

- Poca inversión de tiempo.
- Ningún esfuerzo para el desarrollo.

desventajas:

- Implica un gasto muy alto, ya que deberá comprarse el software y éste es caro.
- Este tipo de paquetes están diseñados para aplicaciones muy generales, implicaría una adaptación del usuario al paquete.
- Inversión de tiempo y esfuerzo en la capacitación de los usuarios

Siendo que la Universidad cuenta con un proceso particular de llevar a cabo el control del almacén, necesidad que esta propuesta no cubre y, que implica un gasto muy alto y no se cuenta con los recursos económicos esta propuesta queda descartada.

- 3) Realizar el desarrollo de un sistema de Almacén para ser utilizado en un microcomputador PC.

ventajas:

- Ofrece un gran ahorro económico, ya que no se invertirá dinero en el desarrollo.
- Aumento en la productividad.
- Obtención de las salidas en forma veraz y oportuna.
- Elaboración del paquete en base a las necesidades específicas de las dependencias.
- Utilización del sistema en forma eficiente .
- Realizará la mayor parte de los procesos que se realizan manualmente.

- Proporcionar los reportes necesarios para facilitar el trabajo al usuario.
- Autorespaldo, recuperación automática, ayuda durante todo el sistema.
- Ejecución de varias tareas por unidad de tiempo.

desventajas:

- Inversión de tiempo y esfuerzo en la capacitación de los usuarios.
- Los usuarios requieren estudios sobre manuales de instalación y uso del paquete.
- Gran cantidad de tiempo invertido en el desarrollo.

Considerando que esta propuesta cubre las necesidades actuales, que no se invertiría económicamente nada, y que el sistema llevaría el proceso de control de los artículos tal y como se lleva en esta universidad proporcionando los resultados en forma oportuna y veraz, se decide realizar esta propuesta.

3.3. ANÁLISIS DE REQUERIMIENTOS

Los requerimientos descubren en detalle como será implementado el sistema, sus funciones: qué puede hacer éste, qué información será transformada y cómo. Proporciona un panorama preciso de lo que requiere la solución viable.

Existen diferentes métodos y técnicas para desarrollar requerimientos funcionales, este capítulo enfatiza una técnica particular llamada "Análisis Estructurado".

3.3.1. REQUERIMIENTOS DEL SISTEMA

Para realizar la aplicación de las técnicas se presenta una lista de los requerimientos del sistema de acuerdo a la descripción del problema visto anteriormente.

- a) Se requiere crear y mantener un archivo de partidas, uno de artículos, uno de departamentos y uno de empleados, que sirvan como catálogos y que a su vez contendrán información referente a cada uno de ellos.
- b) Es necesario obtener las entradas al sistema y además validarlas para evitar posibles errores durante el proceso. Estas entradas son las "Notas de entrada" y los "Vales de salida" y los datos que se deberán verificar de acuerdo al catálogo presupuestal y al catálogo de artículos que proporciona la Dirección General de Proveeduría son: las claves de las partidas y las claves de los artículos.
- c) Si todos los datos capturados son correctos se almacenarán, pero si existieran errores en estas entradas, debe haber la manera de corregirlas, es decir, que la información que se capture deberá almacenarse aunque ésta no sea correcta para evitar doble trabajo en la captura, y posteriormente poder corregirla.
- d) Una vez que se tenga la información de entrada correcta se deben emitir los siguientes reportes durante el período que se este trabajando, en los que se muestran los movimientos registrados y el estado actual del almacén:
 - Reportes de catálogos

- Reporte general del almacén
 - Reporte de entradas y salidas
 - Reporte de consumos y gastos presupuestales
- e) Deberá realizar un cierre de período con el fin de obtener consumos y gastos presupuestales acumulados por período de tiempo.
- f) Se requiere que el sistema permita al usuario consultar en cualquier momento la información capturada, los consumos por artículo, departamento, empleado, partida, los gastos presupuestales por partida y las existencias reales en el almacén por artículo.
- g) Se requiere crear un archivo de ayuda, el cual contenga los textos a desplegar durante la ejecución del sistema cuando así sea solicitada.

Podemos establecer la necesidad de definir siete funciones básicamente durante todo el proceso del control de artículos de consumo, englobadas cada una de ellas en un módulo, como se muestra a continuación:

Módulo 1 " CATALOGOS "

Debe permitir registrar todas las claves de partidas, artículos, departamentos y empleados que serán utilizadas durante todo el proceso de control de artículos de consumo.

Módulo 2 " CAPTURA "

Este módulo es el que se encarga de registrar por medio de la captura toda la información contenida en los documentos antes

mencionados (Nota de entrada/Vale de salida), es quien válida la información para evitar posibles errores de captura que se verían reflejados posteriormente en los resultados.

Módulo 3 " CANCELACIONES "

Este módulo debe permitir realizar cancelaciones de los registros erróneos de las notas de entrada o vales de salida, realizando la actualización de existencias reales por cada artículo.

Módulo 4 " CONSULTAS "

Este proceso debe permitir realizar la consulta por pantalla en cualquier momento , de cualquier vale de salida o nota de entrada, así como la existencia real que se tenga de cada uno de los artículos.

Módulo 5 " REPORTEES "

Este módulo debe permitir emitir reportes de cada uno de los catálogos, así como una serie de reportes que muestren el estado actual del almacén y los movimientos que se hayan registrado.

Módulo 6 " CIERRE DE PERIODO "

Este proceso debe resumir y actualizar la información de forma mensual y/o anual. Y emitir el reporte general de almacén.

Módulo 7 " MANTENIMIENTO A LOS ARCHIVOS "

En este módulo se debe dar la posibilidad al usuario de realizar una protección y restauración de la información actual que se este trabajando en el sistema.

3.3.2. REQUERIMIENTOS DE EQUIPO

Se cuenta con un equipo PC/AT compatible, con el que se desarrollará el sistema, el cual tiene las siguientes características:

- Memoria RAM (Random Access Memory) de 640 KB
- Disco duro de 40 MB
- Monitor a color VGA
- Unidad de disco flexible de 5¼"
- Impresora Estándar marca EPSON
- Papelería
- Diskettes de 5¼ "
- Papel de forma continua

3.4. HERRAMIENTAS DEL ANÁLISIS ESTRUCTURADO

El análisis estructurado se puede definir como una metodología para especificar funcionalmente un sistema.

Las entradas para la fase de requerimientos es la declaración del problema, que en el capítulo anterior se describió y como salida una serie de requisitos, en particular los cuales deben ser:

- a) correctos
- b) completos
- c) concisos
- d) no ambiguos
- e) funcionales
- f) verificables

Ahora bien, estos requisitos pueden lograrse y mejorarse por el uso de herramientas de análisis estructurado, como son : Diagramas de flujo de datos y Diccionario de datos.

3.4.1. DIAGRAMAS DE FLUJO DE DATOS

Conforme la información se mueve a través del software, se modifica mediante una serie de transformaciones. Un diagrama de flujo de datos, es una técnica gráfica que describe el flujo de la información y las transformaciones que se aplican a los datos, conforme se mueven de la entrada a la salida.

La simbología de un diagrama de flujo de datos es la siguiente:



Entrada Externa: Una fuente de entradas al sistema, o fuente de salida del sistema.



Proceso: Ejecuta alguna transformación de sus datos de entrada, produciendo sus datos de salida.



Flujo de datos: Se usa para conectar los procesos entre sí, a las fuentes o a los archivos; la cabeza de las flechas indica la dirección de transferencia de los datos.



Almacén de datos: un depósito de datos, la cabeza de las flechas indica las entradas y salidas al almacén.

Se utiliza un rectángulo para representar una entidad externa, esto es, un elemento del sistema u otro sistema que produce información que ha de ser transformada por el software o que recibe información producida por el software. Un círculo representa un proceso o transformación que se aplica a los datos y que los cambia de alguna manera. Una flecha representa uno o más elementos de datos. Todas ellas deben estar etiquetadas. Una doble línea representa un almacenamiento de datos.

Para realizar un diagrama de flujo de datos se definen primero, los diferentes elementos de entrada y salida del sistema, se realiza un diagrama el cual comience con las fuentes primarias de entrada y contenga las transformaciones lógicas, asegurándose de que esté se acerque lo más posible a los requerimientos. Es necesario que en este diagrama no existan cruces en el flujo de los datos, y al realizar una 'caminata' através de él, la persona involucrada en el uso del proceso quede satisfecha de la secuencia.

A continuación se mencionan las entradas y salidas del sistema:

a) ENTRADAS

El primer documento de entrada al sistema se refiere a la "Nota de entrada"; en la que se sugiere que al capturarse se consideren los siguientes datos:

Fecha de la Nota: Es la fecha de la elaboración de la Nota en el formato día, mes, año.

No. de nota : Es un número consecutivo que contiene la nota.

Clave de artículo: Es un número de siete posiciones, donde los tres primeros dígitos son la partida y los otros cuatro son un número consecutivo.

Unidad: Es la unidad que se maneja en ese artículo.

Descripción: Es la descripción referente al artículo en ella se podrá agregar cualquier comentario adicional.

Precio p/u: Es el precio por unidad del artículo.

El segundo documento de entrada al sistema es el "vale de salida", en esta forma se registra la salida de artículos del almacén, proporcionando la siguiente información necesaria para ello:

Solicitante: Se refiere al R.F.C. de la persona autorizada para sacar artículos del almacén.

Departamento: Es un número que se le asigna a cada uno de los departamentos que tiene la dependencia.

Fecha del Vale: Es la fecha de la elaboración del Vale en el formato día, mes, año.

No. de documento: Es un número consecutivo que contiene el número de vale.

Clave de artículo: Es un número de siete posiciones, donde los tres primeros dígitos son la partida y los otros cuatro son un número consecutivo.

Cantidad: Es un número referente a la cantidad de artículos que se va a sacar del almacén.

Para obtener los datos antes descritos se sugieren los siguientes formatos de captura utilizados en la UNAM:

b) SALIDAS

Como salidas se pueden considerar los reportes y las consultas por pantalla. A continuación se muestran los reportes emitidos y algunas de las pantallas de consulta:

- Catálogo de Partidas.

El catálogo de partidas es un reporte que muestra la clave y descripción de las partidas existentes, con las cuales se trabajará durante el proceso, y tiene el siguiente formato:

NOMBRE DE LA DEPENDENCIA	
REPORTES DE PARTIDAS	
CLAVE DE PARTIDA	DESCRIPCION
.	.
.	.
232	IMPRESIONES GRAFICAS

- Catálogo de Artículos.

El catálogo de artículos es un reporte que muestra la clave y descripción de los artículos existentes, con los cuales se trabajará durante el proceso, y tiene el siguiente formato:

NOMBRE DE LA DEPENDENCIA		
REPORTES DE ARTICULOS		
CLAVE DE ARTICULO	UNIDAD	DESCRIPCION
.	.	.
.	.	.
.	.	.
2321001	PIEZA	BLOCK CONTROL DE ACCESO

- Catálogo de departamentos.

El catálogo de departamentos es un reporte que muestra la clave y descripción de los departamentos que se tienen en la dependencia y tiene el siguiente formato:

NOMBRE DE LA DEPENDENCIA REPORTE DE DEPARTAMENTOS	
CLAVE DE DEPTO.	DESCRIPCION
.	.
.	.
.	.
305	ASESORIA E INPLANTACION DE SIST.

- Catálogo de empleados.

El catálogo de empleados es un reporte que muestra el RFC del empleado, su nombre y departamento al que esta adscrito, y tiene el siguiente formato:

NOMBRE DE LA DEPENDENCIA REPORTE DE EMPLEADOS		
RFC	NOMBRE	DEPTO
.	.	.
.	.	.
.	.	.
NAPAS20164	MARTINEZ PEREZ ANGEL	305

- General de Almacén.

Es un reporte que muestra por cada artículo las entradas y salidas acumuladas, así como la existencia actual, el precio unitario y las existencias máximas y mínimas, y tiene el siguiente formato:

NOMBRE DE LA DEPENDENCIA						
REPORTE GENERAL DEL ALMACEN						
CLAVE ARTICULO	ENTRADAS	SALIDAS	EXIS.ACTUAL	PRECIO UNITARIO	EXIS. MIN.	EXIS. MAX.
.
.
.
.
.
.
2321001	10	4	6	1,200.00	20	2

- Entradas.

Es un reporte que muestra la información que fué capturada en las notas de entrada, y tiene el siguiente formato:

NOMBRE DE LA DEPENDENCIA						
REPORTE DETALLADO DE ENTRADAS						
NOTA	FECHA	CLAVE ART	DESCRIPCION	UNIDAD	CANTIDAD	PRECIO
.
.
.
123456	12/12/91	2311234	CUADERNOS CUADRO	PIEZA	300	\$ 2,350.00

- Salidas.

Es un reporte que muestra la información que fué capturada en los vales de salida, y se presenta en el siguiente formato:

NOMBRE DE LA DEPENDENCIA					
REPORTE GENERAL DEL ALMACEN					
R.F.C.:					
VALE	FECHA	CLAVE ART	DESCRIPCION	UNIDAD	CANTIDAD
.
.
.
.
123456	12/12/91	2311234	CUADERNOS CUADRO	PIEZA	300

Dentro de las salidas que son las emitidas por pantalla, esta por ejemplo la consulta del catálogo de empleados, que presenta el siguiente formato:

DIRECCION GENERAL DE SERVICIOS DE COMPUTO ADMINISTRATIVO																
SISTEMA DE ALMACEN																
CONSULTAS																
CATALOGOS EXIS. DE ARTICULO CONSUMOS GASTOS M. ENTR / V. SALI ART. EXIS=<MIM DETALLE E/S X AN DETALLE S X RFC FIN DE CONSULTAS	<table border="1"> <thead> <tr> <th>R.F.C.</th> <th>NOMBRE</th> <th>DEPTO</th> </tr> </thead> <tbody> <tr> <td>AAA2590626</td> <td>ALMAZAN AGUILERA JUAN</td> <td>306</td> </tr> <tr> <td>BEPE590823</td> <td>BECCERIL PORRAS LUIS</td> <td>301</td> </tr> <tr> <td>HEDA400612</td> <td>HERNANDEZ ORTEGA A. JOSE</td> <td>301</td> </tr> </tbody> </table>	R.F.C.	NOMBRE	DEPTO	AAA2590626	ALMAZAN AGUILERA JUAN	306	BEPE590823	BECCERIL PORRAS LUIS	301	HEDA400612	HERNANDEZ ORTEGA A. JOSE	301	<table border="1"> <tr> <td>C PARA CONTINUAR O RETURN PARA CONTINUAR</td> <td>_</td> </tr> </table>	C PARA CONTINUAR O RETURN PARA CONTINUAR	_
R.F.C.	NOMBRE	DEPTO														
AAA2590626	ALMAZAN AGUILERA JUAN	306														
BEPE590823	BECCERIL PORRAS LUIS	301														
HEDA400612	HERNANDEZ ORTEGA A. JOSE	301														
C PARA CONTINUAR O RETURN PARA CONTINUAR	_															

Una vez descritas las entradas y salidas del sistema se analizarán los procesos globales para cumplir con los requerimientos antes mencionados.

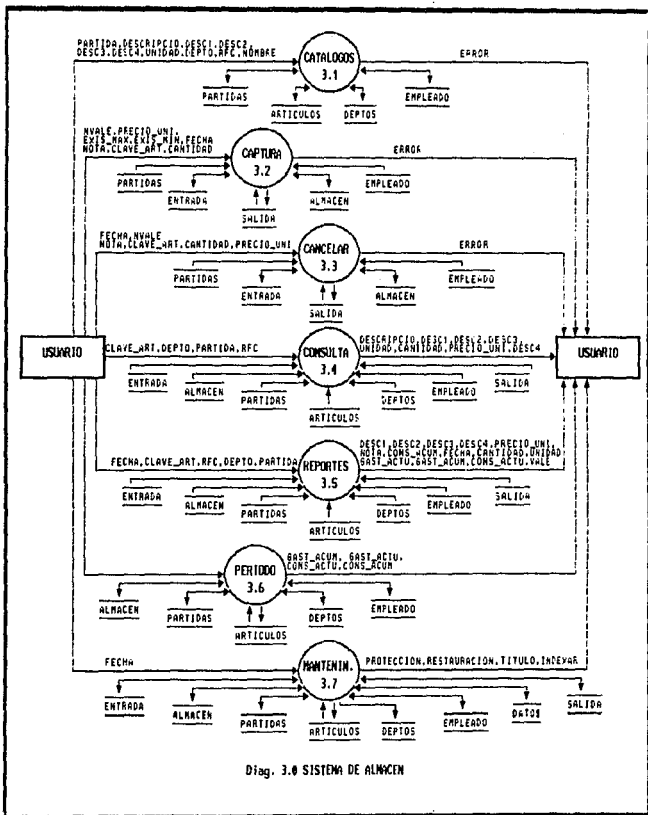
En cuanto a la validación que se realizará de la información cabe hacer notar que ésta debe referirse a las claves que se manejen de los artículos así como verificar que las salidas no sobrepasen a la existencia actual para no tener existencias negativas.

Una vez definidas las entradas y salidas así como los módulos que formarán el sistema, se presentan a continuación los diagramas de flujo de datos.

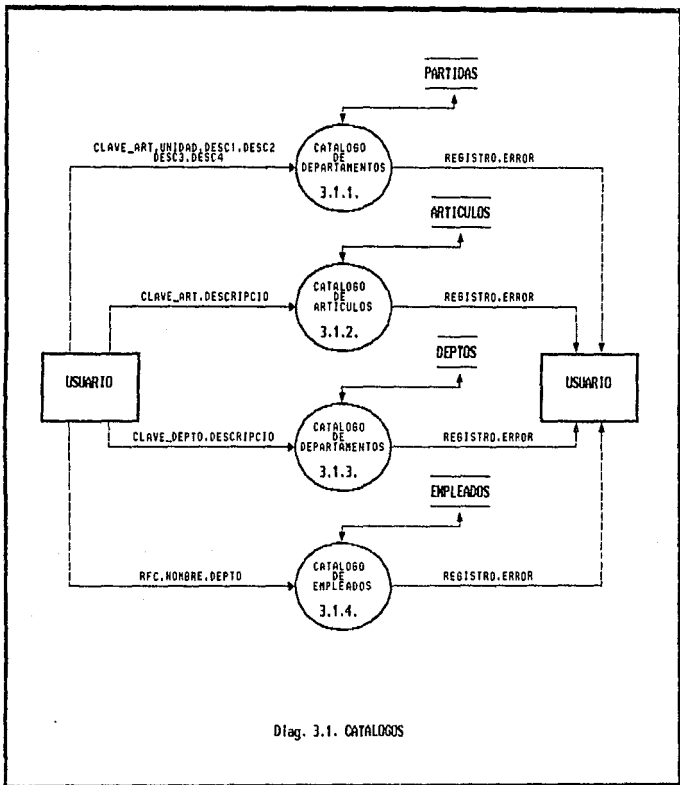
DIAGRAMAS DE FLUJO DE DATOS

DEL

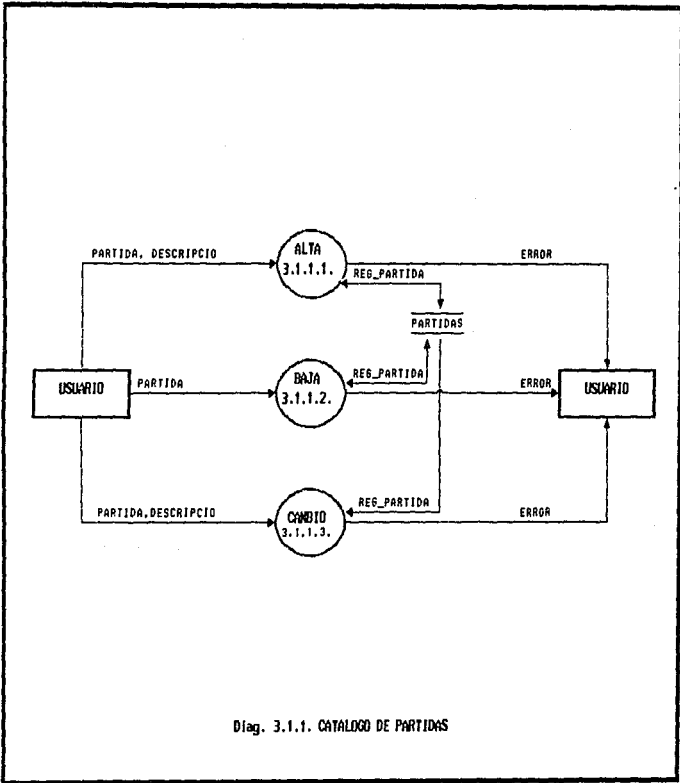
SISTEMA DE ALMACEN



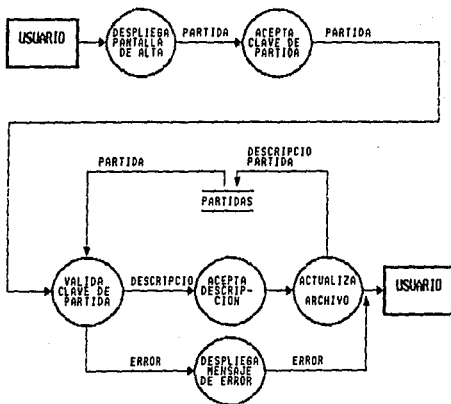
Diag. 3.0 SISTEMA DE ALMACEN



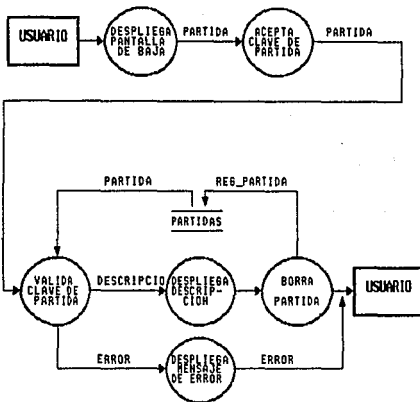
Diag. 3.1. CATALOGOS



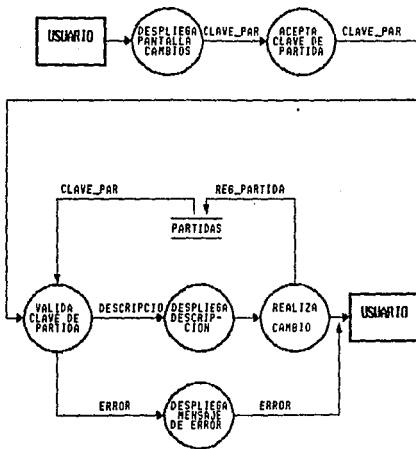
Diag. 3.1.1. CATALOGO DE PARTIDAS



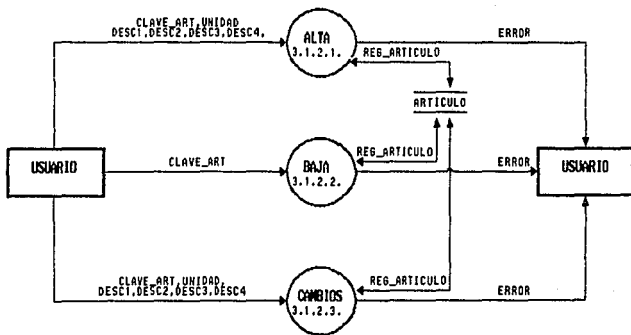
Diag. 3.1.1.1. ALTA DE PARTIDAS



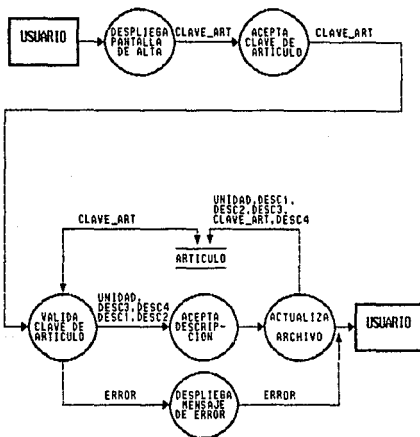
Diag. 3.1.1.2. BAJA DE PARTIDAS



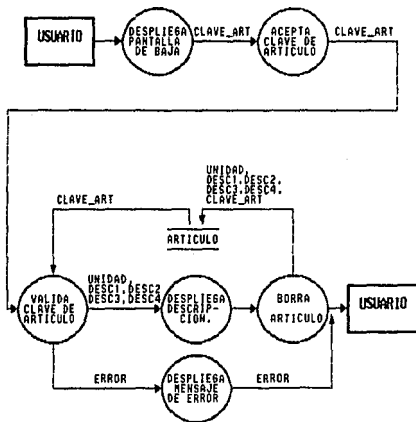
Diag. 3.1.1.3. CAMBIO DE PARTIDAS



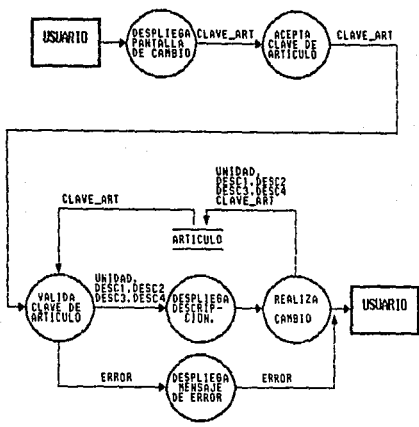
Diag. 3.1.2. CATALOGO DE ARTICULOS



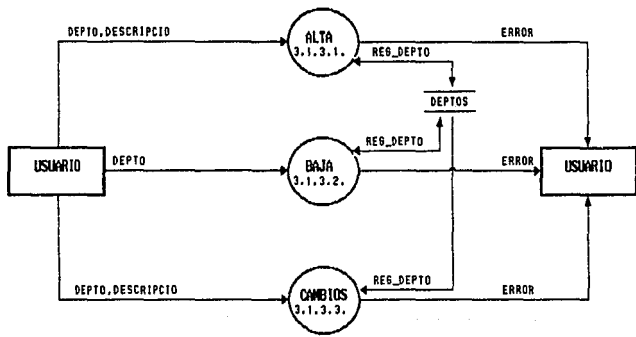
Diag. 3.1.2.1. ALTA DE ARTICULOS



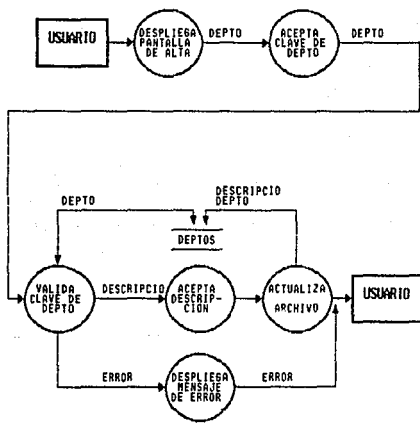
Diag. 3.1.2.2. BAJA DE ARTICULOS



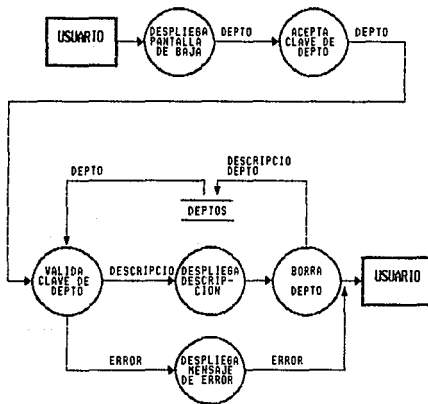
Diag. 3.1.2.3. CAMBIO DE ARTICULOS



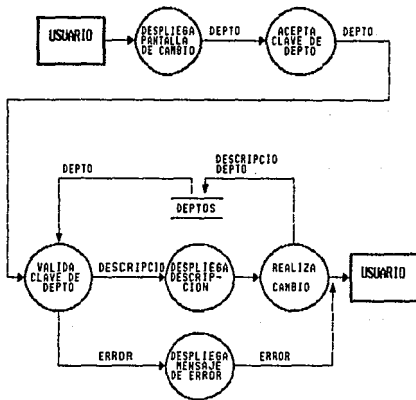
Diag. 3.1.3. CATALOGO DE DEPARTAMENTOS



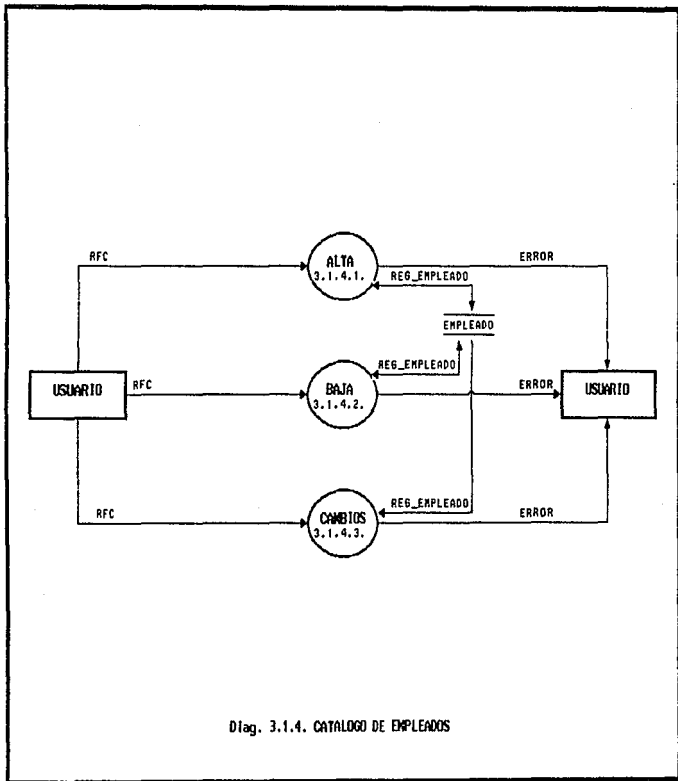
Diag. 3.1.3.1. ALTA DE DEPARTAMENTOS

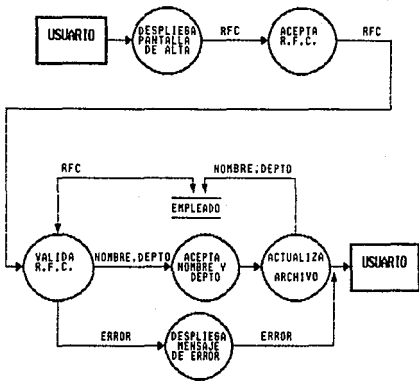


Diag. 3.1.3.2. BAJA DE DEPARTAMENTOS

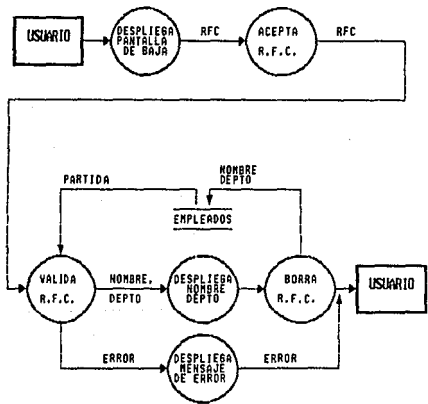


Diag. 3.1.3.3. CAMBIO DE DEPARTAMENTOS

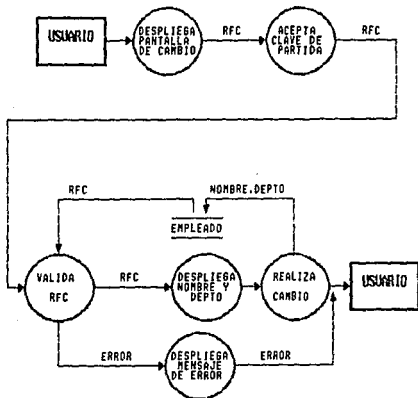




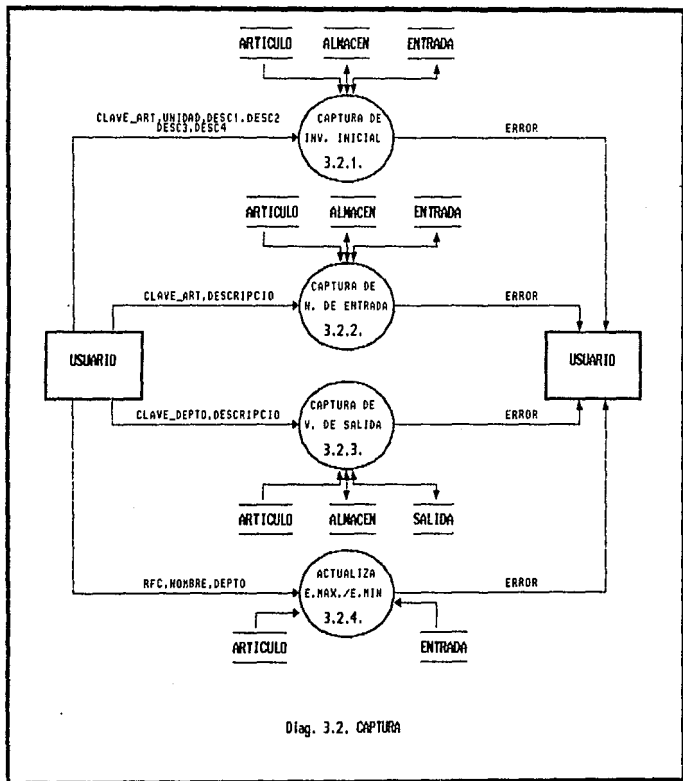
Diag. 3.1.4.1. ALTA DE EMPLEADOS



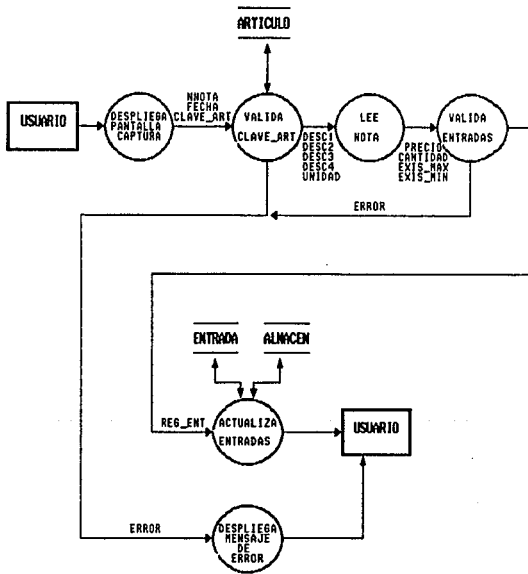
Diag. 3.1.4.2. BAJA DE EMPLEADOS



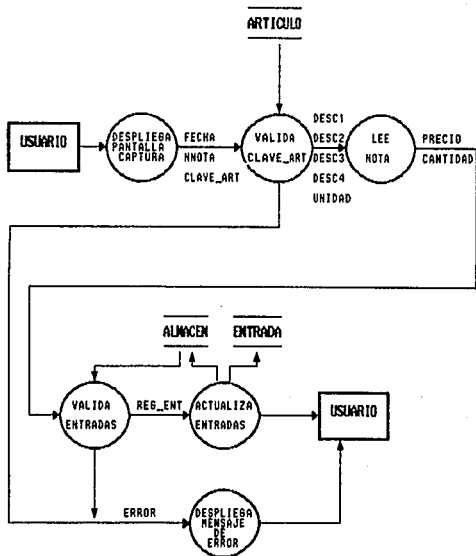
Diag. 3.1.4.3. CAMBIO DE EMPLEADOS



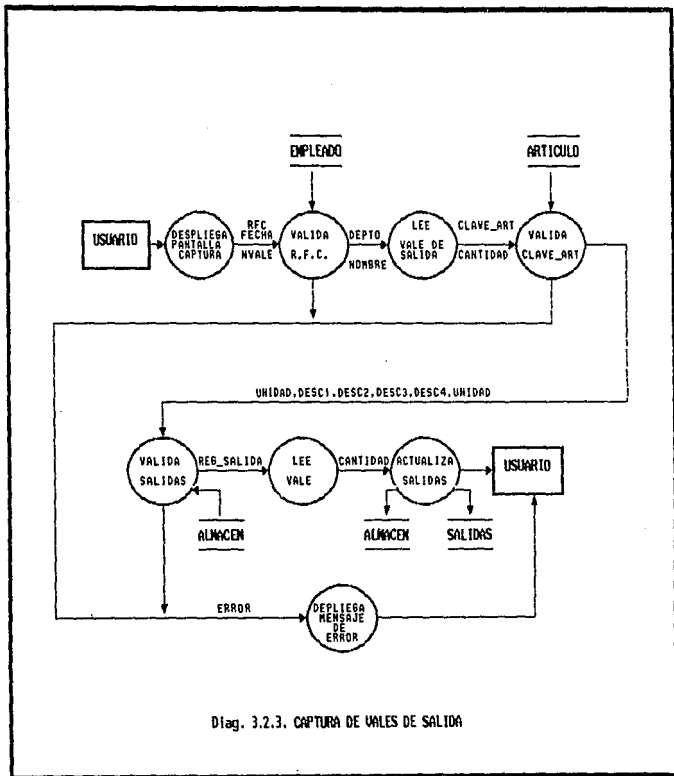
Diag. 3.2. CAPTURA

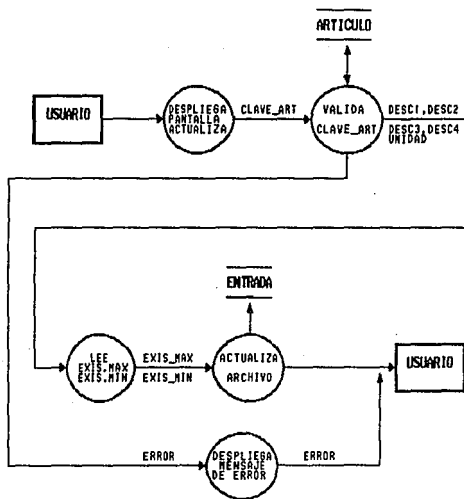


Diag. 3.2.1. CAPTURA DE INVENTARIO INICIAL

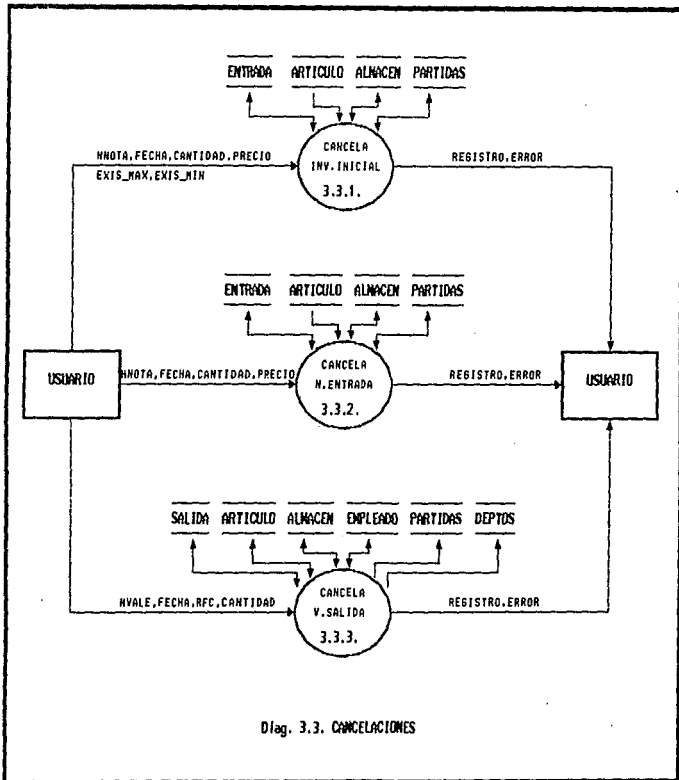


Diag. 3.2.2. CAPTURE DE NOTAS DE ENTRADA

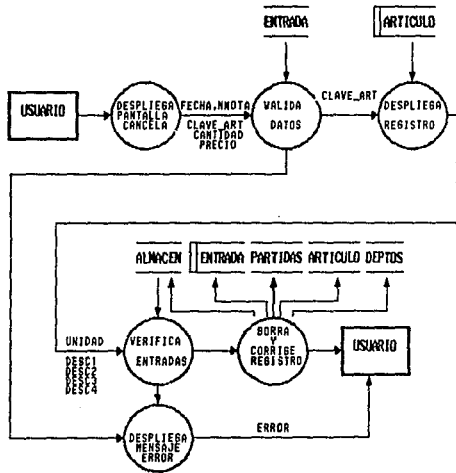




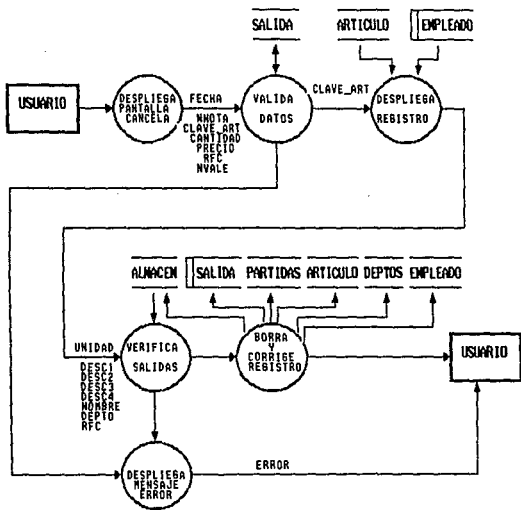
Diag. 3.2.4. ACTUALIZA E. MINIMA / E. MAXIMA



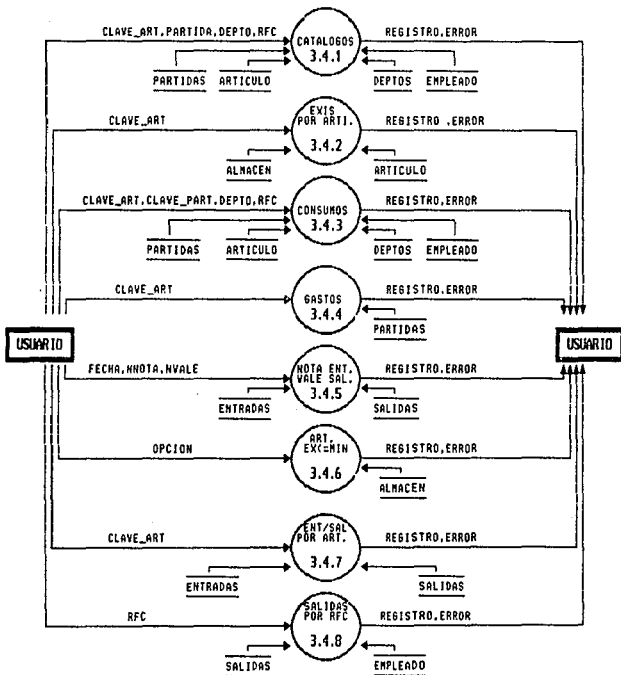
Diag. 3.3. CANCELACIONES



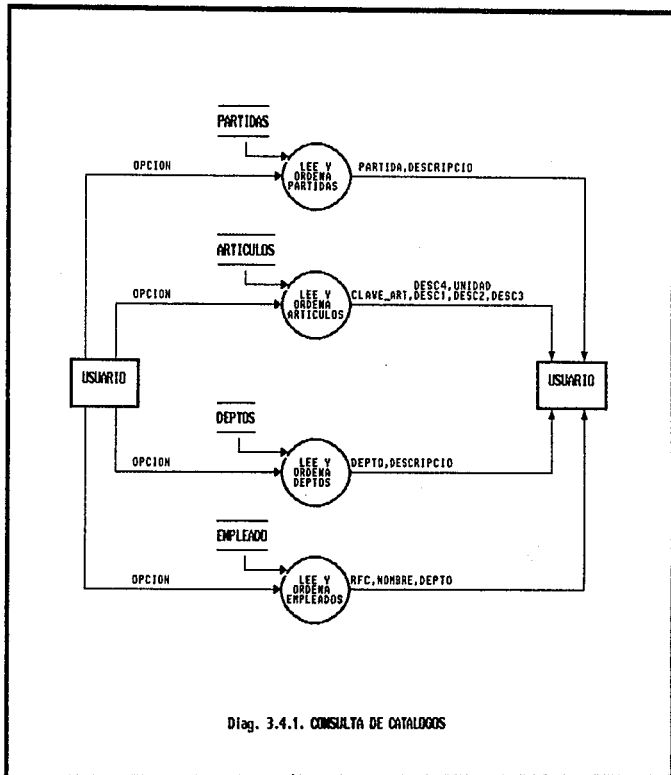
Diag: 3.3.2. CANCELA NOTA DE ENTRADA

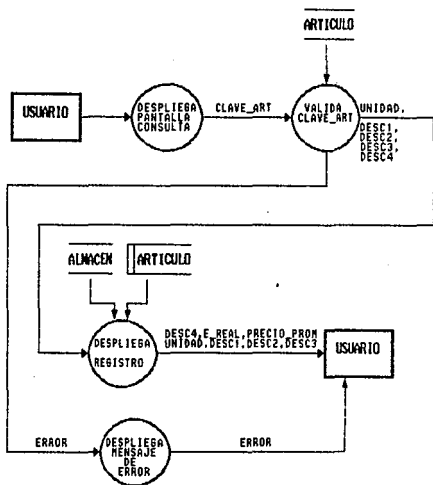


Diag. 3.3.3. CANCELA VALE DE SALIDA

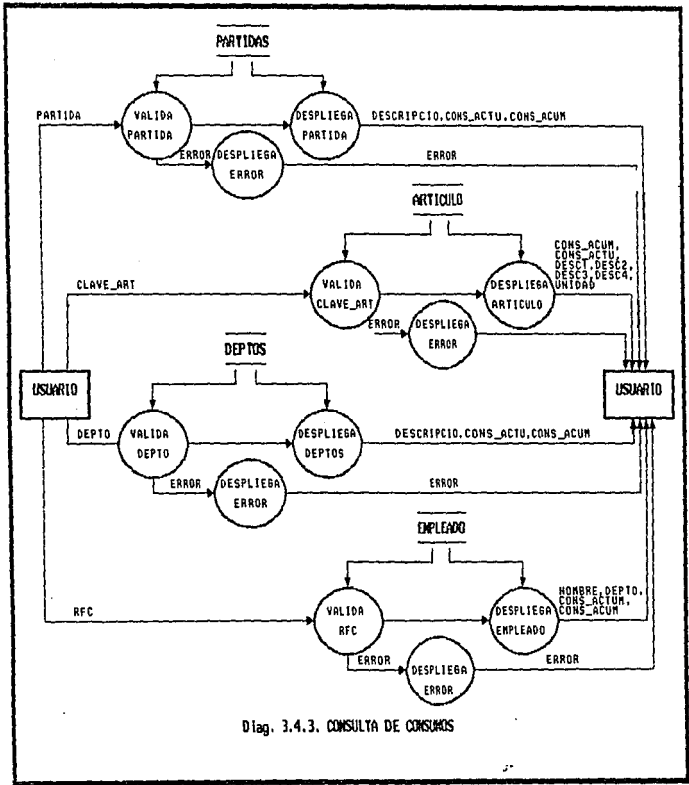


Diag. 3.4. CONSULTAS

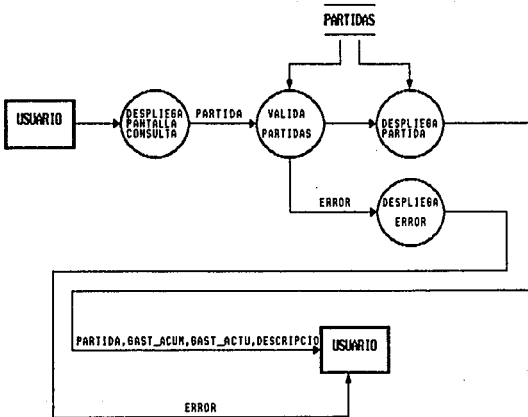




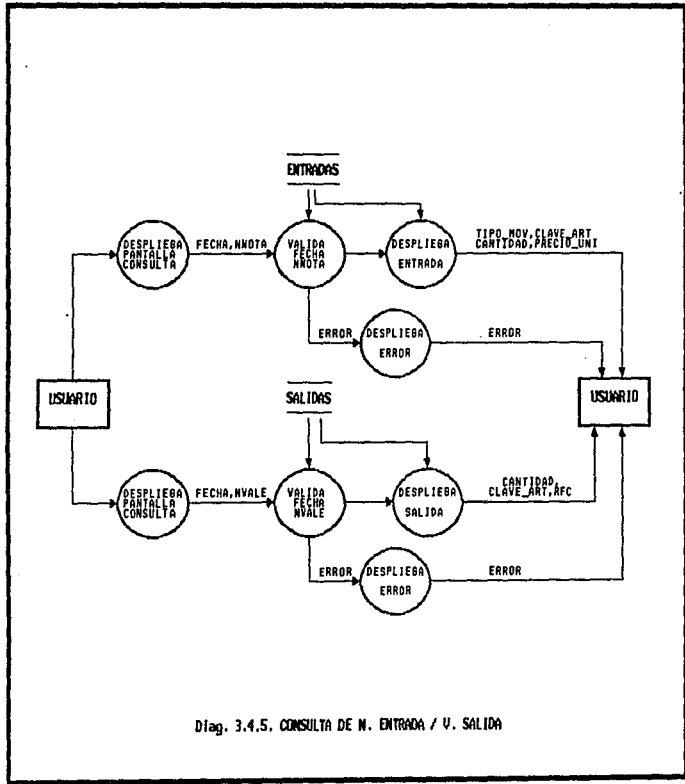
Diag. 3.4.2. CONSULTA DE EXISTENCIA POR ARTICULO



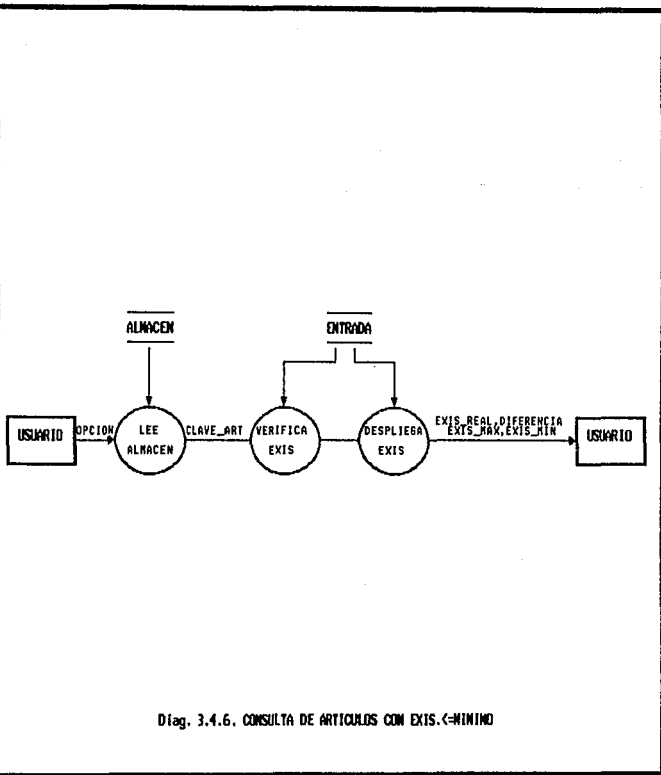
Diag. 3.4.3. CONSULTA DE CONSUMOS



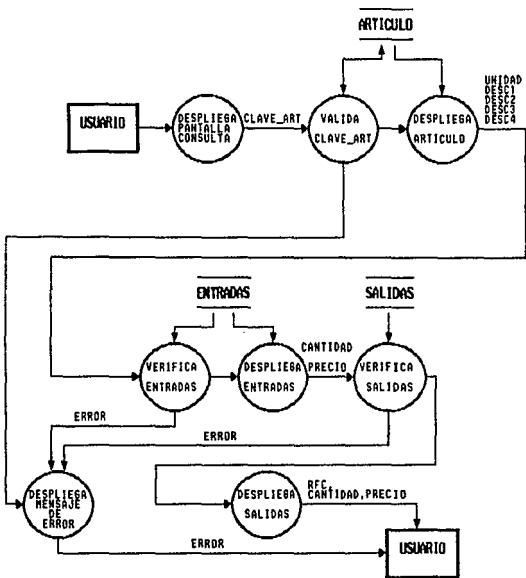
Diag. 3.4.4. CONSULTA DE GASTOS POR PARTIDA



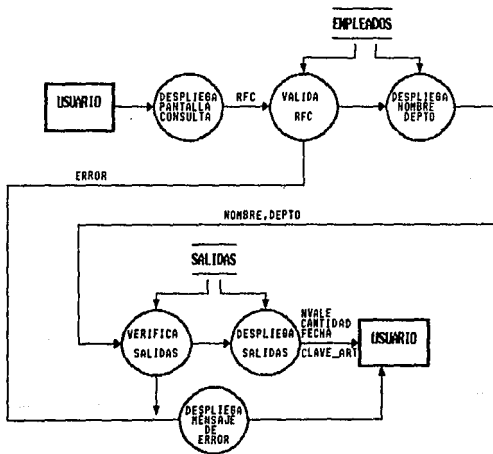
Diag. 3.4.5. CONSULTA DE M. ENTRADA / U. SALIDA



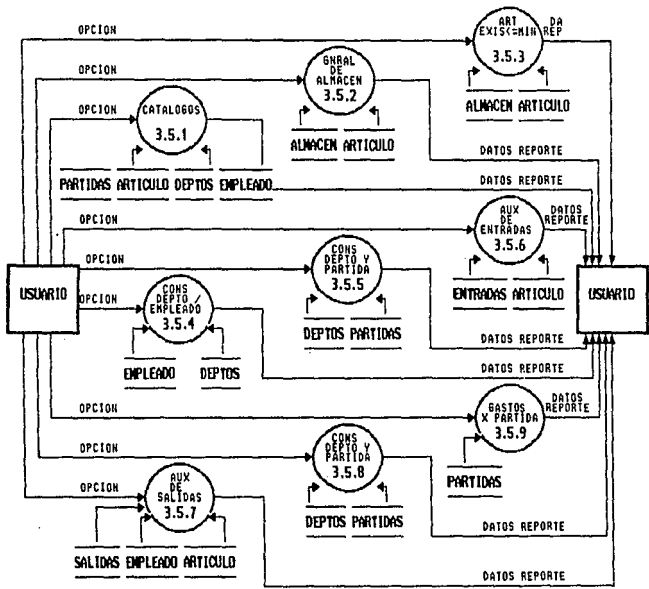
Diag. 3.4.6. CONSULTA DE ARTICULOS CON EXIS.<=MINIMO



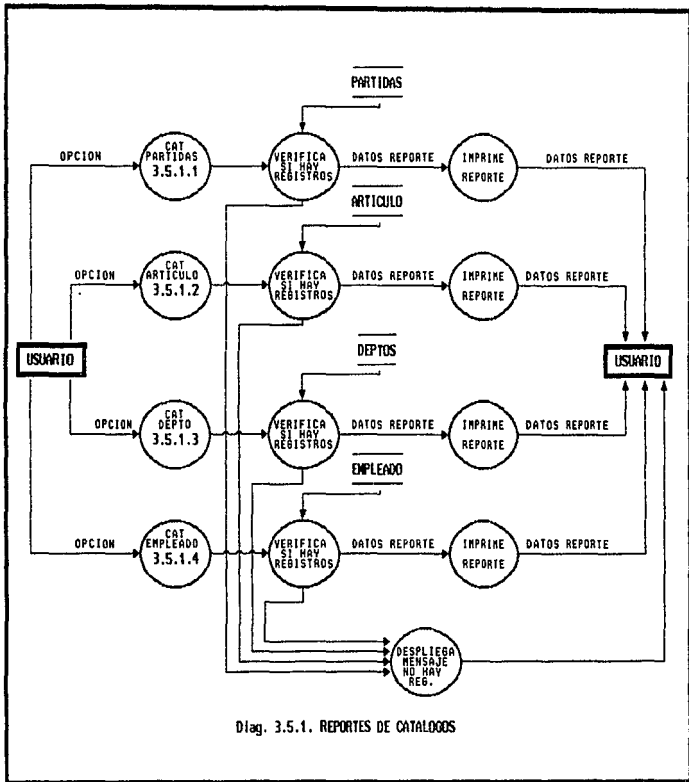
Diag. 3.4.7. CONSULTA DETALLADA DE E/S POR ARTICULO'

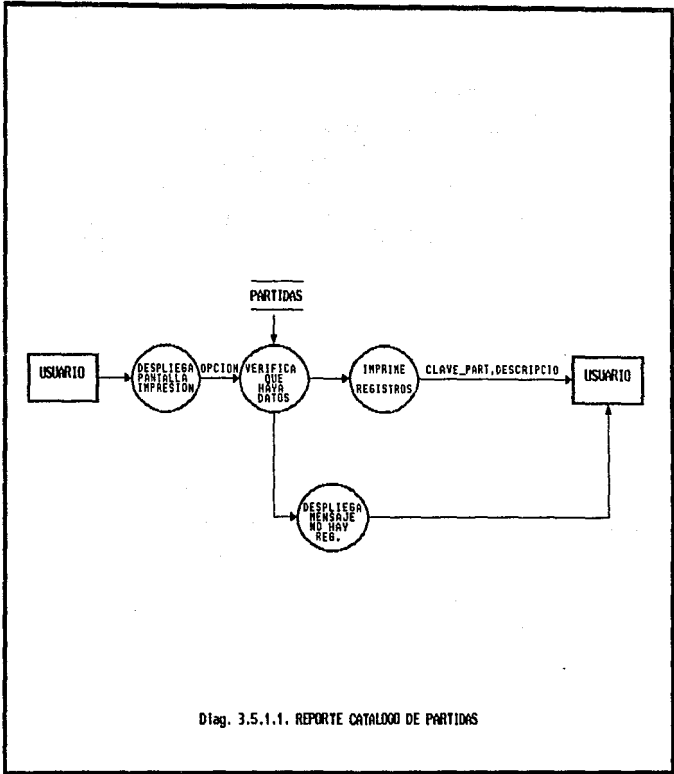


Diag. 3.4.8. CONSULTA DETALLADA DE E/S POR R.F.C.

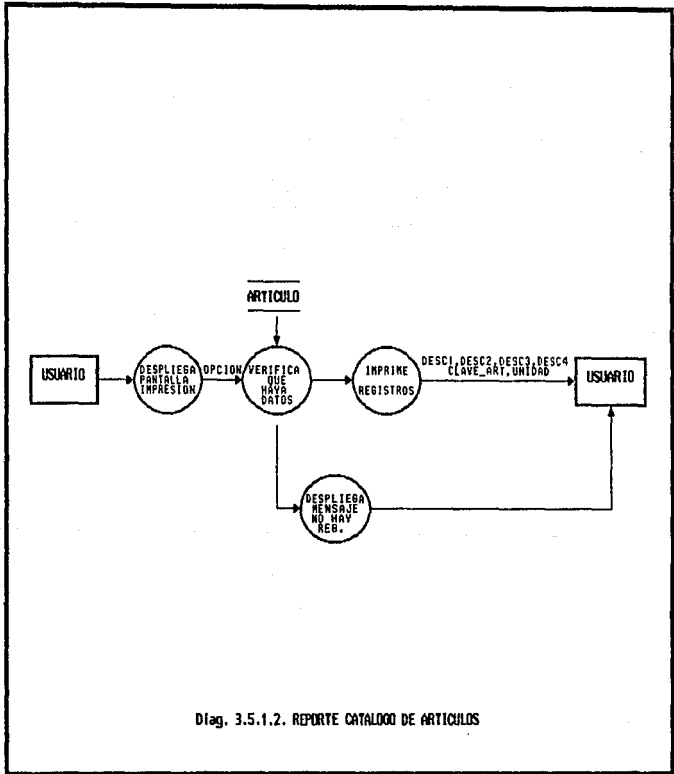


Diag. 3.5. REPORTES

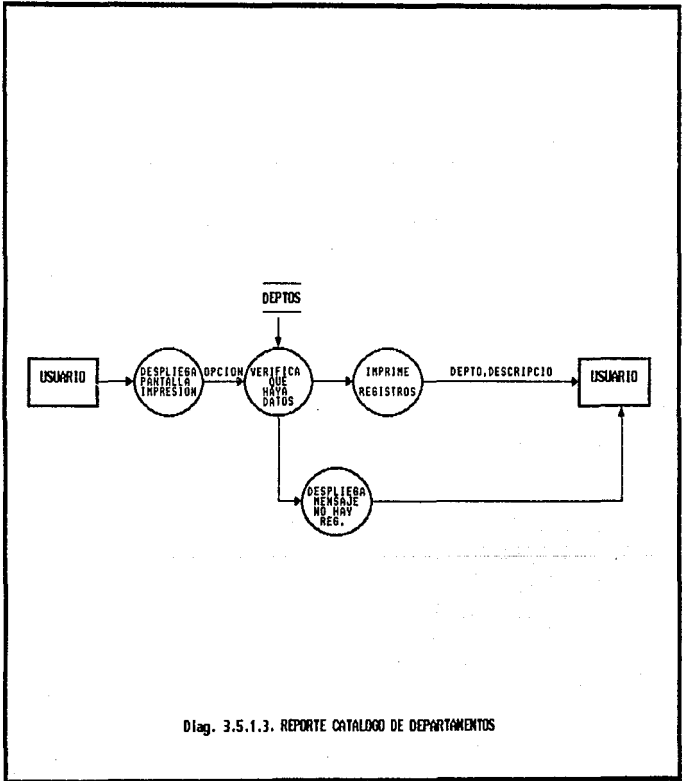




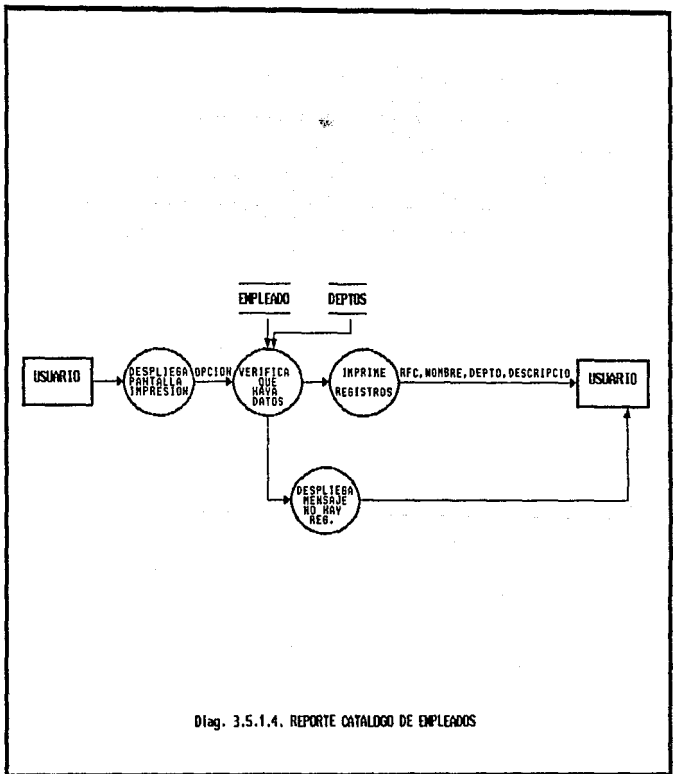
Diag. 3.5.1.1. REPORTE CATALOGO DE PARTIDAS



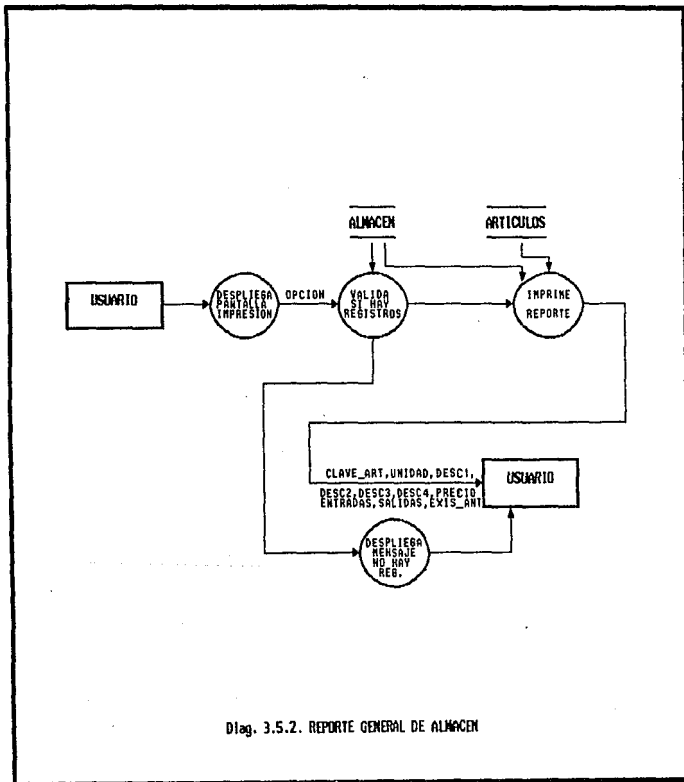
Diag. 3.5.1.2. REPORTE CATALOGO DE ARTICULOS



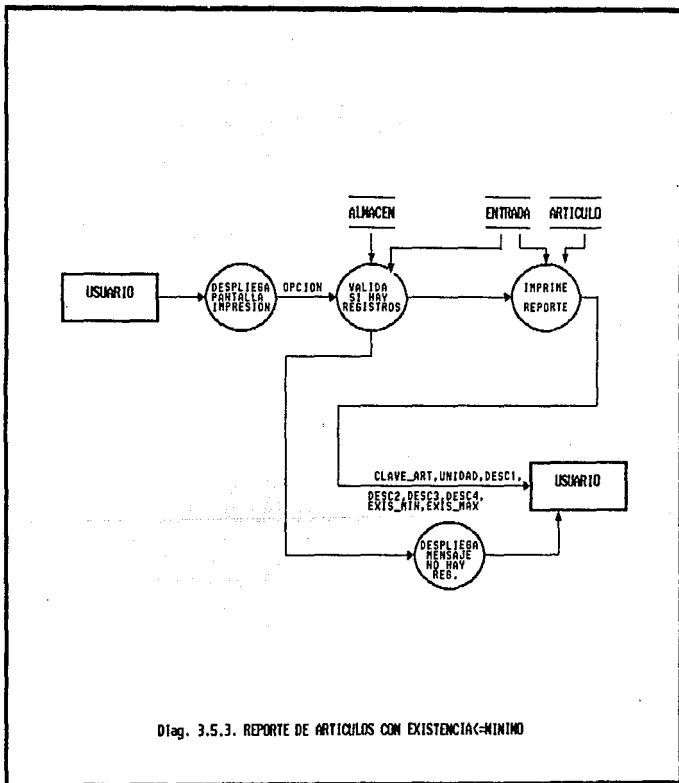
Diag. 3.5.1.3. REPORTE CATALOGO DE DEPARTAMENTOS



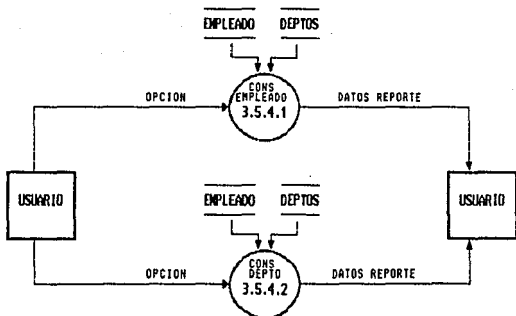
Diag. 3.5.1.4. REPORTE CATALOGO DE EMPLEADOS



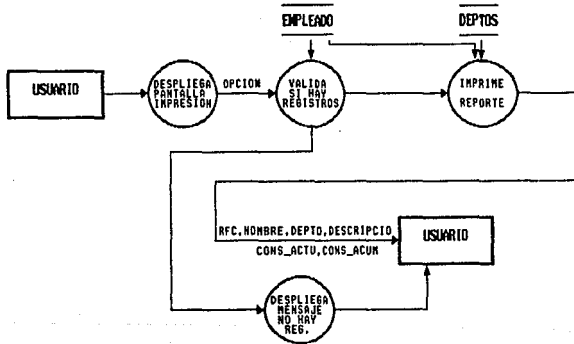
Diag. 3.5.2. REPORTE GENERAL DE ALMACEN



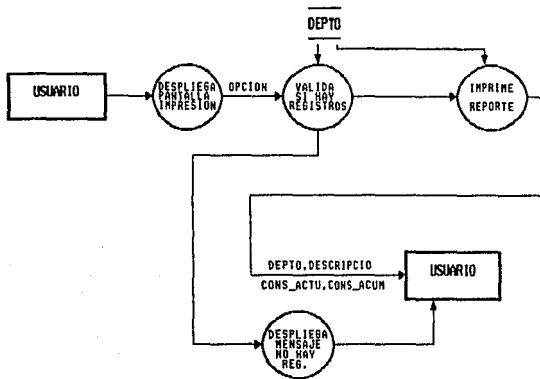
Diag. 3.5.3. REPORTE DE ARTICULOS CON EXISTENCIA<=>MÍNIMO



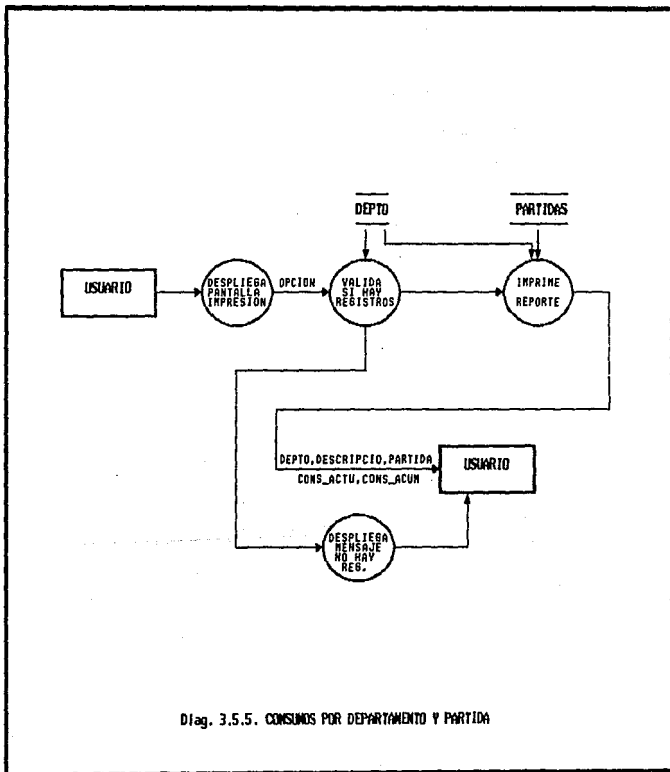
Diag. 3.5.4. REPORTE CONSUMOS POR DEPTO O PARTIDA



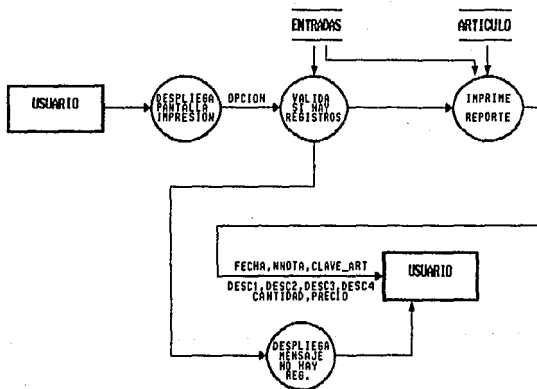
Diag. 3.5.4.1. REPORTE CONSUMOS POR EMPLEADO



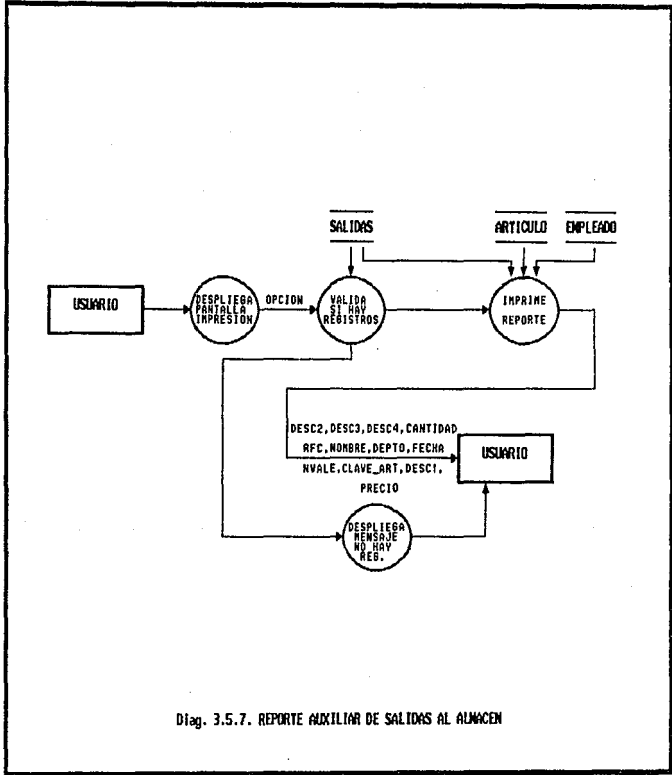
Diag. 3.5.4.2. REPORTE CONSUMOS POR DEPARTAMENTO



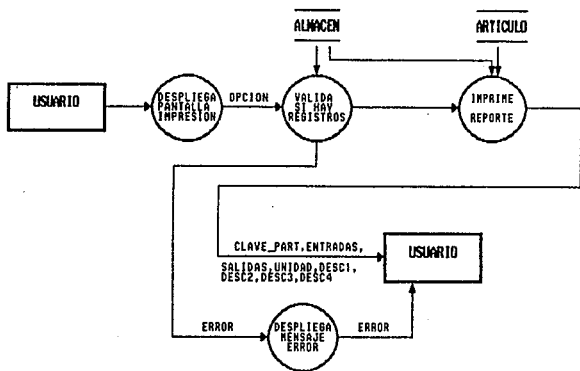
Diag. 3.5.5. CONSUMOS POR DEPARTAMENTO Y PARTIDA



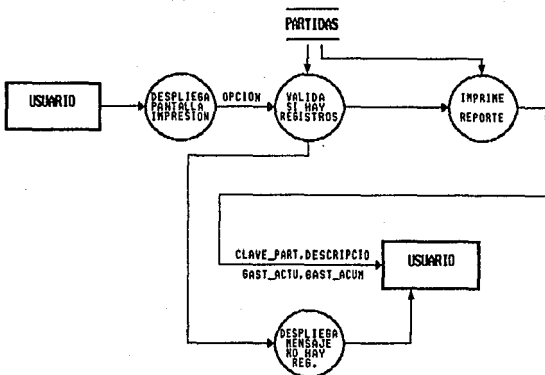
Diag. 3.5.6. REPORTE AUXILIAR DE ENTRADAS AL ALMACEN



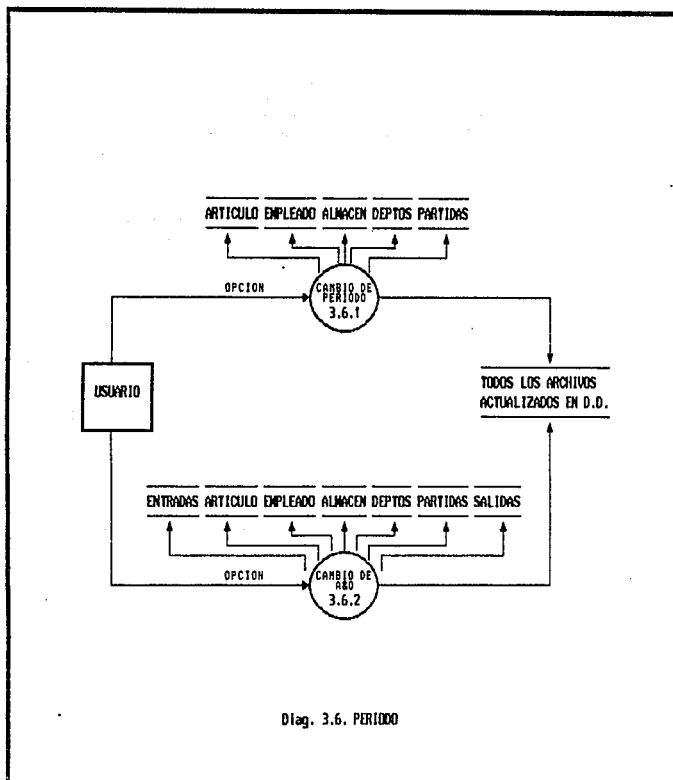
Diag. 3.5.7. REPORTE AUXILIAR DE SALIDAS AL ALMACEN



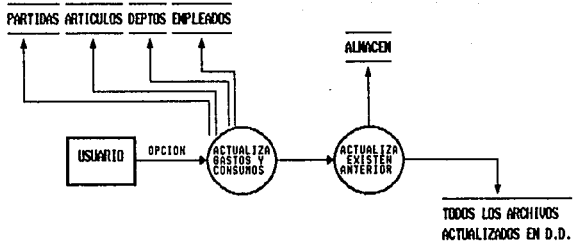
Diag. 3.5.8. REPORTE ACUMULADO DE ENTRADAS Y SALIDAS



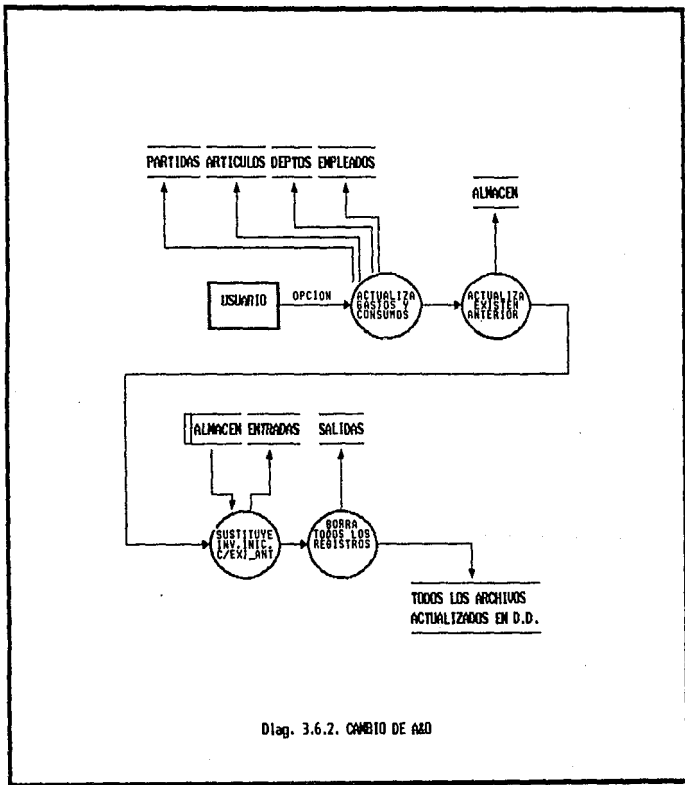
Diag. 3.5.9. REPORTE GASTOS PRESUPUESTALES POR PARTIDA



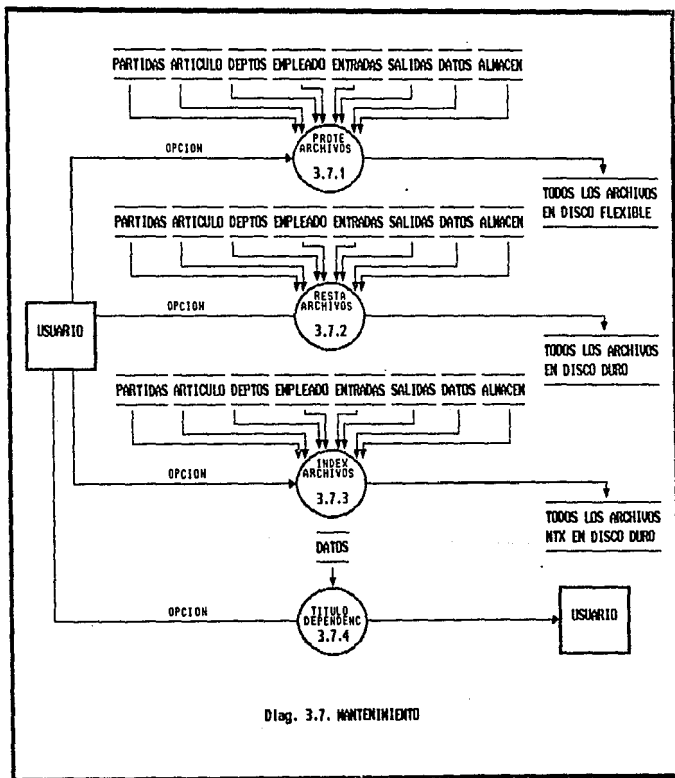
Diag. 3.6. PERIODO



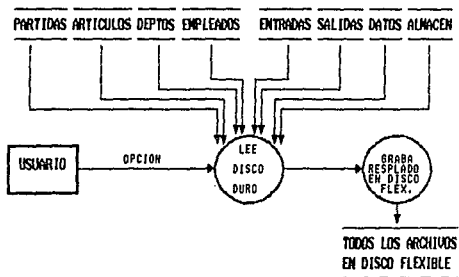
Diag. 3.6.1. CAMBIO DE PERIODO



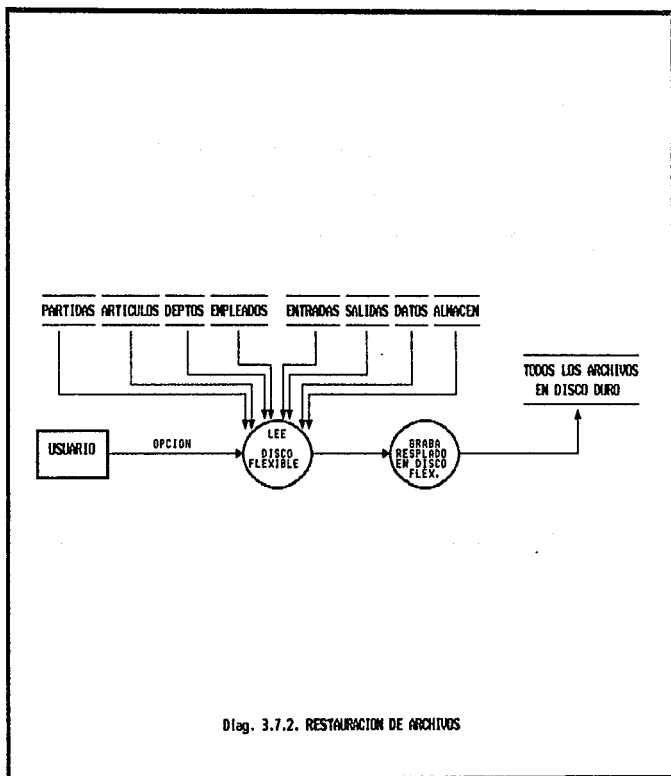
Diag. 3.6.2. CAMBIO DE A&D

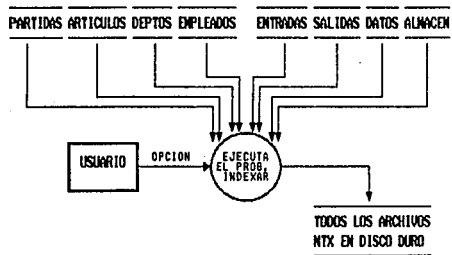


Diag. 3.7. MANTENIMIENTO

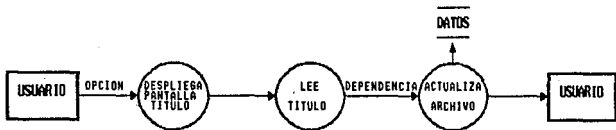


Diag. 3.7.1. PROTECCION DE ARCHIVOS





Diag. 3.7.3. INDEXAR ARCHIVOS



Diag. 3.7.4. TITULO DE LA DEPENDENCIA

3.4.2. DICCIONARIO DE DATOS

Cada flecha de un diagrama de flujo de datos representa uno o más elementos de información. El diccionario de datos es el método utilizado para describir el contenido de los elementos de información y las definiciones de todos los datos mencionados en los D.F.D's; El diccionario esta formado de datos compuestos (datos que pueden ser divididos) y se definen en términos de sus componentes; Los datos elementales (datos que no pueden ser divididos) se definen en términos del significado de cada uno de los valores que puede asumir. Por tanto el diccionario de datos esta formado de definiciones de flujo de datos, archivos (datos almacenados) y datos usados en los procesos (transformaciones). Una de las características fundamentales del diccionario de datos es que no debe existir ambigüedad entre la información.

La notación en un diccionario de datos es la siguiente:

Construcción de datos	Notación	significado
	=	Esta compuesto de
Secuencia	+	y
Selección	[]	Uno u otro
Repetición	{ }	n repeticiones de
	()	datos opcionales

A continuación se presenta el diccionario de datos del sistema en desarrollo:

A) Sección de datos compuestos.

partida=partida+descripción
artículo=clave_art+descripción+unidad
depto=depto+descripción
empleado=rfc+nombre+depto
entrada=fecha+nota+clave_art+precio_uni+cantidad
salida=clave_art+fecha+vale+rfc+cantidad+precio

B) Sección de datos elementales.

clave_art=partida+(dígito)
partida={dígito}
descripción={carácter}
unidad={carácter}
depto={dígito}
rfc={carácter}
nombre={carácter}
fecha=dd+mm+aa
dd=["01"|"02"|"03" |..|"31"]
mm=["01"|"02"|"03" |..|"12"]
aa=["80"|"81"|"82" ..|"99" ..]
nota={dígito}
vale={dígito}
dependenci={carácter}
cantidad={dígito}+"."+(dígito)

```

precio={dígito}+"."+(dígito)
exis_máx={dígito}+"."+(dígito)
exis_mín={dígito}+"."+(dígito)
cons_acum={dígito}+"."+(dígito)
cons_actu={dígito}+"."+(dígito)
gast_acum={dígito}+"."+(dígito)
gast_actu={dígito}+"."+(dígito)
tipo_mov=["I"|"E"]

```

D) Sección de Archivos

```

PARTIDAS=partida+descripción+cons_actu+cons_acum+gast_actu+
          gast_acum

```

```

ARTICULO=clave_Art+descripción+unidad+cons_actu+cons_acum

```

```

DEPTO=depto+descripción+cons_actu+cons_acum

```

```

EMPLEADO=rfc+nombre+depto+cons_actu+cons_acum

```

```

ENTRADA=fecha+nota+clave_art+precio_uni+cantidad+exis_max+
        exis_min+tipo_mov

```

```

SALIDA=clave_art+fecha+vale+rfc+cantidad+precio+depto

```

```

ALMACEN=clave_Art+exis_ante+entradas+salidas+precioprom

```

```

DATOS=dependenci

```

```

AYUDA=campo+ayu+programa

```

3.5. MINIESPECIFICACIONES

Una vez que ha sido representada la información utilizando los DFD's y el Diccionario de datos, debe representarse cada transformación o proceso usando el lenguaje de diseño del programa o proceso (LDP), el cual incorpora construcciones procedimentales

precisas de las funciones representadas dentro de un DFD.

A continuación se presentan cada uno de los procesos representados anteriormente en los DFD's.

PROCESO : CONTROLAR EL MENU PRINCIPAL

OBJETIVO: Seleccionar la función a realizar dependiendo de la opción elegida por el usuario.

SISTEMA DE ALMACEN

REPITE

Desplegar el menú principal
Leer opción

CASO de que la opción sea igual a :

- 1: Actualizar los catálogos
- 2: Capturar y validar los archivos de entradas y salidas
- 3: Corregir las notas de entrada o vales de salida.
- 4: Consultar las entradas y salidas.
- 5: Crear reportes
- 6: Realizar el cierre de período o el cierre de año.
- 7: Realizar mantenimiento a los archivos del sistema.
- 8: Salir del sistema.

HASTA opción igual a 8

FIN DEL SISTEMA.

PROCESO : MANTENIMIENTO DE CATALOGOS

OBJETIVO: Seleccionar la función a realizar dependiendo de la opción elegida por el usuario.

REPITE

Desplegar el menú catálogos
Leer opción

CASO de que la opción sea igual a :

- 1: Partidas
- 2: Artículos
- 3: Departamentos
- 4: Empleados
- 5: Fin

HASTA opción igual a 5

FIN DE MANTENIMIENTO DE CATALOGOS.

PROCESO : MANTENIMIENTO DE CATALOGO DE PARTIDAS

OBJETIVO: Seleccionar la función a realizar dependiendo de la opción elegida por el usuario.

REPITE

Desplegar el menú de mantenimiento de partidas
Leer opción

CASO de que la opción sea igual a :

- 1: Alta
- 2: Baja
- 3: Consulta
- 4: Fin

HASTA opción igual a 4

FIN DE MANTENIMIENTO DE CATALOGO DE PARTIDAS.

PROCESO : REALIZA ALTA DE PARTIDAS

OBJETIVO : Dar de alta registros de partidas en la base de datos.

REPITE

Desplegar pantalla de altas de partidas
Leer clave
Valida clave
Lee descripción
Actualiza base de datos

HASTA clave = espacios

FIN DE REALIZA ALTA DE PARTIDAS.

PROCESO : REALIZA BAJAS DE PARTIDAS

OBJETIVO : Dar de baja registros de partidas en la base de datos.

REPITE

Desplegar pantalla de bajas de partidas
Leer clave
Valida clave
Muestra descripción
Borra el registro en la base de datos
HASTA clave = espacios

FIN DE REALIZA BAJA DE PARTIDAS.

PROCESO : REALIZA CAMBIOS EN PARTIDAS.

OBJETIVO : Modificar registros de partidas en la base de datos.

REPITE

Desplegar pantalla de baja de partidas
Leer clave
Valida clave
Muestra descripción
Sustituye los datos anteriores por los nuevos en la B.D.
HASTA clave = espacios

FIN DE REALIZA CAMBIOS DE PARTIDAS.

PROCESO : MANTENIMIENTO DE CATALOGO DE ARTICULOS

OBJETIVO: Seleccionar la función a realizar dependiendo de la opción elegida por el usuario.

REPITE

Desplegar el menú de mantenimiento de catálogo de artículos
Leer opción

CASO de que la opción sea igual a :
1: Alta
2: Baja
3: Consulta
4: Fin

HASTA opción igual a 4

FIN DE MANTENIMIENTO DE CATALOGO DE ARTICULOS.

PROCESO : REALIZA ALTA DE ARTICULOS

OBJETIVO : Dar de alta registros de artículos en la base de datos.

REPITE

Desplegar pantalla de alta de artículos
Leer clave
Valida clave
Lee unidad y descripción
Actualiza base de datos

HASTA clave = espacios

FIN DE REALIZA ALTA DE ARTICULOS.

PROCESO : REALIZA BAJA DE ARTICULOS

OBJETIVO : Dar de baja registros de artículos en la base de datos.

REPITE

Desplegar pantalla de baja de artículos
Leer clave
Valida clave
Muestra unidad y descripción
Borra el registro en la base de datos

HASTA clave = espacios

FIN DE REALIZA BAJA DE ARTICULOS.

PROCESO : REALIZA CAMBIOS EN ARTICULOS

OBJETIVO : Modificar registros de artículos en la base de datos.

REPITE

Desplegar pantalla de cambios de artículos
Leer clave
Valida clave
Muestra descripción
Sustituir los datos anteriores por los nuevos en la B.D.
HASTA clave = espacios

FIN DE REALIZA CAMBIOS EN ARTICULOS.

PROCESO : MANTENIMIENTO DE CATALOGO DE DEPARTAMENTOS

OBJETIVO: Seleccionar la función a realizar dependiendo de la opción elegida por el usuario.

REPITE

Desplegar el menú de mantenimiento de catálogo de departamentos
Leer opción

CASO de que la opción sea igual a :

- 1: Alta
- 2: Baja
- 3: Consulta
- 4: Fin

HASTA opción igual a 4

FIN DE MANTENIMIENTO DE CATALOGO DE DEPARTAMENTOS.

PROCESO : REALIZA ALTA DE DEPARTAMENTOS

OBJETIVO : Dar de alta registros de departamentos en la base de datos.

REPITE

Desplegar pantalla de alta de departamentos
Leer clave
Valida clave
Lee descripción
Actualiza base de datos

HASTA clave = espacios

FIN DE REALIZA ALTA DE DEPARTAMENTOS.

PROCESO : REALIZA BAJA DE DEPARTAMENTOS

OBJETIVO : Dar de baja registros de departamentos en la base de datos.

REPITE

Desplegar pantalla de baja de departamentos
Leer clave
Valida clave
Muestra descripción
Borra el registro en la base de datos

HASTA clave = espacios

FIN DE REALIZA BAJA DE DEPARTAMENTOS.

PROCESO : REALIZA CAMBIOS EN DEPARTAMENTOS

OBJETIVO : Modificar registros de departamentos en la base de datos.

REPITE

Desplegar pantalla de cambios de departamentos
Leer clave
Valida clave
Muestra descripción
Sustituir los datos anteriores por los nuevos en la B.D.

HASTA clave = espacios

FIN DE REALIZA CAMBIOS EN DEPARTAMENTOS.

PROCESO : MANTENIMIENTO DE CATALOGO DE EMPLEADOS

OBJETIVO: Seleccionar la función a realizar dependiendo de la opción elegida por el usuario.

REPITE

Desplegar el menú de catálogos
Leer opción

CASO de que la opción sea igual a :

- 1: Alta
- 2: Baja
- 3: Consulta
- 4: Fin

HASTA opción igual a 4

FIN DE MANTENIMIENTO DE CATALOGO DE EMPLEADOS.

PROCESO : REALIZA ALTA DE EMPLEADOS

OBJETIVO : Dar de alta registros de empleados en la base de datos.

REPITE

Desplegar pantalla de alta de empleados
Leer rfc
Valida rfc
Lee nombre y clave de departamento
Actualiza base de datos

HASTA rfc = espacios

FIN DE REALIZA ALTA DE EMPLEADOS.

PROCESO : REALIZA BAJA DE EMPLEADOS

OBJETIVO : Dar de baja registros de empleados en la base de datos.

REPITE

Desplegar pantalla de baja de empleados
Leer rfc
Valida rfc
Muestra nombre y clave de departamento
Borra el registro en la base de datos

HASTA clave = espacios

FIN DE REALIZA BAJA DE EMPLEADOS.

PROCESO : REALIZA CAMBIOS EN EMPLEADOS

OBJETIVO : Modificar registros de empleados en la base de datos.

REPITE

Desplegar pantalla de cambios de empleados
Leer rfc
Valida rfc
Muestra nombre y clave de depto
Sustituir los datos anteriores por los nuevos en la B.D.

HASTA clave = espacios

FIN DE REALIZA CAMBIOS EN EMPLEADOS.

PROCESO : CAPTURA

OBJETIVO: Seleccionar la función a realizar dependiendo de la opción elegida por el usuario.

REPITE

Desplegar el menú de captura
Leer opción

CASO de que la opción sea igual a :

- 1: Inventario inicial
- 2: Nota de entrada
- 3: Vale de salida
- 4: Actualiza Máximos y Mínimos
- 5: Fin

HASTA opción igual a 5

FIN DE CAPTURA

PROCESO : REALIZA CAPTURA DE INVENTARIO INICIAL

OBJETIVO : Capturar un registro de inventario inicial.

MIENTRAS No.nota <> espacios y clave de artículo <> espacios
Desplegar pantalla captura de inventario inicial

MIENTRAS clave de artículo ya exista en el archivo
Lee fecha, No.nota y clave de artículo
Valida clave de artículo

FIN MIENTRAS

MIENTRAS precio=0 y cantidad=0
Lee precio y cantidad
Valida precio y cantidad

FIN MIENTRAS

Despliega unidad y descripción
Lee existencia máxima y existencia mínima

FIN MIENTRAS

FIN DE REALIZA CAPTURA DE INVENTARIO INICIAL.

PROCESO : REALIZA CAPTURA DE NOTA DE ENTRADA

OBJETIVO : Capturar un registro de notas de entrada.

MIENTRAS No.nota = espacios y fecha= espacios o clave de
artículo = espacios
Desplegar pantalla captura nota de entrada

MIENTRAS fecha= espacios y No.nota=espacios
Lee fecha, No.nota
Valida fecha y No.nota

FIN MIENTRAS

MIENTRAS clave de artículo = espacios
Lee clave de artículo
Valida clave de artículo

MIENTRAS clave de artículo exista en el catálogo
Despliega unidad y descripción
Lee precio unitario y cantidad

FIN MIENTRAS

FIN MIENTRAS

FIN MIENTRAS

FIN DE REALIZA CAPTURA DE NOTA DE ENTRADA.

PROCESO : REALIZA CAPTURA DE VALE DE SALIDA

OBJETIVO : Capturar un registro de vales de salida.

MIENTRAS No.vale = espacios y fecha= espacios y rfc=espacios
o clave de artículo = espacios
Desplegar pantalla captura vale de salida

MIENTRAS fecha= espacios , No.vale =espacios y
rfc=espacios
Lee fecha,No.nota
Valida fecha y No.nota

FIN MIENTRAS

MIENTRAS rfc = espacios
Lee rfc
Valida rfc

MIENTRAS rfc exista en el catálogo
Despliega nombre y clave de departamento
Lee cantidad

FIN MIENTRAS

MIENTRAS clave de artículo = espacios
Lee clave de artículo
Valida clave de artículo

MIENTRAS clave de artículo exista en el catálogo
Despliega unidad y descripción
Lee precio unitario y cantidad

FIN MIENTRAS

FIN MIENTRAS

FIN MIENTRAS

FIN MIENTRAS

FIN DE REALIZA CAPTURA DE VALES DE SALIDA.

PROCESO : REALIZA ACTUALIZACIÓN DE EXIS. MÁX Y EXIS. MÍN

OBJETIVO : Actualizar un registro de inventario inicial.

MIENTRAS No.nota <> espacios y clave de artículo <> espacios
Desplegar pantalla actualización de exis.máx / exis.mín.
Leer Clave de artículo
Leer precio y cantidad
Leer exis: máx y exis. mín.

MIENTRAS exista el registro en el archivo
Desplegar unidad
Desplegar descripción

FIN MIENTRAS

Actualiza el registro en el archivo de entradas
FIN MIENTRAS

FIN DE REALIZA ACTUALIZACION DE EXISTENCIAS MAX Y MIN.

PROCESO : CANCELACIONES

OBJETIVO: Seleccionar la función a realizar dependiendo de la opción elegida por el usuario.

REPITE

Desplegar el menú de cancelaciones
Leer opción

CASO de que la opción sea igual a :

- 1: Inventario inicial
- 2: Nota de entrada
- 3: Vale de salida
- 4: Fin

HASTA opción igual a 4

FIN DE CANCELACIONES

PROCESO : CANCELACION DE INVENTARIO INICIAL

OBJETIVO: Borrar un registro de inventario inicial

Desplegar pantalla de cancelaciones
Leer datos de la nota de inv. inicial a cancelar
Valida No.nota, fecha, clave de artículo, cantidad, precio y
 exis.máx, exis.mín.

SI los datos de la nota son correctos
 Borra registro de la base de datos

EN CASO CONTRARIO

 Muestra mensaje de error

FIN DE CANCELACION DE INVENTARIO INICIAL.

PROCESO : CANCELACION DE NOTA DE ENTRADA

OBJETIVO: Borrar un registro de nota de entrada

Desplegar pantalla de cancelaciones
Leer datos de la nota de entrada a cancelar
Valida No.nota, fecha, clave de artículo, cantidad y precio

SI los datos de la nota son correctos
 Borra registro de la base de datos

EN CASO CONTRARIO

 Muestra mensaje de error

FIN DE CANCELACION DE NOTA DE ENTRADA.

PROCESO : CANCELACION DE VALE DE SALIDA

OBJETIVO: Borrar un registro de vale de salida

Desplegar pantalla de cancelaciones
Leer datos del vale de salida a cancelar
Valida No.vale, fecha, clave de artículo y cantidad.

SI los datos del vale son correctos
 Borra registro de la base de datos

EN CASO CONTRARIO

 Muestra mensaje de error

FIN DE CANCELACION DE VALE DE SALIDA.

PROCESO : CONSULTAS

OBJETIVO: Seleccionar la función a realizar dependiendo de la opción elegida por el usuario.

Desplegar el menú de consultas
Leer opción

REPITE

CASO de que la opción sea igual a :

- 1: Catálogos
- 2: Existencia de artículo
- 3: Consumos
- 4: Gastos por partida
- 5: Nota de entrada o Vale de salida
- 6: Artículos con exist. <= exist. mínimo
- 7: Detalle de entradas y salidas por artículo
- 8: Detalle de salidas por empleado
- 9: Fin

HASTA opción igual a 9

FIN DE CONSULTAS

PROCESO : CONSULTA DE CATALOGOS

OBJETIVO: Seleccionar la función a realizar dependiendo de la opción elegida por el usuario.

Desplegar el menú de consultas
Leer opción

REPITE

CASO de que la opción sea igual a :

- 1: Lee y ordena Partidas
despliega clave y descripción
- 2: Lee y ordena artículos
despliega clave, unidad y descripción
- 3: Lee y ordena departamentos
despliega clave y descripción
- 4: Lee y ordena empleados
despliega rfc, nombre y clave de departamento
- 5: Fin

HASTA opción igual a 5

FIN DE CONSULTA DE CATALOGOS.

PROCESO : CONSULTA DE EXISTENCIA DE ARTICULOS

OBJETIVO: Consultar la existencia real en almacén de un artículo.

Lee y ordena almacén

MIENTRAS clave artículo = espacios

despliega clave y descripción
despliega existencia real, precio promedio

FIN MIENTRAS

FIN DE CONSULTA DE EXISTENCIA DE ARTICULOS.

PROCESO : CONSULTA DE ENTRADAS Y SALIDAS POR ARTICULO

OBJETIVO: Consultar los movimientos registrados de un artículo en el almacén.

MIENTRAS clave artículo = espacios

Lee clave de artículo
Valida clave de artículo

FIN MIENTRAS

Despliega clave y descripción
Lee y ordena entradas
Despliega no. nota, fecha, cantidad y precio
Lee y ordena salidas
Despliega no. vale, fecha, rfc y cantidad

FIN DE CONSULTA DE ENTRADAS Y SALIDAS POR ARTICULO.

PROCESO : CONSULTA DE SALIDAS POR EMPLEADO

OBJETIVO: Consultar las salidas al almacén registradas por un empleado en un periodo de tiempo.

MIENTRAS rfc = espacios

Lee rfc
Lee fechas del periodo a consultar
Valida rfc

FIN MIENTRAS

Despliega nombre y departamento
Lee y ordena salidas
Despliega no. vale, fecha y cantidad

FIN DE CONSULTA DE SALIDAS POR EMPLEADO.

PROCESO : CONSULTA DE NOTAS DE ENTRADA O VALES DE SALIDA

OBJETIVO: Seleccionar la función a realizar dependiendo de la opción elegida por el usuario.

REPITE

Desplegar el menú de consultas

Leer opción

CASO de que la opción sea igual a :

1: **MIENTRAS** fecha = espacios y nota = espacios

Lee fecha y No. de nota

Valida que exista el registro

FIN MIENTRAS

Despliega clave de artículo, cantidad, precio .

2: **MIENTRAS** fecha = espacios y vale = espacios

Lee fecha y No. de vale

Valida que exista el registro

FIN MIENTRAS

Despliega clave de artículo,cantidad.

3: **FIN**

HASTA opción igual a 3

FIN DE CONSULTA DE N.ENTRADA/V.SALIDA.

PROCESO : CONSULTA DE ARTICULOS CON EXISTENCIA <= EXIS.MÍN

OBJETIVO: Consultar los artículos cuya existencia real sea menor a la existencia mínima especificada en el inv. inicial.

Lee y ordena almacén

MIENTRAS clave artículo = espacios

Despliega clave y descripción

Despliega existencia real, precio promedio

FIN MIENTRAS

FIN DE CONSULTA DE ARTICULOS CON EXISTENCIA <= EXIS.MÍN.

PROCESO : REPORTE

OBJETIVO: Seleccionar la función a realizar dependiendo de la opción elegida por el usuario.

REPITE

Desplegar el menú de reportes
Leer opción

CASO de que la opción sea igual a :

- 1: Catálogos
- 2: General de almacén
- 3: Artículos con existencia menor o igual a su mínimo
- 4: Consumos por empleado o departamento
- 5: Consumos por departamento y partida
- 6: Auxiliar de entradas al almacén
- 7: Auxiliar de salidas al almacén
- 8: Acumulado de entradas y salidas del almacén
- 9: Gastos presupuestales
- 0: Fin

HASTA opción igual a 0

FIN DE REPORTES.

PROCESO : REPORTE DE CATALOGOS

Objetivo: Emitir el reporte de cualquiera de los catálogos.

REPITE

Leer opción

CASO de que la opción sea igual a :

- 1: Lee y ordena partidas
despliega clave y descripción
- 2: Lee y ordena artículos
despliega clave, unidad y descripción
- 3: Lee y ordena departamentos
despliega clave, descripción
- 4: Lee y ordena empleados
despliega rfc, nombre y departamento
- 5: Fin

HASTA opción igual a 5

FIN DE REPORTES DE CATALOGOS.

PROCESO : REPORTE GENERAL DE ALMACEN

Objetivo: Emitir reporte general de almacén
Lee y ordena almacén
Imprime clave de artículo, entradas, salidas,
Precio y existencia real.

FIN DE REPORTE GENERAL DE ALMACEN.

PROCESO : REPORTE DE ARTICULOS CON EXIS. <= EXIS.MÍN

Objetivo: Emitir reporte de artículos con existencia real
menor o igual a la existencia mínima especificada.

Lee y ordena almacén y entradas
Imprime clave de artículo, existencia real, exis.
mínima y exis. máxima, y además la diferencia entre
La existencia real y la mínima.

FIN DE REPORTE DE ARTÍCULOS CON EXIS. <= MÍNIMA.

PROCESO : REPORTE DE CONSUMOS POR EMPLEADO O DEPARTAMENTO

Objetivo: Emitir reporte de consumos globales dependiendo
la opción elegida por el usuario.

REPITE

Leer la opción

CASO de que la opción sea igual a:

- 1: Lee y ordena empleados
imprime rfc, nombre, departamento, consumo actual
y consumo acumulado.
- 2: Lee y ordena departamentos
imprime clave, descripción, consumo actual y
consumo acumulado.
- 3: Fin

HASTA opción = 3

FIN DE REPORTE DE CONSUMOS POR EMPLEADO O DEPARTAMENTO.

PROCESO : REPORTE DE CONSUMOS POR DEPARTAMENTO Y PARTIDA

OBJETIVO: Emitir reporte de consumos por departamento y partida.

Lee y ordena archivo de partidas
Lee y ordena archivo de salidas

FOR CADA REGISTRO DE PARTIDAS HAZ

Lee datos del archivo de salidas
Suma consumos actuales
Suma consumos acumulados
Imprime depto, consumos actuales, consumos acumulados

FIN FOR CADA

FIN DE REPORTE DE CONSUMOS POR DEPARTAMENTO Y PARTIDA.

PROCESO : REPORTE AUXILIAR DE ENTRADAS AL ALMACEN

Objetivo: Emitir reporte detallado de entradas al almacén.

Lee y ordena entradas
Imprime no. nota, fecha, cantidad y precio

FIN DE REPORTE AUXILIAR DE ENTRADAS AL ALMACEN.

PROCESO : REPORTE AUXILIAR DE SALIDAS AL ALMACEN

Objetivo: Emitir reporte detallado de salidas al almacén.

Lee y ordena salidas
Imprime no. vale, fecha, cantidad y rfc

FIN DE REPORTE AUXILIAR DE SALIDAS AL ALMACEN.

PROCESO : REPORTE ACUMULADO DE ENTRADAS Y SALIDAS DEL ALMACEN

Objetivo: Emitir reporte acumulado de entradas y salidas del almacén.

Lee y ordena almacén
Imprime clave artículo, entradas, salidas, precio
Promedio, existencia real.

FIN DE REPORTE ACUMULADO DE ENTRADAS Y SALIDAS DEL ALMACEN.

PROCESO : REPORTE DE GASTOS PRESUPUESTALES POR PARTIDA

Objetivo: Emitir reporte de gastos presupuestales por partida.

Lee y ordena partidas
Imprime clave de partida, descripción, gastos
acumulados y gastos actuales.

FIN DE REPORTE DE GASTOS PRESUPUESTALES POR PARTIDA.

PROCESO : PERIODO

OBJETIVO: Seleccionar la función a realizar dependiendo de la
opción elegida por el usuario.

REPITE
Desplegar el menú de período
Leer opción

CASO de que la opción sea igual a :

- 1: Cambio de período
- 2: Cambio de año
- 3: Fin

HASTA opción igual a 3

FIN DE PERIODO.

PROCESO : CAMBIO DE PERIODO

Objetivo: Actualizar los gastos y consumos de acuerdo a los movimientos registrados durante un periodo.

POR CADA registro partidas HAZ

Lee datos del archivo partidas
Reemplaza consumo acumulado con saldo actual
Reemplaza consumo actual con cero
Reemplaza gasto acumulado con gasto actual
Reemplaza gasto actual con cero

FIN POR CADA**POR CADA registro artículos HAZ**

Lee datos del archivo partidas
Reemplaza consumo acumulado con saldo actual
Reemplaza consumo actual con cero

FIN POR CADA**POR CADA registro departamento HAZ**

Lee datos del archivo departamento
Reemplaza consumo acumulado con saldo actual
Reemplaza consumo actual con cero

FIN POR CADA**POR CADA registro empleado HAZ**

Lee datos del archivo empleado
Reemplaza consumo acumulado con saldo actual
Reemplaza consumo actual con cero

FIN POR CADA**POR CADA registro almacén HAZ**

Lee datos del archivo almacén
Reemplaza existencia anterior con
existencia anterior más entradas menos salidas
Reemplaza entradas y salidas con cero

FIN POR CADA

FIN DE CAMBIO DE PERIODO.

PROCESO 1 CAMBIO DE AÑO

Objetivo: Inicializar los gastos y consumos de los archivos para inicial un nuevo año.

FOR CADA registro partidas HAZ
Borra registro

FIN FOR CADA

FOR CADA registro articulos HAZ
Borra registro

FIN FOR CADA

FOR CADA registro departamento HAZ
Borra registro

FIN FOR CADA

FOR CADA registro empleado HAZ
Borra registro

FIN FOR CADA

FOR CADA registro almacén HAZ
Lee datos del archivo almacén
Reemplaza existencia anterior con
existencia anterior más entradas menos salidas
Reemplaza entradas y salidas con cero

FIN FOR CADA

FIN DE CAMBIO DE AÑO.

PROCESO : MANTENIMIENTO

OBJETIVO: Seleccionar la función a realizar dependiendo de la opción elegida por el usuario.

REPITE

Desplegar el menú de mantenimiento
Leer opción

CASO de que la opción sea igual a :

- 1: Realizar una protección de archivos
- 2: Realizar la restauración de archivos
- 3: Indexar archivos
- 4: Leer el título de la dependencia
reemplazar registro en archivo de datos
- 5: Fin

HASTA opción igual a 5

FIN DE MANTENIMIENTO.

PROCESO : AYUDA

OBJETIVO: Desplegar un texto explicativo que sirva como ayuda al usuario en el momento que se solicite.

Desplegar pantalla con opción de ayuda
leer opción

MIENTRAS opción=ayuda

Desplegar texto explicativo

FIN MIENTRAS

FIN DE AYUDA.

En esta fase del desarrollo del sistema se especifica la estructura y la arquitectura del software, es decir, los despliegues por pantalla, los formatos de reportes, la definición de las entradas y salidas de datos.

El diseño se realiza en dos pasos: el diseño preliminar y el diseño detallado. El diseño preliminar se refiere a la transformación de los requerimientos en datos y arquitectura del software, y el diseño detallado se enfoca hacia los refinamientos de la representación arquitectónica que conduce a una estructura de datos detallada y a representaciones algorítmicas del software.

4.1. CALIDAD DEL SOFTWARE

Para evaluar la calidad de una representación del diseño, se deben establecer ciertos criterios que a continuación se mencionan:

- 1) Un diseño debe presentar una organización jerárquica que haga un uso eficiente de los programas y datos entre los elementos del software, esto se refiere a la estructura de los mismos.

La estructura de los programas representa la organización jerárquica de los componentes del programa (módulos) e implica una jerarquía de control y para representarla se utilizan los "Diagramas de Estructura".

La estructura de datos es una representación de la relación lógica entre elementos individuales de datos. Es quien dicta la

organización, métodos de acceso, grado de asociatividad y alternativas de procesamiento para la información.

2) Un diseño debe ser modular.

Los sistemas modulares consisten en unidades claramente definidas y manejables con las interfaces entre los diversos módulos, y uno de los principales objetivos del diseño es el crear sistemas modulares, en los que cada módulo cuenta con las siguientes características:

- a) Contienen instrucciones lógicas de proceso y estructuras de datos.
- b) Pueden ser compilados en forma independiente.
- c) Pueden quedar incluidos en un solo programa.
- d) Pueden contener a otros módulos.

La modularización puede ser utilizada para aislar dependencias funcionales, para mejorar el desempeño de un producto o para facilitar la depuración, las pruebas, la integración, el ajuste y la modificación del sistema.

- 3) Un diseño debe contener una representación distinta de los datos y los procedimientos de cada módulo.
- 4) Un diseño debe conducir a módulos, es decir hacer uso de subrutinas.
- 5) Un diseño debe derivarse usando un método repetible que esté conducido por la información obtenida durante el análisis de requerimientos del software.

El concepto de independencia funcional es una derivación directa del de modularidad y los conceptos de abstracción y ocultación de la información. La independencia funcional se adquiere desarrollando módulos con una función específica y siendo éste poco interactivo con otros módulos, por este motivo es importante la independencia entre módulos. Ahora bien, la medida de la fuerza funcional relativa de un módulo se llama "Cohesión", y la medida de la interdependencia relativa entre módulos se llama "Acoplamiento".

4.2. ACOPLAMIENTO Y COHESION

Una meta fundamental en el diseño del sistema es la de estructurarlo de tal forma que el número y la complejidad de las interacciones entre los diversos módulos sea minimizada.

1) ACOPLAMIENTO

La interdependencia relativa entre módulos puede ser considerado en el siguiente orden, del mayor al menor:

- a) de contenido
- b) de zonas compartidas
- c) de control
- d) de zonas de datos
- e) de los propios datos

2) COHERSIÓN

La cohesión interna se mide en términos de la fuerza de unión de los elementos dentro del módulo; esta cohesión ocurre dentro de una escala de la menor a la mayor:

- a) coincidental
- b) lógica
- c) temporal
- d) en la comunicación
- e) secuencial
- f) funcional
- g) informacional

4.3. ARQUITECTURA DEL SOFTWARE

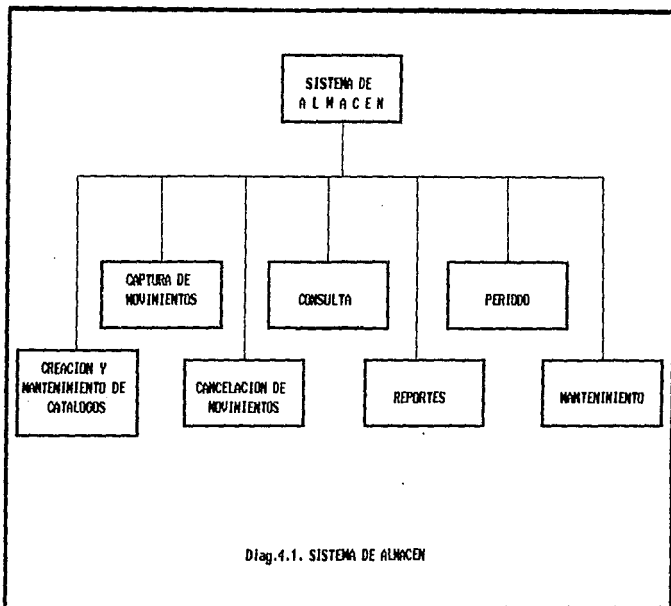
La arquitectura del software alude a dos características importantes de un programa de computadora:

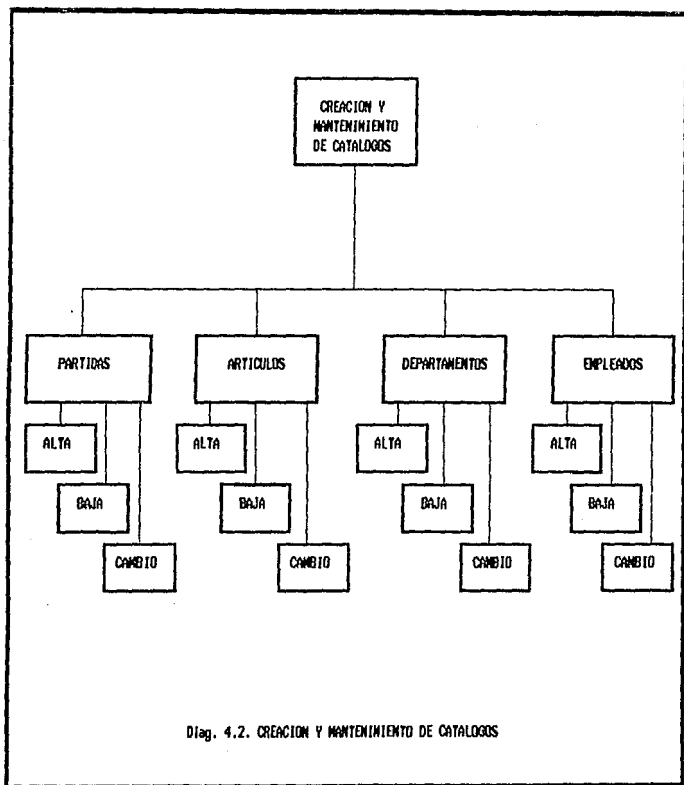
- 1) La estructura jerárquica de los componentes procedimentales (módulos), y
- 2) La estructura de los datos.

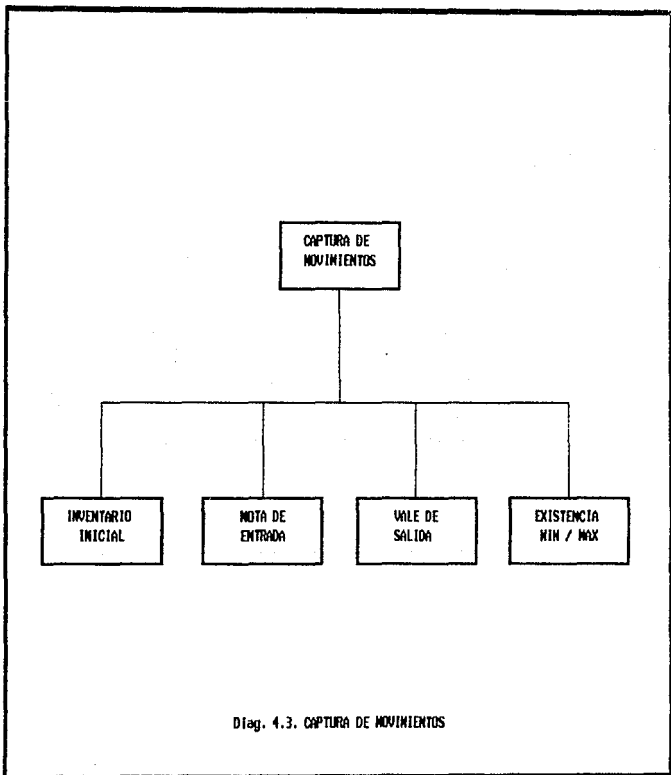
La arquitectura del software implica modularidad; esto es, el software debe dividirse en elementos con nombres y direcciones separadas, llamadas módulos, que se integran posteriormente para satisfacer los requerimientos del sistema. Además esta modularidad proporciona ventajas en el costo y esfuerzo requerido para el desarrollo del programa.

4.3.1. DIAGRAMAS DE ESTRUCTURA DEL SISTEMA PROPUESTO

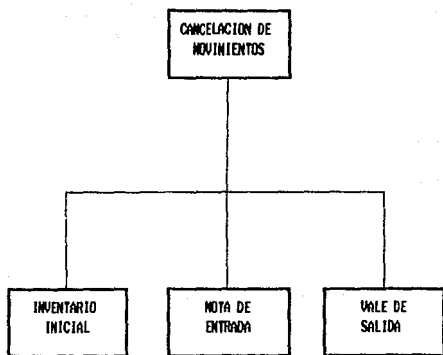
A continuación se muestran los diagramas de estructura del sistema en desarrollo :



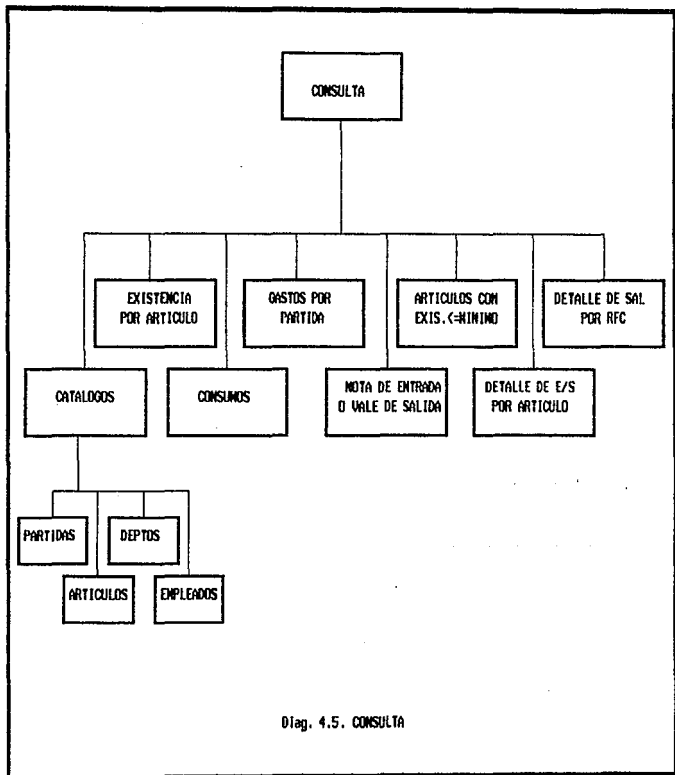




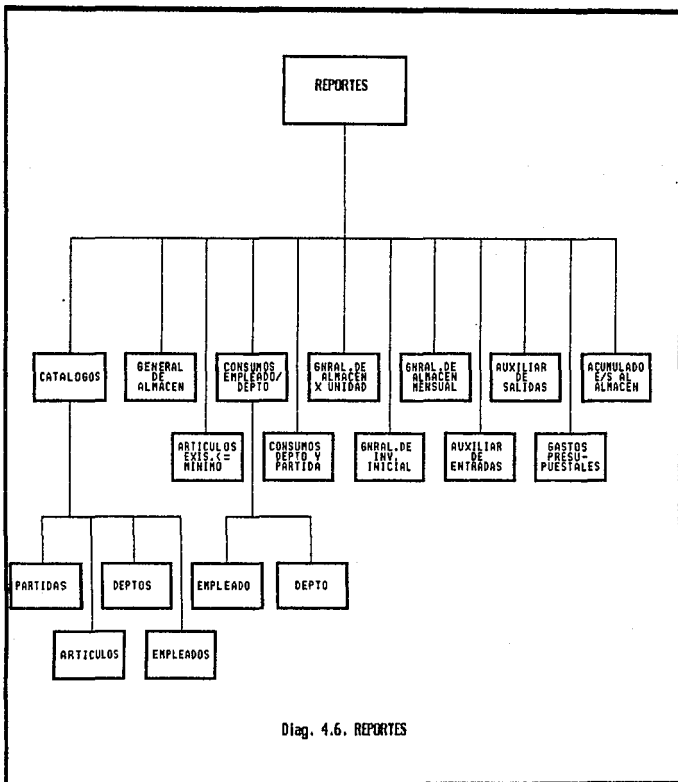
Diag. 4.3. CAPTURA DE MOVIMIENTOS



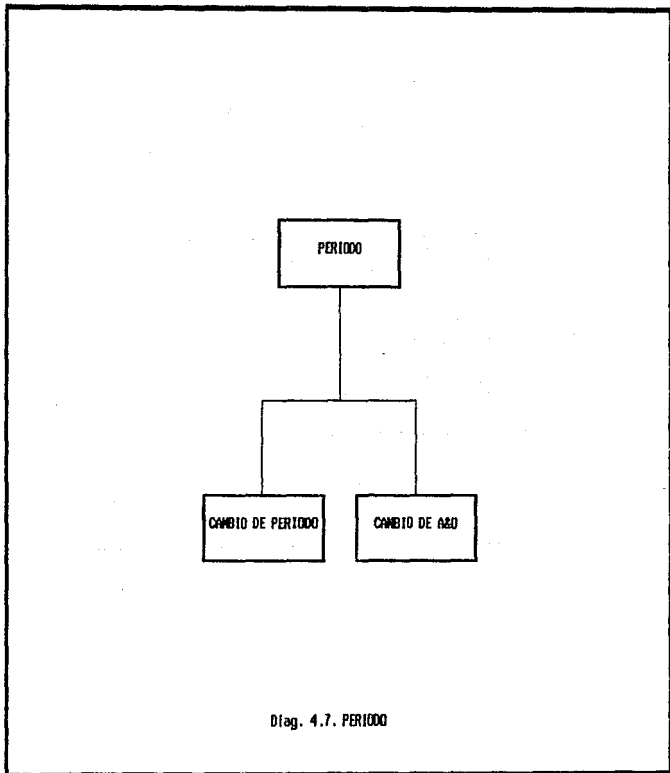
Diag. 4.4. CANCELACION DE MOVIMIENTOS



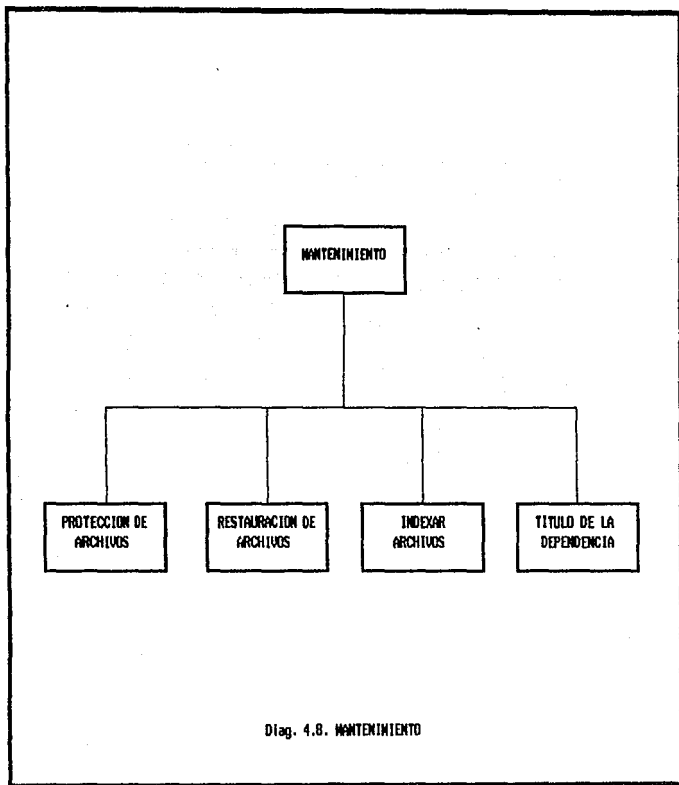
Diag. 4.5. CONSULTA



Diag. 4.6. REPORTES



Diag. 4.7. PERIODO



Diag. 4.8. MANTENIMIENTO

4.3.2. DISEÑO DE LA BASE DE DATOS

El diseño de datos es una representación de la relación lógica entre elementos individuales de datos.

Recordemos que la unidad más pequeña de datos es el campo . Y un conjunto de datos elementales constituye un registro lógico o entidad. Un tipo de registro lógico es un registro con una constitución particular de datos elementales; un archivo es una colección de ocurrencias de un mismo tipo de registros y finalmente una base de datos es un conjunto de ocurrencias de varios tipos de registros, en el cual, tanto los tipos de registros como sus ocurrencias están interrelacionadas mediante relaciones específicas.

Antes de proceder al diseño es importante señalar las características generales de las bases de datos.

Los primeros esfuerzos en manejo de bases de datos se realizaron a finales de los años sesenta, las necesidades que tenían de proporcionar acceso efectivo y eficiente a los usuarios, a los programas, a grandes archivos, los cuales requerían además integrarse. Las estructuras convencionales de archivos, con sus limitaciones de llaves únicas de acceso secuencial, o un solo tipo de registros, resultaban inadecuadas.

Los primeros diseños de sistemas de manejo de bases de datos denominados jerárquicas, invertidas y de red aparecieron a finales de los años sesenta, con lo cual hubo un surgimiento inevitable de aplicaciones de la computadora y de sistemas de información

centrados en bases de datos, en lugar de la orientación tradicional.

Han surgido muchos retos debido a la necesidad de integrar dentro de bases de datos, archivos y datos diversos con mayor efectividad de costos para usuarios con diferentes prioridades y requerimientos. Entre estos retos se tiene la necesidad de estructuración de los datos, organización de los archivos, independencia de datos entre programas de aplicación y estructuras particulares, habilidad para que los usuarios compartan datos, no redundancia de la información, alto nivel de rendimiento, eficiencia, seguridad e integridad.

a) Habilidad de relacionar

Es la habilidad para definir relaciones entre registros o entidades a nivel lógico de manera conveniente.

b) Flexibilidad de acceso

Es la capacidad de lograr acceso a cualquier lugar de la base de datos en base a cualquier llave.

c) Seguridad

Deben existir mecanismos para asignar, controlar y revocar los derechos de acceso de cualquier usuario a cualquier dato.

d) Arquitectura de un sistema de manejo de bases de datos

Existen tres tipos principales de modelos de bases de datos para definir registros o entidades y las relaciones que guardan éstos en las estructuras de las bases de datos lógicas:

1) **Modelo jerárquico**, es una simple red, constituida por

una estructura arborescente en la cual todos los apuntadores apuntan en la dirección desde los hijos hacia el padre, sin embargo en este tipo de estructura no pueden existir hijos compartidos por diferentes padres.

ii) Modelo de datos de red, se caracteriza porque todas las relaciones están restringidas a ser de muchos a uno, es decir que los nodos pueden tener uno o varios padres, permitiendo con esto usar un diagrama de datos simple y la implementación se hace más simple.

iii) Modelo de datos relacional, es un arreglo bidimensional con las siguientes características:

- Cada entrada en la tabla es un elemento de datos o dato elemental donde no hay grupos repetitivos.
- A cada columna, esto es al dominio, se le asigna un nombre diferente y que está constituido por valores del mismo dato elemental.
- Todas las hileras son distintas, no se permiten duplicaciones.
- Las hileras y las columnas pueden ordenarse en cualquier secuencia en cualquier momento sin que afecte el contenido de la información de la semántica implicada.

4.3.2.1. NORMALIZACION

La normalización es un proceso mediante el cual un diseñador de bases de datos puede transformar cualquier estructura no plana, arborea o de red a un conjunto de relaciones normalizadas es decir,

a un conjunto de relaciones planas que no tengan grupos repetitivos. Se dice que una relación no plana es no normalizada. Una relación no normalizada contiene al menos un dominio que es en realidad otra relación, una relación normalizada tiene únicamente dominios simples es decir dominios que no sean a su vez otra relación. De otra forma, el proceso de normalización identifica los datos redundantes que pueden existir en la estructura lógica, determina claves únicas necesarias para el acceso a los elementos de datos y ayuda a establecer las relaciones necesarias entre los elementos de datos.

Pueden establecerse tres formas o niveles de normalización, estas proporcionan una mejoría en las propiedades de la base de datos respecto a cambios en la misma.

El proceso de normalización se puede resumir de la siguiente forma:

- a) Tomar la relación en primera forma normal (se dice que una relación se encuentra en primera forma normal si todos sus dominios son simples, es decir si es una tabla bidimensional o plana), y remover todas las dependencias funcionales incompletas mediante la aplicación de proyecciones apropiadas. Lo que produce un conjunto de relaciones de segunda forma normal, si cada llave candidata consiste de un solo atributo, entonces la relación de primera forma normal se convierte automáticamente en segunda forma normal.
- b) Aplicar proyecciones a las relaciones en segunda forma normal, de tal forma que las relaciones que resulten de cada

determinante sea una llave candidata o un atributo primo (un atributo primo es uno que sea miembro del conjunto de atributos que constituyen a una llave candidata de atributos múltiples) o en otras palabras, tal que los atributos no primos sean mutuamente independientes.

De acuerdo a la teoría de bases de datos se utilizará un proceso de diseño de bases de datos para que éstas cumplan con las características óptimas de un buen diseño. Este proceso se puede desglosar en tres etapas : diseño conceptual, diseño lógico y diseño físico.

4.3.2.2. DISEÑO CONCEPTUAL

El objetivo de este diseño en las bases de datos es obtener un modelo de las necesidades de la información, independientemente de las características del sistema de cómputo, y es un modelo estructurado de las entidades que deben almacenarse y relacionarse. De acuerdo al análisis realizado en el capítulo anterior se necesita almacenar la siguiente información:

- a) Los catálogos deberán almacenarse uno en cada archivo, conteniendo información referente a los consumos y gastos de cada rubro.
- b) Es necesario almacenar a detalle los movimientos registrados: notas de entrada y vales de salida.
- c) Los movimientos incorrectos permanecerán marcados en el archivo correspondiente.

determinante sea una llave candidata o un atributo primo (un atributo primo es uno que sea miembro del conjunto de atributos que constituyen a una llave candidata de atributos múltiples) o en otras palabras, tal que los atributos no primos sean mutuamente independientes.

De acuerdo a la teoría de bases de datos se utilizará un proceso de diseño de bases de datos para que éstas cumplan con las características óptimas de un buen diseño. Este proceso se puede desglosar en tres etapas : diseño conceptual, diseño lógico y diseño físico.

4.3.2.2. DISEÑO CONCEPTUAL

El objetivo de este diseño en las bases de datos es obtener un modelo de las necesidades de la información, independientemente de las características del sistema de cómputo, y es un modelo estructurado de las entidades que deben almacenarse y relacionarse. De acuerdo al análisis realizado en el capítulo anterior se necesita almacenar la siguiente información:

- a) Los catálogos deberán almacenarse uno en cada archivo, conteniendo información referente a los consumos y gastos de cada rubro.
- b) Es necesario almacenar a detalle los movimientos registrados: notas de entrada y vales de salida.
- c) Los movimientos incorrectos permanecerán marcados en el archivo correspondiente.

- d) Los movimientos deben ordenarse según el tipo de movimiento, según el número de vale o nota y según la clave del artículo.
- e) Es necesario acumular las entradas y salidas en un archivo llamado "Almacén".
- f) La relación entre los archivos está dada por la clave del artículo.

4.3.2.3. DISEÑO LOGICO

En el diseño lógico de la base de datos se presentan las entidades y las relaciones del modelo conceptual, utilizando estructuras de algún sistema de administración de datos o las estructuras de un sistema de archivos.

Esas relaciones pueden representarse en un modelo gráfico no implementado en alguna máquina, llamado "Entidad-Relacionamiento", compuesto por tablas y gráficas que utilizan la siguiente nomenclatura:



Representa una entidad.



Representa atributos, los cuales están ligados a las entidades mediante líneas.



Representan a las relaciones entre entidades.

La definición lógica de la base de datos para el sistema de Almacén requiere de los siguientes archivos:

Archivo: PARTIDAS

Contiene: información referente a las partidas

Estructura:

Nombre de campo	tipo	longitud
partida	caracter	3
descripcion	caracter	32
cons_acum	numérico	15,2
cons_actu	numérico	15,2
gest_acum	numérico	15,2
gest_actu	numérico	15,2

Archivo: DEPTOS

Contiene: información referente a los departamentos

Estructura:

Nombre de campo	tipo	longitud
depto	caracter	3
descripcion	caracter	32
cons_acum	numérico	15,2
cons_actu	numérico	15,2

Archivo: ARTICULOS

Contiene: información referente a los artículos

Estructura:

Nombre de campo	tipo	longitud
clave_art	caracter	7
unidad	caracter	8
desc1	caracter	40
desc2	caracter	40
desc3	caracter	40
desc4	caracter	40
gest_acum	numérico	15,2
gest_actu	numérico	15,2

Archivo: EMPLEADO

Contiene: información referente a los empleados

Estructura:

Nombre de campo	tipo	longitud
rfc	caracter	10
nombre	caracter	32
depto	caracter	3
cons_acum	numérico	15,2
cons_actu	numérico	15,2

Archivo: ALMACEN

Contiene: información acumulada de entradas y salidas por artículo

Estructura:

Nombre de campo	tipo	longitud
clave_art	caracter	7
exis_ante	numérico	9,2
entradas	numérico	9,2
salidas	numérico	9,2
precioprom	numérico	10,2

Archivo: ENTRADA

Contiene: información detallada referente a las entradas registradas en el almacén.

Estructura:

Nombre de campo	tipo	longitud
tipo_mov	caracter	1
fecha	fecha	8
nota	caracter	6
clave_art	caracter	7
precio_uni	numérico	10,2
cantidad	numérico	9,2
exis_max	numérico	9,2
exis_min	numérico	9,2

Archivo: SALIDAS

Contiene: información detallada referente a las salidas registradas en el almacén.

Estructura:

Nombre de campo	tipo	longitud
clave_art	caracter	7
fecha	fecha	8
vale	caracter	6
rfc	caracter	7
depto	caracter	3
cantidad	numérico	9,2
precio	numérico	10,2

Archivo: DATOS

Contiene: el nombre de la dependencia usuaria

Estructura:

Nombre de campo	tipo	longitud
dependenci	caracter	78

Archivo: AYUDA

Contiene: el texto que será desplegado al solicitar la ayuda

Estructura:

Nombre de campo	tipo	longitud
campo	caracter	5
ayu	texto	10
programa	caracter	20

Las relaciones entre los archivos se muestran en el siguiente diagrama de Enter-Relación:

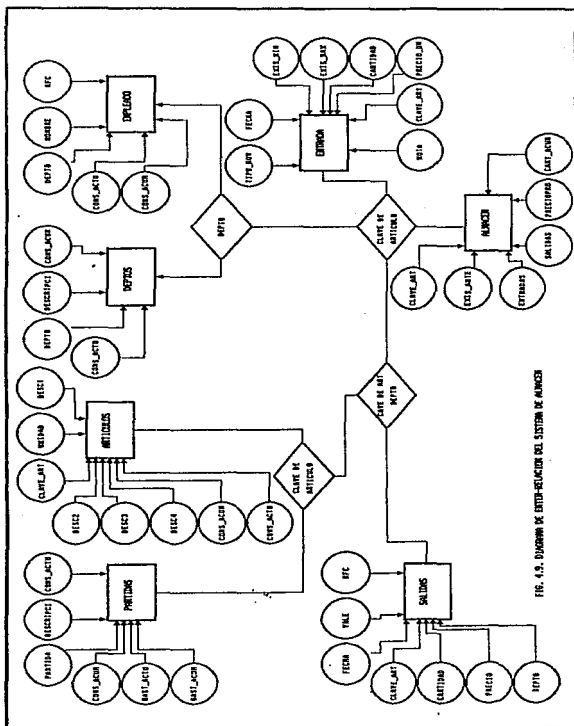


FIG. 4.9. DIAGRAMA DE ENTER-RELACION DEL SISTEMA DE NUMEROS

4.3.2.4. DISEÑO FÍSICO

En el diseño físico de la base de datos se hacen estudios de los tiempos de acceso para ver si corresponden a los requerimientos de tiempo de respuesta especificado por el usuario.

Las dos primeras etapas del diseño de una base de datos son necesarias, pero el diseño físico en ocasiones no es necesario, dependiendo del tipo de sistema de administración de datos o el tipo de sistema de archivos con que se cuente.

En este problema no se tienen requerimientos que demanden un acceso a la base de datos en determinado tiempo, como sucede en sistemas de tiempo real, por lo que no se realizará esta etapa.

4.4. SOFTWARE PARA EL DESARROLLO DEL SISTEMA

4.4.1. SELECCION DEL LENGUAJE DE PROGRAMACION

Es aquí en donde se describen las herramientas a utilizar en el desarrollo del sistema, primero se describen los factores considerados en la elección del lenguaje de programación:

- Mínimos costos y tiempo de programación
- Mínimo tiempo y costo de desarrollo
- Mínimo costo de mantenimiento
- Disponibilidad del compilador
- Versatilidad
- Documentación

El software utilizado deberá funcionar para equipos PC compatibles con IBM además de estar orientado a funciones administrativas, por tanto tenemos dos alternativas en el lenguaje de programación a utilizar:

- a) Dentro de los lenguajes de alto nivel enfocados a cuestiones administrativas se encuentra COBOL. Que gracias a su estandarización, mantenimiento, manejo de pantalla para la captura y presentación de los datos es recomendable. La desventaja de este lenguaje es que los programas fuente presentan mucho código y restan tiempo al desarrollo. Además la experiencia con el sistema anterior muestra problemas en la recuperación de archivos índice, como ya se mencionó en el capítulo uno.
- b) Manejadores de Bases de datos, como: Dbase, Clipper, Foxbase, Dataeas, etc., que existen en el mercado. Los cuales proporcionan ventajas como:
 - Proporcionar un lenguaje de programación para P.C.'s
 - El buen manejo de grandes volúmenes de información así como la administración adecuada de las funciones interrelacionadas en la organización.
 - Presentar la descripción de la información en forma independiente de los procesos lo que permite al usuario desligarse del manejo de datos.
 - No es necesario modificar la estructura de los programas cuando se modifica la estructura de las bases de datos.

- El poder eliminar la redundancia de información mediante la integración de los datos de diferentes aplicaciones en una misma base de datos.
- El permitir la utilización de la base de datos para posteriores aplicaciones.
- Reducir el esfuerzo en la consulta por pantalla

La mayoría de este tipo de paquetes: manejadores de bases de datos, aplican conceptos sobre bases de datos tipo relacional. Este tipo de bases de datos descrito por F. Codd en 1969, cuenta con las siguientes características:

- Maneja los archivos como estructura de tabla
- Los datos se almacenan en forma tabular
- Almacena los datos con una mejor visión para el usuario
- Proporciona un diccionario de datos
- Tiene facilidad para crecer sin causar problemas en encadenamiento o apuntadores.

Y proporciona ciertas ventajas en la seguridad de la información, en la independencia de los datos, en la facilidad de uso, en la actualización y consulta en forma interactiva y presenta el almacenamiento en forma virtual.

Como ya se mencionó existen diferentes manejadores de bases de datos, dentro de los cuales esta Dbase III Plus, el cual fué elaborado por la compañía Asthon Tate, siendo éste el más aceptado en el mercado actualmente. Cabe hacer notar que Dbase proporciona todas las ventajas antes mencionadas, pero presenta también una

desventaja y es que no genera código objeto, por lo que es necesario hacer uso de un compilador y ligador compatible, para generar código ejecutable, en este caso CLIPPER es quien solucionará esta desventaja, ya que cuenta con las características requeridas y además es también un manejador de bases de datos.

4.4.2. CARACTERISTICAS DE DBASE III PLUS Y DE CLIPPER

Ventajas de Dbase III Plus:

- permite el manejo de sistemas modulares
- permite el manejo fácil de archivos de datos
- permite gran almacenamiento de información
- permite la portabilidad a otros sistemas operativos, como MS-DOS ver 2.0 o mayor, o en UNIX.
- permite ordenar en dos formas: indexado o por medio de un sort.
- permite la actualización de archivos índice automáticamente al modificar datos.

Limitaciones de Dbase III Plus :

- 128 campos por registro
- 4000 bytes por registro
- 254 bytes por campo
- 512 bytes por campo tipo memo
- un máximo de 15 ficheros abiertos al mismo tiempo

- 10 ficheros índices abiertos al mismo tiempo
- 1 byte máximo para caracteres tipo memo
- 19 bytes máximo para campos tipo numérico
- 8 bytes máximo para campos tipo fecha

Requerimientos de Dbase III Plus:

- 256 Kbytes de memoria RAM mínimo
- sistema operativo MS-DOS ver 2.0 o posterior

Características de CLIPPER:

- la ejecución de un programa objeto es rápida y posee mayor seguridad el código fuente.
- permite utilizar hasta 64000 variables activas en memoria.
- permite manejar hasta 1024 campos por base de datos.
- se puede utilizar a Dbase como intérprete y a partir de este generar archivos, reportes, etiquetas y pantallas, para posteriormente ejecutarlos desde el compilador.
- provee un menú de selección para el programa, el cual da facilidad de hacer uso de las flechas de navegación para seleccionar la opción deseada.
- también permite tener un archivo de ayuda invocado por las teclas de función. Facilitando la restauración de pantallas mientras se hace el uso de este tipo de ayuda.

Requerimientos de CLIPPER:

- deberá utilizarse en PC/XT , AT o PC compatibles con IBM
- sistema operativo PCDOS o MSDOS versión 2.0 o mayor
- 256 Kbytes de memoria RAM disponible
- la impresora deberá ser de 80 columnas como mínimo.

4.5. DISEÑO DE ENTRADAS Y SALIDAS

Las entradas y las salidas son los medios por los cuales el usuario tiene contacto con el sistema.

- a) **Entradas:** Son datos con los cuales se alimenta el sistema y son requeridos por las salidas. Estas son las pantallas de captura.
- b) **Salidas:** Todo el proceso dentro del sistema está en función de éstas, ya que son las que se desea obtener, siendo las consultas por pantalla y los reportes.

A continuación se presenta un ejemplo de las pantallas de captura (ENTRADA) , consulta (SALIDA) y reportes (SALIDA) :

DIRECCION GENERAL DE SERVICIOS DE COMPUTO ADVMO. SISTEMA DE ALMACEN	
CAPTURA	FECHA: _____ N.VALE: _____ REG: _____ NOMBRE: _____ DEPTO: _____ DESCRIPCION: _____
INV. INICIAL	CLAVE DESCRIPCION:
NOTA ENTRADA	_____
VALE SALIDA	_____
EXIS. MAX. MIN	_____
	CANTIDAD: _____
C PARA CORREGIR O RETURN PARA CONTINUAR	

DIRECCION GENERAL DE SERVICIOS DE COMPUTO ADMVO.
SISTEMA DE ALMACEN

CONSULTA

CATALOGOS EXIS. DE ARTICULO CONSUMOS GASTOS X PARTIDA N. ENT O VAL. SAL ART. EXIS. - MIN. DETALLE E/S X ART DETALLE SAL X RFC FIN DE CONSULTAS	<table style="width: 100%;"> <tr> <td style="width: 20%;">RFC</td> <td>NOMBRE</td> </tr> <tr> <td><input type="text"/></td> <td><input type="text"/></td> </tr> <tr> <td>DEPTO</td> <td>DESCRIPCION</td> </tr> <tr> <td><input type="text"/></td> <td><input type="text"/></td> </tr> <tr> <td colspan="2" style="text-align: center;">CONSUMO ACTUAL CONSUMO ACUMULADO</td> </tr> <tr> <td style="text-align: center;">\$.00</td> <td style="text-align: center;">\$.00</td> </tr> </table>	RFC	NOMBRE	<input type="text"/>	<input type="text"/>	DEPTO	DESCRIPCION	<input type="text"/>	<input type="text"/>	CONSUMO ACTUAL CONSUMO ACUMULADO		\$.00	\$.00
RFC	NOMBRE												
<input type="text"/>	<input type="text"/>												
DEPTO	DESCRIPCION												
<input type="text"/>	<input type="text"/>												
CONSUMO ACTUAL CONSUMO ACUMULADO													
\$.00	\$.00												

C. ARTICULO
 C. EMPLEADO
 C. EMPLEADO
 C. PARTIDA

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

TITULO DE LA DEPENDENCIA

REPORTE GENERAL DE ALMACEN

HOJA: 1
FECHA: 01/09/92

CLAVE	UNIDAD	DESCRIPCION	EX. ANTE	ENT.	SAL.	PRECIO	VALOR	EX. MIN	EX. MAX
.....
.....
.....

En esta fase del desarrollo del sistema se realiza la implementación del sistema, es decir, en esta etapa se debe realizar la codificación y documentación de los programas especificados por el diseño del sistema.

5.1. CODIFICACION

La codificación consiste en la traducción de las especificaciones de proceso de cada módulo en instrucciones ejecutables por un lenguaje de programación.

Este proceso de "traducción" continúa cuando un compilador acepta el código fuente como entrada y produce como salida un código objeto.

Se pueden definir dos tipos o técnicas de codificación:

- a) **BOTTOM-UP** o ascendente, esta técnica permite atender inicialmente los módulos de los niveles jerárquicos más bajos, ya que estos contendrán las operaciones más primitivas del programa y permite sentar la base para de ahí construir las operaciones complejas.
- b) **TOP-DOWN** o descendente, en esta técnica se codificarán primero los niveles más altos de la jerarquía y así sucesivamente hasta llegar a los niveles más bajos.

El sistema de Almacén desarrollado se codificará usando la técnica Top-Down de tal forma que primero se codificará el menú principal, después los menús siguientes en la jerarquía y por último los procesos que son llamados por estos.

5.2. DOCUMENTACION DEL CODIGO

La documentación interna del código fuente comienza con la elección de los nombres de los identificadores como variables y etiquetas, posteriormente continúa con la localización y la composición de los comentarios y termina con la organización visual del programa.

Existe la posibilidad de expresar comentarios en lenguaje natural dentro de cualquier lenguaje de programación, lo cual permite al programador comunicarse con otros lectores del código fuente, y pueden resultar una guía clara durante la fase de mantenimiento. Los comentarios deben aparecer al principio de cada módulo, y deben contener:

- a) Una sentencia de propósito que indique la función del módulo.
- b) Una descripción de la interfaz, que incluya:
 - un ejemplo de secuencia llamada
 - una descripción de los argumentos
 - una lista de los módulos subordinados.
- c) Una discusión de los datos, como variables importantes y su uso, de las restricciones y limitaciones.
- d) Una historia del desarrollo que incluya:
 - el nombre del autor

- el nombre del auditor
- fechas de modificación y de descripción.

5.2.1. DECLARACION DE DATOS

El orden de las declaraciones de datos se debe estandarizar, lo cual permite que los atributos sean fáciles de descubrir, de comprobar, de depurar y de mantener.

5.2.2. CONSTRUCCION DE SENTENCIAS

La construcción de sentencias se debe basar en una regla general: cada sentencia debe ser simple y directa.

5.2.3. ENTRADA/SALIDA

El estilo de la entrada y salida (E/S) se establece durante el análisis de requerimientos y diseño. Sin embargo es una característica importante para la aceptación del sistema por un usuario.

Deben considerarse una serie de principios de estilo para la E/S durante la codificación, que deberá cumplir el programador:

- a) validar los datos de entrada.
- b) comprobar las combinaciones admisibles de elementos de entrada.
- c) mantener el formato de entrada simple.
- d) usar indicativos de fin de dato, en lugar de solicitar al usuario que indique el número de elementos.

- e) etiquetar las peticiones interactivas de entrada, especificando las opciones posibles o valores límite.
- f) mantener el formato de entrada uniforme.
- g) etiquetar todas las salidas y diseñar todos los informes.

El estilo de E/S se ve afectado por las características del hardware ,por ejemplo por el tipo de monitor, dispositivos de gráficos, tipo de impresora, etc. Es por esto que se mencionan a continuación unos principios aplicables a la codificación:

- a) hacer invisibles al usuario los aspectos internos de la computadora.
- b) hacer que el usuario no haga que termine el programa en forma anormal.
- c) advertir al usuario cuando un proceso pueda tener mayores consecuencias.
- d) proporcionar ayuda o asistencia durante toda la ejecución.
- e) adecuar los requerimientos de entrada a las posibilidades del usuario.
- f) adecuar los mensajes de salida a la velocidad de los dispositivos de salida, tales como la impresora, o el monitor.
- g) mantener un tiempo de respuesta consistente.
- h) minimizar el trabajo extra del usuario en casos de error.

Por otra parte, en la codificación se utilizaron las miniespecificaciones obtenidas en la fase de "Análisis del sistema y de requerimientos" , ya que el lenguaje de programación Dbase III Plus seleccionado anteriormente permite utilizar de manera total

las técnicas de programación estructurada, dado que posee dentro de su conjunto de instrucciones los siguientes comandos:

- i) IF. . . ELSE , para configurar estructuras de desición
- ii) CASE , para condiciones mutuamente exclusivas
- iii) DO WHILE, para realizar estructuras de repetición

con lo que permitirá obtener un sistema bien documentado gracias a la claridad de su estructura.

Además, en la implementación de las bases de datos posee instrucciones que permiten manejar de manera sencilla las bases de datos relacionales como es, la creación y modificación de la estructura interna sin necesidad de realizar modificaciones a los programas de aplicación.

5.2.4. EFICIENCIA

La eficiencia es un requerimiento de rendimiento del software y por esto todos los sistemas se catalogan como buenos por la ingeniería desde este punto de vista; es por esto que se consideran tres puntos importantes en la eficiencia de un sistema: la eficiencia en código, la eficiencia en memoria y la eficiencia en la E/S.

La eficiencia en el código, simplifica las expresiones aritméticas y lógicas, así como evita la mezcla de tipos de datos.

La eficiencia en memoria, mantiene al sistema en forma simple, es decir, consigue eficiencia en el tiempo de ejecución con lo que se minimiza el uso de la memoria.

Una vez que se han establecido las características que debe cumplir

el código, para obtener un producto eficaz y de calidad, se muestran algunos de los programas (código fuente) en Dbase III Plus desarrollados para el sistema de Almacén. Estos programas se encuentran ligados con los diagramas de estructura mostrados en la fase de diseño.

NOTA. se anexa a este documento una carpeta con todos los programas elaborados, ejemplo de éstos se muestran en las siguientes páginas:

```

PROGRAMA: ALMACEN1.PRO
MENU PRINCIPAL DEL SISTEMA DE ALMACEN
U . N . A . M .
DIRECCION GENERAL DE SERVICIOS DE COMPUTO PARA LA ADMINISTRACION
AUTOR: CARRILLO VELAZQUEZ MA. DEL ROCIO
FECHA: JULIO DE 1990.

```

```

DO WHILE .T.
SET ECHO OFF
SET WRAP ON
SET DATE FRENCH
SET BELL OFF
SET SCOR OFF
SET DELETED ON
SET TALK OFF
SET COLOR TO
SET ESCAPE ON
CLOSE DATABASES
CLEAR
USE DATOS
GO TOP

```

```

**** DEFINICION DE COLORES PARA CUALQUIER MONITOR ****

```

```

IF !SCOLOR()
IF .NOT. FILE ('$INCOLOR.VID')
COLOR1='M+/B,N/BG'
COLOR2='M++/R+'
COLOR3='GR+/B'
COLOR4='M+/BR,N/BG'
COLOR5='M+/R+'
COLOR6='N/BG'
COLOR7='B+/BG'
COLOR8='GR+/BG'
ELSE
STORE 'M+/M' TO COLOR1,COLOR2,COLOR3,COLOR4,COLOR5,COLOR6,COLOR7,COLOR8
ENDIF
ELSE
**** MONITOR MONOCROMATICO ****
STORE 'M+/M' TO COLOR1,COLOR2,COLOR3,COLOR4,COLOR5,COLOR6,COLOR7,COLOR8
ENDIF

```

```

VERIFICA QUE ESTE DADO DE ALTA EL NOMBRE DE LA DEPENDENCIA USUARIA

```

```

IF EOF()
SET COLOR TO &COLOR2
@ 22,13 SAY " NO SE ENCUENTRA DADO DE ALTA EL TITULO DE LA DEPENDENCIA "
@ 21,12 TO 23,70 DOUBLE
WAIT
SET COLOR TO
@ 21,12 CLEAR TO 23,79
CLEAR
RETURN
ELSE
STORE DEPENDENCI TO TITULO1
ENDIF

```

DESPLIEGA MENU PRINCIPAL DEL SISTEMA DE ALMACEN

```

SET COLOR TO
CLEAR
SET COLOR TO &color1
TITULO2= '          SISTEMA DE ALMACEN
@ 1,(80-LEN(TITULO1))/2 SAY titulo1
@ 2,(80-LEN(TITULO2))/2 SAY TITULO2
@ 2,2 SAY DATE()
@ 2,10 SAY ' '
@ 2,70 SAY TIME()+' '
@ 0, 0 TO 3,79 DOUBLE
SET COLOR TO
SET COLOR TO &color1
S= ''
R= ''
@ 05,1 PROMPT "CATALOGOS"
S=S+'1'
@ 05,10 SAY ' '
@ 5,12 PROMPT "CAPTURA"
S=S+'2'
@ 5,19 SAY ' '
@ 05,21 PROMPT "CANCELAR"
S=S+'3'
@ 5,29 SAY ' '
@ 05,31 PROMPT "CONSULTA"
S=S+'4'
@ 5,39 SAY ' '
@ 05,41 PROMPT "REPORTES"
S=S+'5'
@ 5,49 SAY ' '
@ 05,51 PROMPT "PERIODO"
S=S+'6'
@ 5,58 SAY ' '
@ 05,60 PROMPT "MANTENIMIENTO"
S=S+'7'
@ 5,73 SAY ' '
@ 05,75 PROMPT "FIN"
S=S+'8'
@ 4,0 TO 6,78 DOUBLE
MENU TO SS
IF SS=0
  RETURN
ENDIF
OPC=SUBSTR(S,SS,1)
DO CASE
  CASE OPC=1'
    DO ARCHIVOS
  CASE OPC=2'
    DO CAPTURA
  CASE OPC=3'
    DO CANCELA
  CASE OPC=4'
    DO CONSULTA
  CASE OPC=5'
    DO reportes
  CASE OPC=6'
    DO PERIODO
  CASE OPC=7'
    DO MANTENIM
  CASE OPC=8'
    DO FINAL

```

```
SET COLOR TO  
CLEAR  
RETURN  
ENDCASE  
ENDOO
```

```

PROGRAMA: REPORT81.PRG
REPORTE AUXILIAR POR SALIDAS AL ALMACEN
U . N . A . M .
DIRECCION GENERAL DE SERVICIOS DE COMPUTO PARA LA ADMINISTRACION
AUTOR: CARRILLO VELAZQUEZ MA. DEL ROCIO
FECHA: JULIO DE 1990.

```

```

VERIFICA QUE ESTE DADO DE ALTA EL NOMBRE DE LA DEPENDENCIA USUARIA

```

```

USE DATOS
GO TOP
IF EOF()
SET COLOR TO &COLOR2
@ 22,13 SAY " NO SE ENCUENTRA DADO DE ALTA EL TITULO DE LA DEPENDENCIA "
@ 21,12 TO 23,70 DOUBLE
WAIT " "
SET COLOR TO
@ 21,12 CLEAR TO 23,79
ELSE
STORE DEPENDENCI TO TITULO1
ENDIF

```

```

DESPLIEGA PANTALLA DEL SISTEMA DE ALMACEN

```

```

SET COLOR TO
SET COLOR TO &COLOR1
TITULO2=" SISTEMA DE ALMACEN "
@ 1,(80-LEN(TITULO1))/2 SAY TITULO1
@ 2,(80-LEN(TITULO2))/2 SAY TITULO2
@ 2,2 SAY DATE()
@ 2,10 SAY " "
@ 2,70 SAY TIME()
@ 0, 0 TO 3,79 DOUBLE
SET COLOR TO

```

```

SOLICITA QUE SE CONFIRME LA IMPRESION

```

```

SET CONSOLE OFF
SET COLOR TO &COLOR1
@ 2,70 SAY TIME()
STORE " " TO IMP
DO A
SET COLOR TO &COLOR1
@ 10,60 SAY "¡ PARA IMPRIMIR "
@ 11,60 SAY "RETURN CANCELAR "
@ 12,60 SAY " "
@ 12,66 GET IMP PICTURE " "
@ 9,59 TO 13,79 DOUBLE
READ
CLEAR GETS
SET COLOR TO
IF IMP<>" " AND . IMP<>" "
CLOSE DATABASES
SET COLOR TO

```

```

@ 9,59 CLEAR TO 13,75
@ 21,24 clear to 24,60
RETURN
ENDIF

```

```

# VERIFICA QUE HAYA REGISTROS A IMPRIMIR #

```

```

IF IMP='1'.OR.IMP='11'
  SELE 1
  USE SALIDA INDEX CVEDCS
  GO TOP
  COUNT ALL TO SUMAR
  IF SUMAR=0
    SET COLOR TO
    @ 20,25 CLEAR TO 22,57
    SET COLOR TO &COLOR2
    @ 21,26 SAY = NO HAY REGISTROS A IMPRIMIR *
    @ 20,25 TO 22,57
    WAIT 11
    SET COLOR TO
    @ 20,25 CLEAR TO 22,57
  RETURN
ENDIF

```

```

# INICIALIZA VARIABLES #

```

```

STORE 0 TO SUMAT
store 0 to res,falta
STORE 0 TO RAQUEL
STORE 0 TO RAQUEL1
STORE 1 TO A
STORE 1 TO HOJA
STORE 0 TO LINEAS
store 0 to abas

```

```

# INICIA LA IMPRESION #

```

```

SET PRINT ON
USE SALIDA
GO TOP
INDEX ON RFC TO TRFC
INDEX ON DTOC(FECHA)+VALE TO FECHASS
SELE 1
USE EMPLEADOS INDEX RFC
GO TOP
STORE RFC TO YEYE
DO WHILE .NOT. EOF()
  STORE 0 TO LINEAS
  STORE 1 TO A
  STORE 0 TO ABAS
  STORE RFC TO TIRFC,LLAVE
  STORE NOMBRE TO TNOMBRE
  STORE DEPTO TO TDEPTO
  SELE 2
  USE SALIDA INDEX TRFC,FECHASS
  GO TOP
  SEEK LLAVE
  IF .NOT. EOF()
    DO WHILE .NOT. EOF() .AND. RFC=LLAVE

```

```

STORE FECHA TO TFECHA
STORE VALE TO TVALE
STORE CLAVE_ART TO TCLAVE_ART
STORE CANTIDAD TO TCANTIDAD
STORE PRECIO TO TPRECIO
STORE DEPTO TO TDEPTO

IF RFC=>YEYE.AND. FALTA<=0
?
? CONSUMO TOTAL POR RFC $ '+STR(FALTA,17,2)
  STORE 0 TO RES,FALTA
  @Ject
  store rfc to yeve
  endIf
  IF A=1
?
?SPACE(25)+' UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO '
?
?SPACE(25)+TITULO1
?
?SPACE(25)+' REPORTE AUXILIAR POR SALIDAS AL ALMACEN '
?
?SPACE(80)+'HOJA '+STR(HOJA,3)
?SPACE(80)+'FECHA '+DTC(DATE())
?
? R . F . C . : '+RFC+SPACE(5)+TNOMBRE+SPACE(20)+'DEPARTAMENTO : '+TDEPTO
?
-----
? | DOCTO | FECHA | CLAVE | DESCRIPCION | CANTIDAD | PRECIO | I M P O R T E |
? | | | | | | | UNITA. | |
-----
?

STORE 0 TO A
STORE HOJA+1 TO HOJA
STORE LINEAS+16 TO LINEAS
ENDIF
SELE 3
USE ARTICULOS IMOX CLAVE
GO TOP
SEEK TCLAVE_ART
IF .NOT. EOF()
  STORE CLAVE_ART TO LLAVES
  STORE DESC1 TO TDESC1
  STORE DESC2 TO TDESC2
  STORE DESC3 TO TDESC3
  STORE DESC4 TO TDESC4
ELSE
  STORE SPACE(50) TO TDESC1,TDESC2,TDESC3,TDESC4
ENDIF
SELE 2
STORE TCANTIDAD*TPRECIO TO TTOTAL
? ' +DOCTO+' ' +DTC(FECHA)+' ' +CLAVE_ART+' ' +TDESC1+' ' +STR(CANTIDAD,9,2)+'
$'+STR(PRECIO,10,2)+' $'+STR(TTOTAL,17,2)
STORE SUMAT+TTOTAL TO SUMAT
STORE CANTIDAD*PRECIO TO RES
STORE FALTA+RES TO FALTA
store 1 to abas
store lineas+1 to lineas
STORE RAQUEL+TTOTAL TO RAQUEL
IF TDESC2<>SPACE(50)
  ?SPACE(31)+TDESC2
  store lineas+1 to lineas
ENDIF

```

```
IF tdesc3 <= space(50)
  ?SPACE(31)+TDESC3
  store lines+1 to lines
ENDIF
IF tdesc4 <= space(50)
  ?SPACE(31)+TDESC4
  store lines+1 to lines
ENDIF
IF LINEAS=49
  IF .NOT. EOF()
    STORE 1 TO A
    STORE 0 TO LINEAS
  EJECT
ENDIF
endif
SELE 2
SKIP
ENDDO
ENDIF
IF EOF() .AND. LINEAS<=49
  IF abase=1
    EJECT
    STORE 1 TO A
    STORE 0 TO LINEAS
  endif
ENDIF
SELE 1
SKIP
ENDDO
SET PRINT OFF
SET COLOR TO
CLOSE DATABASES
ENDIF
set color to
@ 9,59 clear to 13,75
@ 21,24 clear to 24,60
close databasee
```

6.1. INTEGRACION

La finalidad de esta fase es, integrar los módulos de un programa, ajustándolos a las particularidades del sistema de explotación de cómputo.

Esta fase esta formada por dos aspectos:

- a) la manera de como se va a combinar los módulos para formar programas, y
- b) el diseño de pruebas que permite identificar errores de codificación.

Dentro de la integración se encuentran los siguientes tipos:

a) Integración tradicional o no incremental.

Esta integración sigue la siguiente secuencia:

- escribir y probar cada módulo separadamente, es lo que se llama 'unit-test'.
- reunir y probar todas las unidades agrupadas en subsistemas es lo que se llama 'subsystem-test'.
- se combinan todos los subsistemas hasta obtener el sistema completamente integrado.
- probar el sistema.

b) Integración incremental.

Esta forma de integración permite detectar los errores en la intercomunicación de los módulos de manera fácil, ya que al integrarse un nuevo módulo a otros ya integrados, puede traer errores en la interconexión modular.

La implementación incremental se puede describir en el siguiente algoritmo:

REPETIR HASTA que el sistema este completo
probar un módulo
adicionar ese módulo a la combinación existente
probar la nueva combinación

FIN REPITE.

liberar el sistema.

Este tipo de integración se puede llevar a cabo por diversos caminos, con lo que se definen tres formas de integración incremental que son:

- Integración Top-Down incremental.
- Integración Bottom-Up incremental.
- Integración Sandwich incremental.

El sistema de Almacén se integró haciendo uso de la técnica de integración incremental Top-Down, considerando los módulos como los de más alto nivel, es decir se probaron por separado los módulos inferiores, integrándolos al módulo o módulos inmediatos superiores, y así hasta tener integrado totalmente el módulo de más alto nivel.

6.2. PRUEBAS

Esta es una de las etapas finales en la que al sistema desarrollado se le aplican una serie de pruebas, y al final de éstas los resultados respecto al funcionamiento y a los objetivos del sistema, deben ser totalmente satisfactorios.

A continuación se definirán los tipos de pruebas que existen:

- a) Pruebas de implementación. Que se aplican a cada versión del sistema, con el objeto de encontrar errores.
- b) Pruebas del sistema. Estas se llevan a cabo después de la integración del sistema y deben cubrir cada una de las facetas de éste.
- c) Pruebas de aceptación. Son muy importantes por que para ello se cuenta con el usuario final, que es quien tendrá la responsabilidad de probar el sistema. Y para tal existen dos técnicas diferentes:

- Pruebas de caja blanca.

Estas pruebas involucran la estructura interna del sistema ya que los casos de prueba se realizan a partir del diseño y código de los programas.

Estas pruebas permiten garantizar que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, así como todas las decisiones lógicas en los dos casos verdadero o falso, que se ejecuten todos los bucles en sus límites y que se ejerciten las estructuras de datos internas para asegurar su validez.

- Pruebas de caja negra.

Este tipo de pruebas nos permite diseñar los casos de prueba en función de las especificaciones del programa sin tener que consultar la lógica interna.

Estas pruebas permiten encontrar errores de las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento, errores de inicialización y de terminación.

A diferencia de la prueba de caja blanca, que se lleva a cabo previamente en el proceso de prueba, la prueba de caja negra suele ser aplicable en fases de prueba posteriores. Ya que la prueba de caja negra ignora la estructura de control y concentra su atención en el dominio de información.

Particularmente, en el sistema de "Almacén" se llevaron a cabo varios tipos de pruebas.

Primero pruebas modulares, lo que permitió depurar cada uno de los módulos que constituyen el sistema, detectandose algunos errores de sintaxis;

Después se continuó con la realización de pruebas integrales, que sirven aparte, para construir la estructura del programa mientras que al mismo tiempo se llevan a cabo pruebas para detectar errores asociados con la interacción. Uno de los objetivos de

construir la estructura del programa con los módulos probados por unidad, es que sea de acuerdo con lo que dicta el diseño. En estas pruebas se obtuvieron resultados satisfactorios.

El siguiente paso son las pruebas funcionales en las que se llevo más tiempo ya que se probaron todas las funciones de cada uno de los procesos.

Cabe mencionar que el sistema se probó paralelamente con datos reales de una dependencia, durante estas pruebas surgieron pequeños detalles y errores que se corrigieron inmediatamente.

6.3. DOCUMENTACION DEL SISTEMA

La documentación en un sistema es importante ya que por medio de ésta se podrán realizar modificaciones al sistema, la conversión de la información de un equipo a otro, la actualización de la información, etc; Debido a que con el paso del tiempo surgen nuevos requerimientos y no siempre la persona que desarrolla es quien los actualiza.

Una buena documentación permite una mejor comunicación entre quien desarrolla el sistema y el usuario del sistema, esta documentación básicamente se enfoca a la realización de una serie de manuales, que a continuación se mencionan :

El manual de organización.- consiste en presentar todos los diagramas de organización así como las características del sistema.

El manual de normas.- consiste en establecer los lineamientos y estructuras con las cuales debe trabajar el personal de mantenimiento.

El manual de procedimientos.- consiste en mostrar cada uno de los procesos que componen el sistema.

El manual de datos.- contiene la definición de los datos utilizados, como su nombre, tipo, longitud, valores máximo y mínimo que debe tomar y una lista de los archivos en donde son utilizados.

El manual de estandares de instalación.- consiste en los tópicos, técnicas de diagramación, lenguajes de programación, organización de los programas y reglas para los encabezados de página.

El manual de usuario y manual de operación.- estos manuales son de gran utilidad para el usuario ya que lo ayudan a operar el sistema, y deben presentarse sencillos y concisos.

El manual de usuario incluye las características de los programas así como la forma de utilizarlos.

El manual de operación incluye información más específica como: la estructura de los procesos y la forma en que deben instalarse.

7.1. MANTENIMIENTO

El mantenimiento es un conjunto de actividades de ingeniería del software que se dan una vez que el software se ha puesto en marcha. Podemos describir tres actividades que se deben llevar a cabo una vez puesto en marcha el sistema.

- 1a. Esta actividad surge debido a que no es razonable asumir que la prueba del software descubre todos los errores en los programas. El proceso que incluye el diagnóstico y la corrección de uno o más errores se denomina "MANTENIMIENTO CORRECTIVO".
- 2a. Esta actividad modifica el software para que interaccione adecuadamente con su entorno cambiante y es llamado "MANTENIMIENTO ADAPTIVO", sucede cuando por ejemplo: surgen nuevas versiones de sistema operativo, se modifican los periféricos, etc.
- 3a. Esta actividad se da cuando el software tiene éxito y a medida que pasa el tiempo los usuarios sugieren otras posibilidades o modificaciones de funciones existentes y sobre mejoras en general. Para satisfacer estas sugerencias se lleva a cabo el "MANTENIMIENTO PERFECTIVO".

Para adaptar o perfeccionar debemos determinar nuevos requerimientos, rediseñar, generar código y probar el software existente, es a esto a lo que se le llama MANTENIMIENTO. De hecho,

aproximadamente la mitad de todo el mantenimiento de software es perfectivo, tal y como se muestra a continuación:

Perfectivo 50%	21% Correctivo
	25% Adaptivo
	4% Otras

MANTENIMIENTO

Se realizó una evaluación del sistema de la cual se desprende lo siguiente:

Los diferentes usuarios detectaron detalles en el sistema los cuales fueron propuestos como modificaciones, pero estos detalles dependen en su mayoría de las necesidades y los datos particulares de cada uno de ellos; por lo que cabe mencionar que las modificaciones se realizarán únicamente si son en forma general, es decir que sean útiles o solicitadas por más dependencias ya que si no sería un verdadero problema tener que preparar una versión por cada una.

El presente sistema proporciona la solución a los problemas que el anterior presentaba, además proporciona las siguientes ventajas útiles al usuario:

- a) La posibilidad de manipular la información, como realizar un respaldo y/o restauración desde el sistema sin tener que realizarlo desde sistema operativo.
- b) Los reportes se obtienen en un menor tiempo, y la posibilidad de obtenerlos en cualquier momento.

- c) La rapidez de respuesta en los procesos, debido a la utilización de Clipper.
- d) El sistema es transportable y compatible con cualquier equipo PC, XT o AT.

Ya que el sistema ha cumplido con estas características y con los objetivos predefinidos al inicio, nos damos por satisfechos y con esto se concluye el desarrollo del sistema de Almacén para un microcomputador.

CONCLUSIONES

Hay que realizar una valorización de qué tanto los objetivos especificados inicialmente que impulsaron al desarrollo de este sistema para el control de los artículos de consumo a nivel dependencia en un microcomputador, han sido cubiertos.

Se puede decir que estos objetivos han sido cubiertos satisfactoriamente, la posibilidad de utilizar el sistema en forma sencilla por los usuarios, la forma en que el sistema brinda la posibilidad de obtener una serie de reportes y consultas en forma inmediata, la forma de vaciar la información de las notas de entrada y vales de salida tal y como se realiza en la Universidad y la forma en que su operación y uso pueden ser realizados plenamente gracias al manual de usuario.

Además el sistema se puede evaluar en los siguientes términos:

La operación del sistema en forma modular y estandar, garantizando así la consistencia de los datos, y la facilidad de realizar modificaciones o actualizaciones al sistema.

Los procesos de consulta y reportes proporcionan al usuario la información correcta, actualizada y concisa de manera rápida.

Satisface plenamente las necesidades de los usuarios que de alguna u otra forma están vinculados con el control de los artículos de consumo.

El sistema brinda seguridad en la información, con las opciones de protección y restauración, además es compatible y transportable.

Actualmente el sistema se implanto ya en más de ochenta dependencias las cuales han propuesto modificaciones, que han sido efectuadas, las cuales han cubierto las necesidades de los usuarios. Estas dependencias actualmente estan utilizandolo sin ningun problema.

Por lo anterior, se puede concluir que los resultados son satisfactorios y se han logrado las metas propuestas, que era el objetivo de esta tesis.

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

- 1.-INGENIERIA DE SOFTWARE
RICHARD FAIRLEY
EDIT. MC. GRAW HILL
MÉxico, 1988
- 2.-INGENIERIA DEL SOFTWARE "UN ENFOQUE PRACTICO"
ROGER S. PRESSMAN
EDIT. MC. GRAW HILL
España, Madrid 1988.
- 3.-MANUAL DE USUARIO SISTEMA DE ALMACEN
D.G.S.C.Ad.
DEPTO. DE ASESORIA E IMPLANTACION DE SISTEMAS
MÉxico, 1990.
- 4.-APLIQUE EL DBASE III PLUS
EDWARD JONES
ED. OSBORNE MC.GRAW HILL
- 5.-NOTAS DE LAS MATERIAS DE :
INGENIERIA DE PROGRAMACION
BASES DE DATOS