

49
2ej

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS



PLANTEAMIENTO DE UN SISTEMA AUXILIAR EN PC BASADO EN INFORMACION GENERADA POR UNA MACROCOMPUTADORA ENFOCADO AL AREA DE PRODUCCION Y VENTAS

T E S I S

QUE PARA OBTENER EL TITULO DE
LICENCIADO EN ACTUARIA
P R E S E N T A

MARIA ANTONIA OLMO SILICEO

DIRECTORA DE TESIS : MAT. MARIA DE LA LUZ GASCA SOTO

MEXICO, D. F.

1993

TESIS CON
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.

**PLANTEAMIENTO DE UN SISTEMA AUXILIAR EN PC
BASADO EN INFORMACION GENERADA POR UNA MACROCOMPUTADORA
ENFOCADO AL AREA DE PRODUCCION Y VENTAS.**

INTRODUCCION	3
I.SISTEMAS DE INFORMACION	
I.1 CONCEPTOS BASICOS.	5
I.2 CICLO DE VIDA DE UN SISTEMA.	8
I.3 PROBLEMATICA DEL DESARROLLO DE SISTEMAS.	11
II.BASES DE DATOS	
II.1 ANTECEDENTES	14
II.2 GENERALIDADES	15
II.3 ELEMENTOS	16
II.4 SISTEMA MANEJADOR DE BASE DE DATOS (DBMS)	17
II.5 OBJETIVOS DE UN MANEJADOR DE BASE DE DATOS	19
II.6 ORGANIZACIONES LOGICAS	20
II.7 DISEÑO DE UNA BASE DE DATOS	24
III. COMUNICACION ENTRE MICROCOMPUTADORAS Y MACROCOMPUTADORAS	
III.1 INTRODUCCION	26
III.2 PANLINK	26
III.3 A R B I T E R	29
IV. CASO PRACTICO	
IV.1 ANALISIS DEL SISTEMA AUXILIAR DE PRODUCCION Y VENTAS	

III.3	A R B I T E R	29
IV. CASO PRACTICO		
IV.1	ANALISIS DEL SISTEMA AUXILIAR DE PRODUCCION Y VENTAS DE UNA EMPRESA AUTOMOTRIZ	31
IV.2	DISEÑO DE LA BASE DE DATOS	46
VI.	CONCLUSIONES	59
VII.	BIBLIOGRAFIA	61
APENDICE A: METODOLOGIA PARA EL DISEÑO DE UNA BASE DE DATOS		
		63
APENDICE B. ALGEBRA RELACIONAL Y NORMALIZACION		
		73

INTRODUCCION

El objetivo de este trabajo es diseñar un Sistema de Información, en ambiente de computadoras personales (PC), que permita conocer la distribución de la producción de una industria en base a una o más características de los productos, tomando la información desde los sistemas que residen en una Macrocomputadora.

Esto es posible y a su vez una consecuencia de los avances tecnológicos en el área de computación debido a las facilidades para tener acceso a una computadora personal, y en el caso de las empresas, de no tener que depender totalmente del Departamento de Sistemas para obtener alguna información, por encontrarse ésta disponible a través de reportes generados en las mismas por los distintos Departamentos que la componen.

Esto da como resultado que la información de un área sobre un mismo concepto sea diferente a la generada por otra, ya que puede ser que el concepto no este a un mismo nivel de actualización entre las distintas Microcomputadoras, lo que ocasiona la pérdida de integridad en la información que circula dentro de la empresa.

Con este trabajo se pretende optimizar la forma en que se usa la información dentro de las empresas, propiciando el hecho de compartir la información entre las máquinas y que los usuarios de computadoras personales, enlazados a una red de computadoras, no sientan la necesidad de crear su propia información sino de utilizar la generada por los sistemas que apoyan la operación de los procesos de la empresa.

No se va a desarrollar el sistema, sino se dejará en la fase de diseño, pues solo se pretenden mostrar las herramientas necesarias para lograr un modelo de esa naturaleza.

Este tipo de estructura está orientado a empresas grandes con redes de comunicación.

En el capítulo cuatro se desarrolla un caso práctico tomando a una empresa ensambladora y estableciendo la comunicación entre una macrocomputadora y una computadora personal.

Por lo anterior en capítulo tercero se analizarán solo los paquetes de comunicación que permiten enlazar este tipo de máquinas (IBM - PC).

La metodología usada para diseñar la base de datos se incluye en el apéndice A.

CAPITULO I.

SISTEMAS DE INFOMACION

I.1 CONCEPTOS BASICOS

Un sistema es un conjunto de elementos que se interrelacionan para conseguir un objetivo, donde las propiedades esenciales de él se pierden al considerar las partes que lo componen de manera separada. Los elementos que lo forman pueden ser: objetos, sujetos, procedimientos, políticas o cualquier combinación de ellas.

Un sistema de información es un instrumento lógico que se emplea para producir información que sirva de soporte a las actividades de una organización, a partir de un conjunto de datos que son elaborados de acuerdo a ciertas reglas preestablecidas.

Generalmente las palabras datos e información se emplean indistintamente, y aunque se refieren a diferentes conceptos, están muy relacionados ya que la información se produce a partir de los datos.

Los datos son entes del lenguaje matemático o simbólico sobre los que generalmente se está de acuerdo para representar personas, objetos, acontecimientos y conceptos, mientras que la información es el resultado de modelar, organizar, convertir y, en general, elaborar datos de tal manera que aumente el nivel de comprensión de la realidad.

Otras características que deben tener los sistemas de información son:

FLEXIBILIDAD: permite incluir modificaciones futuras sin alterar su estructura básica.

MODULARIDAD: es la posibilidad de ser desarrollado e implantado en forma gradual.

NEUTRALIDAD: su implantación no debe tener una repercusión

significante en la organización a causa de su diseño.

ADAPTABILIDAD: se debe apegar a las necesidades de información de los usuarios.

CONFIABILIDAD: cuando las fuentes de información son confiables, entonces la información producida lo será.

EFICIENCIA: la información que brinda al usuario debe ser eficiente.

INTEGRABILIDAD: evitar que en el sistema un mismo dato se alimente más de una vez, procurando que estén disponibles.

Un gran avance producido en las últimas décadas en la concepción y diseño de sistemas de información consiste en la posibilidad de ser implantados en una computadora. A éstos se les llama Sistemas Informáticos.

Un Sistema Informático, también llamado Sistema de Información Computarizado, es un conjunto de Hardware (recursos materiales) y Software (recursos lógicos) que sirve de soporte al sistema de información.

El estudio de un sistema de información debe ser previo al estudio del sistema informático debe existir un alto grado de coordinación e interacción entre ambos.

Los componentes de un sistema informático normalmente son:

Sistemas de información: de él provienen los requerimientos del usuario que permiten diseñar el sistema.

Bases de datos: se definen a partir de las clases de datos identificados durante el estudio del sistema de información y la estructura lógica de los datos.

Comunicación: comprende todo el Sistema de Comunicación y frecuentemente los elementos de control del sistema.

Hardware: comprende los elementos físicos como son: la computadora, terminales, etcétera.

Software: Comprende los elementos lógicos y las bases de datos.

Las cualidades de un sistema informático son:

SEGURIDAD.

El sistema realizará las funciones requeridas por el usuario bajo condiciones normales.

GARANTIA.

Se realizan pruebas que aseguren la correcta satisfacción de los requerimientos del usuario.

FLEXIBILIDAD.

Para que a los programadores les sea fácil modificar y probar los cambios que necesite el sistema, ya sea por nuevos requerimientos, para rectificar deficiencias o para corregir errores.

EFICIENCIA.

Se pretende satisfacer los requerimientos del usuario sin exceder los recursos destinados para lograrlo.

AMIGABILIDAD.

Se intenta que el manejo del sistema sea fácil para el usuario y que la presentación de resultados sea de tal forma que el usuario los pueda entender y dominar.

PORTABILIDAD.

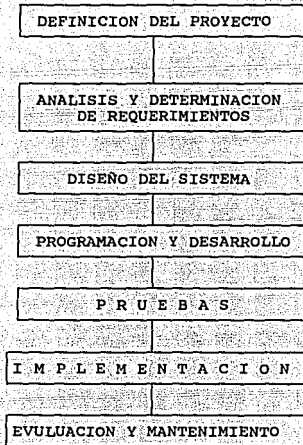
Las características del diseño del sistema deberán permitir cambiarlo fácilmente a otro ambiente de cómputo.

Además de estas cualidades, el sistema debe producirse dentro del tiempo y costo planeado. Todo esto es difícil de satisfacer, debido a las características heterogéneas de la gente que está involucrada en el proyecto, por ejemplo la deficiencia técnica y comunicativa entre el usuario y el analista.

1.2 CICLO DE VIDA DE UN SISTEMA

Desde el inicio del análisis del problema, hasta que se encuentre en operación, el sistema pasa por diferentes etapas (fig. 1.1).

Durante el análisis o desarrollo del sistema no se debe olvidar al factor humano, ya que el sistema a desarrollar va a tener interfaz de procedimientos manuales que muchas veces son fundamentales para el buen funcionamiento del mismo.



F I G U R A 1.1

Es importante mencionar que la documentación del sistema se debe elaborar en cada una de las etapas y aunque es difícil determinarlas, en seguida se describen brevemente las funciones que las comprenden:

DEFINICION DEL PROYECTO

En esta etapa se definen el objetivo, los alcances y limitaciones del sistema; se analizan los procedimientos existentes y se elabora un análisis de costo-beneficio.

ANALISIS Y DETERMINACION DE REQUERIMIENTOS

Durante esta etapa se analizan los procedimientos existentes y se entrevistan a los usuarios para tener una concepción clara del sistema, y así obtener su diseño conceptual apoyado en los objetivos definidos en la etapa anterior. En éste se muestra el flujo de datos, las entradas, salidas y procesos.

Cuando el sistema se divide en módulos, a cada uno se le denomina subsistema, y al primero se le ve como el conjunto de subsistemas que lo forman y cada uno puede verse como un sistema independiente.

DISEÑO DEL SISTEMA

En esta fase se desarrolla la estructura modular del sistema, se definen las interfases entre sus módulos y se selecciona el Hardware y Software que se va a utilizar.

Además se generan las especificaciones detalladas de los programas que forman cada módulo, de los procedimientos manuales y de la información que tendrá cada archivo, así como de las interfases entre los subsistemas. Es decir, se completa toda la lógica para desarrollar el sistema en la computadora.

Al diseñar el sistema, es importante considerar su modularidad. Un sistema modularizado es aquel que se ha dividido en partes independientes que pueden ser instaladas o modificadas sin afectar a todo el sistema.

PROGRAMACION Y DESARROLLO

Aquí se desarrollan los programas de cómputo que forman al sistema, y se desarrollan las tareas específicas para los procedimientos manuales.

PRUEBAS

PRUEBAS

Las pruebas del sistema se hacen a diferentes niveles, empezando por cada uno de los programas, hasta revisar el funcionamiento de todos los subsistemas del sistema en conjunto.

Si existe un sistema equivalente, es recomendable realizar una prueba paralela entre éste y el nuevo, para revisar la eficiencia del nuevo.

IMPLEMENTACION

Esta etapa empieza cuando se considera que el sistema está listo para lograr los objetivos de operación.

Es importante considerar en esta etapa la conversión de los datos existentes a la nueva estructura. Muchas veces a esta actividad no se le da la importancia debida, lo cual acarrea muchos problemas. En caso de ser posibles se recomendable implementar los sistemas por subsistemas.

EVALUACION Y MANTENIMIENTO

La evaluación y el mantenimiento duran todo el tiempo que el sistema está en operación.

Se recomienda tener un procedimiento de control para el mantenimiento del sistema, además de contar con un plan para recuperar la información en caso de pérdida. También hay que monitorear al sistema para detectar indicadores sobre su crecimiento y obsolescencia.

La mayoría de las veces estos procedimientos son menospreciados, poniéndose en peligro la seguridad de la información.

1.3 PROBLEMATICA DEL DESARROLLO DE SISTEMAS

En los últimos tiempos la computación ha tenido mucho auge por la

facilidad para almacenar, manejar y procesar altos volúmenes de información en un mínimo de tiempo. Por lo tanto los avances tecnológicos han permitido aumentar la capacidad de los equipos para almacenar información, y facilitar el acceso a ella, por lo que, tanto los costos de equipos y programas han decrecido a un ritmo muy rápido.

La computadora es, sin duda, una de las grandes contribuciones que han producido las inteligencias tecnológicas de nuestra época. En los países industrializados se está convirtiendo en un instrumento indispensable en la vida de los ciudadanos comunes. Ya no se trata de una herramienta solamente utilizada por las grandes empresas para realizar operaciones y controles importantes, sino que se usa como cualquier aparato doméstico que realiza tareas que progresivamente se transforman en indispensables. Así como resulta inconcebible la inexistencia de un aparato de televisión en la mayoría de los hogares, la computadora está volviéndose un objeto imprescindible.

En las grandes empresa a medida que las aplicaciones de la computación se han multiplicado en tamaño y complejidad, la tarea de administrar los recursos se ha complicado cada vez más pues es difícil controlar el espacio en disco y las conexiones para que funcionen de manera óptima las redes de computadoras.

El crecimiento sin control ha originado que se desarrollen sistemas de manera independiente, sin tomar en cuenta que se puede compartir información con otras áreas de la compañía ya que en la mayoría de los casos sólo se utiliza para soportar unidades simples de operación. Esto ha originado una concentración de automatización en macrocomputadoras donde se encuentran archivos fraccionados y redundantes lo que da como resultado la falta de integridad de la información que circula en la empresa.

Otro factor que ha influido considerablemente, es la facilidad con que se cuenta actualmente para la obtención de Hardware y Software, pues a través de ellas las diferentes áreas satisfacen sus necesidades de información a nivel gerencia principalmente.

Ya que la mayoría de los sistemas de información en macrocomputadora están dirigidos a las actividades operativas, se deja sin tiempo al Departamento de Sistemas para atender otros requerimientos.

Todo lo anterior da como resultado un crecimiento poco planeado entre los sistemas desarrollados en microcomputadoras y los sistemas en la macrocomputadoras.

Debido a esto se pretende que la información se comparta por todos los sistemas.

CAPITULO II

BASES DE DATOS

II.1 ANTECEDENTES

En los años sesentas, se presenta un avance en todo lo que se refiere a computadoras, permitiendo que éstas tengan aplicaciones diferentes a las simplemente de cálculo numérico. Esta década se caracteriza por el surgimiento de los lenguajes de alto nivel, como son: COBOL, FORTRAN Y ALGOL entre otros, los que en un principio, a pesar del mejoramiento en el almacenamiento de datos, tenían grandes restricciones para tener acceso a la información, los tipos más comunes son:

- ACCESO SECUENCIAL, que obliga a recorrer el archivo desde el inicio para consultar cualquier información, resultando más lento su acceso en la medida en que los archivos aumentan su tamaño y lo que se desea consultar, no esté al principio.
- ACCESO DIRECTO, que accesa a la información dándole la dirección física del lugar donde está el registro. Es más rápido, pero tiene la limitante de que se debe conocer la dirección del registro lógico, lo cual se dificulta en la medida que se incrementa el volumen de registros.
- ACCESO INDEXADO, en donde los archivos se accesan mediante llaves o claves. En él se utilizan dos archivos, uno que contiene los datos, y otro con las direcciones de la información y los campos que conforman la llave. Además los registros están clasificados, lo cual agiliza las consultas.

Posteriormente, se vio la necesidad de contar con otra herramienta que permitiera manejar varios archivos, y que facilite la relación entre ellos. Está fue la creación de los Sistemas Manejadores de

Datos (DBMS: Data Base Management System).

II.2 GENERALIDADES

Una Base de Datos es una colección integrada de datos en la cuál se puede tener acceso y modificar cada uno de ellos en particular y minimizar la cantidad de información repetida.

El utilizar las estructuras comunes de archivos implica, por lo general, crear diferentes conjuntos de información, diferentes tipos de archivos en los que puede haber redundancia entre los datos. En la estructura de Bases de Datos, se pueden extraer diferentes conjuntos de información de un fondo común de datos, además de existir independencia entre los datos, con lo que se facilita la adición o eliminación de ellos.

A continuación se presentan conceptos que se relacionan con las Bases de Datos:

- REDUNDANCIA.- en la cuál se repiten los datos continuamente.
- INCONSISTENCIA.- implica obtener diferentes resultados de los mismos procesos y que se obtengan diferentes valores para un mismo dato cuando éste exista en diferentes archivos.
- INTEGRIDAD.- los datos están integrados en un conjunto de archivos y las aplicaciones los comparten.

Un avance relativamente nuevo en el área de bases de datos lo constituyen las BASES DE DATOS INTEGRADAS Y LAS BASES DISTRIBUIDAS. Una base de datos integrada está formada por una o más bases, aunque desde un punto de vista didáctico puede considerarse como una sola. En las bases distribuidas los datos no residen físicamente en un solo lugar, sino que se distribuyen a lo largo de una red de computadoras. Los datos se almacenan en el sitio donde se ocupan con mayor frecuencia, pero están disponibles para todos los usuarios de la red. El objetivo de un sistema así, consiste en que el usuario lo vea como un sistema centralizado, sin preocuparse donde se encuentran físicamente los datos.

La ventaja de distribuir la información radica en que se combina la eficiencia local, con la de poder compartir la información, pero existe la desventaja de que los costos de comunicación pueden ser muy altos y de existir dificultades técnicas significativas para instrumentar un sistema.

II.3 ELEMENTOS.

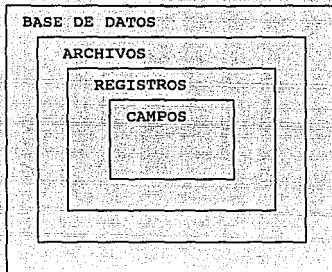
Una Base de Datos es un conjunto de archivos relacionados entre sí, en la cuál, cada uno tiene una colección de registros de un mismo tipo, y a su vez, cada registro tiene un conjunto de campos afines.

Básicamente, la estructura de una Base de Datos está compuesta por (fig. 2.1):

- **CAMPOS.** Es la unidad elemental y están formados por un conjunto de caracteres que representan alguna entidad del mundo real.

- **Registros.** Es una colección de campos relacionados entre sí considerados como una unidad.

- **Archivos.** Es un conjunto de registros con un mismo tipo de datos.



F I G U R A 2.1

II.4 SISTEMA MANEJADOR DE BASE DE DATOS (DBMS).

Un Sistema Manejador de Bases de Datos es un conjunto de programas computacionales que se integran para darle mantenimiento a la información teniendola disponible cuando y donde se le solicite. Se va hacer referencia a él como DBMS.

En general organiza los datos de estructuras predefinidas, realizando relaciones entre los registros, teniendo la posibilidad de mantener y manejar los datos de una o múltiples aplicaciones a un mismo tiempo y para diversos propósitos independientemente de la clase de dispositivos de almacenamiento o métodos de acceso.

Un sistema de esta naturaleza debe de considerar:

- Independencia entre los programas de aplicación y datos con respecto a cambios en la estructura de los datos.
- Soportar lenguajes de programación que se usen en el medio ambiente en que trabajará.
- Tener herramientas de apoyo que faciliten la creación, mantenimiento, reestructuración y reorganización de los

datos.

- Opcionalmente puede incluir un diccionario de datos que ayudará en la documentación de las aplicaciones y en la estructuración de los archivos de la Base de Datos.

Un DBMS incluye cuatro componentes principales: datos, hardware, software y usuarios.

DAIOS

Los datos estan almacenados de tal forma que pueden considerarse como una unificación de varios archivos independientes de datos, donde se elimina total o parcialmente cualquier redundancia entre los mismos. Además pueden compartirse entre varios usuarios, de tal forma que cada uno puede tener acceso a la misma parte de la base de datos y utilizarla con propósitos diferentes.

HARDWARE

El hardware comprende todos los componentes físicos que se necesitan para almacenar y manejar los datos.

USUARIOS

Se consideran tres clases de usuarios: programador de aplicaciones, usuario final y el administrador.

El programador de aplicaciones es la persona encargada de desarrollar las aplicaciones para el usuario final que accesa la base desde una terminal. El usuario final también puede consultar la base utilizando un Lenguaje de Consulta.

Ambos usuarios pueden realizar funciones de recuperación, creación, supresión y modificación de datos.

La tercera clase , el administrador de la base de datos, es el encargado de controlar los datos. Se requiere que tenga un elevado

nivel de destreza técnica y capacidad para controlar e interpretar los requerimientos.

Entre sus responsabilidades están:

- a) Decidir el contenido de la información de la base de datos, identificando las entidades de interés para la empresa y la información que debe registrarse.
- b) Dar mantenimiento al Diccionario de datos. En el se almacena información sobre los datos y sus características.
- c) Decidir la estructura de almacenamiento y la estrategia de acceso.
- d) Vincularse con los usuarios para garantizar que los datos que requieren estén disponibles.
- e) Definir una estrategia de respaldo y recuperación, para reparar los daños causados por una falla de hardware, o por el sistema operativo; reduciendo al mínimo las repercusiones de ella.
- f) Controlar el desempeño y responder a los cambios de requerimientos.

SOFTWARE

El software está constituido por el conjunto de programas que permiten darle mantenimiento a los datos y recuperarlos para que puedan ser usados por los programas de aplicación.

II.5 OBJETIVOS DE UN MANEJADOR DE BASE DE DATOS

El manejador de datos debe perseguir los siguientes objetivos para que se permita el desarrollo de sistemas de información integrados, alrededor de él:

- a) **INDEPENDENCIA:** Los programas de aplicación y las consultas del usuario no se vean afectados al cambiar la estructura lógica o física de la base.
- b) **SINCRONIZACION:** A menudo, más de un usuario puede estar

- ejecutando alguna aplicación al mismo tiempo, pero el DBMS debe de controlar los accesos para evitar inconsistencias al tratar de actualizar un mismo registro simultáneamente.
- c) **IRREDUNDANCIA:** Se debe de evitar la redundancia de los datos lo más posible aunque en algunos casos es necesaria para relacionar información de diferentes bases de datos.
- d) **INTEGRIDAD:** Debe tener la capacidad de garantizar que la información y las relaciones no puedan destruirse al adicionar, borrar o cambiar un dato.
- e) **FLEXIBILIDAD:** Esta es la capacidad de hacer consultas por medio de cualquier llave de acceso o de alguna relación lógica de los datos.
- f) **SEGURIDAD:** No todos los usuarios deben tener acceso a la información, debe protegerse contra el acceso accidental o mal intencionado de algunos usuarios.
- g) **DESEMPEÑO:** Las bases de datos diseñados para usarse por operadores de terminal debe asegurar un tiempo adecuado de respuesta entre el hombre y la terminal. También se debe tomar en cuenta el volumen de transacciones que va a manejar la base, pues este hecho puede poner ciertas restricciones para su diseño.

II.6 ORGANIZACIONES LOGICAS.

Dependiendo de la estructura lógica que se utilice para relacionar los datos de la base, éstas se pueden clasificar en tres:

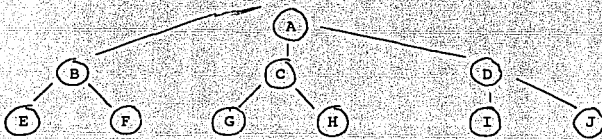
Modelo Jerárquico
Modelo de Redes
Modelo Relacional

MODELO JERÁRQUICO:

Se base en manejar la información como una estructura en forma de árboles.

Estos últimos están compuestos por una jerarquía de elementos denominados nodos. El nivel más alto de la jerarquía tiene solo un nodo, al que denominamos raíz.

Con excepción de la raíz, todo nodo está vinculado con otro nodo de un nivel superior al que llama padre o antecesor. Cada antecesor puede tener uno o más elementos de nivel inferior relacionado con el (hijo o sucesores) pero solo puede tener un padre (Fig. 2.2).



F I G U R A 2.2

A los nodos que no tienen sucesores se les llaman hojas.

Se llamará rama a la conexión que existe entre el sucesor y su predecesor (ejem. en la fig. 2.2 una rama lo constituyen los nodos B, C y D).

El nivel de un árbol lo determina el número de nodos que precedan a una hoja para llegar a la raíz. (En árbol de la fig. 2.2 tiene nivel 3).

Se llamará archivos jerárquicos a aquellos que muestren relaciones de tipo árbol entre sus registros.

Algunos paquetes de base de datos están diseñados para manejar solo archivos planos y archivos jerárquicos.

El problema es que no muchas estructuras de datos pueden respetar el hecho de que solo provengan de un padre. Por lo que en algunos casos se ve limitado el uso de este tipo de modelos.

MODELO DE REDES:

Este modelo está basado en la teoría de redes y permite a cada nodo tener más de un predecedor, aunque esto origina estructuras complejas, como son los llamados ciclos y lazos.

Los ciclos ocurren cuando un hijo tiene como descendiente a un antecesor (fig. 2.3).

A

B

F I G U R A 2.3

Los lazos se forman cuando un hijo tiene un solo registro. Todas estas estructuras complejas pueden simplificarse introduciendo cierta redundancia, pero en algunos casos ésta puede ser excesiva y podrá confundir a quien examine el modelo.

El modelo de redes requiere de un sistema más elaborado para recuperar la información pues su forma de almacenamiento físico es más sofisticada.

MODELO RELACIONAL

El enfoque relacional de base de datos constituye un concepto diferente para la descripción y manipulación lógica de los datos. Esta basado en los principios del álgebra relacional y normalizaciones. Por lo que el conjunto de operaciones válidas en ella se pueden aplicar al modelo proporcionando resultados predecibles.

En el modelo relacional se ve a la base de datos como una colección de tablas bidimensionales llamadas relaciones.

A cada relación se le da un nombre y se denota como $R(X,Y,Z)$, donde R es el nombre de la relación y esta compuesta por los atributos X, Y y Z. Ejemplo:

PROVEEDOR(CVE-PROVEEDOR, NOMBRE, DIRECCION, TELEFONO)

Cada columna de la relación tiene una etiqueta llamada atributo que sirve para describir la información que contiene.

En este modelo no se involucran conceptos como apuntadores, o aspectos posicionales como se ve en los otros modelos. Gracias a esto el usuario puede vislumbrar mejor la configuración lógica de los datos sin necesidad de un entrenamiento profundo.

Cualquiera de los otros dos modelos se pueden transformar en un modelo relacional adicionando relaciones.

Las ventajas de este modelo son que:

- Es entendible por el usuario.

- Existe mayor independencia de como estan almacenados físicamente los datos y su uso dentro de los programas de aplicación.
- Para tener acceso a la información no se tiene que entrar por la raíz como en el caso de los modelos anteriores, sino que se pueden consultar directamente.
- Es flexible, pues a través de algunas operaciones básicas se pueden derivar otras relaciones equivalentes. Aunque esto puede afectar la eficiencia de la máquina.

II.7 DISEÑO DE UNA BASE DE DATOS.

Como se mencionó anteriormente, para iniciar el diseño de la base de datos se requiere tener definido el MODELO CONCEPTUAL del sistema. El cuál proporciona una descripción de los datos que forman la base, las operaciones que se necesitan para crearlos y manipularlos, así como las reglas que gobiernan estas operaciones. El modelo debe ser estable, la descripción de un elemento particular, no tiene que cambiar una vez que se ha incorporado al modelo, a menos que ocurra un cambio en la parte del mundo real descrita por esa entrada; si el modelo conceptual no es estable en este sentido, entonces es probable que las aplicaciones tampoco lo sean y ello confundirá al usuario, aumentará la necesidad de reprogramar y entrañaría un riesgo mayor de error.

Tampoco debe estar influido por consideraciones relativas a la manera en que se van a almacenar y tener acceso físicamente los datos.

Las propiedades más importantes del esquema son:

- 1) Debe ser tan sencillo como sea posible en la práctica.
- 2) Debe tener una base teórica sólida.

Sencillez.

Esta parte se refiere a que debe ser fácil de entender y manejar. Para que el esquema sea fácil de comprender, se deben considerar los siguientes aspectos:

- Debe considerarse construirlo con un número de bloques que mantengan un tamaño manejable.
- Debe haber independencia en los datos, los conceptos distintos deben estar separados, pues de otra manera, resultaría difícil definir cuál es el objeto de esa construcción en una situación dada.
- Debe controlarse la redundancia, es decir, un mismo dato no debe aparecer en dos sitios a la vez, aunque a veces no es posible eliminarlo. En este caso, el esquema debe incluir una declaración que aclare exactamente cuál es la redundancia.
- Debe estar a un nivel tal que el usuario pueda entender que operaciones son posibles.
- Debe asegurar la integridad de la información.

Base teórica.

Dada la importancia del esquema, es esencial que se funde en una base teórica sólida, para poder predecir totalmente su comportamiento y, en la medida de lo posible, debe concordar con las expectativas intuitivas del usuario.

CAPITULO III

COMUNICACION ENTRE MICROCOMPUTADORAS Y MACROCOMPUTADORAS

III.1 INTRODUCCION

Dentro de los avances tecnológicos en el área de la computación, podemos citar el desarrollo de paquetes que permiten establecer comunicación entre las Microcomputadoras y las Macrocomputadoras. Algunos de los ellos solo permiten a la PC emular ser un terminal (XTALK, PCTOOL) ; mientras que otros la habilitan para trabajar en el ambiente de la macrocomputadora (PANLINK, ARBITER).

En este estudio nos enfocaremos a la conexión entre una macrocomputadora IBM y otras computadoras personales compatibles. En este trabajo se analizarán los siguientes paquetes: PANLINK, ARBITER.

Ellos funcionan a través de reservar una área física en la macrocomputadora y usarla desde la microcomputadora, está la incluye en su configuración como discos virtuales.

Además éstos paquetes cuentan con rutinas especiales para transformar los archivos en código de la macrocomputadora (EBCDIC), al código de la microcomputadora (ASCII), para así poder acceder la información desde cualquier aplicación de microcomputadora como son: Lotus, DbaseIII, Clipper, en general cualquiera que use acceso a discos con sistema operativo MS-DOS son compatibles.

III.2 PANLINK.

Panlink es una aplicación que se ejecuta en la macrocomputadora como un subsistema de manejador de sistemas en línea de la

macrocomputadora o desde su editor, y como se dijo anteriormente supone la existencia de discos virtuales desde 32 K hasta 30 M. Para su instalación no se requiere cambios en la red de comunicación, o de entrenamiento especializado.

El software de este paquete se puede dividir en dos: una parte que reside en la microcomputadora y la parte de la macrocomputadora. Una de las desventajas del paquete es que consume muchos recursos tanto en la macrocomputadora como en la microcomputadora, por lo que no es conveniente que muchos usuarios lo accesen a la vez, o que las aplicaciones ocupen mucha memoria.

En la macrocomputadora los discos se definen como archivos secuenciales, el paquete soporta hasta 255. Cada uno de ellos se subdivide en cinco áreas, las cuáles corresponden a un disco virtual de la microcomputadora y cada una posee el directorio de los archivos que residen en ellas.

Con ayuda de comandos de Panlink, las áreas pueden cargarse a uno de los discos definidos previamente en la configuración de la PC. Antes de que la PC pueda tener acceso a algún archivo secuencial, se debe haber establecido comunicación con la macrocomputadora a través de las subrutinas de interfase con las que cuenta Panlink. Estas pueden realizar las siguientes funciones en ambos sentidos:

- Copiar, borrar, renombrar o crear archivos entre los discos virtuales.
- Extraer información del área compartida para enviarla a otro dispositivo único de cualquiera de los ambientes.
- Grabar información en algún disco virtual, solo una persona puede accederlo a un tiempo para escribir en él, aunque varias pueden estar leyendolo a la vez.
- Las aplicaciones en la microcomputadora pueden intercambiar datos de forma conversacional con una o más rutinas a través de dos Buffer, en uno la PC envía datos a la macrocomputadora, y en el otro,

ésta se los envía a la PC.

También tiene un tercer Buffer para guardar los valores que tiene la subrutina en la macrocomputadora, en caso de que se le indique a Panlink que volverá a ser llamada.

- Subrutinas de la PC que estén escritas en los lenguajes BASIC, ENSAMBLADOR O LENGUAJE C, pueden abrir, cerrar o montar discos virtuales con simples comandos de Panlink dados desde ella.

RECURSOS HUMANOS

- El operador de PC debe conocer al menos los comandos del Sistema operativo MS-DOS y las instrucciones de interfase de Panlink.
- El programador de la IBM debe conocer el manejador de trabajos de lectura y escritura, y en caso de tener aplicaciones en línea, debe conocer el paquete que las controle, además de la utilería de Panlink.

HARDWARE

Para la PC:

- Una microcomputadora compatible con IBM con disco duro.
- Al menos 128 k en la memoria de la PC La cantidad de memoria requerida depende del programa de aplicación y del tamaño del programa emulador.
- Para comunicarse emulando una terminal se necesita:
 - . Una tarjeta emuladora.
 - . Estar conectado a la red de comunicación por de cable o por modem.
 - . Definir los disco virtuales en la configuración.

Para la IBM:

- Definir el archivo secuencial, con sus cinco subdivisiones.
- Definir la terminal que va a monitorear al sistema de Panlink. Esto puede ser desde algún sistema en línea o desde el editor de la maquina.

SOFTWARE

La microcomputadora debe tener:

- Sistema operativo MS-DOS.
- Las subrutinas de interfase de Panlink para interfase entre PC y Macrocomputadora.

En la macrocomputadora IBM:

- Las subrutinas de interfase de Panlink para interfase entre la Macrocomputadora y PC.
- Contar con una región de aplicación en línea.

III.3 ARBITER

ARBITER, es otro paquete que permite establecer la comunicación entre una macrocomputadora IBM y una microcomputadora, estableciendo una interfase que les permite tener acceso a datos y tareas en común, simulando discos virtuales.

Arbiter es un sistema de comunicación que puede soportar a cientos de microcomputadoras simultaneamente por estar en interfase directa con el paquete de acceso virtual de telecomunicación de IBM (ACF/VTAM). El ser un subsistema de éste y no ejecutarse en una aplicación de ella le da ventajas en tres aspectos principalmente: en eficiencia, seguridad e integridad.

EFICIENCIA. Arbiter da una mejora al tiempo de respuesta de los usuarios de PC, sin afectar el tiempo que dan otras aplicaciones en línea. Además permite utilizar más eficientemente a la red, por su avanzado método para comprimir los datos.

SEGURIDAD.

A diferencia de otros paquetes no se ven afectados por fallas propias de la aplicación en la que se ejecutan o por no encontrarse activa.

Además no requiere un sistema de seguridad independiente porque utiliza el paquete de seguridad de la IBM (ACF2), por lo que no se requiere de un mantenimiento especial.

Por último cuenta con un sistema detallado de auditoria con el cual se puede dar seguimiento a las actividades realizadas en la macrocomputadora por el usuario de PC y además llevar una contabilidad para poder cargar a cada usuario el uso del sistema en la macrocomputadora.

CAPITULO IV

CASO PRACTICO

IV.1 ANALISIS DEL SISTEMA AUXILIAR DE PRODUCCION Y VENTAS DE UNA EMPRESA AUTOMOTRIZ.

PROBLEMA

Existe incongruencia en los reportes que genera cada área sobre la cantidad de unidades que hay en cada uno de los procesos de producción, ya que cada una de ellas los genera utilizando la información residente en su PC, además de utilizar diferentes fuentes de información.

OBJETIVO.

El sistema deberá dar a conocer la forma en que se distribuyen los vehículos de una planta ensambladora durante un periodo, considerando los siguientes puntos de vista:

- Tipo de Modelo: Se consideran las características de las unidades que se ensamblan en la planta, como son: la línea, el paquete básico y opcional, el tipo de motor, la transmisión, etc.
- Distribución: Dependiendo de la forma en que se organicen los distribuidores de la red y su ubicación.
- Proceso de Producción: Se consideran cuatro fases dentro del proceso: vehículos ordenados, ensamblados, fabricados y los facturados durante un año-modelo.

Se le llama año-modelo al periodo comprendido desde que se ensambló el primer vehículo de una serie hasta que se deja de producir.

ALCANCE

El sistema se desarrollará en un ambiente de PC compatible con IBM y está basado en la información de los Sistemas de Producción que residen en un computadora IBM 4381.

También permite seleccionar varios modelos y comparar el total de vehículos por proceso, tomando periodos de 4 semanas o hasta 4 meses. No permite hacer consultas por unidad.

MODULOS DEL SISTEMA

En la figura IV.1 se presentan los módulos principales del sistemas, ellos son:

1. Generación de información de Producción
2. Control de Catálogos.
3. Consultas de Producción y Ventas.
4. Mantenimiento.

A continuación se describirán cada uno de ellos.

DIAGRAMA FUNCIONAL DE UN SISTEMA AUXILIAR DE CONTROL DE PRODUCCION DE AUTOS.

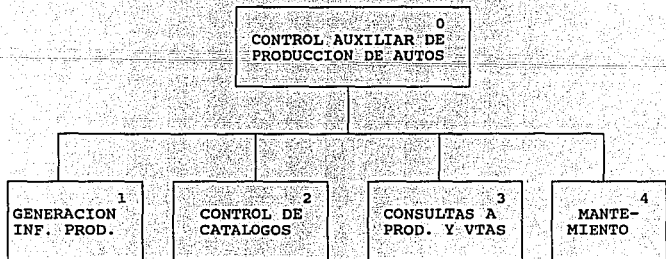


FIGURA IV.1

1. GENERACION DE INFORMACION DE PRODUCCION.

Es el módulo encargado de recibir la información extraída de la macrocomputadora con el estatus de la producción, desde el inicio del año-modelo a la fecha. Además la valida y convierte a las claves de sus catálogos.

Los submódulos que lo forman son:

- 1.1) Transformación de información a las claves de PC.
- 1.2) Generación de archivos de producción.

En la figura IV.2 se muestra el diagrama funcional del módulo y a continuación se describe el contenido de ellos.

DIAGRAMA FUNCIONAL DEL MODULO 'GENERACION DE INFORMACION DE PRODUCCION'.

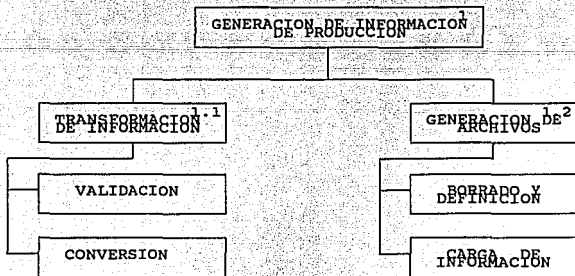


FIGURA IV.2

1.1 TRANSFORMACION DE INFORMACION A LAS CLAVES DE PC.

Modifica la información recibida de la macrocomputadora a las claves de sus catálogos para generar posteriormente los archivos que contiene información sobre la producción.

Se maneja una clave diferente a la de la macrocomputadora para ahorrar espacio y para optimizar el tiempo de transmisión de los datos de una máquina a otra.

Además valida que todas los registros puedan ser interpretados por sus catálogos; en caso de no ser así, lo reporta para que se tomen las medidas pertinentes.

Copia los archivos de producción del área compartida con la PC, para posteriormente desde esta última recuperarlos y copiarlos a disco duro.

Cabe hacer notar que previamente en la macrocomputadora se debe ejecutar un proceso para extraer la información de los Sistemas de Producción y la deje en el área común.

1.2 GENERACION DE ARCHIVOS DE PRODUCCION.

Este modulo comprende las funciones necesarias para borrar del disco el archivo de producción, definirlo y cargarle la información, ya validada y transformada en el módulo anterior.

2. CONTROL DE CATALOGOS.

Comprende las funciones necesarias para dar mantenimiento a los catálogos usados para controlar:

- 2.1 Modelos.
- 2.2 Distribuidores.
- 2.3 Zonas de Distribución.
- 2.4 Regiones de Distribución.
- 2.5 Países y Estados de ventas.

Cada uno de ellos forma un submódulo, el diagrama funcional de el se encuentra en la figura IV.3.

Cada submódulo a su vez cuenta con las siguientes funciones para: altas, bajas, consultas y cambios. El catálogo de modelos presenta otras opciones para definir los diferentes tipos de modelos que ensambla la planta.

Además existe la posibilidad de generar reportes de cada uno de ellos.

DIAGRAMA DEL MODULO DE CONTROL DE CATALOGOS

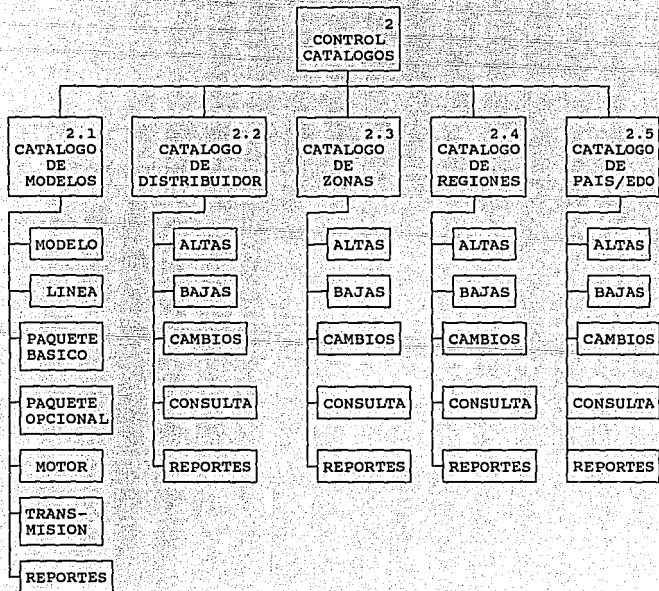


FIGURA IV.3

2.1 CATALOGO DE MODELOS.

Aqui se le da mantenimiento a los diferentes tipos de modelos que ensambla la planta y cada una de las opciones que lo forman como son:

<u>DESCRIPCION</u>	<u>CLAVE</u>	<u>CLAVES</u>
	<u>PC</u>	<u>MACRO</u>
Línea:		
COMPACTO	S	
ESPECIAL	E	
DEPORTIVO	P	
DE LUJO	N	
Paquete Básico:		
TÍPICO	T	AME
EQUIPADO	E	AMG
DEPORTIVO	R	AGG

<u>DESCRIPCION</u>	<u>CLAVE</u>	<u>CLAVES</u>
	<u>PC</u>	<u>MACRO</u>
Paquete Opcional:		
PREMIUN	P	ALS
LUJO	L	AFF
PATRULLA	T	AHB
CONVERTIBLE	C	
DEPORTIVO	D	

Motor:		
TURBO	T	EDC
16 VALVULAS	V	EDS

2.5	D	EDC, EDX
Transmisión:		
AUTOMATICA	A	DGA
MANUAL	M	DGH
Número de puertas:		
	2	2
	4	4
Tipo de Venta:		
FLOTILLA	F	4KT
MENUDEO	M	
MAYOREO	A	

El tipo de modelo se forma tomando uno de los elementos de las opciones anteriores, aunque en el caso del paquete opcional puede omitirse.

2.2 CATALOGO DE DISTRIBUIDORES.

Clasifica a los distribuidores de acuerdo a la ubicación de la agencia. Además cuenta con información para poder comunicarse a ella.

El catálogo nos proporciona la siguiente información:

- Clave de Distribuidor
- Nombre del distribuidor.
- Ubicación
- Nombre y teléfono de Gerente General.
- Región y Zona de ventas a la que pertenece.
- Estado y País.

2.3 CATALOGO DE ZONAS DE DISTRIBUCION.

Contiene información sobre las posibles zonas de ventas asignadas a los distribuidores. La información que proporciona es:

- Clave de la zona.
- Nombre de la zona.
- Región de Ventas asignada.
- Información sobre el Gerente de Ventas dentro de la empresa responsable de ella.

2.4 CATALOGO DE REGIONES DE DISTRIBUCION.

Contiene las diferentes regiones de ventas en que se dividen las zonas, además de:

- Clave de la región.
- Nombre de la región
- Datos sobre el Responsable de Ventas dentro de la empresa en la región.

2.5 CATALOGO DE PAIS / ESTADO.

Asocia una clave al nombre del país y lo relaciona con la región de ventas que le corresponde.

En el caso de México además cuenta con un desglose por estados.

3. CONSULTAS SOBRE PRODUCCION Y VENTAS.

Permite conocer el estado de la producción y establecer comparaciones entre periodos y tipos de unidades seleccionando un proceso de producción. Los periodos pueden comprender de una semana a cuatro meses.

Los procesos de producción son:

- ORDENES. Comprende todos los vehiculos desde que son pedidos por el distribuidor hasta que se aprueba su ensamble por la planta.
- ENSAMBLE. Incluye las unidades que la planta puede iniciar a ensamblar, y aquellas que están en la línea de producción sin terminar.
- FABRICADAS. Se consideran aquellas que están terminadas físicamente, pero que no han sido facturadas aún.
- FACTURADAS. Comprende las unidades facturadas durante el periodo.

Las consultas se realizan desde tres puntos de vista:

- 3.1 Por Tipo de Unidad.
- 3.2 Por Distribución Geográfica.
- 3.3 Por Regiones de Distribución.

Cada una corresponde a un submódulo, los cuales se describen a continuación.

En la figura IV.4 se proporciona el diagrama funcional del modulo.

DIAGRAMA FUNCIONAL DEL MODULO DE CONSULTAS SOBRE PRODUCCION Y VENTAS.

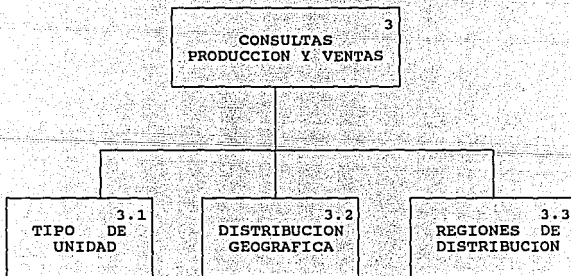


FIGURA IV.4

3.1 CONSULTAS POR TIPO DE UNIDAD.

Permite hacer consultas sobre los diferentes productos fijando:

- El tipo de unidad o fijando algunas opciones de él.
- El tipo de proceso.
- El mercado de ventas.
- El periodo a analizar.

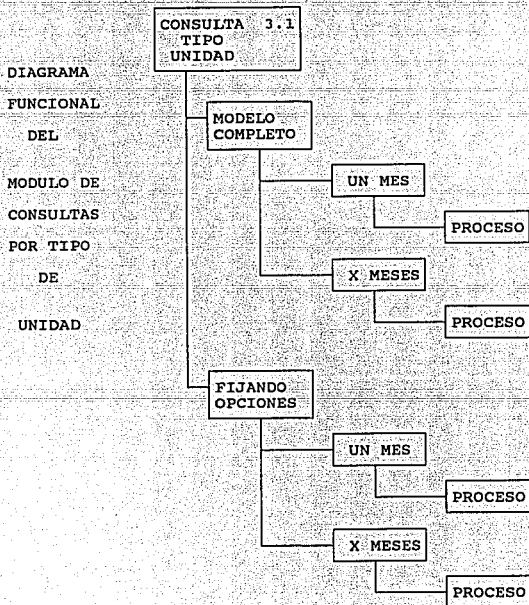


FIGURA IV.5

3.2 CONSULTAS POR DISTRIBUCION GEOGRAFICA.

Permite conocer como está repartido el producto desde un punto de vista geográfico, seleccionando:

- Los vehiculos ensamblados para un pais.
- En caso de que el pais sea México, se puede seleccionar un estado.
- El tipo o características de los modelos. (pueden ser hasta 10)
- El proceso de producción de interés.

El diagrama funcional se presenta en la figura IV.6.

3.3 POR REGIONES DE DISTRIBUCION.

Permite obtener información seleccionando:

- La región de ventas.
- Una de las zonas de la región o todas.
- El tipo o características de los modelos. (pueden ser hasta 10)
- El proceso de producción.

En la figura IV.7 se encuentra el diagrama funcional.

DIAGRAMA FUNCIONAL DE CONSULTAS POR DISTRIBUCION GEOGRAFICA

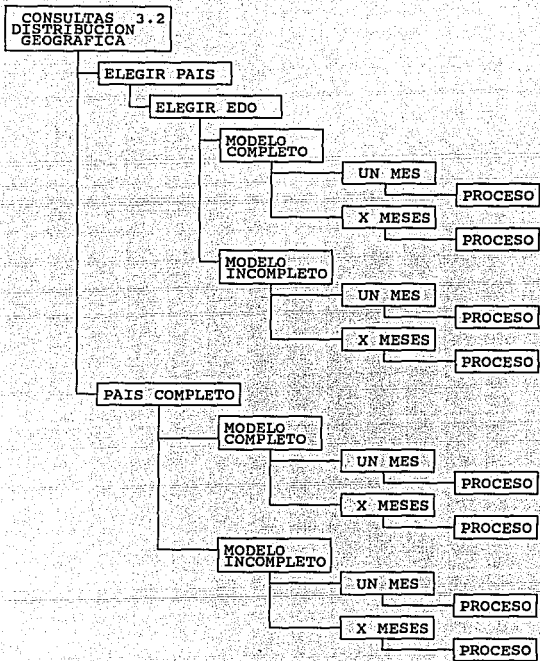
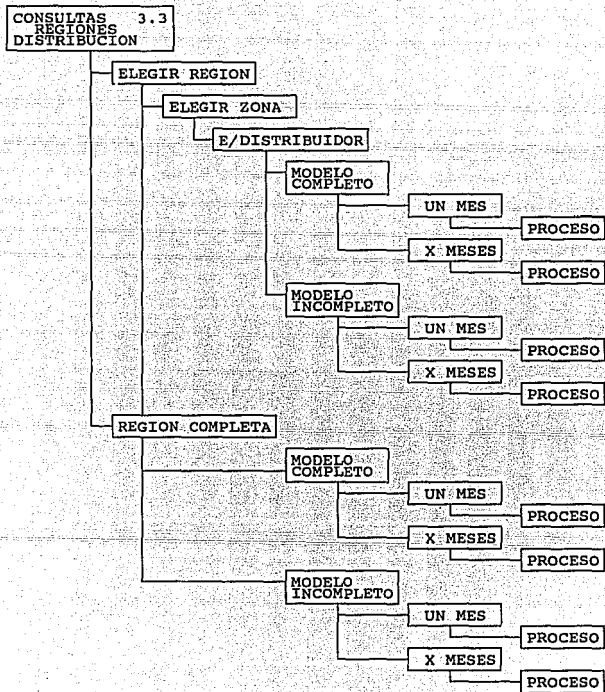


FIGURA IV.6

DIAGRAMA FUNCIONAL DE CONSULTAS POR REGIONES DE DISTRIBUCION



F I G U R A IV.7

4. MANTENIMIENTO.

Cuenta con funciones para mantener los archivos que radican en la PC en óptimas condiciones y asegurar su regeneración en caso de que ocurriese algún percance (fig.IV.8).Estas funciones son:

- 4.1 Reorganizar los archivos.
- 4.2 Respaldar la información.
- 4.3 Reinstalarla a partir de un respaldo.

Cada función corresponde a un submódulo, los cuáles serán descritos posteriormente.

4.1 REORGANIZA LOS ARCHIVOS.

Comprende los procedimientos para reorganizar el área de los archivos debido a la eliminación de registros y para optimizar el espacio en disco.

Para lograrlo respalda la información, borra y define los archivos, y posteriormente les vuelve a carga la información que tenían.

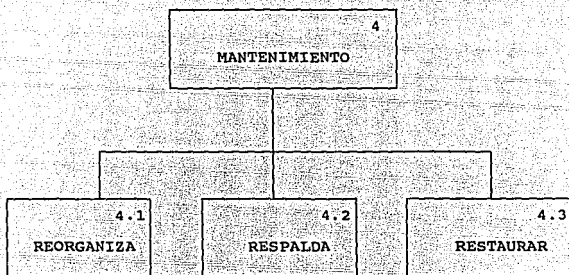
4.2 RESPALDA LA INFORMACION.

Se encarga de respaldar en disco flexible periodicamente los archivos, para poder restaurarlos en caso de algún percance.

4.3 RESTAURAR A PARTIR DE UN RESPALDO.

Permite definir el archivo y cargar la información desde un disco flexible.

DIAGRAMA FUNCIONAL DEL MODULO DE MANTENIMIENTO.



F I G U R A IV.8

IV.2 DISEÑO DE LA BASE DE DATOS.

IV.2.1 PLANTEAMIENTO DE LA ESTRATEGIA.

El diseño del Sistema Auxiliar de Producción y Ventas se hará utilizando un modelo relacional de Base de Datos siguiendo el proceso de Síntesis Canónica propuesto por James Martin (8), el cual se explica en el apéndice A.

El Software a utilizar para el manejo de la base de datos en la PC será Clipper, por las facilidades de manejo que aporta.

Para intercambiar datos entre la PC y la Computadora IBM 4381, se usará Panlink, y para extraerlos de la Macrocomputadora, programas en COBOL.

Los usuarios de la base serán:

- Personal del Departamento de Control de Producción.
- Personal del Departamento de Ventas y Mercadotecnia.
- Personal de la Dirección General.

IV.2.2 PROCESO DE MODELADO DE DATOS.

El Sistema contará con la facilidad de enviar sus resultados a una impresora o a la pantalla a través de Cuadros de Resultados con el siguiente formato (FIGURA IV.9):

UNIDADES FACTURADAS			
PERIODO: ENERO - MARZO 1992			
PAIS: MEXICO			
ESTADO: GUANAJUATO			
TIPO UNIDAD	ENERO	FEBRERO	MARZO
TIPICO DEPORTIVO DE LUJO			

FIGURA IV.9

A continuación se presenta el Diccionario de Datos del Sistema:

NOMBRE	DESCRIPCION DEL CAMPO	TIPO	TAMAÑO
CVE-TMOD	Clave del modelo	Car	6
CVE-LINEA	Clave de la línea	Car	1
M-CVE-LINEA	Clave de línea en Macro	Car	6
LINEA	Descripción de línea	Car	10
CVE_PB	Clave de Paq. Básico	Car	1
M-CVE-PB	Clave Paq. Básico en Macro	Car	3
PAQ-BASICO	Descripción Paq. Básico	Car	10
CVE-PO	Clave de Paq. Opcinal	Car	1
M-CVE-PO	Clave Paq. Opcional (MACRO)	Car	3
PAQ-OPCIONAL	Descripción Paq. Opcional	Car	10
CVE-MOTOR	Clave del motor	Car	1
M-CVE-MOTOR	Clave del motor en Macro	Car	6
MOTOR	Descripción del motor	Car	10
CVE-TRANS	Clave de la transmisión	Car	1
M-CVE-TRANS	Clave de transmisión (m)	Car	6
TRANSMISION	Descripción transmisión	Car	10
NUN-PTAS	Número de puertas	Car	1
TIPO-VENTA	Tipo de venta	Car	1
CANT-ORD	Total de unidades ordenadas durante una semana.	Car	5
CANT-FAB	Total de unidades fabricadas durante una semana.	Car	5
CANT-ENS	Total de unidades ensambladas durante una semana.	Car	5

Car : carácter

Num : Numérico

NOMBRE	DESCRIPCION DEL CAMPO	TIPO	TAMAÑO
CANT-FACT	Total de unidades facturadas durante una semana.	Car	5
PAIS	Descripción del país.	Car	10
CVE-PAIS	Clave del país.	Car	2
ESTADO	Nombre del estado	Car	15
CVE-EDO	Clave del estado	Car	2
MES-SEM	Año, mes y semana	Car	5
DISTRIBUIDOR	Nombre distribuidora	Car	20
CVE-DIST	Clave distribuidora	Car	5
GER-DIST	Gerente distribuidora	Car	15
DIREC-DIST	Dirección distribuidora	Car	20
TEL-DIST	Teléfono del gerente	Car	9
REGION	Nombre de la región	Car	15
CVE-REG	Clave de la región	Car	1
RES-REG	Responsable de la región	Car	15
DIREC-REG	Dirección del Responsable	Car	20
TEL-REG	Teléfono del Responsable	Car	9
ZONA	Nombre	Car	15
CVE-ZN	Clave de la zona	Car	2
RES-ZN	Responsable de la zona	Car	15
DIREC-ZN	Dirección del Responsable	Car	20
TEL-ZN	Telefono del Responsable	Car	9

Car : carácter

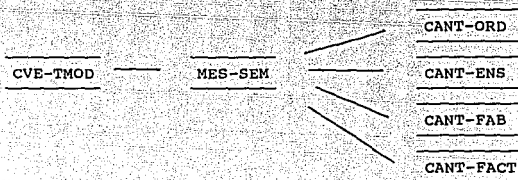
Num : Numérico

IV.2.3 CREACION DEL MODELO DE DATOS.

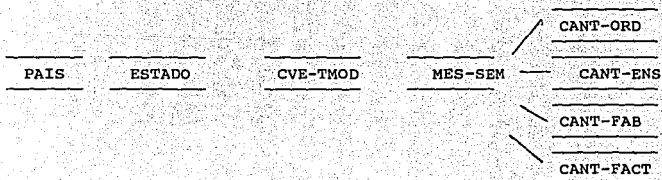
A continuación se explican los pasos seguidos para el desarrollo del modelo:

A. Se representan las diferentes vistas del sistema como Cartas de Burbuja señalando los atributos, las llaves primarias y las asociaciones entre ellos.

- POR TIPO DE UNIDAD.



- POR DISTRIBUCION GEOGRAFICA.



- POR DISTRIBUCION GEOGRAFICA

MES SEM	PAIS	ESTA- DO	CVE TMOD	CANT ORD	CANT ENS	CANT FAB	CANT FACT
------------	------	-------------	-------------	-------------	-------------	-------------	--------------

CVE TMOD	CVE PAIS	CVE EDO	LLA PEM
-------------	-------------	------------	------------

MES SEM	LLA PEM	CANT ORD	CANT ENS	CANT FAB	CANT FACT
------------	------------	-------------	-------------	-------------	--------------

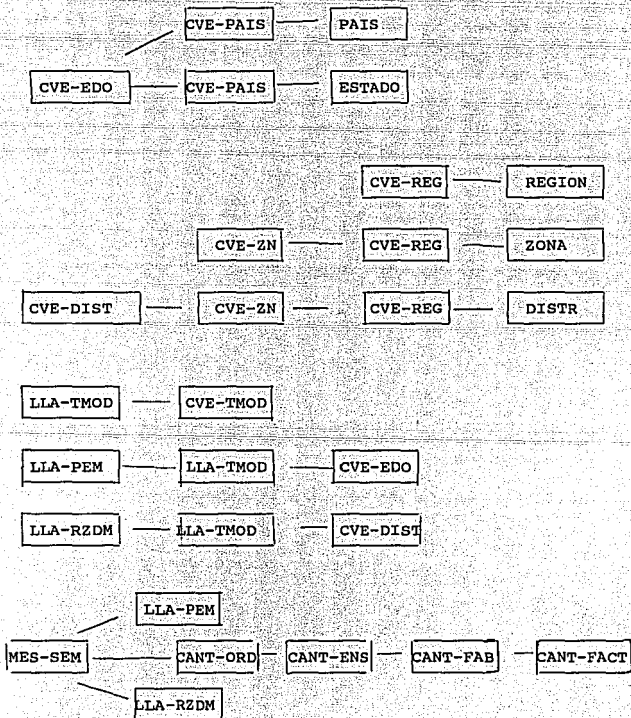
- POR REGION DE DISTRIBUCION.

MES SEM	RE- GION	ZONA	DIST.	CVE TMOD	CANT ORD	CANT ENS	CANT FAB	CANT FACT
------------	-------------	------	-------	-------------	-------------	-------------	-------------	--------------

CVE TMOD	CVE REG	CVE ZN	CVE DIST	LLA RZDM
-------------	------------	-----------	-------------	-------------

MES SEM	LLA RZDM	CANT ORD	CANT ENS	CANT FAB	CANT FACT
------------	-------------	-------------	-------------	-------------	--------------

C. Los registros obtenidos se agrupan en una sola Carta de Burbuja. Esta se volverá a normalizar, se revisarán las asociaciones M entre las llaves, se revisarán los atributos aislados y se eliminarán los intersectados.



Al revisar las asociaciones M:N, se ve que es poco probable que existan en un futuro las relaciones inversas, por lo que no se van a incluir.

Los atributos intersectados, Can-ord, Can-ens, Can-fab y Can-fact, se van a eliminar adicionando una liga entre las CVE-PED y la CVE-RZD.

- D. El modelo resultante está ya normalizado, no tiene atributos aislados, ni intersectados, ni asociaciones redundantes; por lo que se pasa a redibujar dejando al principio las llaves raíz seguidas de los atributos que componen al modelo. Teniendo en cuenta las funciones que debe satisfacer el modelo se da una cuenta que conviene formar la CVE-RZD-MV con los códigos de los elementos que la forman, para facilitar la recuperación de la información.
- A continuación se presenta el modelo de datos resultante.

MODELO DE LA BASE DE DATOS

CVE TMOD	CVE LINEA	CVE PB	CVE PO	CVE MOTOR	CVE TRANS	TIPO VENTA
-------------	--------------	-----------	-----------	--------------	--------------	---------------

CVE DISTR	CVE REG	CVE ZN	CVE EDO	DISTRIBUIDOR	GER DISTR	TEL DISTR	DIREC DISTR
--------------	------------	-----------	------------	--------------	--------------	--------------	----------------

CVE REG	REGION	RES REG	DIREC REG	TEL REG
------------	--------	------------	--------------	------------

CVE ZN	CVE REG	ZONA	RES ZN	DIREC ZN	TEL ZN
-----------	------------	------	-----------	-------------	-----------

ESTADO	CVE PAIS	CVE EDO
--------	-------------	------------

PAIS	CVE PAIS
------	-------------

CVE TMOV	CVE DISTR	CVE EDO	LLA RZDPM
-------------	--------------	------------	--------------



MES SEM	LLA RZDPM	CANT ORD	CANT ENS	CANT FAB	CANT FACT
------------	--------------	-------------	-------------	-------------	--------------

IV.2.4. INSPECCION DE LA SALIDA DEL MODELO.

Una vez creado el modelo se va a revisar que satisfaga los objetivos de cada uno de sus modulos, además se va a revisar con los posibles usuarios para comprobar que responden a las necesidades de información.

Finalmente se convierte el Modelo a un esquema lógico, que puede ser usado para dar de alta los archivos en CLIPPER, creando los siguientes archivos:

<u>MODELO</u>	<u>LINEA</u>	<u>PAOOPC</u>	<u>TRANS</u>
CVE-TMOD	CVE-LINEA	CVE-PO	CVE-TRANS
CVE-LINEA	LINEA	PAQ-OPCIONAL	TRANSMISION
CVE-PB	M-CVE-LINEA	M-CVE-PO	M-CVE-TRANS
CVE-PO			
CVE-MOTOR	<u>PAQBAS</u>	<u>MOTOR</u>	<u>TVENTA</u>
CVE-TRANS	CVE-PB	CVE-MOTOR	CVE-VTA
TIPO-VENTA	PAQ-BASICO	MOTOR	TIPO-VENTA
	M-CVE-PB	M-CVE-MOTOR	

<u>DISTR</u>	<u>REGION</u>	<u>ZONA</u>
CVE-DIST	CVE-REG	CVE-ZN
CVE-REG	REGION	CVE-REG
CVE-ZN	RES-REG	ZONA
CVE-EDO	DIREC-REG	RES-ZN
DISTRIBUIDOR	TEL-REG	DIREC-ZN
GER-DIST		TEL-ZN

<u>ESTADO</u>	<u>PAIS</u>	<u>PRODUCCION</u>
CVE-ESTADO	CVE-PAIS	MES-SEM
CVE-PAIS	PAIS	LLA-RZDPM
ESTADO		CANT-ORD
TEL-DIST		CANT-ENS
DIREC-DIST		CANT-FAB
		CANT-FACT

IV.2.5. CONSIDERACIONES FISICAS.

Dado las restricciones de espacio en lenguaje CLIPPER resulta necesario crear un archivo por cada uno de los meses para disminuir la cantidad de registros que tiene que estar en memoria en un momento determinado.

CAPITULO V

V. CONCLUSIONES

Al concluir este trabajo puedo decir que:

- La idea de que la información está reducida a permanecer en un centro de computo ha quedado desterrada gracias a los avances tecnológicos.
- Es importante coordinar el desarrollo de sistemas que intercambien información entre las PC y las macrocomputadoras para poder lograr integridad en la información que circula en la empresa.
- Es conveniente usar el método de Síntesis Canónica para el diseño de bases de datos porque es fácil de aplicar, ya que permite visualizar la relación entre los datos y las dependencias entre ellos.
- Aunque el aplicar este método para el diseño de bases con muchas variables y relaciones entre ellas se vería limitado en la elaboración de los diagramas de burbujas.
- En cuanto al paquete para intercambiar la información entre las máquinas, Panlik, tiene la desventaja que utiliza muchos recursos tanto en la PC como en la macrocomputadora.
Debido a esto las rutinas para copiar la información del disco virtual al disco duro, no pueden estar incluidas en el menú principal del sistema.
En este estudio se va a utilizar PANLINK, debido a

problemas de costo con respecto a ARBITER.

- En cuanto al análisis del sistema, falta considerar los objetivos de venta de cada distribuidor para cada uno de los meses, pues es común el querer compararlos con las ventas del mes.

Además habría de considerar la posibilidad de tener un archivo con las unidades y sus características principales para usarse en futuras aplicaciones.

- En cuanto al uso del sistema auxiliar evita que el departamento de sistemas elabore un sinnúmero de reportes y que el usuario capture la información de ellos en su PC.

CAPITULO VI

VII. BIBLIOGRAFIA

- 1.- Date C.J.
Introducción a los Sistemas de Bases de Datos.
Addison-Wesley Iberoamericana.
México, 1986.
- 2.- Wiederhold Gio
Diseño de Bases de Datos.
McGraw Hill.
México, 1985.
- 3.- Rivero Cornelio E.
Bases de Datos Relacionales.
Paraninfo.
Madrid, 1988.
- 4.- Cárdenas Alfonso F.
Sistemas de administración de bancos de datos.
Limusa.
México D.F., 1983.
- 5.- Ullman Jeffrey D.
Principles of Data Base Systems.
Computer Science Press, Inc.
U.S.A., 1982.
- 6.- McClanahan David R.
Relational Data Base Design.
Artículo publicado en los números 11 & 12 de la revista
DBMS.
U.S.A., Octubre-Noviembre, 1991.

7.- Business Systems Planning.
Information Systems Planning Guide.
Artículo publicado por IBM., 1984.

8.- Martin James.
Managing the Data Base Enviroment.
Prentice Hall Inc..
U.S.A., 1983.

TESIS

9.- Otaola Esquivel María del Pilar.
Bases de Datos y su aplicación en el diseño de un sistema
escolar utilizando dBase III.
Tesis para obtener el título de actuario.
México, 1987.

10.- Gasca Soto María de Luz.
La importancia del Diseño Estructurado en el desarrollo de
Sistemas de Información.
Tesis para obtener el título de matemático.
México, 1990.

APENDICE A: METODOLOGIA PARA EL DISEÑO DE UNA BASE DE DATOS.

A.1 METODOLOGIA PARA LA CREACION DE UN MODELO CONFIABLE.

La metodologia usada es la de Sintesis Canónica, presentada por James Martin (8) y comprende cinco etapas:

- I. Definir la estrategia a seguir.
- II. Entrada al proceso de modelado de datos.
- III. Creación del modelo de datos.
- IV. Inspección de la salida del modelo de datos.
- V. Consideraciones físicas.

Antes de puntualizar a cada una de ellas es importante dar algunas definiciones y explicar el proceso para generar un modelo lógico de la base, en este caso se plantea el de Sintesis Canónica, por la facilidad para interpretación.

A.1 DEFINICIONES.

ATRIBUTO. Es un dato que representa un dato elemental del sistema.

ATRIBUTO AISLADO. Un atributo que no está identificado por ninguna llave primaria, es decir a el no llegan, ni salen flechas.

ATRIBUTO INTERSECTADO. Es aquel al que llegan más de una flecha con una sola punta.

BURBUJA.

Se utiliza para representar un atributo.



FLECHA CON 1 PUNTA Una flecha con una punta de A a B, sirven para ligar dos burbujas, donde para cada valor de A tiene uno y sólo un valor de B asociado con él (Asociación 1), y se denota como:



FLECHA CON 2 PUNTAS Este tipo de flechas de A a B significa que un valor de A no tiene uno correspondiente en B, o tienen uno o muchos valores asociados con él (Asociación M).

RELACION ENTRE BURBUJAS Existen tres tipos de relación entre burbujas:

Relación uno a uno (1:1) :



Relación uno a muchos (1:M) :



Relación muchos a muchos (M:N) :



LLAVE

Atributo que sirve para identificar a otros campos asociados a ella.



LLAVE PRIMARIA

Es una burbuja de la que sale una flecha o más flechas pero todas de una sola punta.

**LLAVE SECUNDARIA.**

Es una burbuja de la que parte al menos una flecha con doble punta hacia otra burbuja.

**LLAVE CONCATENADA.**

Cuando la llave primaria esta formada por la composicion de dos o más atributos.

A +
B

LLAVE CANDIDATA.

Es la llave primaria de un registro normalizado. Ejemplo 'A'.

	A	B	C	D	E	F
--	---	---	---	---	---	---

LLAVE RAIZ.

Es una llave primaria de la que salen una o varias flechas, pero ninguna llega a ella.



REGISTRO NORMALIZADO. Cuando todos los campos que lo forman dependen

funcionalmente de la llave primaria, y el

regis
tro se
encue
ntra
e n
tercera
forma
normal.



CARTAS DE BURBUJAS: Son diagramas que permiten representar los datos de un sistema y las asociaciones que hay entre ellos usando burbujas, y flechas.

A.2. MODELO DE SINTESIS CANONICA.

El proceso de Sintesis Canónica se usa para crear un modelo lógico de datos, esta basado en las Cartas de Burbujas y en el proceso de Normalización, así el diagrama resultante de este proceso estará en tercera forma normal.

Los pasos a seguir para generar el modelo son:

- 1.- Se toma una vista de los datos, y se representa como Carta de Burbuja.
- 2.- Distinguir en el diagrama resultante los atributos, y las llaves primarias.
- 3.- Para cada asociación M, entre llaves, agregar la asociación inversa, en caso de no estar en la gráfica. Si la relación resultante es M:M, determinar si tiene verdadera posibilidad de ser usada en un futuro; en caso de que si, reemplazarla introduciendo una Llave Concatenada.

4.- Eliminar los atributos intersectados siguiendo alguno de los tres caminos siguiente:

- a) Reemplazarlo con ligas equivalentes a través de una llave existente.
- b) Duplicar el atributo, en caso de que no exista la posibilidad de utilizarlo como llave primaria en un futuro.
- d) Tratarlo como una llave solitaria.

5.- La Carta resultante se lleva a tercera forma normal. Es decir:

- Eliminar las Asociaciones Redundantes. (Primera forma Normal).
- Eliminar las dependencias funcionales entre los campos llave y los atributos dependientes. (Segunda Forma Normal).
- Suprimir las dependencias transitivas entre los atributos no llave. (Tercera Forma Normal)

6.- Repetir los seis pasos anteriores para cada una de las vistas, y después agruparlas en un mismo diagrama, hasta asegurándose que todas están incluidas en él, y aplicar los pasos anteriores al nuevo modelo.

7.- Identificar las Llaves Raíz.

8.- Revisar los Atributos Aislados de la gráfica, y eliminarlos de alguna de las siguientes maneras:

- a) Implementarlo como un atributo repetido en un registro de longitud variable.
- b) Considerarlo como llave solitaria.
- c) Revisar si no corresponde a un error de interpretacion de los datos, y si este es el caso, corregirlo.

9.- Redibujar los registro, dejando la llave primaria y sus atributos asociados.

10.- Identificar las llaves secundarias.

11.- Procurar que el modelo resultante sea lo más estable posible, tomando en cuenta:

- a) Como se van a utilizar los datos en un futuro.
- b) Revisando la posibilidad de que un atributo llegue a se llave.
- c) Borrando las flechas que sean redundantes.
- d) Que el modelo puede tener relaciones 1:1.
- e) Eliminando los ciclos, en caso de haberse formado.

A.3 PLANTEAMIENTO DE LA METODOLOGIA PARA EL DISEÑO DE UNA BASE DE DATOS.

I. Planteamiento de la Estrategia.

- Determinar las características de la base de datos a diseñar.
- Determinar todos los tipos posibles de usuarios finales de una Base de Datos.
- Seleccionar una herramienta de modelado y un diccionario de datos.
- Seleccionar el Software adecuado.

II. Proceso de modelado de datos.

- Recolectar todos los documentos que serán derivados de la Base de Datos o que servirán como entrada a ésta.
- Determinar con los usuarios finales que tipo de datos quieren obtener de la Base de Datos tanto actualmente como a futuro.
- Determinar a partir de un proceso de análisis de sistemas si alguno de los nuevos requerimientos son urgentes.
- Planear si los archivos o las Bases de Datos podran coexistir con la nueva Base, o si deberan de ser convertidos.
- Emplear un diccionario de datos para documentar los datos y la descripción del significado de cada uno.

III. Creacion del Modelo de Datos.

- Determinar las características de la base de datos a diseñar.
- Determinar todos los tipos posibles de usuarios finales de una Base de Datos.
- Seleccionar una herramienta de modelado y un diccionario de datos.
- Seleccionar el Software adecuado.

II. Proceso de modelado de datos.

- Recolectar todos los documentos que serán derivados de la Base de Datos o que servirán como entrada a ésta.
- Determinar con los usuarios finales que tipo de datos quieren obtener de la Base de Datos tanto actualmente como a futuro.
- Determinar a partir de un proceso de análisis de sistemas si alguno de los nuevos requerimientos son urgentes.
- Planear si los archivos o las Bases de Datos podran coexistir con la nueva Base, o si deberan de ser convertidos.
- Emplear un diccionario de datos para documentar los datos y la descripción del significado de cada uno.

III. Creacion del Modelo de Datos.

- Emplear un proceso para la generacion de un modelo que se encuentre en tercera forma normal. Se puede usar el de Sintesis Canónica.
- Inspeccionar cada entrada para ver si puede ser simplificada.
- Revisar que no se dupliquen los nombres de los atributos en el diccionario de datos.
- Emplear estandares convencionales para la seleccion de los nombres de los elementos que forman la base.
- Asegurarse que todos los atributos de entrada sean unicamente dependientes de la llave que los identifica.
- Asegurarse que los grupos de datos empleados no contienen dependencias transitivas.

IV. Inspección de la Salida del Modelo.

- Revisar con el usuario el diccionario de datos para asegurar que concuerdan las definiciones.
- Revisar el modelo resultante para tener la seguridad de que están incluidos todos los requerimientos.
- Establecer con los usuarios las posibles futuras asociaciones entre los datos.

- Revisar cada uno de los campos del modelo para determinar si pueden convertirse en llave primaria.
- Revisar las asociaciones inversas entre los datos, para asegurarnos de que nuestro modelo no tendrá relaciones M:N.
- Revisar las relaciones redundantes que se eliminaron para confirmar su redundancia.
- Verificar que todas las llaves Candidatas se conservarán así en el futuro.
- Confirmar el tratamiento dado a la eliminación de atributos de intersección.
- Inspeccionar los ciclos, y ver la posibilidad de existir una liga futura para romperlos.
- Hacer un rediseño del modelo, con todos los posibles cambios.

V. Consideraciones Físicas.

- Para cada patrón de utilización de los datos del modelo hay que añadir el volumen de utilización.
- Para cada liga A \longrightarrow B, determinar cuantos valores de B en promedio son asociados con un valor de A.

- Revisar la facilidad que da el software usado para desarrollar la base de datos en cuanto: adicionar nuevos atributos, o de ligas.
- Analizar el uso de las llaves secundarias, y ver la posibilidad de dividir la Base en dos.

APENDICE B. ALGEBRA RELACIONAL Y NORMALIZACION

Como ya se ha mencionado para poder construir un modelo relacional es necesario definir la estructura de las relaciones y sus operaciones básicas.

A continuación se mencionan los conceptos principales del álgebra relacional y de la teoría de Normalización.

R E L A C I O N E S

Las relaciones se definen en términos de parejas ordenadas y de lo que es el producto cartesiano.

PAREJA ORDENADA Si $a, b \in X$ la pareja ordenada formada por a y b se denota como (a, b) . Cuando se tiene más de dos elementos ordenados en lugar de pareja son eneadas ordenadas.

PRODUCTO CARTESIANO Sean A y B conjuntos. El producto cartesiano de A y B , $A \times B$, es el conjunto de parejas ordenadas.

$$A \times B = \{ (a, b) \mid a \in A \text{ y } b \in B \}$$

RELACION Sean A y B conjuntos. Una relación entre A y B es un subconjunto del producto cartesiano $A \times B$.

Se llama grado de una relación al número de componentes que tiene sus eneadas.

Dentro de una eneada se puede distinguir dos tipos de atributos: unos que identifican al objeto descrito en ella, la parte rectora y la otra que proporciona datos referentes al objeto, parte dependiente.

Se usa la siguiente notación para separar la parte rectora de la parte dependiente " :> ". La parte rectora se escribe antes del signo, mientras que la otra irá después de él.

DEPENDENCIAS DE LAS RELACIONES

Supongamos que se tienen dos relaciones A y B. En la que los valores de B están determinados por los valores que tome A, en cuyo caso se dice que B depende funcionalmente de A, y se denota de la siguiente manera:

$$FD(A_1, A_2, A_3, \dots, A_n) = B_1, B_2, \dots, B_n$$

Si el valor de B_i en cualquier enada depende totalmente de los valores A_i .

Entre dos o más relaciones podemos tener diferentes tipos de dependencias como:

DEPENDENCIA NO INVERSA. Aquella en la que no existe una función de B que de como resultado los valores de A.

DEPENDENCIA MULTIVALUADA Una dependencia multivaluada es aquella en que un elemento de A, determina un conjunto de valores de B.

DEPENDENCIA INVERSA Cuando existe una función de B que de como resultado los valores de A.

DEPENDENCIA TRANSITIVA Cuando existe una dependencia funcional entre los elementos de la parte

dependiente de una relación.

Cuando se tiene dependencias inversas entre la parte rectora y la parte dependiente de la relación, se da origen a una estructura circular. Ejem.

FD(codigo_postal)= ciudad

FD(ciudad)=codigo_postal.

Las dependencias multivaluadas no se pueden representar dentro de un modelo, utilizando solo una relación.

OPERACIONES BASICAS DE LAS RELACIONES

El álgebra relacional es un sistema cerrado de operaciones definidas sobre relaciones. Es cerrado porque, tanto los operandos, como los resultados son relaciones.

DEF. Dado un conjunto de dominios $D=(D_1, D_2, \dots, D_n)$, sea R el conjunto de todas las relaciones posibles que se puedan construir sobre D . El álgebra relacional es un sistema formado por R y unos operadores cuyos operandos y resultados también pertenecen a R .

Las operaciones básicas son las siguientes: unión, diferencia, intersección, producto.

Suponemos que existe R y S conjuntos de relaciones sobre los conjuntos $R=(R_1, R_2, \dots, R_n)$ y $S=(S_1, S_2, \dots, S_n)$.

Entonces diremos que :

UNION.

$R \cup S$ es el conjunto formado por todas las todas las eneadas de R y de S . Pero R y S deben ser del mismo grado.

DIFERENCIA

$R - S$ es el conjunto de las eneadas que están en R y no están en S . R y S deben tener el mismo grado.

PRODUCTO

$R \times S$ es el conjunto de parejas obtenidas al concatenar una eneada de R y otra de S .

PROPIEDADES DE LAS OPERACIONES

Las siguientes propiedades permiten transformar las relaciones de en otras equivalentes.

1) ASOCIATIVA.- La union y la interseccion son asociativas.

$$(A \cup B) \cup C = A \cup (B \cup C)$$

2) CONMUTATIVA.- La union y la interseccion son conmutativas.

$$A \cup B = B \cup A$$

3) DISTRIBUTIVA.- Propiedad distributiva de la union e interseccion.

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

VISTAS

Vista es un subconjunto del sistema original al que se le da un nombre. Este nombre puede utilizarse como operando en la construcción de una relación. El uso de vistas disminuye la complejidad de las relaciones.

REGLAS PARA LA TRANSFORMACION DE RELACIONES

Como se mencionó anteriormente, las relaciones pueden transformarse en otras equivalentes, que conserven la consistencia del esquema original.

Las siguientes transformaciones se conocen como axiomas de Armstrong :

i) La reflexibilidad.

Los subconjuntos de un conjunto de atributos son funcionalmente dependientes de su conjunto completo

$$FD(A_1, A_2, \dots, A_j) = A_k, A_m, \dots, A_o \text{ para } k, m, \dots, o \in \{1, 2, 3, \dots, j\}$$

ii) Aumento

Las partes rectoras y dependientes pueden aumentarse con los mismos atributos.

$$FD(A_1, A_2, \dots, A_j) = B_1, B_2, \dots, B_m \text{ entonces}$$

$$FD(A_1, A_2, \dots, A_j, C) = B_1, B_2, \dots, B_m, C$$

iii) La transitividad.

Las dependencias se pueden unir en cadenas.

$$\text{Si } FD(A_1, A_2, \dots, A_j) = B_1, B_2, \dots, B_k \quad Y$$

$$FD(B_1, B_2, \dots, B_k) = C_1, C_2, \dots, C_m$$

decimos que F es transitiva si :

$$FD(A_1, A_2, \dots, A_j) = C_1, C_2, \dots, C_m$$

iv) Descomposición.

Las partes dependientes pueden dividirse de manera que es posible transformar una relación en varias que tengan una misma parte rectora.

Si $FD(A_1, A_2, \dots, A_J) = B_1, B_2, \dots, B_K$

entonces $FD(A_1, A_2, \dots, A_J) = B_1, B_2, \dots, B_1$

$FD(A_1, A_2, \dots, A_J) = B_{i+1}, B_{i+2}, \dots, B_J$

La selección para que un atributo dependiente de B pertenezca a la primera o segunda relación es arbitrario.

N O R M A L I Z A C I O N

La normalización es el proceso que permite eliminar las dependencias funcionales del dominio de una relación y construir relaciones con dominios simples, que son a su vez otra relación.

Para normalizar una relación a veces es necesario eliminar el grupo repetitivo y formar con él una nueva relación.

Los conceptos de la normalización surgen como un resultado de observar las propiedades que tienen algunos conjuntos de relaciones en un medio dinámico, para construir relaciones equivalentes, más fáciles de manejar pero que respetan las relaciones iniciales.

La normalización es un proceso formado por tres diferentes niveles (Primera, Segunda y Tercera forma normal).

Las ideas básicas de estas dependencias funcionales y de normalización fueron dadas por Codd principalmente, pero el tema se abordó de una manera muy formal y rigurosa que no se describirá.

PRIMERA FORMA NORMAL.

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

Se dice que una relación está en primera forma normal si no existe una dependencia funcional en los elementos que forman a la relación. Esto permite expresar a las relaciones en tablas bidimensionales.

SEGUNDA FORMA NORMAL

Una relación está en segunda forma normal si está ya en primera forma normal y cada elemento de la relación depende por completo de la parte rectora.

TERCERA FORMA NORMAL

Una relación que está en segunda forma normal y que no existen dependencias funcionales en la parte dependiente de ella. Con ella se eliminan las dependencias transitivas de la relación.