

Nº 63
2 EV.



**UNIVERSIDAD NACIONAL
AUTONOMA DE MEXICO**

FACULTAD DE CIENCIAS

**"BASE DE DATOS PARA EL CONTROL INTEGRAL DE BIENES
Y SERVICIOS EN UN LABORATORIO DE ELECTRONICA."**

**T E S I S
P R O F E S I O N A L
QUE PARA OBTENER EL TITULO DE:
A C T U A R I O
P R E S E N T A**

LAURA MARCELA SALMAR SERRANO LOPEZ

FALLA DE ORIGEN





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

BASE DE DATOS PARA EL CONTROL INTEGRAL DE BIENES
Y SERVICIOS EN UN LABORATORIO DE ELECTRONICA.

CONTENIDO	Página
1. Introducción	3
1.1 Antecedentes	6
1.2 Objetivos	7
2. Estructura de una Base de Datos	
2.1 Sistemas de Información	8
2.2 Sistemas de Bases de Datos	13
2.3 Tipos de Bases de Datos	21
2.3.1 Base de Datos Relacional	21
2.3.2 Base de Datos Jerárquica	25
2.3.3 Base de Datos de Redes	28
2.4 Manejador de Bases de Datos	31
3. Selección del Software y Hardware para el Desarrollo de la Base de Datos. Software.	
3.1 Cobol	36
3.2 DMSII	36
3.3 Facilidad para el Diseño de Pantallas Hardware.	48
3.4 Equipos Disponibles	53

4.	Configuración de la Base de Datos para el Control de Servicios en un Laboratorio de Electrónica.	
4.1	Diagrama de Bloques	54
4.2	Descripción de los diferentes módulos	54
4.3	Implementación	82
4.4	Forma de Operación del Sistema	89
5.	Conclusiones	91
6.	Bibliografía	94

1. INTRODUCCION:

La toma de decisiones es un factor fundamental en toda organización, algunas veces las situaciones que se presentan son complicadas e involucran a gran cantidad de personas, lo cual hace indispensable el hecho de contar con información que tenga un significado relevante y que se encuentre disponible en cualquier momento para que sea de utilidad. Esta información es la interpretación significativa de la correlación de datos.

En una empresa los tres elementos más importantes son los recursos económicos, el personal y los datos, éstos últimos representan la forma en que se conducen los proyectos y abastecen de información para la planeación, y si se manejan apropiadamente, se contará siempre con los recursos necesarios, actualizados como apoyo a una decisión.

Un Sistema de Manejo de Información permite obtener la información que se necesita. Da soporte a la planeación, control y función operativa, proporcionando información uniforme en el tiempo preciso para auxiliar al proceso de toma de decisiones.

Estos sistemas permiten recuperar y almacenar registros, formados por campos en un archivo. La limitación

de este tipo de sistemas es que no permite combinar fácilmente información de un archivo con información en otro archivo diferente, esto limita la combinación de datos a ser procesados y existe entonces la posibilidad de que la información requerida no pueda obtenerse. Haciendo uso de una Base de Datos este proceso se simplifica notablemente, además de brindar una serie de facilidades para utilizar diferentes opciones, cuya implantación no es complicada, como resultado puede obtenerse un sistema que puede contar con diversas características como son la concurrencia, el control de acceso a algunos usuarios, información en red controlada, acceso de información rápida, dependiendo de las necesidades que se tengan. Podemos considerar a una Base de Datos como un archivero central donde se encuentra toda la información que se necesita, es como un banco central de archivos que se organiza en forma lógica evitando repetir información, el hecho de tener la información centralizada hace más fácil la tarea de encontrar lo que cada uno necesita, lo cual resulta mejor que tener varios archivos dispersos con información duplicada. El objetivo de esta tesis es contar con un sistema que dé apoyo al control de servicios e inventario de el Laboratorio de Electrónica de la Dirección General de Servicios de Cómputo para la Administración (DGSCA), cumpliendo con las características más sobresalientes de una base de datos, ya que las tareas que es

capaz de realizar satisfacen los requerimientos que se tienen en dicho departamento, al utilizar las facilidades descritas anteriormente, dando así una solución rápida y eficiente a las requisiciones que se van presentando en las diferentes actividades que se tienen en el departamento. Estas tareas están involucradas con el mantenimiento de equipos de cómputo pertenecientes a las diferentes dependencias de la U.N.A.M., así como los equipos con los que cuentan, además de proporcionar información respecto a las soluciones que se dan a los problemas que éstos han ido presentando, proporciona también el inventario de herramienta y materiales que se tienen para así poder brindar un servicio adecuado.

1.1 ANTECEDENTES :

Anteriormente el control de servicios e inventario en el Departamento de Electrónica se llevaba mediante un proceso manual, en el que se elaboraban reportes escritos diariamente, éste era un proceso muy lento. Las necesidades crecieron debido al volumen y variedad de la información, las expectativas de los usuarios aumentaron, hasta que la demanda era tal que el manejo de la información no se tenía actualizado en el tiempo requerido, por lo que era necesario automatizar los procesos de consulta, actualización y validación de la información para así poder resolver estos problemas cotidianos. La creación de la Base de Datos para el Control de Servicios del Laboratorio de Electrónica (SERVI) ha logrado controlar el proceso de manejo de información, apoyando así la operación diaria al proporcionar información actualizada y ágil de los servicios, equipos y partes disponibles.

1.2 OBJETIVOS :

Los principales objetivos de este sistema son :

- Suministrar información de las fallas más frecuentes que se dan en los equipos para conocer la solución aplicable en cada caso.
- Disminuir el tiempo de atención de las fallas.
- Control de existencias y partes para su mejor utilización.
- Planeación de compras a futuro.
- Contar con información actualizada en el momento que se desee.
- Proporcionar al usuario una forma amigable y rápida de consulta de información.

2. ESTRUCTURA DE UNA BASE DE DATOS.

2.1 SISTEMAS DE INFORMACION.

Un sistema es un conjunto de elementos y procedimientos íntimamente relacionados para lograr uno o más propósitos en común.

Un sistema de información puede definirse como un conjunto de componentes que realizan operaciones de procesamiento de datos para proporcionar información en apoyo a diferentes actividades, satisfaciendo las necesidades que se requieran. La función principal de estos sistemas consiste en aumentar el conocimiento del usuario.

Los datos se manejan y deben situarse en un contexto significativo para que puedan ser útiles, para ello se necesita ordenarlos y así poder obtener resultados comprensibles. Deben efectuarse una serie de operaciones básicas como captar la información, verificarla, o sea asegurarse de que los datos han sido registrados en forma correcta, colocarlos en una secuencia específica predeterminada, efectuar los cálculos que sean necesarios, y almacenar la información en algún dispositivo para que esté disponible y pueda consultarse. Es muy importante copiar los datos de un dispositivo a otro para tener una mayor seguridad y para proporcionar información al usuario, ya que esta es la finalidad de todo sistema de información.

Existen algunas características que son comunes a muchos sistemas, una de estas es que están formados por distintos componentes que se encuentran relacionados y tienen interacciones definidas o interdependencias tal que, al hacer un cambio a cualquiera de estas partes se produce alguna modificación en las demás y en el sistema completo. Todas estas partes trabajan para alguna función u objetivo primario de los sistemas, éstos registran datos, los manipulan de alguna forma y producen resultados que sirven como base para realizar alguna acción.

El desarrollo de estos sistemas se hace en distintas fases, las cuales se describen en seguida :

Primero se perciben las necesidades y objetivos de manera informal.

En seguida se fijan el alcance y objetivos del sistema aún de manera imprecisa, pero habiendo ya un contacto entre las personas que desean el sistema y las que lo llevarán a cabo, estudiando las restricciones que existen para lograr lo que se desea.

El análisis es la tercera fase y la mas importante, es donde hay una verdadera comunicación entre el personal de la organización y el analista de sistemas que determina conjuntamente con el usuario, que es lo que hará el sistema, definiendo las salidas y la descripción del proceso.

La cuarta fase es el diseño que debe describir completamente el formato de los archivos a utilizar, sus características, funciones de los programas, recomendaciones sobre la utilización del sistema operativo, criterios para la optimización técnica del equipo. Aquí se elabora el desarrollo detallado de las aplicaciones, tomando en cuenta los recursos y limitaciones que se tienen, planeando el método de solución. El diseño puede clasificarse en tradicional y estructurado. Dentro del tradicional están los diagramas de flujo, en estos se siguen una serie de procedimientos en línea, son representaciones gráficas de los procedimientos usados en un programa en forma detallada para la realización del procesamiento de datos, a través de símbolos que son sencillos de entender, a diferencia del estructurado, que utiliza reglas para dividir funcionalmente la programación en módulos, haciendo que los programas sean fácilmente actualizables. En la mayor parte del diseño que utiliza programación estructurada, el almacenamiento de los elementos de información se hace usando bases de datos, ya que parece ser lo más apropiado, éstas se usan para denotar datos coleccionados y conservados en uno, dos ó mas archivos que pueden estar coordinados y ser utilizados por uno ó más sistemas de información. El resultado del diseño son una serie de procedimientos que el programador

convierte en un lenguaje que sea entendido por la computadora, aquí es donde da inicio la fase de programación, a la que le siguen las pruebas en paralelo para detectar si ya no existen fallas, y en su caso corregirlas, para alimentar al sistema ya de una manera formal y darle mantenimiento.

Para poder definir claramente el problema de información que se tiene, se deben conocer la situación actual, lo que se desea y las limitantes que existen, y en esta forma especificar lo que se va a hacer, en que forma y el costo que tendrá. Antes de implantar un sistema nuevo, se debe de determinar su viabilidad, desde los puntos de vista operativo y económico. Un sistema es viable operativamente si aumenta la eficiencia, por ejemplo búsqueda de los datos en el menor tiempo posible, el poder almacenar mayores volúmenes de información, y económicamente, si incrementa la utilidad reduciendo los costos. Los factores del costo son el del equipo, el trabajo empleado en el desarrollo del sistema, el de espacio y control de las funciones ambientales y el de las operaciones.

Se deben analizar diferentes caminos para llegar al objetivo, haciendo una estimación de lo que implica la implantación y operación del sistema en relación con los beneficios que va a aportar. En algunos casos sucede que el costo es igual o mayor que los beneficios, pero se acepta

por el factor tiempo.

Antes de la aparición de las Bases de Datos, la mayor parte de los archivos se organizaban en forma secuencial simple, el software ejecutaba operaciones de entrada y salida de los dispositivos de almacenamiento. No había independencia de datos, entonces si se modificaban ya sea la organización de éstos o si se cambiaban los dispositivos de almacenamiento, se tenían que reescribir los programas, repitiendo los procesos de compilación y depuración. La mayor parte de los archivos que se tenían servían para una sola aplicación, y en caso de que los mismos datos sirvieran para otras aplicaciones, pero organizados de diferente forma, se tenían que crear otros archivos con los mismos datos, lo cual llevaba a un alto índice de redundancia. Las Bases de Datos son herramientas que pueden utilizarse en el diseño de sistemas de información utilizando generalmente técnicas de programación estructurada.

2.2 SISTEMAS DE BASES DE DATOS.

Una Base de Datos es una colección de archivos integrados que pueden relacionarse entre sí por medio de los registros de los diferentes archivos. Estos registros se pueden definir como un conjunto de campos, son conjuntos lógicos de caracteres que contienen toda la información relativa a una partida dentro de un archivo. Los campos se componen a su vez de uno o más caracteres y contienen una pieza individual de información. Una base de datos puede considerarse también como una entidad global que puede ser accedida y actualizada por diversas aplicaciones en forma concurrente y todos los datos concernientes a una aplicación en particular deben concentrarse en una sola base de datos. Contienen una serie de características diseñadas para dar soporte a una aplicación.

Para poder calificar a dos ó más archivos como una base de datos, deben tener uno ó más campos en común y en el caso más simple, todos los registros que la conforman son accesibles con el mismo registro llave, es decir un campo que hace único al registro, o sea, permite identificarlo de los demás, así cualquier registro en cada uno de estos archivos puede ser actualizado o accedido por medio de la llave común que se haya definido, en cuyo caso la estructuración del sistema debe evitar contener registros muy largos.

Existe otro tipo de estructura de Base de Datos que es mas compleja y comprende la relación de algunos archivos de datos pero que tienen diferentes llaves de acceso.

La creación del tipo de Base de Datos compuesta por algunos archivos relacionados tiene dos objetivos principales, evitar lo mas posible la duplicación de datos, y optimizar el acceso a la información, para este último punto puede hacerse el almacenamiento de campos individuales en una sola localización dentro del sistema completo, y al hacer esta reducción la cantidad total de almacenamiento de archivos y el costo en la utilización del equipo se reducirán notablemente.

Las bases de datos tienen tres distintos niveles de abstracción, el nivel físico, el nivel conceptual y las diferentes vistas o representaciones, éstos deben poder modificarse en forma independiente, o sea que los cambios a uno de ellos no deben afectar otro nivel. El nivel físico se refiere a como se registran los datos en el medio de almacenamiento, o sea como están arreglados físicamente, esta representación es creada automáticamente por el manejador de bases de datos. Mediante el esquema o diseño conceptual se pueden describir y nombrar todos los datos del sistema, definiendo de que tipo son, como es su longitud y dominio. Contiene la base de datos completa en términos de unas cuantas estructuras.

En ocasiones no se desea que todos los usuarios tengan acceso a la base de datos completa y para ello se crean los subesquemas o vistas que son subconjuntos de los esquemas y describen el modelo de una aplicación en particular, al seleccionar y modificar partes del esquema para satisfacer a determinados usuarios.

En estos sistemas intervienen una serie de componentes que interactúan para satisfacer distintas necesidades, y son el Hardware, Programas, Datos, Gente y Procedimientos.

En cuanto a el Hardware, estos sistemas requieren gran cantidad de memoria principal, así como una unidad de procesamiento central bastante rápida además de un acceso inmediato al almacenamiento.

Los programas que intervienen son de diferentes tipos, cuando los usuarios hacen, por ejemplo, alguna requisición, ésta es enviada a la computadora, la cual procesa a través de líneas de comunicación, un programa de control de comunicaciones se encarga de enviar y recibir las requisiciones, coordinando las actividades de la terminal, es el que guía los mensajes a su destino siguiente y ejecuta además otras actividades de comunicación orientada. Los programas de aplicación son creados para satisfacer necesidades específicas. Los programas de utilerías generalmente son proporcionados por el manejador

de la base de datos y se ocupan de diferentes servicios, como por ejemplo crear y mantener la base de datos, contar con una recuperación generalizada de la información, obtener reportes con una mínima programación, almacenar y recuperar datos del sistema, reformatar y limpiar archivos, etc. El Sistema Operativo es el programa que controla los recursos de la computadora, hace que se ejecuten los servicios solicitados por los programas. Controla todos los recursos que los programas solicitan.

La alimentación de los datos puede ser en línea o mediante un proceso llamado "batch", en este no se utiliza un sistema de comunicación, sino que los datos se coleccionan en grupos que se procesan uno tras otro, por lotes, y se envían directamente a los programas de aplicación.

La Base de Datos es descriptiva por ella misma, ya que contiene la descripción de su estructura, lo cual hace posible la independencia entre programas y datos, difiere con el procesamiento de archivos en que la estructura de éstos no tiene que estar presente en los programas de aplicación. Esta descripción de los datos es una de las funciones más importantes, mediante la cual los requerimientos de datos de los programas de aplicación se comunican con el manejador de bases de datos. Aunque la base de datos haya sido diseñada para ser usada por un solo sistema de información ésta debe ser una entidad

separada y distinta, aparte de cualquier aplicación particular, entonces, sus archivos son un conjunto de datos comunes que está disponible a cualquier programa de aplicación. La independencia física y lógica que contienen este tipo de sistemas es un factor muy importante ya que con la primera es posible modificar el esquema físico sin tener que escribir nuevamente los programas de aplicación y con la segunda se puede modificar el esquema conceptual sin tener que reescribir los programas. Otra diferencia importante con el procesamiento de archivos es que en éstos, los registros se agrupan generalmente juntos, teniendo así que el agrupamiento físico y lógico sean los mismos, en las Bases de Datos estas colecciones pueden ser diferentes.

La aplicación del concepto de bases de datos al diseño de sistemas de información implica generalmente un acceso a la información de tipo aleatorio, aunque no es un requerimiento indispensable, ya que la forma en que se pueden procesar es variable, puede ser secuencialmente dentro de un archivo, aleatoriamente por el valor de un campo o mediante relaciones con otros registros. El orden en que aparecen los registros es el que parece convenir al manejador, si uno no le indica en que forma lo desea, y en caso de no hacerlo así, los registros pueden ordenarse externamente utilizando una llave. El procesamiento a través de relaciones permite obtener todos los registros referidos

a una misma situación.

El personal que utiliza las Bases de Datos son los clientes, para quienes el sistema fue desarrollado, ellos no participan activamente, son a quienes se entregan los resultados. Los usuarios emplean el sistema para satisfacer las necesidades de los clientes. El Personal de Operación se encarga de operar la máquina y del control de los datos. El Personal de Desarrollo de Sistemas diseña e implementa el sistema determinando los requerimientos y alternativas, son analistas de sistemas y programadores, le dan soporte y mantenimiento al sistema operativo, al programa de control de comunicaciones y al manejador de la base de datos . El personal de Administración controla el diseño y uso del sistema. Los usuarios y el personal de operación requieren de algunos procedimientos a seguir en caso de fallas en el sistema.

En una Base de Datos es indispensable contar con un Diccionario de Datos, el cual cataloga todos los nombres de datos, así como su significado, permitiendo a los usuarios investigar el contenido de los nombres que se están utilizando, con lo cual no se tendrán diferentes definiciones para un mismo nombre.

El tiempo de respuesta de estos sistemas debe ser inmediato, debe contar con algunos requisitos de

seguridad en caso necesario, y debe además ser un proceso concurrente. La seguridad se refiere a que el sistema debe estar protegido contra accesos no autorizados que pueden tener fines indebidos como pueden ser lecturas no permitidas, o bien modificación o destrucción de los datos y contra pérdidas accidentales como pueden ser cortes de corriente eléctrica durante el procesamiento de las transacciones, o bien anomalías debidas a un acceso concurrente. Para prevenir el uso mal intencionado, pueden usarse "candados" de seguridad para los diferentes procesos, se le puede permitir a determinado usuario la lectura de la información, pero sin opción de hacer modificaciones, o se le puede permitir la modificación pero no la eliminación, o bien, se le puede permitir la creación de nuevas relaciones, o agregar y eliminar atributos, etc.

La concurrencia se refiere a la ejecución de varias transacciones al mismo tiempo, compartiendo al procesador. Esto mejora la eficiencia total del sistema pues puede trabajarse más en menos tiempo. Cuando se está realizando una transacción, ésta llega a un punto en el que debe esperar a que se termine una operación que generalmente es de entrada o salida, en un sistema uniprogramado, el procesador permanece sin hacer nada, es un tiempo de espera que se desperdicia, en cambio con el multiproceso se aprovecha ese tiempo, con lo cual se tiene una productividad total de las transacciones. Los sistemas

interactivos se componen en general de varias transacciones cortas, y el usuario espera el resultado de cada una, entonces el tiempo de respuesta debe ser muy corto, solo se requiere una porción de tiempo muy pequeña del procesador para cada usuario. El sistema cambia muy rápido de usuario, entonces se tiene la idea de que cada uno está trabajando en una computadora diferente, cuando en realidad es solamente una, que se está compartiendo. Si solo se permitiera una requisición a un tiempo para acceder la base de datos, todas las demás tendrían que esperar hasta que la primera fuera terminada y se formaría un cuello de botella. Si las requisiciones llegaran más rápido de lo que pueden ser procesadas secuencialmente, entonces el tiempo que esperan antes de ser procesadas aumenta y por lo tanto el tiempo de respuesta se retardaría mucho, y podrían darse problemas de inconsistencia en la información.

La base de datos debe crear un ambiente tal, que permita una forma adecuada y sencilla de recuperación y almacenamiento de información.

2.3 TIPOS DE BASES DE DATOS.

2.3.1 BASE DE DATOS RELACIONAL.

Las Bases de Datos Relacionales representan la Base de Datos como un conjunto de tablas que tienen asignado un nombre único, donde las columnas representan relaciones entre un conjunto de valores. Estas tablas de datos pueden verse como relaciones, en las cuales las columnas se llaman "atributos" y las filas o registros "tuples".

Una relación es una correspondencia o mapeo entre los miembros de dos conjuntos, es una asociación entre un conjunto de entidades, y un atributo es la función que mapea este conjunto a un cierto dominio.

Las relaciones pueden ser de 1 a 1, (de uno a uno), de 1 a N (de uno a muchos), ó de N a M (de muchos a muchos). Estas se ajustan a distintos tipos de estructuras y entre ellas están los archivos planos, en los cuales cada registro tiene un conjunto similar de campos y se pueden representar como matrices bidimensionales en las que no existen grupos repetitivos, son de medida fija y se pueden construir organizaciones secuenciales, indexadas o directas, se considera que poseen una forma simple de almacenamiento y en esto se basa la base de datos relacional.

Las relaciones se establecen de acuerdo a las necesidades del usuario y dependiendo de sus exigencias es

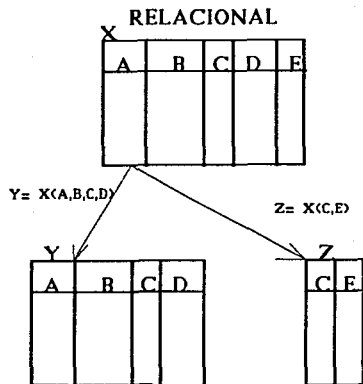
necesario extraer de las tablas los subconjuntos pedidos para crear otras de mayor o menor grado. Estas operaciones proporcionan un grado de flexibilidad que no se puede lograr con la mayor parte de estructuras de árbol o red. Las relaciones pueden tener varias llaves, pero una es designada como primaria y no puede tener un valor indefinido debido al papel tan importante que ésta tiene en el proceso de búsqueda. El lenguaje de datos relacional permite formar nuevas relaciones de las ya existentes, algunas tienen una existencia independiente, se definen inicialmente y los registros ("tuples") pueden insertarse en forma independiente, se les llama relaciones primarias, en cambio las relaciones definidas que usan operadores relacionales son relaciones derivadas, después de que fueron creadas pueden también tener una existencia independiente. Este lenguaje debe permitir el relacionar cualquier atributo de una relación con otros compatibles en otras relaciones para obtener nuevas relaciones derivadas.

Las relaciones se ajustan a lo que matemáticamente es una relación, un subconjunto del producto cartesiano de sus correspondientes dominios, pero en las Bases de Datos éstas difieren de este concepto, solamente en el hecho de que varían con el tiempo, al insertar, borrar o modificar registros ("tuples").

La representación lógica, utilizando una estructura relacional es a través de tablas bidimensionales con operaciones para extraer y cambiar columnas. (figura 1).

EJEMPLO DE UNA BASE DE DATOS

23



(Fig. 1)

En este ejemplo aparece una de las operaciones que pueden utilizarse, llamada Proyección y denotada con el símbolo π . En esta base de datos existe una tabla llamada X, de la cual se proyectan los atributos A,B,C y D, para crear una subtabla llamada Y, de igual forma se crea otra tabla llamada Z, al proyectar C y E.

Para describir las relaciones y operar con ellas se puede hacer uso del Algebra ó Cálculo Relacional, aunque pueden usarse otros métodos que lleven a la obtención del mismo resultado. El Algebra Relacional es un lenguaje de consulta donde el usuario debe proporcionar al sistema una secuencia de operaciones para obtener el resultado que desea. Cuenta con cinco operaciones fundamentales que son : elegir, proyectar, producto cartesiano, unión y diferencia de campos. El Cálculo Relacional es un lenguaje donde se da la descripción de la operación que se desea, sin especificar como obtenerla.

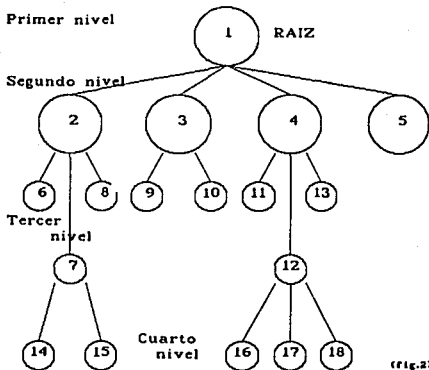
2.3.2 BASE DE DATOS JERARQUICA.

Consiste de un conjunto de registros que se conectan entre sí por medio de relaciones llamadas ligas, en esto son similares a las bases de datos basadas en modelos de red, pero difieren de éstas en que los registros se organizan para formar conjuntos de árboles en vez de gráficas arbitrarias (figura 2).

Las estructuras de árbol están compuestas por una jerarquía de elementos llamados nodos en donde el nivel más alto tiene un solo nodo y se llama raíz, todos los nodos están conectados a otro de nivel más alto que se conoce como propietario o padre, pero en un nivel más bajo todo elemento puede tener uno ó mas elementos relacionados, y son los miembros o hijos, así, las ligas están restringidas a ser de 1 a 1 (uno a uno) ó de 1 a N (uno a muchos). Estas estructuras lógicas marcan relaciones entre los registros, y se pueden describir como una jerarquía de nodos con relaciones bimodales, de tal forma que el nivel más alto de la jerarquía tiene un solo nodo, y los restantes se reparten en cero ó más conjuntos disjuntos (no conectados), cada uno de los cuales constituye a su vez un subárbol. Cada subárbol disjunto constituye un árbol de la Base de Datos y consiste de un registro que es la Raíz, donde todos los demás registros son dependientes de éste. En estas bases de

EJEMPLO DE UNA ESTRUCTURA TIPO JERARQUICA

26



(fig.2)

datos el orden relativo de los subárboles es un factor importante y restrictivo, en ellos los nodos representan registros tipo y las ligas enlaces que solo pueden existir entre dos nodos en un árbol. Cada registro miembro no puede existir sin un registro propietario, que debe ser único, por lo cual las relaciones entre padre e hijo solo pueden ser de uno a muchos o de uno a uno, de ahí se dice que existe una correspondencia simple de miembro a propietario y una correspondencia compleja a la inversa. Así, al insertar un registro nuevo, puede conectarse a lo más a un registro existente. En general en las Bases de Datos Jerárquicas puede haber un número variado de ocurrencias de cada registro-tipo en cada nivel de un árbol. La raíz es el único registro tipo que tiene una ocurrencia independiente. El diagrama de estructuras de árbol es el esquema para representar a las bases de datos jerárquicas y especificar la estructura lógica general del sistema. Se considera que este modelo es fácil de entender, ya que proporciona al usuario pocos comandos por lo que el manejo debe resultar menos complicado, además, debido a la limitación que existe en los tipos de relaciones que permite, su implementación puede resultar mas sencilla que las de otros modelos, aunque presenta algunas desventajas entre las cuales se puede encontrar que operaciones como inserción o borrado pueden complicarse debido al ordenamiento estrictamente jerárquico.

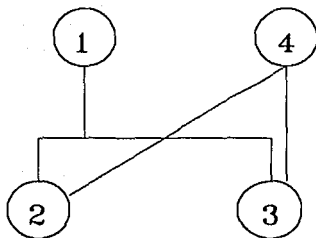
2.3.3 BASE DE DATOS DE REDES.

El modelo de red es distinto al modelo relacional en que el primero se expresa por medio de registros y las relaciones entre los datos es a través de ligas. Todo registro es un conjunto de campos o atributos, y cada uno contiene el valor de un dato, siendo las ligas las asociaciones entre ellos. Es un modelo formal para representar relaciones de atributos de un conjunto de entidades y la asociación entre éstos. Consiste de registros tipo y de ligas, un objeto representa una entidad y una liga es una asociación entre entidades. Estas estructuras permiten que cualquier componente se vincule con cualquier otro, o sea que las relaciones pueden ser de 1 a 1 (uno a uno), de 1 a N (uno a muchos), o de N a M (muchos a muchos) registros, esto es, un registro tipo puede participar en muchos enlaces con otros registros tipo, lo cual puede describirse en términos de padres a hijos, como se puede observar en la figura 3, aquí, cada hijo tiene 2 padres, así, al insertar un nuevo registro, debe estar conectado a algunos ya existentes. Aquí los registros pueden tener una existencia independiente, o sea que un registro no tiene que ser conectado a otro aunque se haya definido una liga entre ellos. Al borrar un registro implica el borrado de todos los que están conectados a él, lo mismo que sucede en una base de datos jerárquica. El modelo de datos de redes

EJEMPLO DE UNA ESTRUCTURA

29

TIPO RED



(Fig. 3)

permite que los registros puedan ser borrados, insertados, modificados y recuperados, así como también permite hacer conexiones entre registros a ser creados o removidos de acuerdo a las ligas entre los registros tipo.

2.4 MANEJADOR DE BASES DE DATOS.

El Software que se utilizó para la creación de este sistema incluye un manejador de bases de datos.

Un sistema manejador de bases de datos es un sistema especializado de software que se usa para describir una base de datos y las relaciones entre los datos que la conforman. Estos sistemas cuentan con algunas facilidades, almacenan los datos que las aplicaciones requieren en una o mas bases de datos. La vista o estructura que se le quiere dar a la aplicación depende del modelo de datos que se haya elegido.

Las facilidades provistas por un manejador de bases de datos en general son para dos tipos de usuario, el administrador, y los usuarios. El administrador es el encargado de definir esquemas y subesquemas, además de crear y mantener las estructuras físicas de almacenamiento que les dan soporte a éstos. Es el que toma las decisiones en lo referente al manejador de bases de datos, ya que los usuarios únicamente se involucran con los aspectos lógicos de la organización de los datos y con su acceso. Los usuarios utilizan las características que contiene el manejador tales como el lenguaje de datos para poder tener el acceso a la información, este lenguaje consta de un conjunto de operaciones que permiten acceder la información en la base de datos, esta facilidad de

manipular los datos es el principal enlace entre el usuario y el manejador y consiste de algunos comandos primitivos que corresponden a las operaciones que pueden realizarse. Estos comandos ejecutan operaciones básicas con la información así como el tipo de respuesta que se hará, recuperando los datos y regresándolos al medio ambiente del programa, cambiando y borrando datos o haciendo algunas otras operaciones. Los tres tipos de comandos primitivos son de control, recuperación y modificación, siendo los principales los dos últimos. Los comandos de control se refieren a la identificación de una aplicación, determinando que base de datos se desea acceder, distribuyendo los recursos del sistema tal como los archivos y los "buffers" y estableciendo el tipo de comandos que una aplicación puede utilizar. Con los comandos de recuperación se obtienen valores de los datos y en algunos casos de las relaciones entre ellos. Los comandos de modificación cambian los valores de los datos o de las relaciones. Antes de realizar estas operaciones, se deben de seleccionar los datos, identificando los subconjuntos que se requieren mediante las instrucciones que da el usuario a través de algunos valores o de relaciones entre los datos.

Para poder utilizar el lenguaje de datos, se requiere de un lenguaje de programación, ya que es el que da soporte al primero y el programa de aplicación puede contener los comandos del lenguaje de datos como una parte

integral, o pueden tomar la forma de un procedimiento de llamada. Los lenguajes de datos permiten una gran flexibilidad en el manejo de datos, pero el formato que utilizan es rígido, algunos usan más procedimientos que otros, ya que existen lenguajes en los que solamente se especifica qué es lo que uno desea obtener y el sistema decide cómo proporcionar los datos y optimizar la búsqueda, en otros es necesario decirle al sistema manejador de bases de datos lo que tiene que hacer, paso por paso.

El lenguaje de datos procesa partes específicas de la base de datos, antes de hacer cualquier operación, los datos deben ser seleccionados, esto es, al recuperar la información, la localización se hace a través de la especificación de algunos valores que el usuario solicita y mediante un proceso de selección, se encuentra el conjunto de datos requerido, es el proceso de identificar un subconjunto de los datos en la base de datos. El criterio de acuerdo al cual el sistema manejador elige los datos consiste de un conjunto de condiciones conectadas por operadores booleanos, éstas constan del nombre de una unidad lógica de datos, un operador condicional y el valor de un dato. La selección se hace de acuerdo al contenido de los datos, y una vez hecha ésta, los datos relacionados con los que fueron seleccionados pueden obtenerse en forma inmediata.

La identificación de los registros o proceso de búsqueda se puede definir como un mapeo :

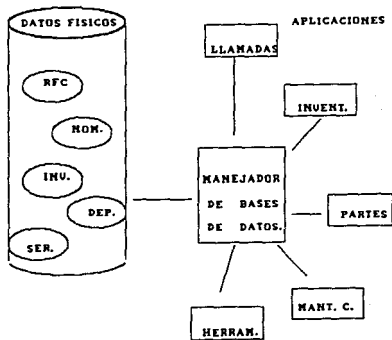
$$M : V - (i_1, i_2, i_3, \dots, i_n)$$

Dado un valor V para un elemento de datos (atributo) el mapeo debe dar él o los registros (tuples) de los identificadores o direcciones $i_1, i_2, i_3, \dots, i_n$ de los campos que tienen el valor V como su contenido.

Dentro de las funciones del manejador de bases de datos se encuentra la definición de éstos, con ella se comunican los requerimientos del programa de aplicación con el manejador , el cual actúa como intermediario entre las aplicaciones y los datos físicos (figura 4). Otras de las funciones que contiene son el almacenamiento, recuperación y actualización de la información, debe permitir el acceso simultáneo a los datos, en tal forma que el resultado de dos requerimientos a la vez sea como si éstos se hubieran pedido por separado.

Como ya se mencionó, una de las razones principales para hacer uso de un manejador de bases de datos es la independencia entre sus representaciones física y lógica. Para asegurar esto se deben de tomar en cuenta las representaciones que los usuarios desean utilizar para sus distintos requerimientos, así como la descripción detallada de la estructura física de los datos.

MANEJADOR DE BASES DE DATOS



(Fig. 4)

3.- SELECCION DEL SOFTWARE Y DEL HARDWARE PARA EL DESARROLLO DE LA BASE DE DATOS.

SOFTWARE.

3.1 COBOL.

Es el lenguaje común orientado a los negocios, es uno de los lenguajes de programación accesibles para dar soporte al lenguaje de datos que se ha utilizado. Permite escribir los procedimientos que la computadora debe seguir en términos parecidos al inglés, además de que disminuye los costos de cambiar computadoras, ya que un programa escrito en COBOL puede transportarse a casi cualquier computadora realizando solamente pequeños cambios.

3.2 MANEJADOR DE BASES DE DATOS DMSII.

(DATA BASE MANAGEMENT SYSTEM II).

El acceso al DMSII es mediante instrucciones que pueden declararse en programas escritos en ALGOL, COBOL, o PLI, y todas ellas son incorporadas en los compiladores para el lenguaje huésped, siendo así innecesario preprocesar estas declaraciones. Cada declaración de manejo de datos que es ejecutada por algún programa, invoca a algunas rutinas para que ejecuten todas las funciones de manejo de archivo que se requieran. Esto ayuda a que el programador no tenga que involucrarse con manejo de archivos, conversión de llaves, etc.

El DMSII tiene como características las siguientes funciones :

1) Tiene definido un lenguaje llamado DASDL (DATA AND STRUCTURE DEFINITION LANGUAGE). Este permite hacer la descripción de los datos que se van a manejar.

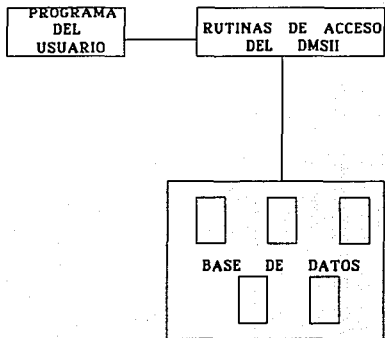
2) Proporciona Rutinas de Acceso que contienen funciones específicas, las cuales permiten a muchos usuarios el manejo de la información en forma concurrente, al usarlas los registros pueden ser añadidos, modificados y borrados, por medio de instrucciones sencillas que acceden a estas rutinas, se pueden dar de alta registros, darse de baja y hacer cambios a los mismos sin hacer grandes esfuerzos de programación. Se encargan del manejo físico y lógico de los datos, o sea del acceso directo a los archivos y de la extracción de un registro contenido en algunos archivos. (figura 5).

3) Proporciona Rutinas de Verificación y Recuperación.

Son rutinas de autoverificación y autodiagnóstico, las cuales se encargan de reconocer las ligas físicas que existen entre los archivos y registros que componen la Base de Datos. Cuando existen fallas en los procesos que causan una terminación anormal, este procedimiento es cancelado mediante una rutina de acceso llamada " abortar " . Este sistema puede ser recuperado usando unas cintas de respaldo

RUTINAS DE ACCESO

38



(Fig. 5)

producidas por alguna utilería y el diagnóstico dado por las rutinas de acceso.

4) Establece un lenguaje de Interconexión con los lenguajes huésped. Es el enlace que permite que programas escritos en COBOL, ALGOL o PLI puedan tener acceso a la Base de Datos.

5) Hace posible la Reorganización de la Base de Datos. Permite hacer cambios al formato de una Base de Datos ya existente, puede usarse para reordenar y consolidar estructuras de datos (sets) y subestructuras (subsets) automáticamente.

6) Construye un Archivo de Control. Este archivo contiene el registro de impresiones de fecha y tiempo, se usa para asegurar que los archivos de la Base de Datos se están utilizando.

7) Investigación. Permite realizar consultas de los datos o información del sistema a través de un programa separado de éste. También se encarga de proporcionar un método conveniente mediante el cual el usuario puede examinar la información vía terminal (en línea).

Componentes de la Base de Datos DMSII.

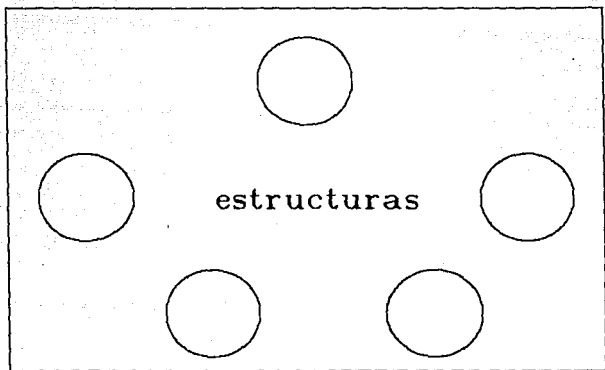
Estructura (data set) .- Es una colección de registros de datos relacionados en un archivo y puede ser de acceso aleatorio. Es parecido a un archivo convencional, contiene elementos de datos y tiene propiedades físicas y lógicas similares a ellos, pero difiere en que puede contener otras estructuras (data sets) y subestructuras (sets). En la figura 6, el cuadro representa la base de datos y los círculos las estructuras, o sea los conjuntos de registros que la conforman.

SUBESTRUCTURA (set).- Es quien permite el acceso a todos los registros de una estructura en alguna secuencia lógica, contiene una entrada para cada uno de ellos, es como un índice que localiza los datos. En la figura 7, cada círculo representa un registro y las flechas indican la dirección del recorrido que la subestructura puede tomar. Si las llaves se especifican en la estructura, el acceso de los registros se basa en las llaves de recuperación declaradas, de otra forma el acceso es secuencial. Para una sola estructura se pueden declarar muchas subestructuras, esto permite que los datos en una estructura puedan accederse en diferentes secuencias.

Campo (data item).- Es un campo en un registro, usado para contener una pieza individual de información.

BASE DE DATOS

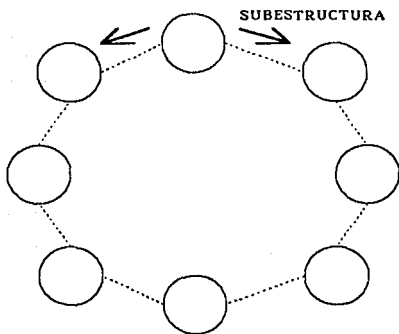
41



(Fig. 6)

ESTRUCTURA

42



(Fig. 7)

INFORMACION GLOBAL (global data).- Contiene información tal como totales y poblacionales de control de registros accedidos, modificados, etc. con el fin de autocontrolarse y autodiagnosticarse. Esta información se aplica a los sistemas completos.

El programa de aplicación para la recuperación de información, presenta los datos al usuario en un formato claro, este puede tomar distintas formas, una de ellas es leer a través de una ó mas estructuras enteras y generar una línea de reporte para cada uno de los registros que contiene. Otra es la búsqueda de registros dependiendo de las requisiciones que se han metido vía terminal, y una tercera forma de consulta es cuando se prepara un archivo para actualizar requisiciones que sea la entrada a un programa de actualización y validar estos datos en los registros que ya se tienen en existencia. Los comandos de recuperación disponibles en los lenguajes huésped pueden dividirse en dos categorías, los de localización y los de selección.

El proceso de trabajo de una Base de Datos sigue una secuencia típica:

- 1 Abrir.
- 2 Recuperación de datos.
- 3 Cerrar.

Cuando un programa de aplicación abre un sistema de este tipo, el sistema abre sus archivos físicos, localiza "buffers", etc. y el DMSII establece una trayectoria para cada estructura que es invocada, la cual es como un apuntador al registro particular de la estructura, que el programa está utilizando. Funciona en forma análoga al apuntador de registro estándar que se mantiene cuando un sistema abre un archivo de datos ordinario, al abrir el sistema cada trayectoria apunta al inicio de su estructura asociada. Si el programa procede a realizar la búsqueda en forma de serie a través de las estructuras, manejará las trayectorias igual que lo haría un apuntador de registro para un archivo de datos procesado secuencialmente, éste avanzará una posición de registro para cada registro accedido, si el programa hace la búsqueda brincando de registro a registro, entonces la trayectoria se manejará como la que se usa para un archivo accedido en forma aleatoria, por lo cual puede decirse que es poca la diferencia entre el manejo de una trayectoria DMSII y el apuntador de registro para un archivo de datos simple. Cuando la trayectoria se encuentra al inicio de cualquiera de las estructuras, la única forma de recuperar los datos es hacia adelante, y al llegar al final del archivo, puede hacerse la recuperación en sentido inverso, lo cual no

está permitido en archivos convencionales. Cuando una estructura se abre, al programa se le da acceso a diferentes archivos, a la estructura que se está tratando y a las asociadas a la primera, cada subestructura declarada contiene índices que apuntan a los registros de la estructura, que son almacenados en las secuencias marcadas por la subestructura que se esté tratando. El programa puede hacer la lectura de la subestructura en forma secuencial, y recuperar los registros en esta secuencia, también puede leerlos directamente y recibirlos en una secuencia no especificada. La interacción entre los programas de COBOL y DASDL es simple, ya que los elementos de la estructura forman parte del programa en COBOL. Cuando este programa contiene invocaciones a la Base de Datos y se compila, el compilador de COBOL llama al archivo de Interfaz de la Base de Datos para obtener toda la información de la Base invocada, éste regresa al compilador todos los nombres y descripciones de las estructuras, y son a su vez añadidos directamente a la lista de nombres de datos declarados en el programa en COBOL, se consideran como variables del programa que hubieran sido declaradas a través de alguna librería de copiado.

La estructura es equivalente a un archivo, y sus entidades son las subestructuras, y para hacer referencia a ellas se usan declaraciones de extensión del DMSII.

El lenguaje DASDL considera los componentes de una Base de Datos como una estructura jerárquica creciente, donde ésta juega el papel del elemento supremo, siendo la estructura la subordinada inmediata, que a su vez puede ser propietaria de algunos campos. Existe una pertenencia en cadena por parte de la base de datos hacia cada elemento de datos definido, y el manejador explota esta relación jerárquica para permitir mayor libertad en el nombrado de los elementos de datos. Los campos se usan como cualquier nombre de datos definido en el programa. Cuando en el programa se abre una de las estructuras, los registros son almacenados en una secuencia arbitraria que depende del programa de aplicación. Cada subestructura contiene índices que apuntan a los registros de la estructura. La búsqueda se puede hacer directamente a la estructura o bien a la misma pero vía una subestructura. Cuando dos o más programas de aplicación abren la Base en forma concurrente, se establece una subestructura de trayectoria para cada programa, cada uno de éstos realizará sus funciones como si fuera un solo programa el que esta accediendo información, ya que el manejador se encarga de las dificultades que implica el acceso simultáneo. Cuando el programa de aplicación toma un registro existente, se le da una copia de éste para que lo maneje como quiera, y se deja el registro original en

la base de datos hasta que sea alterado por un comando de borrado o de almacenamiento. En cada declaración de manejo de datos que ejecuta el usuario, se llama a las rutinas de acceso para ejecutar las funciones que se necesitan.

3.3 FACILIDAD PARA EL DISEÑO DE PANTALLAS.

(SCREENDESIGN FACILITY).

Para la construcción de este sistema se utilizó también un paquete de aplicación de Unisys, conocido como "Facilidad para el diseño de pantallas" (SDF), el cual es una herramienta para ayudar a los programadores a diseñar y procesar formas para sus aplicaciones de una manera simple y eficiente. SDF presenta un conjunto de pantallas y menús que se usan para crear formas interactivas. Las primeras son imágenes que son desplegadas en la terminal y presentan información a ser seleccionada, en tanto que los menús son imágenes que presentan opciones para actividades o procesos de sistemas. Las formas pueden crearse, diseñándolas con textos literales y delimitadores de campos en la forma en que uno lo desee.

SDF consta de un formato en el cual se teclea la opción de lo que se desea hacer, y en el menú principal se pueden seleccionar actividades que se encuentran en diferentes categorías :

- Librerías de Formas
- Formas
- Campos
- Generación
- Utilerías

Una de las cosas importantes al crear formas cuando se está usando SDF es que ningún campo ni forma puede existir por sí misma si no están relacionadas con una entidad más alta. Es como una estructura de arriba hacia abajo. Esta entidad es conocida como una librería de formas, que es una colección de formas y se genera mediante una opción de utilidad, las formas deben estar relacionadas a ella, y éstas a su vez actúan como entidades de los campos. Así, lo primero que se definió para el sistema fue una librería, luego las formas y posteriormente los campos. Una vez que se generan, se convierten en archivos en código de ALGOL, y contienen todas las especificaciones lógicas para poder procesar la colección de formas. Estas son usadas por los programas de aplicación, mediante procedimientos normales de lectura y escritura, lo que permite prestar una mayor atención en la forma lógica de la programación en vez de tener un formato de lenguaje complicado para poder procesar la forma.

SDF genera esencialmente dos "campos", el que se encuentra en la forma o sea el que el usuario de la aplicación ve en la pantalla, y un registro de programa. De cualquier modo, se puede cambiar el orden de los campos en un registro o en la forma, de manera independiente, lo cual es una ventaja para manejar valores de entrada y salida. Los pasos a seguir en la creación de una librería de formas son :

- 1) Definir la librería y sus atributos.
- 2) Definir las formas y sus atributos.
- 3) Diseñar las formas.
- 4) Definir los campos y sus atributos.
- 5) Validar la forma.
- 6) Generar la forma.

Una vez que las formas han sido generadas, ya pueden ser invocadas por los programas de aplicación.

Algunas de las características que ofrece SDF son :

- Creación de formas de pantalla por medio de técnicas de diseño.
- Construir utilerías integradas para la producción de documentación en copias.
- Permitir interfaz de aplicaciones simplificada con -- COBOL 74.
- Definición de controles programáticos, lo cual significa el control de la forma por el programa de aplicación.
- Definición de campos sin imagen.- Son los campos que existen en el registro del programa, pero no se despliegan en la forma.

- Definición de campos llamados "OCCURS".- Son definidos para que ocurran más de una vez en una forma.
- Permite realizar validaciones.- Valida campos, fechas y rangos de valores.
- Permite la substitución de Entrada - Salida.- Ofrece substituir un valor por otro en entrada o salida.

SDF también permite incluir descripciones textuales de campos y de formas que uno crea.

Cuando las librerías de formas son llamadas por un programa, éstas formatean y validan automáticamente las formas que contienen, ejecutan estos procesos y manipulan los datos antes de que la forma o los datos sean desplegados en la terminal. La respuesta de la terminal es regresada a través de la librería de formas al programa original. La librería de formas de SDF regresa la misma forma al mismo programa, como sigue :

PROGRAMA DE
APLICACION

LIBRERIA DE
FORMAS

PROGRAMA
EN LA TERMINAL

Con el diseño de formas se elimina la programación de éstas y como se ha dicho, apoya a la validación de campos y a la documentación en línea. Reduce en gran medida los esfuerzos de programación, permitiendo al usuario mejorar significativamente el desarrollo de pantallas para que los sistemas en línea puedan operar a la brevedad, lo cual hace al sistema mucho más aceptable por parte del usuario, además de contar con procedimientos de verificación que aseguran un nivel muy elevado de la integridad del sistema. Todos los menús de este sistema fueron creados con SDF.

3.4 EQUIPOS DISPONIBLES.

El sistema se desarrolló en una computadora modelo A3 de UNISYS, debido a que es un equipo con suficiente capacidad para los distintos requerimientos que se tenían como son gran volumen de información, gran velocidad de acceso a los archivos y velocidad de ejecución en los programas, además este equipo permite la concurrencia y es usado para el desarrollo de Software, está conectado a dos computadoras A9 a través de BNA (Binary Network Architecture), lo cual permite consultar la información de la Base de Datos desde cualquier terminal T27 ó computadoras personales emulando este tipo de terminales conectadas a uno de estos equipos. Actualmente esto puede implementarse con las facilidades de las redes locales.

El sistema de cómputo A3 de Unysis se caracteriza por la ejecución eficiente de lenguajes de alto nivel. El Software que utiliza minimiza el tiempo y esfuerzos que se requieren para el desarrollo y mantenimiento de los sistemas.

El sistema pudo haberse diseñado con el paquete LINK II, pero no se tenía disponible en el momento de la implementación del sistema, y actualmente se cuenta ya con paquetes de software para PC , (DB4, CLIPPER), que permiten implementarlo fácilmente, lo cual tiene la restricción de no poderse consultar en cualquier terminal de la dependencia.

4. CONFIGURACION DE LA BASE DE DATOS PARA EL CONTROL DE SERVICIOS DEL DEPARTAMENTO DE ELECTRONICA.

4.1 DIAGRAMA DE BLOQUES. (figura 8) pág. 56.

4.2 DESCRIPCION DE LOS DIFERENTES MODULOS.

El sistema cuenta con cinco módulos :

- Control de Llamadas
- Control de Inventarios
- Control de Partes
- Mantenimiento Correctivo
- Control de Herramientas

Control de Llamadas.- Permite registrar las llamadas telefónicas que se reciben en el Departamento de Electrónica de la DGSCA para reportar fallas en los equipos de cómputo de las dependencias a las que da servicio este centro. (figura 9).

Control de Inventarios.- Permite obtener en forma inmediata la información acerca de las existencias de equipo con que cuenta la dependencia, así como su asignación y ubicación con cada uno de los diferentes usuarios.

(figura 10).

Control de Partes.- Este módulo proporciona información sobre las existencias de partes que se tienen en el departamento de electrónica, de esta forma es fácil saber si existe la refacción requerida para un equipo y conocer la lista de partes que deben ser compradas por existencia mínima o consumo estimado.

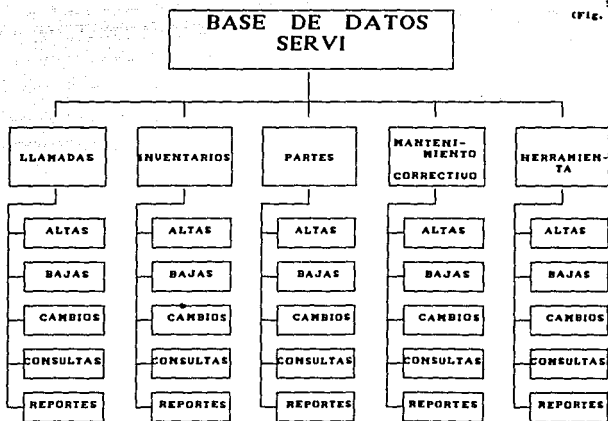
(figura 11).

Mantenimiento Correctivo.- En este módulo se lleva un control sobre las fallas más frecuentes, señalando cual es su solución, y las partes empleadas para así proporcionar un servicio más eficiente.

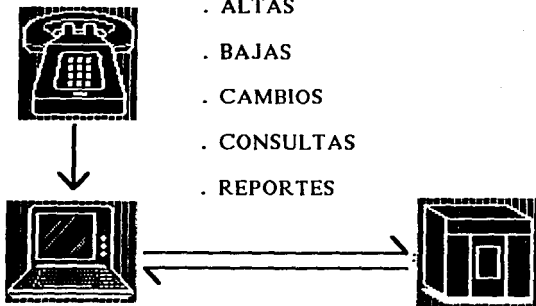
(figura 12).

Control de Herramientas.- Con este módulo pueden tenerse las existencias de la herramienta con que cuenta el departamento de electrónica, lo cual permite la planeación de compras de éstas para requerimientos futuros.

(figura 13).



MODULO : CONTROL DE LLAMADAS⁵⁷



(Fig. 9)

MODULO : CONTROL DE INVENTARIO⁵⁹

- . EXISTENCIAS DE EQUIPO
- . ASIGNACION
- . UBICACION



Terminales



Computadoras Personales



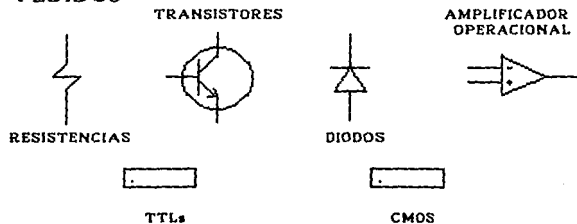
Impresoras

(Fig. 10)

MODULO : CONTROL DE PARTES ⁵⁹

- EXISTENCIAS

- PEDIDOS



(Fig. 11)

MODULO : MANTENIMIENTO CORRECTIVO

- . CONTROL Y ESTADISTICA DE FALLAS**
- . SOLUCION DADA**
- . MATERIAL EMPLEADO**

(Fig. 12)

**MODULO : CONTROL DE
HERRAMIENTA**

- . EXISTENCIAS**
- . ASIGNACION**
- . REQUISICIONES**

(Fig. 13)

La programación del sistema esta hecha en lenguaje COBOL, utilizando los programas de aplicación SDF y DMSII de los que se habló anteriormente.

Dentro del programa de aplicación se especificó una sección dentro de la división de datos (Data Division), referida a la base de datos. Con esta declaración, el compilador de COBOL localiza la descripción completa que existe en el sistema, incluyendo todas las estructuras y sus subestructuras asociadas, con lo cual ya se puede hacer referencia a cualquier información que el programa requiera para la manipulación de los datos en la división de procedimientos (Procedure Division). Cuando el compilador encuentra una invocación a la base de datos, una corrutina se encarga de buscar el archivo de descripción del sistema, desde aqui se dan especificaciones para cada una de las estructuras que van a ser utilizadas, posteriormente estos elementos de datos son añadidos a algunas tablas como si fueran declaraciones comunes a las del programa, el compilador de COBOL asigna un número de nivel a cada una de las estructuras que se usan en el lenguaje de datos y las lista en formato COBOL al momento de la compilación.

A continuación se muestra como está constituida la base de datos y cada uno de los módulos que la conforman, especificando la descripción, longitud y tipo de cada uno de los campos que intervienen en ella.

(MODULO PRINCIPAL)

Nombre de los módulos	Tamaño (en bytes)	Descripción
Control de Llamadas	129	LLamadas telefónicas para reportar fallas en los equipos de cómputo de la UNAM.
Control de Inventario	111	Control de los bienes que pertenecen al departamento de Electrónica.
Control de Partes	127	Existencias de partes en el Departamento de Electrónica.
Mantenimiento Correctivo	125	Fallas frecuentes , solución y refacciones empleadas.
Control de Herramientas	110	Existencias de herramienta en el departamento.

MODULOS.

 MODULO DE CONTROL DE LLAMADAS

Nombre del	Tamaño		Tipo	Descripción
Campo	long.	decim.		
Numlin	4		alfanum.	Número de línea del equipo.
Depend	32		alfanum.	Dependencia a la que pertenece el equipo.
LSNBD	6		alfanum.	Número de Estación Lógica.
Fechrep	8		alfanum.	Fecha en que se reportó la falla.
Horarep	5		alfanum.	Hora del reporte.
Fechrest	8		alfanum.	Fecha en que se dió solución a la falla.
Horarest	5		alfanum.	Hora de solución.
Descfall	43		alfanum.	Que tipo de falla se dió.
Estad	15		alfanum.	Estado actual del equipo.
Atendidobd	3		alfanum.	Persona que recibe el recado.

 M O D U L O D E C O N T R O L D E I N V E N T A R I O

Nombre del Campo.	Tamaño long.	Tipo Decim.	Descripción
Inventario	10	alfanum.	Número de inventario del equipo.
Marca	15	alfanum	Marca del equipo.
Modelo	10	alfanum	Modelo del equipo
Número	10	alfanum	No. consecutivo que se asigna a los equipos de la misma marca.
Serie	12	alfanum	No. de Serie.
Precio	12	numérico	Precio del equipo.
Dependen	32	alfanum.	Dependencia a la que pertenece el equipo.
RFC	10	alfanum	Responsable del equipo.

MODULO DE MANTENIMIENTO CORRECTIVO			
Nombre del Campo	Tamaño long.	Tipo	Descripción
Serie-BD	10	alfanum	No. de serie del equipo.
Marca-BD	15	alfanum	Marca del equipo.
Instalado-BD	32	alfanum.	Dependencia.
Parte-BD	30	alfanum	Partes arregladas.
Material-BD	35	alfanum	Materiales empleados.
Reparado-BD	3	alfanum	Personal que reparó la falla.

MODULO DE CONTROL DE PARTES

Nombre del Campo.	Tamaño long.	Tipo	Descripción
NUM-PARTBD	10	alfanum	Número de parte.
DESC-PARTBD	25	alfanum	Descripción de la parte.
CARA-PARTBD	25	alfanum	Características.
ATRI-PARTBD	25	alfanum	Atributos.
EXIS-PARTBD	5	alfanum	No. en existencia.
PREC-PARTBD	12	alfanum	Precio de parte.
EQUI-PARTBD	10	alfanum	Equipo.
EXISTI-PARTBD	5	alfanum	Existencias.

MODULO DE CONTROL DE HERRAMIENTA

Nombre del Campo.	Tamaño long.	Tipo	Descripción
DESCRIPCIONBD	30	alfanum.	Tipo de herramienta.
CANTIDADBD	45	alfanum.	Cantidad en existencia.
ASIGNACIONBD	35	alfanum.	Responsable de la herramienta.

El esquema del sistema contempla la información de los cinco módulos o estructuras que son los conjuntos de registros que contienen los datos pertinentes a las aplicaciones, cada uno de ellos se ve como un archivo separado pero todos están coordinados por el manejador en tal forma que juntos actúan como una unidad. Son archivos diferentes que al ser fusionados y estar juntos sirven como una base de datos. Cada registro de las estructuras está compuesto por 1 ó más elementos de datos.

El acceso a ellos es en forma aleatoria a través de las correspondientes subestructuras lógicas que permiten tener un acceso directo a cada uno de los registros de las estructuras. Este acceso puede ser en orden ascendente o descendente y los registros son ordenados en relación a la llave definida en la subestructura. Esta subestructura es un archivo enteramente separado que se usa para conservar las posiciones físicas de los registros en las estructuras. También está compuesto de registros individuales, pero cada uno de éstos sirve como un índice para hacer referencia a la información, en vez de actuar como un lugar para almacenar datos. Los registros de las subestructuras están compuestos de dos piezas, un elemento de datos que es la llave y una dirección. Esta llave es mediante la cual los registros de la estructura se van a secuenciar y la dirección que apunta a la localización actual que tiene ese registro.

Un ejemplo es el siguiente :

La estructura Control de Inventarios indexada por la subestructura a la que se llamó IPARSET, la cual realiza la recuperación mediante la llave declarada que es el número de inventario (Ver diagrama en pag. 70).

IPARSET SET

ESTRUCTURA INVENTARIOS

No. INV.	DIR.	No. INV.	MCA. MODEL.	NUM. SERIE.	RFC.
0000000001	4	0000000003	IBM IB34	IB-0165768	DASD600918
0000000002	2	0000000300	APPLEAP01	AP-1732456	EASOS581024
0000000003	0	0000000002	PRINTPT05	PR-0923451	SALA450819
0000000004	25	0000000011	UNISYBU20	BU-7656476	DAFE471008
0000000005	356	0000000001	TELEVTV10	TE-3567878	FAXE630512
.		.		.	.
.		.		.	.
.		.		.	.

Para realizar el objetivo primario del DBMS los registros de datos deben ser almacenados en un formato accesible, el cual debe proveer de un fácil reemplazo de registros, así como borrar y agregar en puntos arbitrarios dentro de las estructuras, y para ello lo más fácil es conservar los registros en secuencia por medio de una llave de recuperación.

Así los registros que se añaden a las estructuras van siendo colocados donde hay espacio disponible, la subestructura es el archivo que se usa para conservar las pistas de las posiciones físicas de los registros.

Los nombres de las subestructuras que se han usado se listan a continuación :

ESTRUCTURAS	SUB ESTRUCTURAS	LLAVES
Llamadas	Parset	Número de Línea
Inventarios	Iparset	No. de Inventario
	Npparset	Modelo
	Rpparset	RFC
Mantenimiento	Mparset	No. de Serie
Herramientas	Hparset	Descripción
Partes	Pparset	Número de Parte

Si en este programa no se hubiera usado DMSII, sino solamente COBOL, la selección de información sería mediante una llave única en caso de que se deseara recuperar la información en forma aleatoria, pero si existiera otro campo por el cual se quisieran seleccionar estos mismos datos, se tendría que hacer antes una clasificación por el campo que se desea consultar y la selección sería en forma secuencial. En este sistema, mediante las facilidades que brinda el manejador se declararon varias llaves en un módulo, o sea en una misma estructura, siendo el acceso aleatorio, esto evita el tener que realizar un ordenamiento de los datos antes de reportarlos.

En archivos convencionales como en COBOL la lectura siempre se hace del inicio al fin, mientras que en este sistema esto se puede hacer a la inversa, lo cual trae como ventaja mayor velocidad en el acceso a la información, además en COBOL no se puede clasificar la información en subesquemas y en DMSII si, lo cual en caso de que se desee, hace posible que un usuario particular tenga acceso a solo una parte del sistema.

En cuanto a la concurrencia, nuestro sistema puede ejecutar varias transacciones al mismo tiempo compartiendo el procesador, con lo cual puede trabajarse más en menos tiempo, esto lo ejecuta el manejador automáticamente.

El sistema opera en línea, lo que trae como ventaja no tener que codificar y elaborar cada reporte a máquina y posteriormente archivarlo, esto representa un gasto inútil de tiempo y de papelería, además en muchos casos no es necesario codificar la información, ya que conforme la van reportando por vía telefónica, esta se va metiendo en forma inmediata a la computadora, con lo que los datos se mantienen actualizados, siendo el proceso mucho mas efectivo. Otra de las grandes ventajas es la respuesta rápida a los usuarios y aunque el número de terminales y de requisiciones se incrementen, se tiene un proceso concurrente que permite abarcar a todas, manteniendo un tiempo de respuesta inmediato.

Los comandos de DASDL simplifican la programación y por lo tanto todos los tiempos de respuesta en el acceso a la información son mas rápidos.

Un ejemplo de consulta usando COBOL, y la misma pero utilizando DMSII es el siguiente :

Buscar en el archivo de Herramientas si existe algún artículo con el nombre "tijeras".

Usando COBOL :

Inicio.

Open input Herramientas.

(Abre el archivo de Herramientas)

Perform Lee-Registro.

(Manda a ejecutar la rutina de lectura)

Move 0 to X.

(Inicia una variable llamada x que servirá para continuar el proceso de búsqueda en caso de tener el valor 0 o bien detenerlo si es uno)

Perform Busca until X = 1.

(Manda a ejecutar la rutina de búsqueda hasta que la variable x tenga el valor 1)

Close Herramientas.

(Cierra el archivo de Herramientas)

Stop run.

Busca.

If descripcion = "tijeras"

(Si la descripción = tijeras)

display "se encontro articulo"

(Despliega que el artículo ya fue encontrado)

move 1 to X.

(Mueve el valor 1 a la variable x)

Perform Lee-Registro.

(Ejecuta la rutina de lectura)

Lee-Registro.

Read Herramientas at end

(Lee el archivo de Herramientas hasta el fin)

move 1 to X.

(Mueve el valor 1 a la variable x)

Usando DASDL :

Inicio.

Open query herramientas.

(Abre el archivo de Herramientas)

Move 0 to X.

(Mueve el valor 0 a la variable x)

Set Hparset to beginning.

(Mueve el índice al inicio de la estructura Hparset)

Find next Hparset at descripcion = "tijeras"

on exception move 1 to X.

(Encuentra el siguiente registro tal que la descripción sea igual a tijeras, en caso de que la descripción no sea igual a tijeras, mueve el valor 1 a X)

If X = 0

(Si la variable X es cero)

Display "se encontro articulo".

(Despliega que el artículo ya fue encontrado)

Close Herramientas.

(Cierra el archivo de Herramientas)

Stop run.

En este ejemplo se ve que el número de pasos es menor y menos complicado al utilizar DASDL en este proceso de búsqueda.

El proceso de recuperación de datos se efectúa mediante comandos de DASDL que forman parte del programa COBOL.

Cuando el programa de aplicación abre la base de datos, el sistema localiza al archivo de control, que contiene el archivo de las rutinas de acceso, se determina automáticamente si están activas, lo primero que hacen es verificar la continuidad del programa, después ejecutan una prueba de seguridad, y si al programa le está permitido el acceso al sistema, entonces proceden a abrirlo. En seguida las rutinas realizan una revisión a los archivos de datos que componen la base, para ver si están intactos, o sea que en este momento se verifica como fue terminada la última transacción realizada en el sistema, si los archivos están en un estado correcto, entonces las rutinas deciden que la base de datos se encuentra lista para su acceso. Cada estructura y subestructura requieren de un archivo físico abierto al ser invocadas por los programas de aplicación.

Al abrir el sistema, el tiempo de respuesta es directamente proporcional al número de estructuras invocadas por el programa. Existen dos formas de abrir el sistema, en

modo de actualización ó en modo de investigación, en este último no se permite alterar ninguna información que esté contenida en el sistema y en el modo de actualización se permite realizar cualquier tipo de movimiento a los datos, siempre y cuando no se trate de archivos completos. Existe un comando que se usa para alterar la trayectoria de las estructuras y subestructuras, a tres diferentes posiciones : puede asignarla al principio, al final o al registro que es referenciado por la subestructura asociada, esto se hace antes de iniciar la búsqueda de información. Entre las instrucciones más importantes con que cuenta el manejador esta la instrucción " encontrar " que implica una selección, y abarca todas las llaves que fueron descritas, se usa para buscar un registro ya sea mediante un acceso aleatorio o secuencial. En el procesamiento secuencial, la declaración principal para la recuperación es " encuentra el siguiente ", con este comando, la trayectoria para la estructura designada se va a incrementar en uno, una vez que se encontró, este registro es regresado al programa. También puede leerse en forma secuencial pero en sentido inverso a través de otro comando que es " encuentra el anterior ", si nos estamos refiriendo a una estructura entonces los registros son tomados en el orden contrario al que fueron almacenados físicamente, pero si se trata de una subestructura entonces los registros son tomados en el orden contrario al que fueron acomodados a partir de alguna llave.

Aparte de las instrucciones " abrir " y " cerrar", esta instrucción tiene mucha importancia en un programa de recuperación de datos.

Para anexar un nuevo registro a una estructura se hace uso de una declaración que se encarga de iniciar un área (de registro) en el formato que se haya definido. A cada elemento de datos le es asignado un valor inicial que es determinado por el lenguaje de datos. Con esto el programa tiene un registro de datos en el cual puede colocar los nuevos valores. Para agregar el nuevo registro a la estructura, se utiliza otro comando que se encarga de meterlo en el área correspondiente, además de que también lo hará dentro de cada una de las subestructuras asociadas con la estructura para el nuevo registro. Para modificar el contenido de un registro, el programa debe tener el uso exclusivo de dicho registro, una vez hecho esto se pueden realizar los cambios, los valores de los campos especificados son reemplazados con los nuevos valores, al terminar de hacerlos, el registro es regresado a su estructura correspondiente. La instrucción para la eliminación de información, localiza el registro designado y lo borra de su estructura asociada, éste no existirá mas, ya que se cancela la ocurrencia del registro especificado. Esta declaración utiliza una expresión de selección para localizar el registro que va a ser borrado, ejecuta la búsqueda y lo remueve, si se borró un registro equivocado, no se puede recuperar, habría que

volver a darlo de alta. El espacio que deja en el archivo este proceso, desaparecerá al momento que el programa realice cualquier otra operación de restauración o compresión, ya que el sistema actualizará sus tablas, y nada quedará del registro que fue suprimido. La instrucción "almacenar" guarda una nueva ocurrencia de registro, creando los vínculos de relación que sean necesarios.

Una vez que el programa de aplicación ha completado su procesamiento, debe cerrar el sistema, al cerrarlo, el sistema separa este programa de todos los archivos de la base de datos, el programa es desligado de las rutinas de acceso que estaba utilizando, los programas de aplicación ahora quedan desconectados de la base de datos, que se cerrará siempre automáticamente al término de éstos, esto lo hace como una medida de protección.

Las opciones del DMSII que se han utilizado en este programa de aplicación incluyen algunas estadísticas (Statistics), con las cuales se mantiene información referente al número de transacciones que se han hecho durante todo el tiempo de corrida del sistema, contienen el número de lecturas y escrituras de registros encontrados y borrados y se imprimen automáticamente cuando la base de datos se cierra.

Otra opción (Allowedcore) controla cuantas palabras de la memoria principal pueden utilizar las rutinas de acceso para los "buffers" de la B. de D. Si el área especificada es totalmente ocupada por ellos, se empieza una recolocación.

Existe también un valor (Checksum) que se calcula para cada bloque de la Base de Datos, y detecta errores de entrada y salida, aplicando un operador de equivalencia en cada palabra en el bloque, cuando éstos son escritos físicamente, se almacena este valor y al ser leído se recalcula comparándolo con el almacenado, debiendo ser iguales.

Con la opción (Reblock) el sistema permite optimizar el acceso aleatorio y secuencial puro a una estructura, o sea que dos tamaños de bloque pueden ser mantenidos por una misma estructura simultáneamente, usando un tamaño pequeño del bloque para el acceso aleatorio y uno grande para el secuencial. Controla si las rutinas de acceso emplean el rebloqueo al tiempo de la corrida.

Otra opción utilizada (Lock to Modify Details) se encarga de que los programas de aplicación aseguren al registro llave antes de realizar alguna actualización de

datos que les pertenecen, con lo que se garantiza que dos programas no hagan modificaciones en forma concurrente.

La Comparación de Llaves (keycompare) verifica que la llave especificada en la estructura principal corresponda a la llave en la subestructura a través de la cual es accedida, en caso de que no sean las mismas, el sistema termina la sesión, emitiendo un error.

Mediante la opción de actualización (Update) se puede modificar la descripción del sistema, ya que las demandas que se tienen pueden cambiar, y la base de datos es redescrita completamente. Permite algunas modificaciones que no requieren cambios físicos a los archivos del sistema. Al utilizar esta opción es posible cambiar parámetros, opciones y atributos, el adicionar y borrar estructuras y subestructuras disjuntas.

4.3 IMPLEMENTACION.

Primero se diseñó un esquema general del sistema, como lo muestra la figura no. 14, después se realizó el programa fuente que genera la base de datos, o sea el programa en DASDL, que es un programa con un formato especial, que especifica las propiedades que tienen cada una de las estructuras, subestructuras, y los elementos de datos que lo conforman. Una vez que se diseñó este programa se hizo la compilación. El compilador del lenguaje de datos DASDL lee y verifica la sintáxis y en ese momento crea un archivo de descripción fuera del esquema de la base de datos (figura 15-a), éste contiene la descripción completa del sistema en forma de código, es la fuente central de información para la generación del software, los compiladores lo utilizan para crear las rutinas de acceso y las utilerías y también al compilar los programas de los usuarios, en el se encuentran todas las variables que se están utilizando. Se verifica el programa completamente al pasar esta compilación.

Una vez que este archivo fue generado, y al terminar de compilar las rutinas de acceso y las utilerías, se crea el archivo de control (figura 15-b), para su generación, la base de datos utiliza un sistema llamado "Dmcontrol", el cual se encarga de actualizarlo, o sea, localiza el archivo

ESQUEMA GENERAL DEL SISTEMA

83

LLAMADAS

NUM. FOLIO	DEPENDEN- CIA	L S N	FECH. REP.	HR REP.	FECH REST.	HR. RESDE	DESCRIPCION LA FALLA	ESTADO ACTUAL	ATEN- DIDO
---------------	------------------	-------------	---------------	------------	---------------	--------------	-------------------------	------------------	---------------

INVENTARIO

NUM. INVENT.	MARCA	MODELO	NUMERO	SERIE	PRECIO	DEPENDENCIA	R F C
-----------------	-------	--------	--------	-------	--------	-------------	-------------

PARTES

NUMERO PARTE	DESCRIPCION	CARACTE- RISTICAS	ATRI- BUTOS	EXISTEN- CIA	PRECIO UNITARIO	USADO EN EQUIP.	EXISTEN. INICIAL
-----------------	-------------	----------------------	----------------	-----------------	--------------------	-----------------------	---------------------

MANTENIMIENTO CORRECTIVO

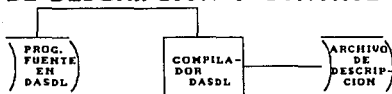
NUM. INVENT.	MARCA	DEPENDEN- CIA	PARTE REPARADA	MATERIAL EMPLEADO	REPARADO POR
-----------------	-------	------------------	-------------------	----------------------	-----------------

HERRAMIENTA

DESCRIPCION	CANTIDAD	ASIGNACION
-------------	----------	------------

(Fig. 14)

GENERACION DE LOS ARCHIVOS DE DESCRIPCION Y CONTROL



COMPILACION DEL PROGRAMA EN DASDL

(Fig. 15-a)



GENERACION DEL ARCHIVO DE CONTROL

(Fig. 15-b)

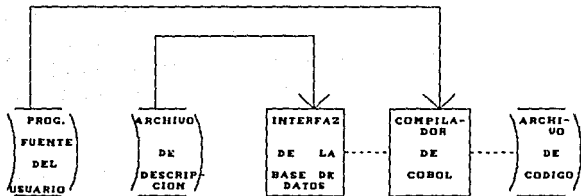
control que se utilizaba anteriormente, lo actualiza y lo copia en el nuevo dispositivo para poder operarlo normalmente, este sistema se ocupa de crear y mantener este archivo que realiza algunas funciones importantes del DMSII, debe estar presente cuando se va a hacer uso de la base de datos. Se encarga del control de seguridad, o sea que algunas funciones que así lo requieren tengan el uso exclusivo del sistema. Verifica la compatibilidad entre los programas y archivos. Revisa básicamente la consistencia física, o sea que las estructuras de la base correspondan a una misma fecha de creación, lleva el control con respecto a las versiones de las estructuras del sistema, y cualquier acceso queda registrado en este archivo, el cual se actualiza por todos los sistemas de software que usa la base de datos. En seguida se deben inicializar las estructuras, mediante una opción que hace que las utilerías inicien todos los archivos del sistema, esto se hizo al crear la base de datos, solo en este momento, ya que esto no debe hacerse cuando una base de datos está siendo modificada, una vez hecho esto, se cargaron algunos datos a las diferentes estructuras, enseguida se llama al programa de aplicación, se compila, hasta que no tenga ningún error, el compilador de Cobol llama a una rutina llamada interfaz de la base de datos, la cual proporciona los nombres y descripciones de todas las estructuras y elementos de datos declarados en el sistema y que son invocados por los programas de aplicación.

(figura 16). En seguida se manda a ejecutar el programa y en este momento la base de datos está lista para usarse. (figura 17).

El compilador DASDL puede ejecutar otras funciones como son listados impresos de la entrada fuente que muestra cualquier error detectado, y crea también un archivo simbólico actualizado que puede ser entrada a compilaciones subsiguientes.

El programa de aplicación para la recuperación de información, presenta los datos al usuario en un formato claro, éste puede tomar distintas formas, una de ellas es leer a través de una ó más estructuras enteras y generar una línea de reporte para cada uno de los registros que contiene. Otra es la búsqueda de registros dependiendo de las requisiciones que se han metido vía terminal, y una tercera forma de consulta es cuando se prepara un archivo para actualizar requisiciones que sea la entrada a un programa de actualización y validar estos datos en los registros que ya se tienen en existencia.

COMPILACION DEL PROGRAMA DEL USUARIO 87



(Fig. 16)

EJECUCION DEL PROGRAMA DEL USUARIO

88



(Fig. 17)

4.4 FORMA DE OPERACION DEL SISTEMA.

La forma de operación del sistema es a través de un menú principal que permite elegir entre los cinco distintos módulos, con solo teclear un número, en la pantalla aparece inmediatamente otro menú el cual se refiere a las diferentes actividades que pueden ejecutarse, altas, bajas, cambios, consultas o reportes dentro de cada uno de los módulos. Si se escoge la opción de altas, aparece el formato definido con los nombres de los campos correspondientes a fin de teclear los valores del nuevo registro que se va a insertar, en cuanto se ha terminado de dar esta información, se presenta nuevamente el formato por si se desea continuar dando de alta nuevos registros hasta terminar, si ya no se desea seguir con este proceso se da " FIN " en el valor que pertenece al primer campo. En el módulo de bajas, se presenta una pantalla de rangos correspondientes a los valores de las llaves de los registros que se desean borrar, los registros elegidos van siendo mostrados en la pantalla, al momento de ser eliminados, solo se permite borrar registros completos, no es posible quitar solamente valores de determinados atributos. Para cambiar algunos valores de los registros se utiliza el proceso de modificación, éste proporciona al usuario un formato para dar la llave del registro que va a ser transformado, posteriormente aparece el registro y se posiciona el cursor en el o los valores que

se van a alterar, hecho esto regresa al formato inicial por si se desea continuar realizando la misma actividad , la modificación queda registrada en forma inmediata.

En caso de haber elegido la opción de consulta, hay una pantalla donde se teclean los rangos que corresponden a las llaves de los registros a consultar, o bien si se desea ver toda la información que esta cargada se teclea " TODO " en el valor que corresponde al primer campo, si ya no se desean ver los registros restantes, el proceso termina, escribiendo " FIN " en el primer campo. En los reportes aparece también la pantalla de rangos para mandar a impresión la información elegida, el sistema regresa al menú anterior y se obtiene el reporte en forma inmediata.

Al estar dentro de la ejecución de este sistema se puede pasar de un módulo a otro y de una opción a otra indistintamente, al alimentar datos, estos pueden consultarse en forma inmediata dentro de la misma sesión. Todos los módulos cuentan con las mismas opciones, y solo varían dependiendo de que algunos cuentan con mas de una llave, lo cual hace posible consultar la información en distintas secuencias.

5. CONCLUSIONES.

Si este sistema se siguiera llevando en forma manual, tendría serias desventajas, ya que el número de errores en la información sería mayor, el manejo de grandes volúmenes de información sería lento y engorroso, el hecho de elaborar reportes resultaría laborioso, principalmente en la consulta de archivos. Al aumentar el trabajo, tendría que aumentar el número de personal. Esto justifica la automatización de la información actualizada, que trae como resultado versatilidad y ahorro de tiempo, facilitando el manejo y almacenamiento de grandes volúmenes de información, además de proporcionar confiabilidad en la misma.

Al utilizar el manejador de bases de datos, el acceso a la información es mucho mas simple, ya que cuenta con funciones preprogramadas para el manejo de archivos, que los lenguajes de alto nivel no tienen. Cuenta con estructuras ya definidas que de no haberse utilizado, se tendrían que haber hecho programas para implementar cada función específica. Además en el ciclo de vida de estos sistemas, si las demandas varían, existe la posibilidad de cambiar la descripción existente, sin tener una sola pérdida en la información con que se cuenta.

El sistema proporciona información al momento que se necesita, además de que cuenta con la posibilidad de poder combinarla para obtener mayor conocimiento sobre los datos que se están consultando, cuenta con un tiempo de respuesta rápido, ya que al eliminar lo mas posible la duplicación de datos, el tiempo de procesamiento disminuye, su manejo es simple y eficiente, además de que al tener los datos centralizados, es mas fácil encontrar lo que se necesita, y distintas personas pueden dedicarse a la actualización de la información en cada uno de los módulos.

Una de las razones por las cuales se decidió utilizar el equipo A3 fue por las necesidades de almacenamiento masivo que se tienen, si llega a presentarse el caso de que hubiera una saturación de datos, se puede transferir la información a una computadora A9, además el acceso se puede realizar desde cualquier terminal T27 de cualquier dependencia usuaria.

Actualmente, el desarrollo de la tecnología de las bases de datos permiten enfocar directamente el problema de información en una organización, se puede pensar que el avance puede ser lento respecto al hardware, pero entre más masivo sea el uso de esta tecnología, proporcionalmente sera más exigible la rapidez en el desarrollo de nuevas técnicas de organización de bases de datos.

6. BIBLIOGRAFIA.

Fundamentos de Bases de Datos.

Henry F. Korth / Abraham Silberschatz.

Litográfica México, S.A.

México, D.F. 1987.

Data Base Management Systems.

Dionysios C. Tsichritzis / Frederick H. Lochovsky.

Academic Press, INC.

Newyork, N.Y. 1977.

Organización de las Bases de Datos.

James Martin.

Ediciones del Castillo, S.A.

Madrid, España. 1977.

Data Base Processing.

David Kroenke.

Science Research Associates, Inc.

Newyork, N.Y. 1983.

Sistemas de Información.

John G. Burch Jr. / Felix R. Strater.

Editorial Limusa, S.A. de C.V.

Mexico, D.F. 1985.

Manual del Taller de Metodología y Estándares para el desarrollo de Sistemas.

México, D.F. 1980

Data and Structure Definition Language.

Burroughs Corporation.

Detroit, Michigan. 1984.

The DMSII Primer.

Gregory Publishing Documents.

A.C. Ridlon, Distributor.

6090 So. Sterne Pkury.

Littleton. Co. 1976.

Screen Design Facility.

Burroughs Corporation.

Detroit, Michigan. 1985.