



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

REDISEÑO DE UNA APLICACION PARA  
EL CONTROL Y SOPORTE EN SISTEMAS  
DE INFORMACION

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

EDUARDO CARBAJAL DIAZ DEL CAMPO



FACULTAD DE CIENCIAS  
UNAM

Director de Tesis:  
M. en C. ELISA VISO GUROVICH

265609

1998

DIVISION DE ESTUDIOS PROFESIONALES



FACULTAD DE CIENCIAS  
SECCION ESCOLAR

TESIS CON  
FALLA DE ORIGEN



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

M. en C. Virginia Abrín Batule  
Jefe de la División de Estudios Profesionales de la  
Facultad de Ciencias  
P r e s e n t e

Comunicamos a usted que hemos revisado el trabajo de Tesis:

Rediseño de una aplicación para el control y soporte en sistemas de  
Información.

realizado por : Carbajal Díaz del Campo Eduardo

con número de cuenta 8651977-5 , pasante de la carrera de Actuaría

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis  
Propietario

M. EN C. ELISA VISO GUROVICH

Propietario

M. EN C. JAVIER GARCIA GARCIA

Propietario

M. EN C. VIRGINIA ABRIN BATULE

Suplente

M. EN C. JOSE GUERRERO GRAJEDA

Suplente

ACT. MAURICIO AGUILAR GONZALEZ

M. EN A.F. MA. DEL PILAR ALONSO REYES  
Consejo Departamental de Matemáticas

PAGUNA

## **AGRADECIMIENTOS ...**

**\*\* ANTES QUE NADA, QUIERO AGRADECER LA DIRECCIÓN DE ESTA TESIS A LA M. EN C. ELISA VISO GUROVICH, YA QUE SIN SUS CONOCIMIENTOS, NO HUBIERA PODIDO DESARROLLAR ESTE TRABAJO \*\***

**\*\* AGRADEZCO ASÍ MISMO, LA VALIOSA APORTACIÓN DE LAS SIGUIENTES PERSONAS: M. EN C. JAVIER GARCÍA GARCÍA, M. EN C. VIRGINIA ABRIN BATULE, M. EN C. JOSÉ GUERRERO GRAJEDA Y ACT. MAURICIO AGUILAR GONZÁLEZ. SU INTERÉS EN QUE TERMINARA ESTA TESIS FUÉ MUY IMPORTANTE PARA LLEGAR A CONCLUIRLA \*\***

**\*\* ME SIENTO AGRADECIDO POR LA COLABORACIÓN DE LA GERENCIA DE SOPORTE A SISTEMAS DE INFORMACIÓN, POR EL APOYO QUE ME BRINDARON DURANTE TODA LA REALIZACIÓN DE ESTE TRABAJO \*\***

**\*\* UN AGRADECIMIENTO ESPECIAL A MI HERMANA MA. LUISA, POR SU AYUDA EN EL PRESENTE TRABAJO \*\***

## **DEDICATORIAS...**

**\*\* A MIS PADRES CARLOS Y GRACIA CON ESPECIAL CARÍÑO, POR EL APOYO QUE ME  
DIERON DURANTE TODOS MIS ESTUDIOS \*\***

**\*\* A MIS HERMANOS CARLOS, ALEJANDRO, MA. LUISA, JAVIER Y RICARDO CON  
CARÍÑO, QUIENES CON TODAS SUS VIRTUDES Y DEFECTOS ME HAN BRINDADO  
SU EXPERIENCIA Y SABIDURÍA \*\***

**\*\* ESPECIALMENTE Y CON MUCHO CARÍÑO A VANESSA, POR SU COMPRENSIÓN Y AMOR  
QUE ME HA OTORGADO DURANTE ESTOS ULTIMOS AÑOS \*\***

**\*\* POR SU ACOMPAÑAMIENTO A MIS AMIGOS HECTOR, EMMANUEL, FERNANDO, LUIS,  
CLEMENCIA, NATASHA, CARLOTA, PATY, CARLOS, JULIO, PILAR, DULCE, MARCO,  
Y A TODOS AQUELLOS QUE POR AMNESIA SE ME ESCAPAN \*\***



<b>CAPÍTULO III.- ANÁLISIS Y DISEÑO</b>	<b>48</b>
<b>III.1 Introducción y Metodología</b>	<b>49</b>
<b>III.2 Análisis de los Procedimientos</b>	<b>52</b>
<b>III.3 Requerimientos del Sistema</b>	<b>58</b>
<b>III.4 Bases y Tareas Administrativas del Sistema</b>	<b>59</b>
<b>III.5 Diseño del Sistema</b>	<b>62</b>
III.5.a Listas de Archivos y Campos	62
III.5.b Conexión de Archivos	67
III.5.c Catálogo de Causas	70
III.5.d Diagramas de Flujo de Datos	73
 <b>CONCLUSIONES</b>	 <b>85</b>
 <b>ÍNDICE GRÁFICO</b>	 <b>89</b>
 <b>BIBLIOGRAFÍA</b>	 <b>91</b>

# I N T R O D U C C I Ó N

En el ciclo diario de los sistemas de información para la producción de una compañía, se tiene una diversidad de problemas relacionados con las aplicaciones desarrolladas para el funcionamiento y toma de decisiones de la empresa en sus propias instalaciones. Estos problemas, basados en el tiempo de respuesta que se le da al usuario externo (cliente) y/o usuario interno (usuario), pueden causar reducciones en el desempeño laboral para la empresa, y éstas pueden ser en su mayoría muy costosas.

Con el fin de proporcionar al usuario un mejor servicio, se crean áreas dedicadas al soporte de sistemas de información ya liberados a producción por las áreas de desarrollo; las cuales tienen como propósito el corregir todo tipo de anomalía y verificar el buen funcionamiento de cada aplicación dentro de la compañía. Además de las fallas en los sistemas, también se tienen fallas en el equipo central que se originan en las comunicaciones. Sin embargo, dado que estos problemas son más costosos y difíciles de corregir a corto plazo, y a su vez se tienen que medir o controlar de otra forma, se crean otras áreas, que den el soporte respectivo, y así controlar el flujo de la producción. Algunas de las áreas involucradas con estos últimos problemas son Administración de Recursos, Ingeniería de Sistemas, Teleproceso, entre otras. Este trabajo está enfocado a aquellos problemas que tienen que ver con el soporte a sistemas de información, con el objetivo principal de controlar, de alguna manera, este soporte o semi-desarrollo de los sistemas.

La problemática, tema de este trabajo, está ubicada en la compañía de seguros Grupo Nacional Provincial, la cual desempeña su labor desde 1902 con el nombre de La Nacional de Seguros

sobre la Vida S.A.. Posteriormente, en 1973 se formaliza el convenio de constitución de Grupo Nacional Provincial. Éste y otros acontecimientos, como el que en 1992 inauguran en Plaza GNP una de las centrales de cómputo más modernas de México, han hecho que la compañía tenga fuertes adelantos en el desarrollo de sistemas de información y un papel preponderante en la vida de los seguros en México. Después de conocer y desempeñar mi labor para el Centro de Cómputo de la compañía, me avoqué a la tarea de analizar algunas de las fallas que se tienen al darle soporte a los sistemas de información.

Para comenzar con el tema, que tenemos bien ubicado dentro del control y soporte de cada aplicación desarrollada en sistemas de información dentro de la compañía, y para atender los problemas que se le presentan al usuario, tomaré cuatro áreas muy importantes para lograr mi objetivo. Éstas son:

**Control de la Producción.** Encargada de monitorear el flujo diario de la producción de estos sistemas.

**Mesa de Ayuda (Help Desk).** La cual se dedica a registrar los problemas que se tienen con las transacciones en línea y por los procesos por lote de cada aplicación. Ésta es una atención a primer nivel, con trato directo con el cliente.

**Administración de Problemas.** Esta área se dedica exclusivamente a llevar el control de todos los problemas que se presentan con los sistemas desarrollados. Aquí se toman los reportes del usuario y se pasan al área respectiva, para que le dé solución. Ésta lleva un orden y obtiene estadísticas de todos aquellos problemas que estén relacionados con los sistemas de información, para una mejor medición y optimización de la producción.

**Soporte a la Producción.** Su objetivo es el control y soporte de la producción y de las aplicaciones desarrolladas en la compañía. Es una atención a segundo nivel, cuyo trato es con el usuario interno.

También existe un tercer nivel de soporte, el cual está ubicado con el experto que desarrolló el sistema en cuestión, o bien en el área correspondiente al desarrollo de este sistema, pero no se toma esta área como parte principal del proyecto, ya que sólo utilizará el producto como consulta. Por ello ésta no será mencionada con la misma frecuencia que las otras áreas ya citadas.

Falta mencionar algunos de los problemas más recurrentes que se tienen, como tiempos de respuesta en línea, tiempos de ejecución en los procesos por lote, errores de códigos productivos existentes, cancelaciones por datos mal capturados, faltantes y/o pérdida de impresión, descuadres de información, etc... Por éstas y otras razones habrá que controlar el soporte ya establecido anteriormente. Para lograr este control nace la idea de crear una base de datos, que contenga todo el histórico de problemas ocurridos; y a su vez un sistema que se dedique a administrar y mejorar el soporte que se tiene.

De una serie de problemas con los sistemas que cada una de las cuatro áreas mencionadas tienen actualmente para el control y soporte de la producción, surge el objetivo principal de este trabajo: rediseñar un solo sistema para la captura de los problemas anteriormente mencionados y la manipulación adecuada y productiva de los datos así obtenidos.

El tener distintos sistemas puede causar un gran problema al manejar en conjunto las cifras o estadísticas de las anomalías ocurridas. Por ejemplo, que el número de reportes que se tienen en el área de Administración de Problemas sea distinto al número de reportes que se tienen en Soporte a la Producción, causan un descontrol a la gente que está dedicada a solucionar esas fallas, y ocasiona no poder arreglar definitivamente la aplicación o sistema desarrollado para su flujo perfecto dentro de la compañía. Un sistema único como el que estoy proponiendo, es útil no sólo para almacenar este tipo de datos, sino también para sacar estadísticas de tiempos y cancelaciones, o para llevar una referencia de problemas que ya se han suscitado anteriormente; asimismo se pueden sacar estadísticas del número de reportes que se han realizado por cada sistema de información desarrollado en la compañía. También, un aspecto muy relevante: podremos medir, optimizar y centralizar toda la información, para tener veracidad de la misma.

# **CAPÍTULO I**

## **REQUERIMIENTOS DEL USUARIO**

### **I.1 SITUACIÓN ACTUAL**

I.1.a Atención de problemas reportados a Mesa de Ayuda

I.1.b Atención de problemas reportados por Control de la Producción

### **I.2 RECEPCIÓN DE SOLICITUDES ADICIONALES DEL USUARIO**

I.2.a Para Mesa de Ayuda

I.2.b Para Control de la Producción

I.2.c Para Soporte a la Producción

I.2.d Para Administración de Problemas

### **3.- ANÁLISIS Y SÍNTESIS DE LAS SOLICITUDES.**

## **I.1 SITUACIÓN ACTUAL**

En los sistemas de información existen dos tipos de procesos: en línea y por lote. Los problemas en línea son aquellos que tiene el usuario al manipular los datos de cualquier aplicación desarrollada durante su ejecución, y por lote, los que se presentan cuando el flujo diario de la producción manipula estos datos automáticamente, alejados en el tiempo del usuario.

La situación actual que presenta la compañía para resolver estos problemas, está sustentada principalmente en tres niveles básicos, los cuales están controlados por el área de Administración de Problemas a partir de un sistema.

- a) Primer Nivel. - Está ubicado en las áreas de Mesa de Ayuda y Control de la Producción. Aquí se reciben los problemas en línea y por lote respectivamente, se intentan resolver de manera inmediata, y en caso de no poder hacerlo se canalizan al siguiente nivel por medio de Administración de Problemas, manteniendo informado al usuario sobre el estado de su consulta.

Se tiene como meta para estas dos áreas, el resolver al menos el 80% de los casos que se les reportan.

No ofrece soporte en aplicaciones que no hayan sido liberadas a producción, así como problemas que impliquen actualizar el ambiente productivo.

- b) Segundo Nivel. Soporte a Producción. - Este nivel cuenta con un perfil más técnico que el de primer nivel. El segundo nivel puede aplicar una solución temporal o resolver el problema de raíz. Si no hay solución, se canaliza al experto o tercer nivel.

Soporte a Producción tiene la meta de resolver el 95% de los casos que recibe. Esto significa que de cada 100 problemas reportados a Administración de Problemas, 80 son solucionados a primer nivel, 19 son resueltos por segundo nivel, y uno es atendido por el tercer nivel.

- c) Tercer Nivel. - Este nivel cuenta con personal especializado en las áreas de desarrollo de cada sistema. Éste es el último nivel de solución y el experto tiene la responsabilidad de hacer un análisis más profundo del problema, para poder aplicar una solución definitiva y resolverlo de raíz.

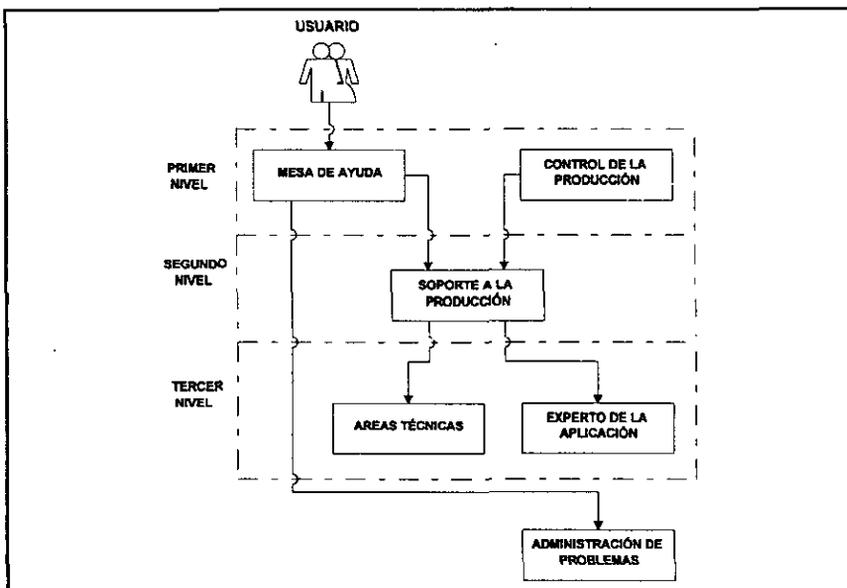


Figura 1.1 Niveles de atención para problemas recibidos.

Como ya se mencionó anteriormente, Administración de Problemas está involucrada en todo el proceso de solución y su objetivo principal es asegurar que el 100% de los casos sean atendidos y resueltos. Entre sus principales funciones están:

- Asegurarse de que los problemas sean registrados en el Sistema para Administración de Problemas (SAP).
- Dar seguimiento a consultas registrados en el SAP, hasta obtener la solución definitiva.
- Mantener informado al usuario sobre el estado de los problemas que reporta.
- Obtener estadísticas del SAP, que permitan identificar áreas de oportunidad.
- Coordinar grupos de trabajo para la solución de cancelaciones.

#### **1.1.a Atención de problemas reportados a Mesa de Ayuda.**

Una de las entradas para problemas es Mesa de Ayuda. El flujo correspondiente se presenta en la figura 1.2.

1. El usuario reporta el problema a Mesa de Ayuda vía telefónica.
2. Mesa de Ayuda registra la llamada, le asigna número de folio en el SAP (que se identifica con una H), hace un diagnóstico del problema y trata de resolverlo (solución a primer nivel).
3. En caso de que no se haya podido solucionar la falla, transfiere la llamada a Administración de Problemas, quien a su vez lo canaliza a segundo nivel, proporcionando al responsable el número de folio respectivo.
4. Una vez resuelto el problema, el responsable notificará de ello a Administración de Problemas vía telefónica o por correo electrónico, proporcionando número de folio del mismo, causa y solución aplicada.

5. Si por algún motivo no pudiera aplicar la solución inmediatamente, deberá dar una fecha compromiso para la solución. Así mismo, indicará si la solución aplicada es definitiva o temporal y por lo tanto es necesario recanalizarlo a un tercer nivel.
6. Una vez que Administración de Problemas reciba la respuesta por segundo o tercer nivel, ya sea telefónicamente o por correo electrónico, cerrará el folio, y notificará a Mesa de Ayuda, quien a su vez se encargará de comunicárselo al usuario y solicitar que verifique si ha sido resuelto. Ningún problema puede cerrarse sin el visto bueno del usuario.

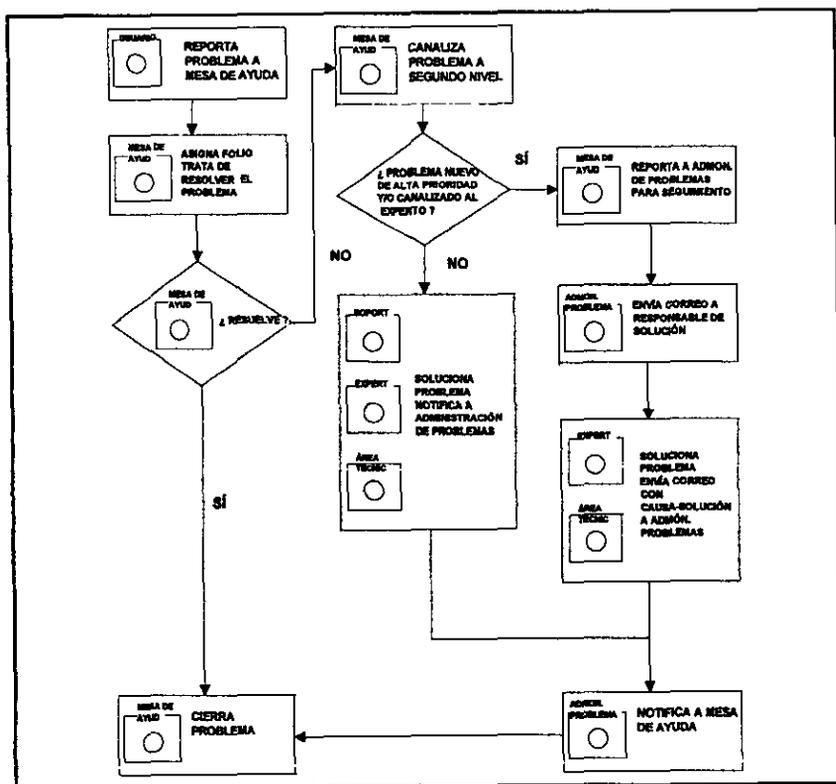


Figura 1.2 Atención de problemas reportados a Mesa de Ayuda

### **I.1.b Atención de problemas reportados por Control de la Producción.**

Los problemas relativos a la producción por lote son reportados por la misma área de Control de la Producción.

1. Control de la Producción detecta, registra el problema en el SAP y trata de resolverlo.
2. Si Control de la Producción no puede resolverlo, lo reporta a Soporte a Producción.
3. Soporte a Producción intenta resolverlo, aunque sólo aplique solución temporal. En caso de no poder hacerlo se comunica con el experto del negocio para solicitar ayuda. El experto del negocio puede orientar al operador por teléfono o, dependiendo de la causa, trasladarse a Plaza GNP para atender el problema personalmente.
4. Diariamente Control de la Producción entrega a Soporte todos los reportes que se presentaron durante la producción por lote y que fueron solucionados.
5. Soporte a Producción clasifica los folios como: cerrados, casos que atenderá él mismo durante el resto del día y folios que deben canalizarse a tercer nivel.
6. Soporte a Producción entrega los folios a Administración de Problemas a primera hora.
7. Administración de Problemas canaliza los reportes pendientes a tercer nivel vía telefónica o por correo electrónico con los siguientes datos: número de folio, descripción del problema y solución aplicada por Soporte a Producción, si se tuvo ésta.
8. El experto del negocio lo atiende. En caso de resolverlo envía un correo a Administración de Problemas con los siguientes datos: número de folio, causa y solución aplicada. En caso de no poder dar una solución inmediata, el experto deberá enviar el correo dando una fecha compromiso de solución.
9. Una vez que Administración de Problemas recibe la notificación de solución por parte del experto, cierra el folio y registra la fecha y hora de este aviso, para posteriormente poder calcular tiempo total de solución y tiempo de respuesta por parte del experto.

Un folio nunca podrá cerrarse, hasta que no se reciba la notificación por parte del responsable de que este caso fue resuelto. A continuación se muestra en la figura 1.3 el diagrama para la atención y solución de estos.

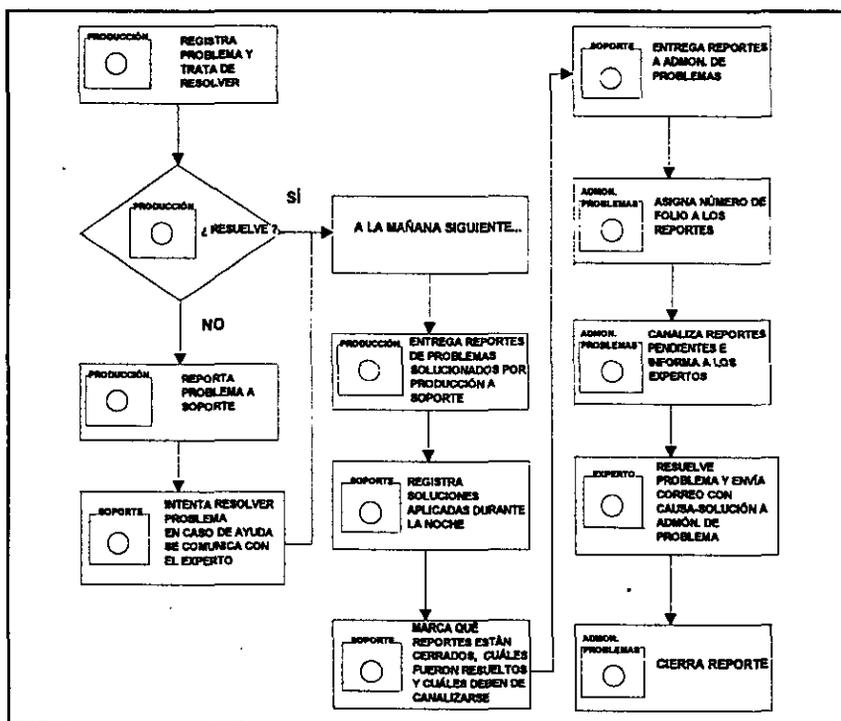


Figura 1.3 Atención de problemas reportados por Producción.

Los problemas reportados por Control de la Producción tienen el formato P-folio y los reportados por Mesa de Ayuda H-folio, para evitar confusiones y conocer el origen del reporte.

## **I.2 RECEPCIÓN DE SOLICITUDES ADICIONALES DEL USUARIO**

A fin de conocer las deficiencias que tiene el Sistema para Administración de Problemas (SAP) actualmente, y siguiendo la metodología de Ingeniería de Software para la etapa de especificación de requerimientos, se realizaron una serie de entrevistas con el personal de las áreas de soporte. A través de éstas, ellos dieron sus puntos de vista y sugerencias para mejorar el sistema.

### **I.2.a Para Mesa de Ayuda.**

- 1.- Consideran que les hacen falta campos para asentar los datos del usuario y así poder identificarlo fácilmente. Datos como: área en la que se encuentra ubicado, subdirección a la que pertenece, etcétera.
- 2.- Hace falta un catálogo de causas que han originado los problemas. De tal forma, que se obtiene una relación concreta de todas las consecuencias que originaron estas cancelaciones o fallas, siendo más sencilla su solución y permitiendo al operador consultar posteriormente distintas causas con una sencilla referencia.
- 3.- Contar con una sola base de datos para los reportes de primero, segundo y tercer nivel, ya que en la actualidad hay una base de datos para el primer nivel en un servidor y otra para segundo y tercer nivel en otro servidor. Esto ocasiona confusiones, porque la información que se presenta está desfasada con respecto al consecutivo de folios para los distintos niveles.

### **I.2.b Para Control de la Producción.**

- 1.- Todos los problemas por lote se resuelven. Sin embargo, hay fallas de control, porque no se levantan folios de algunos problemas, muchas veces por descuido del operador.
- 2.- No existe una consulta de cada proceso por lote que ha tenido una o varias cancelaciones durante cierto rango de fechas, para en un momento determinado sacar una referencia de una cancelación similar y resolver de manera temporal el problema.

### **I.2.c Para Soporte a la Producción.**

- 1.- Se vuelve a repetir el mismo problema que manifestó el personal de Control de la Producción. Hay desfase entre folios levantados y cancelaciones por lote durante el día, creándose desorden y falta de control para todas las áreas de soporte.
- 2.- Las peticiones de consulta por proceso y/o aplicación por rango de fechas y tiempos, son las mismas que presentó Control de la Producción.
- 3.- El especialista o experto de la aplicación de segundo y tercer nivel, carece de información directa. Por lo tanto cuando quiere saber cuántos problemas y qué tiempo se ha llevado resolverlos, necesita acudir a Administración de Problemas para que le proporcione esta información. Esto ocasiona un retraso en el reporte de actividades.
- 4.- Un problema recientemente surgido se ha presentado desde que las peticiones especiales se registran por folio, ya que no hay manera de distinguir entre un mantenimiento eventual (solución a un problema en segundo o tercer nivel), y el mantenimiento correctivo o petición del usuario.

### **1.2.d Para Administración de Problemas.**

- 1.- La inquietud que presentó el personal de esta área, se refirió a la diferencia que hay entre los reportes registrados por Mesa de Ayuda y los registrados en segundo y tercer nivel. Esto se origina por la existencia de dos bases de datos distintas en diferentes servidores.
  
- 2.- Son para ellos muy importantes los requerimientos de las distintas áreas, ya que podrán realizar sus mantenimientos, altas, bajas y consultas con mayor eficiencia y rapidez, para obtener todos los datos de la persona indicada, tanto el que reportó, como el que solucionó, dando un mayor control de los folios que se tienen cerrados o abiertos.

### **I.3 ANÁLISIS Y SÍNTESIS DE LAS SOLICITUDES.**

En el punto anterior se describieron detalladamente las necesidades de cada área de soporte. Partiendo de todos estos requerimientos se presenta un resumen de lo que se considera que son las deficiencias, que van a quedar resueltas al rediseñar el Sistema para Administración de Problemas y control de sistemas de información.

Entre lo que se pudo observar, la mayor parte de solicitudes son de consulta, como: catálogo de causas, histórico de soluciones y relaciones de tiempo-solución.

Otra falla muy importante es el factor humano, ya que el operador se preocupa más por el problema en sí, que por el aspecto administrativo, por lo cual se le olvida registrar el problema. Esto sólo ocurre en la producción por lote; en línea no ocurre porque se registra al mismo tiempo que lo reporta el usuario.

Para la administración de ciertas peticiones especiales hacen falta campos de captura, que distinguan quién reporta la falla o de dónde surge el requerimiento, a la vez que determinen si son requerimientos o problemas de la aplicación que esté manejando el usuario.

Se cree que la mayor desventaja del sistema es que tenga diferentes bases de datos en distintos servidores, porque no hay una relación exacta de los reportes que se hacen en cualquiera de los niveles de solución. Si contáramos con una sola base de datos, solamente habría un consecutivo de folios y no se desfazaría la información.

Una vez obtenidos los requerimientos del usuario, pasaremos a definir la metodología que utilizaremos dentro de este proyecto, en la cual describiremos algunas de las herramientas a utilizar, la forma de diseño a la cual estaremos avocados, y una especie de introducción general para definir lo que es un sistema de información.

## **CAPÍTULO II**

# **A N Á L I S I S D E R E Q U I S I T O S D E L S I S T E M A**

### **II.1 NOMENCLATURA Y HERRAMIENTAS**

II.1.a Tipo de Herramientas

### **II.2 MODELOS Y TÉCNICAS PARA EL DISEÑO**

II.2.a Diagramas de Flujo de Datos

II.2.b Diagramas Entidad-Relación

II.2.c Balance y Diagramas de Flujo Estructurado

### **II.3 DISEÑO ORIENTADO AL FLUJO DE DATOS**

## II.1 NOMENCLATURA Y HERRAMIENTAS

Al hablar en este proyecto de **rediseño de sistemas** y **sistemas de información**, términos muy usados en varios contextos y con diversos significados, vale la pena aclarar ciertos conceptos básicos de sistemas, y comenzar describiendo explícitamente - corriendo el riesgo de aburrir al lector - el significado, sentido e intención de la terminología utilizada en el desarrollo del trabajo.

La palabra **sistema** es un término sobreutilizado técnicamente. Existen sistemas educativos, de fabricación, bancarios, y en sí la palabra nos dice poco. El diccionario Webster la define así:

*1. un conjunto u ordenación de cosas relacionadas de tal manera que forman una unidad o un todo orgánico; 2. un conjunto de hechos, principios, reglas, etc... clasificados y ordenados de tal manera que muestran un plan lógico uniendo las diferentes partes; 3. un método o plan de clasificación u ordenación; 4. una forma establecida de hacer algo; un método; un procedimiento...*

Tomando como referencia las definiciones anteriores del diccionario Webster, definimos un **sistema basado en computadora** como:

" un conjunto de elementos organizados para llevar a cabo algún método o procedimiento mediante el procesamiento de información."

En la siguiente figura 2.1[\*1] se muestran los elementos principales de un sistema de información basado en computadora.

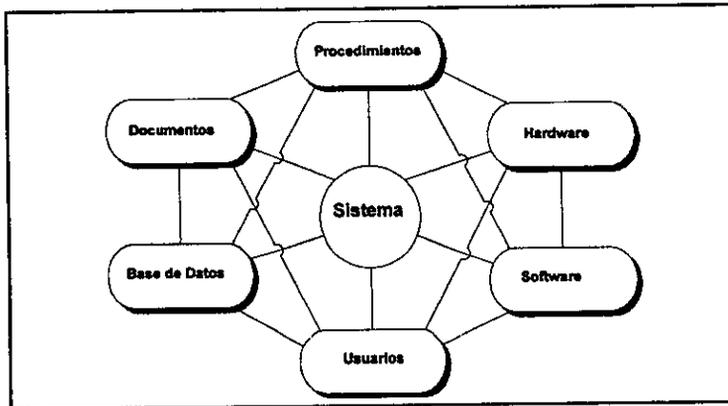


Figura 2.1 Elementos del sistema.

De la figura anterior describimos al **software**, como el conjunto de programas de computadora, las estructuras de datos y la documentación asociada, que sirve para obtener el método lógico o procedimiento requerido. **Hardware**, se refiere a todos los dispositivos electrónicos que proporcionan la capacidad de manejar procedimientos para computadora, por ejemplo: CPU, memoria, etcétera.

Todos los manuales, las impresiones y la información descriptiva de la operación del sistema se llama **documentación**; y los **procedimientos**, son los pasos que definen el uso específico de cada elemento del sistema o el contexto del mismo.

[\*1] Roger S. Pressman, Ingeniería del Software, 3ª edición, Mc Graw Hill, Octubre 1995.

A la agrupación organizada de información a la que se accede con software y que es parte integral del funcionamiento de un sistema, se le llama **base de datos**.

A las personas que trabajan creando, procesando, administrando o intercambiando información, se les puede llamar trabajadores de la información; sin embargo, este término es muy abstracto y nunca se emplea, por lo que el nombre más común que se les ha dado a estas personas es **usuarios**. De manera más concreta se puede definir a los usuarios del sistema como todas aquellas personas que están involucradas en el sistema, ya sea al operarlo, al administrarlo o al obtener información y beneficios adicionales de él. En la siguiente tabla (figura 2.2) se pueden ver los tipos de usuarios que existen.

TIPO DE USUARIO	CARACTERÍSTICAS
USUARIO FINAL DIRECTO	Operan el sistema en forma directa a través del equipo.
USUARIO FINAL INDIRECTO	Emplea los reportes y la información generada por el sistema pero no opera el equipo.
ADMINISTRADORES	Supervisan la inversión en el desarrollo del sistema. Tienen la responsabilidad de controlar las actividades del sistema.
DIRECTIVOS	Evalúan los riesgos que son originados por fallas en los sistemas de información, incorporando planes estratégicos.

**Figura 2.2** Esquema con tipos de usuarios.

Existe un enfoque para evitar el mal funcionamiento del software, mediante métodos completos para todas las fases de su desarrollo, mejores herramientas para automatizar estos métodos, excelentes técnicas para garantizar la calidad del producto y una filosofía para la coordinación y control del mismo. Esta disciplina se llama **ingeniería del software**.

Los tres elementos claves para la ingeniería del software son: métodos, herramientas y procesos. Los **métodos** indican cómo construir técnicamente el producto, incluyendo: planificación, análisis de los requisitos, diseño de estructuras de datos, codificación, prueba y mantenimiento.

Las **herramientas** suministran un soporte automático para los métodos, como es la *ingeniería del software asistida por computadora* (en inglés CASE). Los **procedimientos** son la unión de los métodos con las herramientas, los cuales facilitan un desarrollo oportuno del software de computadora y definen la secuencia en la que se aplican los métodos.

Esta ingeniería de software tiene un ciclo de vida figura 2.3[\*2], el cual lleva un enfoque secuencial del desarrollo del sistema a través del análisis, diseño, codificación, prueba y mantenimiento. El proceso de recopilación de los requisitos, tanto del sistema como del software se documentan y se revisan con el usuario. A esto se le llama **análisis de requisitos**. El **diseño** es un proceso multipaso que traduce los requisitos en una representación que pueda ser establecida de tal manera que obtenga la calidad requerida antes de la **codificación**, la cual realiza la tarea mecánica del diseño. Una vez que se ha generado el código, comienza la **prueba** del programa, asegurando que la entrada definida produce los resultados que realmente se requieren. Por último se menciona el **mantenimiento** como la parte de cambios que sufrirá el producto después de que se entregue al usuario, aplicando cada uno de los pasos del ciclo de vida clásico para la ingeniería del software.

Cabe mencionar que en la fase de desarrollo el **diseño del software** transforma los requerimientos a un conjunto de representaciones que describen la estructura de los datos, la arquitectura, el procedimiento algorítmico y sus características de interface. Estas representaciones necesitan de una codificación para ser traducidas a un lenguaje artificial, dando como resultado unas instrucciones ejecutables por la computadora.

[\*2] Roger S. Pressman, Ingeniería del Software, 3° edición, Mc Graw Hill, Octubre 1995.

En la fase de mantenimiento o verificación se encuentran tres tipos de cambios: **corrección**, el cual modifica el software para corregir los defectos; **adaptación**, que consiste en modificar el software para acomodarlo a los cambios de su entorno externo; y **mejora**, amplía el software más allá de sus requisitos originales.

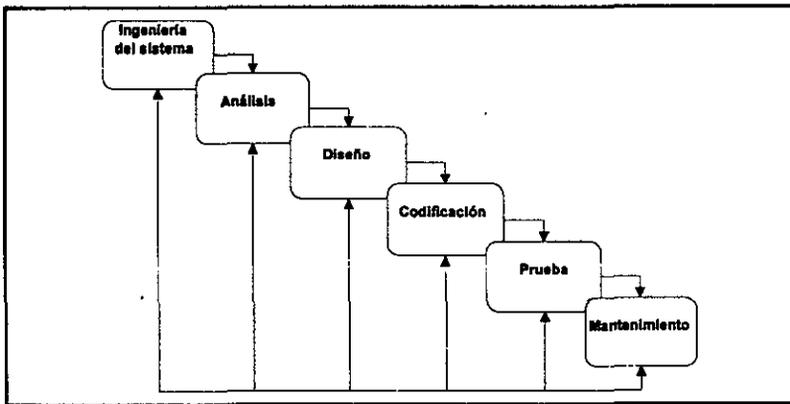


Figura 2.3 Ciclo de vida clásico.

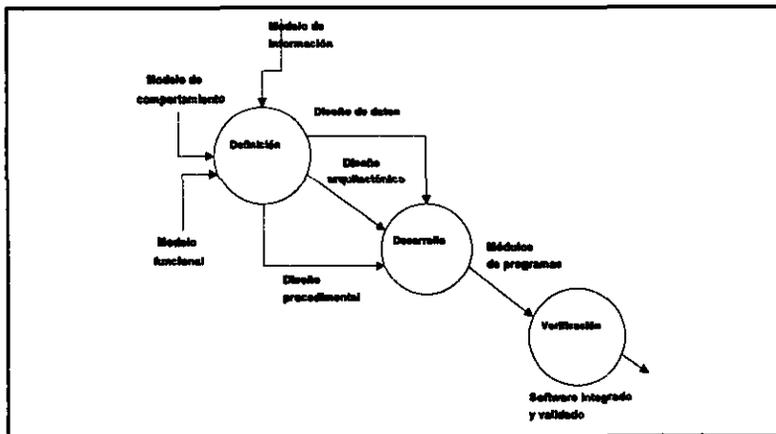


Figura 2.4 Ingeniería de software.

Como podemos ver en la figura 2.4, la ingeniería de software está dividida en tres fases: de definición, de desarrollo y de verificación, tal como lo muestra la siguiente figura.

En este trabajo nos ocuparemos del análisis y diseño del sistema, en base a los requerimientos de los usuarios, de tal manera se checarán los diagramas del sistema actual y compararemos con la propuesta del nuevo sistema. Por tanto conviene definir bien todos los modelos de información y funcionales; y a su vez el desarrollo del diseño de datos, para concluir con la verificación, comparando el sistema actual con los diagramas del sistema propuesto.

### II.1.a Tipo de Herramientas

Entre los diferentes tipos de diseño se encuentran el **diseño de datos**, que transforma el modelo del campo de la información, creado durante el análisis, en las estructuras de datos que se van a requerir para implementar el software. El **diseño arquitectónico** es otro tipo de diseño que define las relaciones entre los principales elementos estructurales del programa. Otro tipo es el **diseño procedimental**, que transforma los elementos estructurales en una descripción procedimental del software.

La **estructura de datos** es una representación de la relación lógica existente entre los elementos individuales de datos; es tan importante como la estructura del programa en la representación de la arquitectura del software y del diseño procedimental.

Sin duda las herramientas gráficas como los diagramas de flujo y de entidad-relación, son una excelente forma para describir los detalles del diseño procedimental. El **diagrama de flujo de datos** (DFD) es una técnica en la que se representa el flujo de la información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida.

Es también conocido como diagrama de burbujas. En la figura 2.5 se ilustra la notación básica que se usa para crear un DFD.

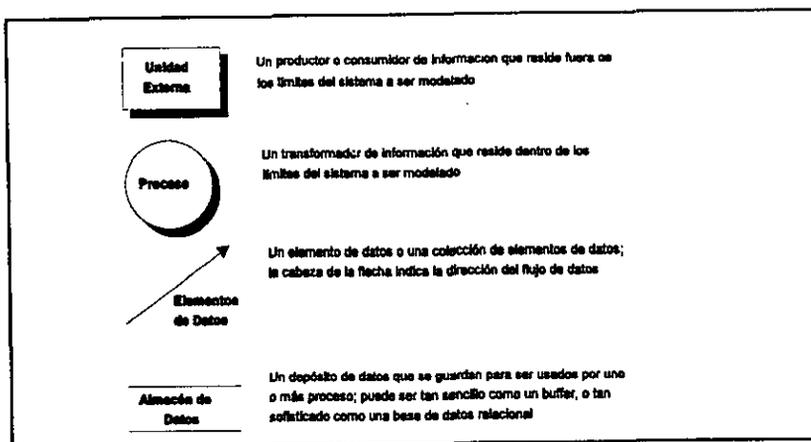


Figura 2.5 Notación básica de un diagrama de flujo de datos.

Otra técnica o herramienta es el **diagrama entidad-relación**. Éste se utiliza como base en modelos de datos para el diseño de aplicaciones; es un componente para el modelo entidad/relación, donde se puede identificar: asociación, cardinalidad, atributos, ocurrencia, identificador, tipos de datos, y subtipos/supertipos.

Hoy en día una de las herramientas más importante son las **CASE**, ingeniería de software asistida por computadora, la que proporciona al ingeniero la capacidad de automatizar las actividades manuales y mejorar el enfoque de trabajo. La expresión se ha convertido en un término bastante conocido por los profesionales dedicados al desarrollo y al mantenimiento de software.

Para este proyecto decidimos utilizar diagramas de flujo de datos, por la facilidad y experiencia que se tiene al desarrollar otro tipo de sistemas actualmente optimizados. Esta herramienta va a facilitar las comparaciones entre el sistema actual y el sistema propuesto.

Cabe aclarar que dentro de la empresa se utilizan distintas herramientas, de las cuales se propone utilizar dentro del desarrollo lo siguiente:

- a) La bitácora de JOBTRAC, con la cual podremos verificar el histórico de cancelaciones por lote del diario.
  
- b) Para manejar copias, respaldos y reportes de la información, algunas utilerías de IBM, y de CA, como son entre otras: IEBGENER, IEBCOPY, CA-SORT, etcétera.
  
- c) Para hacer cálculos y manejo de datos, programaremos en NATURAL, PL1 y COBOL 370.
  
- d) Para el almacenamiento de datos manejaremos archivos secuenciales en discos y respaldos en cintas, a su vez ADABAS como manejador de bases de datos.
  
- e) Para la comunicación del Equipo Central con Computadoras Personales, una interface DTF y *Netware File Transfer*, los cuales nos proporcionan la transferencia de datos tal y como la dejan los procesos por lote.
  
- f) Por último como administrador principal se utilizara el sistema actual en PARADOX para conservar el sistema en su desarrollo original.

Es importante mencionar que el modelo en cascada no es el único para la ingeniería del software, existen otros modelos que han sido desarrollados para cubrir todas las características y requisitos del sistema, tenemos por ejemplo el modelo en espiral que añade el análisis de riesgo, en el cual se da un análisis de todas las alternativas y resoluciones de cualquier riesgo que se tenga durante la planificación o determinación de los objetivos. El modelo, representado mediante la espiral de la figura 2.6, define cuatro actividades principales, representadas en los cuatro cuadrantes de la figura:

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

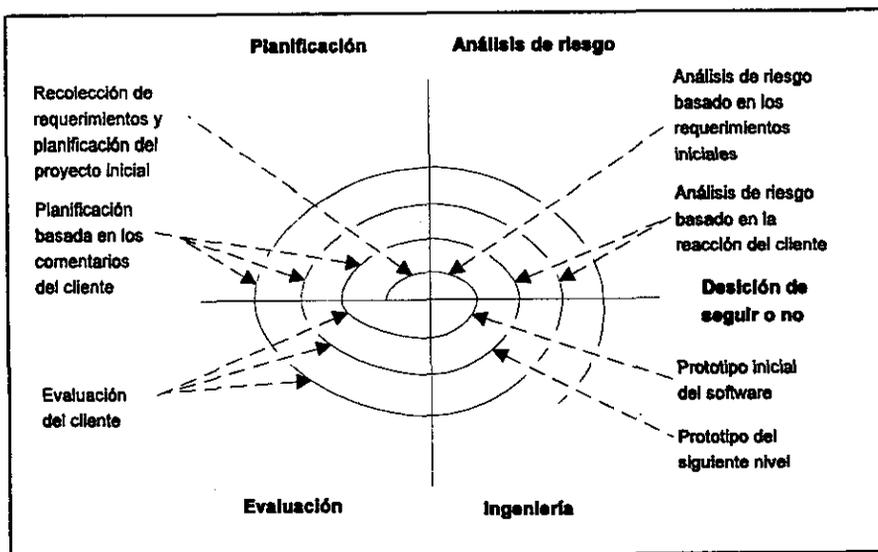


Figura 2.6 El modelo en espiral.

## **II.2 MODELOS Y TÉCNICAS PARA EL DISEÑO**

Para comenzar con el análisis, debemos definir algunos modelos del análisis estructurado, el cual determinará cómo implantaremos el diseño de la aplicación. Será el esquema teórico que usaremos para diseñar el sistema de control y soporte a sistemas de información.

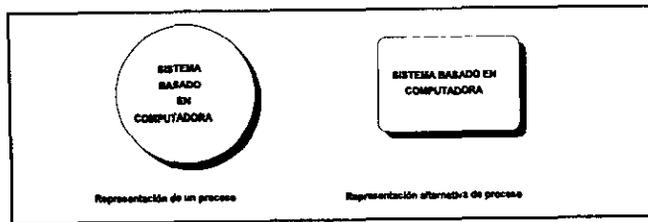
Uno de los modelos del análisis estructurado más sencillo y fácil de entender es el de diagramas de flujo de datos, uno de los métodos más extendidos en modelado conceptual de procesos. Esta técnica, conocida también como diagramas de burbuja, fue propuesta por Yourdon a mediados de los setentas.

El objetivo fundamental de esta técnica es descomponer un problema complejo en varios sencillos y manejables, facilitando la modularidad del sistema y su comunicación entre módulos, representados gráficamente. A su vez pretende separar la estructura física del sistema de la estructura lógica, y con esto dar mantenimiento más sencillo al análisis del sistema.

### **II.2.a Diagramas de Flujo de Datos**

Los Diagramas de Flujo de Datos, que denotaremos por DFD, constan de los siguientes elementos:

- 1.- **Procesos:** son aquellos lugares donde se transforma o descompone la información de un flujo de datos; muestra el lugar donde una o varias entradas se transforman en salidas. El proceso está representado gráficamente por un círculo, como se muestra en la figura 2.7, indicando el nombre del proceso dentro de aquél; algunos analistas lo representan con un rectángulo con las esquinas redondeadas.



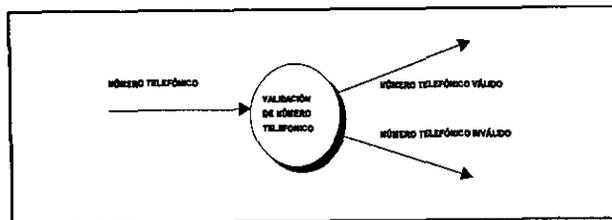
**Figura 2.7** Ejemplos de procesos.

- 2.- Flujos: son los caminos por donde fluye la información entre los diferentes elementos del sistema, es decir, describe el movimiento de la información desde una parte del sistema a otra parte distinta. Éstos se representan a través de un arco orientado hacia un proceso o un almacén de datos como se puede observar en la figura 2.9.



**Figura 2.8** Representación de un flujo.

El flujo representa datos, caracteres, mensajes y otra clase de información que las computadoras puedan procesar.



**Figura 2.9** Representación de un DFD con sus respectivos flujos.

- 3.- Almacenes: son aquellos lugares donde se guarda de forma estática la información dentro del sistema. Se representan gráficamente con dos líneas paralelas, como se ve en la figura 2.10.

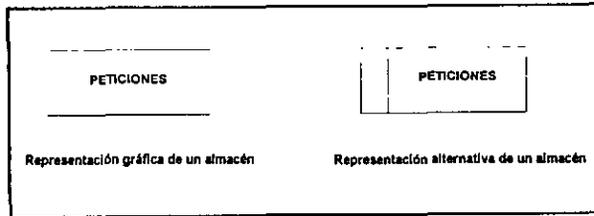


Figura 2.10 Ejemplos de almacenes.

Para el analista de sistemas, los almacenes se refieren a archivos o bases de datos (IDMS, DB2, ADABAS, o cualquier otro manejador de bases de datos).

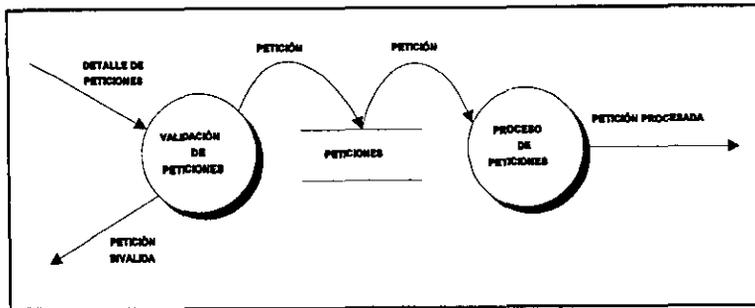
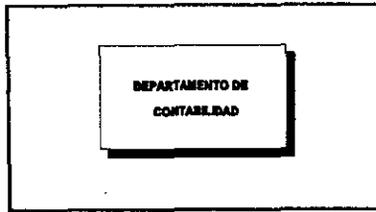


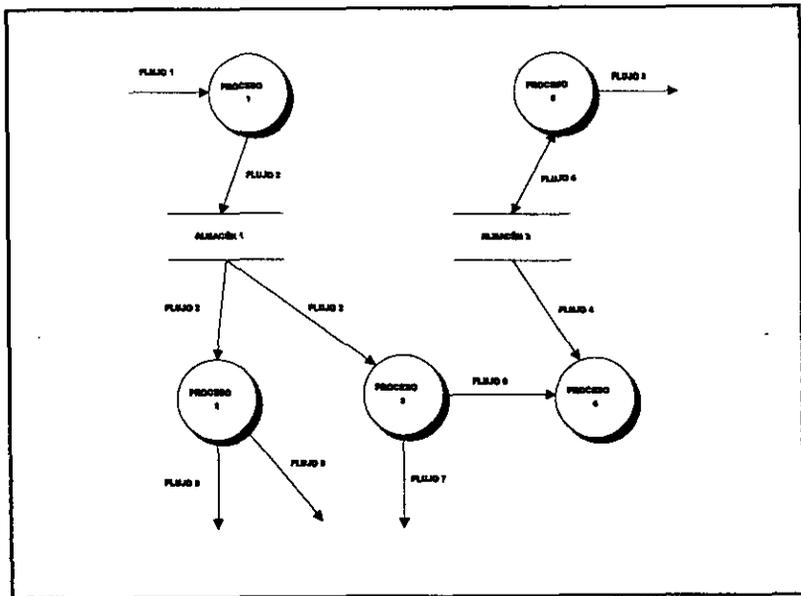
Figura 2.11 Representación de un almacén implementado.

- 4.- Entidades Externas: son aquellas organizaciones, personas, o entidades en general que no pertenecen al sistema de información, pero tienen relación con éste. De ellas es de donde proceden los datos que necesita el sistema o hacia donde van los generados por él. Sólo aparecen en el diagrama de contexto inicial. Se representan mediante un rectángulo con el nombre de la entidad en el interior (figura 2.12).



**Figura 2.12** Representación de una entidad externa.

El proceso de análisis es sencillo. Consiste en un refinamiento a través de la construcción del diagrama de contexto o bien exponer el DFD compuesto por un solo proceso que represente todo el sistema, las entidades externas que participan en él y los flujos de entrada y salida. Según la notación que dimos anteriormente (de Yourdon) se puede observar la siguiente figura 2.13 :



**Figura 2.13** Representación completa de un Diagrama de Flujo de Datos.

Cabe señalar que existe una serie de características generales a tener en cuenta a la hora de diseñar un DFD, como son:

- a) Evitar la utilización de estructuras ilegales.
- Un flujo de datos que se subdivide en el diagrama, es decir, para que un flujo de datos se pueda dividir, es necesario un proceso que realice esta operación.
  - Señales de control. No pueden aparecer flujos de datos cuyo único contenido sean banderas (datos booleanos, pasos de control por llamadas a procedimientos).
  - Ciclos. Si un flujo entra y sale en el mismo proceso, significa que es algo que sólo le interesa a dicho proceso, por lo tanto debe quedar oculto.
  - Flujo de datos entre dos almacenes. Para poder hacer una lectura en un almacén y una escritura en otro, es necesaria la existencia de un proceso que realice dicha función.
  - Procesos aislados. No tiene sentido la existencia de un proceso que no emita o reciba flujos de datos sin conexión alguna con el resto del sistema.
  - Flujos entre entidades externas. El paso de información que pueda existir entre dos entidades externas, es algo que no interesa al sistema que se está diseñando.
- b) Procurar mantener la propiedad de la conservación de los datos.
- Los datos no se crean, sólo se transforman.
  - Balanceo entre niveles. Todas las entradas y salidas del diagrama  $n + 1$  deben estar en el diagrama  $n$ , excepto los mensajes de error.
  - Dividir el sistema de forma natural. Sólo debemos descomponer si es necesario.
  - Establecer conexiones simples. Entre menos flujo de datos mejor.
  - Limitar el número de procesos dentro de un nivel de descomposición (a lo mucho 7)
  - Nombrar correctamente todos los objetos que surjan en los diagramas.
    - . Flujos : con un sustantivo más un adjetivo
    - . Procesos: con un verbo más un sustantivo
    - . Almacenes: con un solo sustantivo.

## II.2.b Diagramas Entidad-Relación

El diagrama entidad relación, también conocido como DER, es un modelo de trabajo de alto nivel de abstracción, que describe el almacenamiento de los datos de un sistema. Esto es muy diferente del DFD, en el cual las funciones son la parte importante del sistema. El DER es estático, mientras que el DFD representa la parte dinámica.

Estamos interesados en el modelo de datos del sistema (DER), porque la estructura de datos y sus relaciones, aisladas de los procesos en los que participan, son menos complejas y fáciles de examinar, que si intentáramos revisarlas inmersas en las funciones que las afecten. Adicionalmente los usuarios suelen estar familiarizados con los datos y se preguntan : ¿ qué datos necesitamos para nuestro negocio ? o, ¿ cómo están relacionados los datos con otros datos?, ¿ quién es el propietario de los datos ? Algunas de estas preguntas acerca de datos son quizás responsabilidad de un grupo de administradores de datos, quienes manejan y controlan esencialmente la información del negocio. El DER es una herramienta para realizar este tipo de organización de la información.

Existe otro grupo de administradores con un nombre similar, Administrador de Bases de Datos (DBA`s - conocido por sus siglas en inglés), cuya función es asegurar que las bases de datos computarizadas sean organizadas y controladas efectivamente. Tiene la responsabilidad de tomar un modelo esencial y traducirlo a un diseño físico eficiente de una base de datos para IDMS, ADABAS, u otro manejador de base de datos. El diagrama entidad-relación es una herramienta modelo para comunicarse con los DBA`s. Basado en la información presentada por un DER, el administrador de bases de datos puede comenzar a ver qué tipo de llaves o índices o apuntadores necesitan para acceder eficientemente los registros de base de datos.

La relación entre DFD y DER al actualizar o requerir datos se puede ver solamente en la especificación de los procesos. Por ejemplo, un DER típico se muestra en la siguiente figura, de la cual las cajas rectangulares corresponden a la actualización de datos en un DFD, y podemos ver que existe una conexión, la cual no tenemos normalmente en un diagrama de datos. Esto es porque el foco de atención de un DFD está en las funciones que el sistema realiza, y no en los datos que se necesitan.

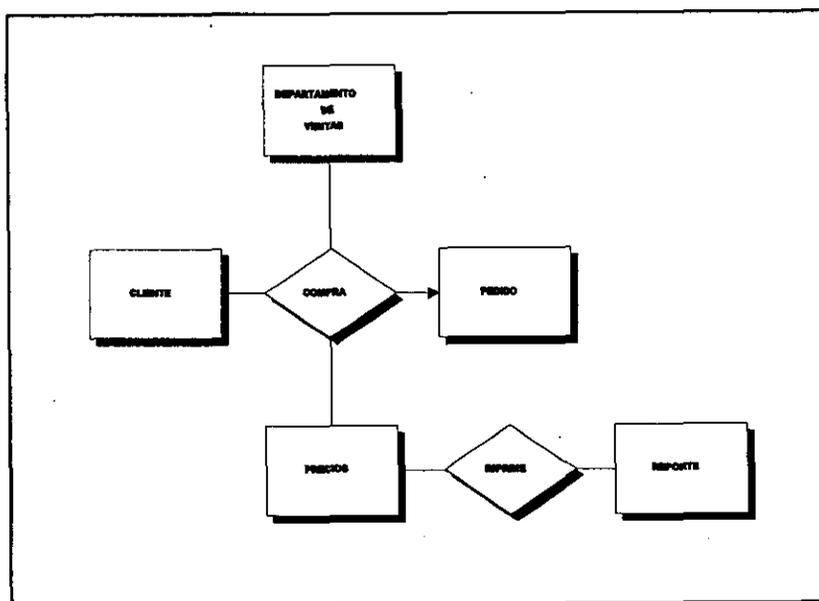


Figura 2.14 Representación completa de un Diagrama Entidad-Relación.

Más adelante especificaremos el significado de cada uno de los componentes del DER, de acuerdo a la notación introducida por Chen en 1976. En 1982, Martin propuso la distinción entre el Modelo entidad-relación, y lo que hasta el momento tiene el diagrama entidad-relación (figura 2.14). No por ello el DER deja de ser una de las técnicas más difundidas y populares

entre analistas e ingenieros de software. En un DER se pueden identificar los siguientes conceptos: entidad, relación, cardinalidad, atributos, ocurrencias, identificadores, tipos de datos, y dependiendo del nivel de desarrollo, se utilizarán otros conceptos que enlazarán directamente con la técnica de diseño de bases de datos (con elementos como subtipos, supertipos, dominios, etcétera).

Existen cuatro componentes de un diagrama entidad relación:

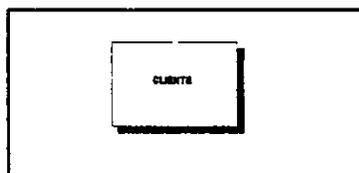
- 1.- Tipos de objetos o entidades
- 2.- Relaciones
- 3.- Indicadores asociativos de tipos de objetos
- 4.- Indicadores de relaciones de Supertipos/Subtipos.

### **Tipos de objetos**

Un tipo de objeto se representa con una caja rectangular en un DER; un ejemplo se muestra en la figura 2.15. Ésta presenta una colección de objetos en el mundo real como miembros individuales que tienen las siguientes características:

- Pueden ser identificados de una sola manera. Sólo existe una manera de distinguir entre instancias individuales de tipos de objetos. Por ejemplo, si nosotros tenemos un tipo de objeto conocido como CLIENTE, debemos distinguir un cliente de otro. Si todos los clientes son uno solo no va a ser nuestro tipo de objeto principal.
- Juega un papel necesario en el sistema que estamos construyendo. Podemos decir que el sistema no puede operar sin el acceso a estos miembros. Obviamente esto es algo que debemos verificar con el usuario al construir el modelo.

- Se puede describir como uno o más elementos de datos. Un cliente puede describirse como cualquier elemento de datos, por ejemplo: nombre, dirección, límite de crédito y teléfono.

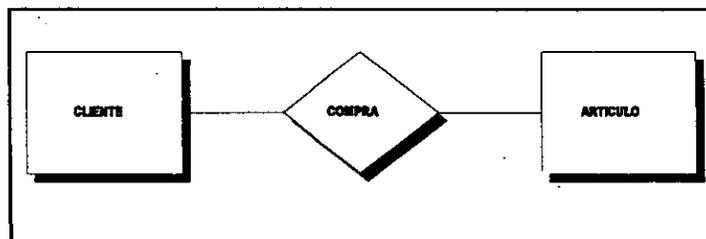


**Figura 2.15** Representación de un tipo de objeto.

Existe una correspondencia entre objetos en un DER y almacenes en el DFD, la cual veremos posteriormente al realizar un balance entre ambos. Es decir, si hay un objeto CLIENTE en el DER, debería haber un almacén CLIENTES en el DFD.

### Relaciones

Los objetos están conectados unos a otros por relaciones. Una relación representa un conjunto de conexiones entre objetos y está representado por un diamante. La figura 2.16 muestra una relación simple que puede existir entre dos o más objetos.



**Figura 2.16** Representación gráfica de relaciones.

La relación representa una asociación entre una o más ocurrencias de un objeto y una o más ocurrencias de otro objeto. La relación de la compra pudiera contener algo como lo siguiente:

- 1 : cliente 1 compra artículo 1,
- 2 : cliente 2 compra artículo 2 y 3,
- 3 : cliente 3 compra artículo 4,
- 4 : cliente 4 compra artículo 5, 6 y 7, etcétera

Como podemos ver, una relación puede conectar dos o más ejemplares del mismo objeto. También puede haber más de una relación entre dos objetos como en la figura 2.17, donde se muestran dos diferentes relaciones entre un paciente y un doctor, el doctor trata al paciente, y a su vez el paciente consulta al doctor.

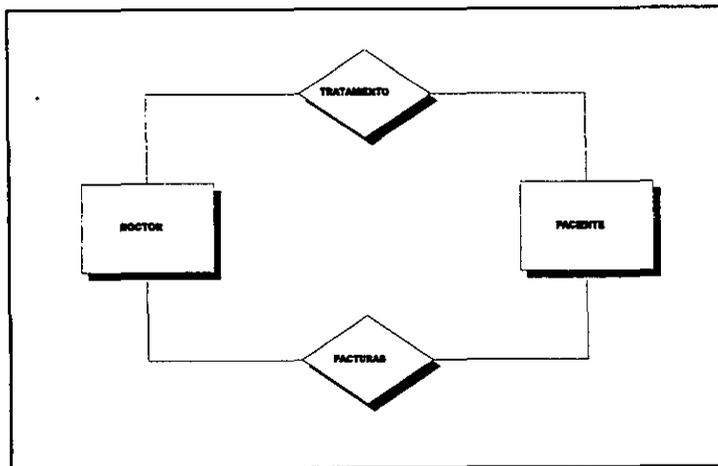
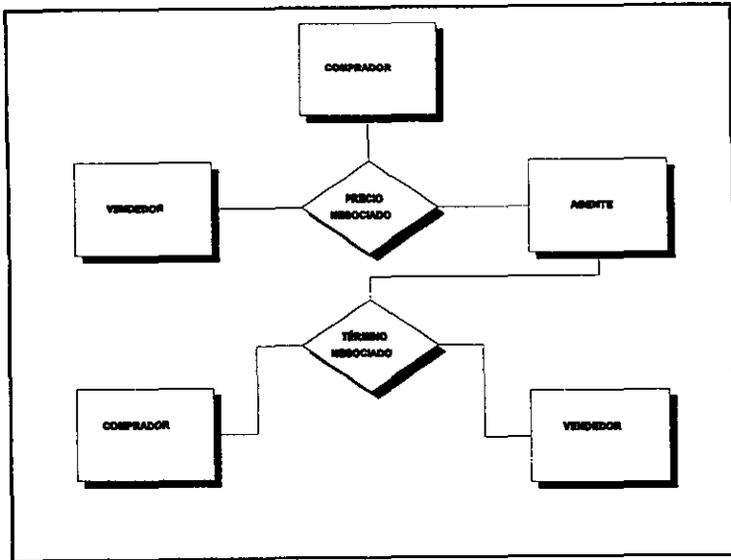


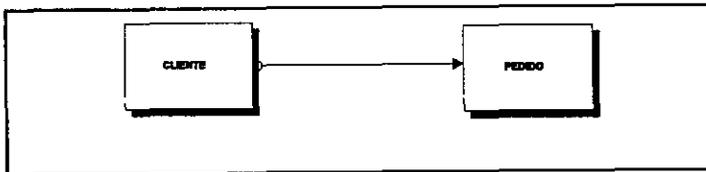
Figura 2.17 Múltiples relaciones entre objetos.

La situación mas común es ver múltiples relaciones entre múltiples objetos, como se muestra en la figura 2.18: entre un comprador, un vendedor, un agente, un departamento de compras y un departamento de ventas.



**Figura 2.18** Múltiples relaciones entre múltiples objetos.

La forma de denotar una relación de uno a N, se muestra en la figura 2.19, en la cual se entiende que si la relación es uno a uno la flecha sólo tendrá una cabeza al final de la flecha, también se puede usar una flecha con doble cabeza.



**Figura 2.19** Representación de relaciones uno a uno.

### Indicadores asociativos de tipos de objetos.

Una notación especial en el DER es la asociación entre tipos de objetos, la cual representa algo que funciona tanto para un objeto como para la relación. Dicho de otra forma, representa a una relación sobre la cual quisiéramos mantener alguna información. Digamos que en el ejemplo anterior, el punto principal en la relación de compras no es más que una asociación entre un cliente y uno o más artículos. Pero supongamos que hay algún dato que nosotros quisiéramos recordar acerca de cierta instancia de una compra (ejemplo, a qué hora se realizó en el día). La pregunta es: ¿ dónde podremos almacenar esta información ? "Hora" no es un atributo del Cliente, ni tampoco del Artículo, por lo tanto nosotros atribuimos esto a la compra, como se muestra en la figura 2.20.

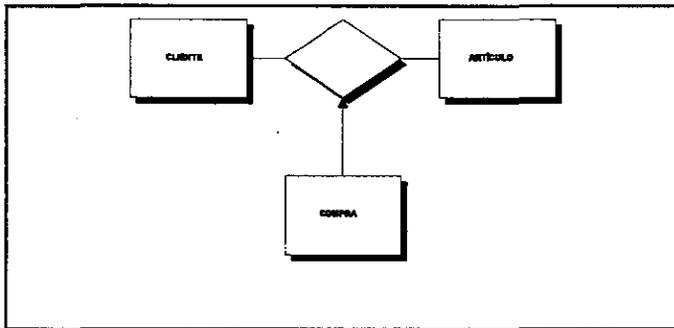
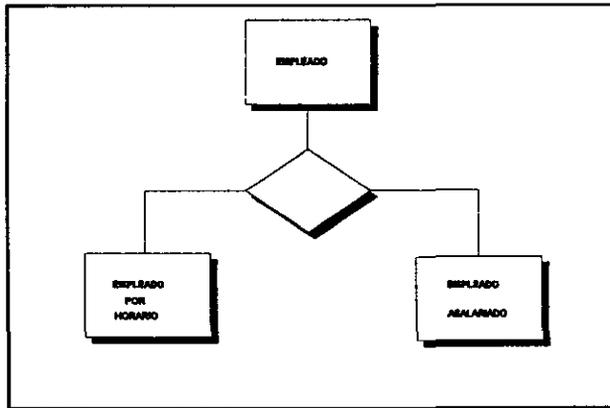


Figura 2.20 Representación gráfica de indicadores asociativos de objetos.

Nótese que la compra está dentro de un rectángulo y que se conecta por una línea directa a una relación sin nombre. Esto significa que la compra funciona como un tipo de objeto (donde queremos almacenar información), y una relación conectando a dos objetos (Cliente y Artículo).

### Indicadores de relaciones de Supertipo/Subtipo

Estos consisten en un tipo de objeto y una o más categorías, conectadas por una relación. La figura 2.21 muestra un Supertipo/Subtipo, donde la categoría general es Empleado y sus subcategorías son Salario del empleado y Horario del empleado. Nótese que los Subtipos están conectados al Supertipo por una relación sin nombre, además de que el Supertipo conecta la relación con una línea cruzada.



**Figura 2.21** Representación gráfica de indicadores de Supertipo/Subtipo.

En esta notación, describimos al Supertipo por elementos de datos que pertenecen a todos los Subtipos. Ejemplo: Nombre, Años de servicio, Dirección, Nombre del supervisor, etcétera, y a su vez los Subtipos contienen diferentes elementos de datos (salario mensual, bonos, hora de entrada y de salida).

## II.2.c BALANCE Y DIAGRAMAS DE FLUJO ESTRUCTURADO

Para terminar, sólo nos falta hacer un balance entre un Diagrama de Flujo de Datos y un diagrama entidad-relación. Como se ha visto en este capítulo, existen varios puntos de vista diferentes entre dos modelos. Pero a su vez hay alguna relación, la cual deberá mantener en orden el modelo del sistema para ser correcto, completo y consistente:

- Todo almacén en un DFD debe corresponder a un tipo de objeto, o a una relación, o a una combinación de un objeto y una relación (asociación de objetos) de un DER. Si existe algún almacén del DFD que no corresponda al DER, algo está mal, y similarmente, si existe alguna relación u objeto que no corresponda al DFD.
- Los nombres de los objetos del DER deben ser iguales a los de los almacenes de datos de un DFD.
- Si existe un diccionario de datos, sus entradas deberán ser tanto para el DFD como para el DER. Ejemplo:

CLIENTES = {CLIENTE}

CLIENTE = nombre + dirección + teléfono + .....

El diagrama de flujo estructurado es la representación gráfica más ampliamente usada para el diseño procedimental. Para representar un paso de procesamiento se utiliza un cuadro, un rombo para representar una condición lógica y flechas para mostrar el flujo de control. La secuencia se representa como dos cuadros de procesamiento conectados por una línea de control; la condición, también denominada *If-then-else* (si-entonces-sino), se dibuja como un rombo de decisión que, si es verdad, hace que se realice el procesamiento de la parte *then* y, si es falso, pasa al de la parte *else*. La construcción estructurada puede estar anidada una a otra desarrollando esquemas lógicos complejos.

## II.3 DISEÑO ORIENTADO AL FLUJO DE DATOS.

El diseño es el segundo paso de la fase de desarrollo de cualquier producto o sistema de ingeniería. El diseño de datos transforma el modelo del dominio de la información, creado durante el análisis, en las estructuras de datos que se van a requerir para implementar el software. Se describe el diseño como un proceso multipaso, en el que a partir de los requisitos de información para el usuario, se resume la forma de la estructura de datos, de la estructura de programa y del procedimiento. Freeman [\*3] amplía esta descripción:

*... El diseño es una actividad en la que se toman decisiones importantes, frecuentemente de una naturaleza estructural. Con la programación, comparte los aspectos relativos a la abstracción de la representación de la información y de las secuencias de procesamiento, pero el nivel de detalle es muy diferente en ambos casos. El diseño construye representaciones coherentes y bien planificadas de los programas, centrándose en las interrelaciones de los componentes de mayor nivel y en las operaciones lógicas involucradas en los niveles inferiores...*

La representación del flujo de la información es parte del análisis de requisitos. Partiendo de un modelo del sistema, la información puede representarse como un flujo continuo que sufre una serie de transformaciones (procesos) conforme va del punto de entrada al punto de salida. El diagrama de flujo de datos (DFD) se utiliza como herramienta gráfica para la descripción del flujo de la información, y el diagrama entidad relación (DER) se utiliza como herramientas gráficas para la descripción estática de los datos; con estas herramientas transformamos el flujo de la información en la estructura del programa.

[\*3] Roger S. Pressman, Ingeniería de Software, 3ª edición, Mc Graw Hill, Octubre 1995.

El enfoque del diseño orientado al flujo de datos (denominado como "diseño estructurado") amplió sus técnicas, integrando el flujo de la información en el proceso de diseño. En 1974, Stevens, Myers y Constantine fueron los primeros que propusieron el diseño de software basado en el flujo de los datos a través de un sistema. Los primeros trabajos fueron presentados en los libros de Yourdon y Constantine publicados en 1979. El método que presentaremos es una síntesis de este material.

El diseño orientado al flujo de datos puede utilizarse en un amplio rango de áreas de aplicación, de hecho, el método de diseño que hace uso de ese diagrama podría aplicarse a cualquier desarrollo de software. Cuando la información se procesa secuencialmente es útil el diseño orientado al flujo de datos. Por ejemplo, el control de procesos, las aplicaciones de control con microprocesadores y otras aplicaciones de ingeniería y científicas caen dentro de esta categoría. Sin embargo, existen otros métodos de diseño para sistemas expertos e interfaces orientadas a los objetos que pueden ser más apropiados.

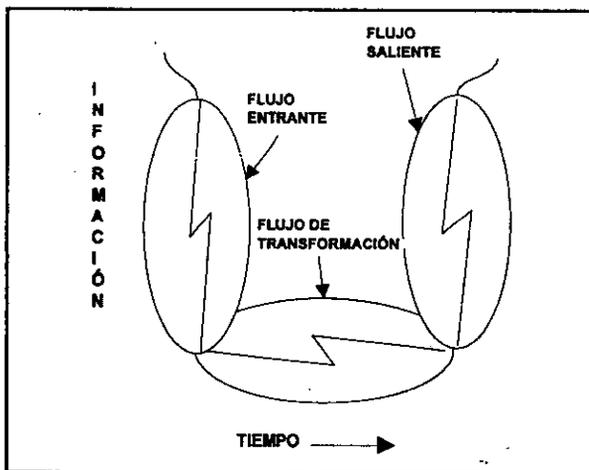


Figura 2.22 Representación de un flujo de transformación.

El modelo fundamental del sistema o el diseño orientado al flujo de datos implica un flujo de transformación, donde la información entra al sistema mediante caminos que transforman los datos externos a una forma interna, como se ilustra en la figura 2.22, identificándose como *flujo entrante*. Los datos de entrada pasan a través de un *centro de transformación*, moviéndose a lo largo de caminos que conducen a la salida del sistema. Los datos que se mueven por estos caminos se llaman *flujo saliente*. Por lo tanto, todo flujo de datos se puede identificar con este concepto.

Sin embargo, el flujo de la información está caracterizado por un elemento denominado *transacción*, que desencadena otro flujo de datos a través de uno entre varios caminos. Cuando un DFD toma la forma que muestra la figura 2.23, lo que tenemos es un flujo de transacción, donde el movimiento de los datos pasa por un camino de llegada que convierte la información en una transacción. En el *centro de transacción* se evalúa la información, y de acuerdo con su valor, el flujo continúa por una de los *caminos de acción*.

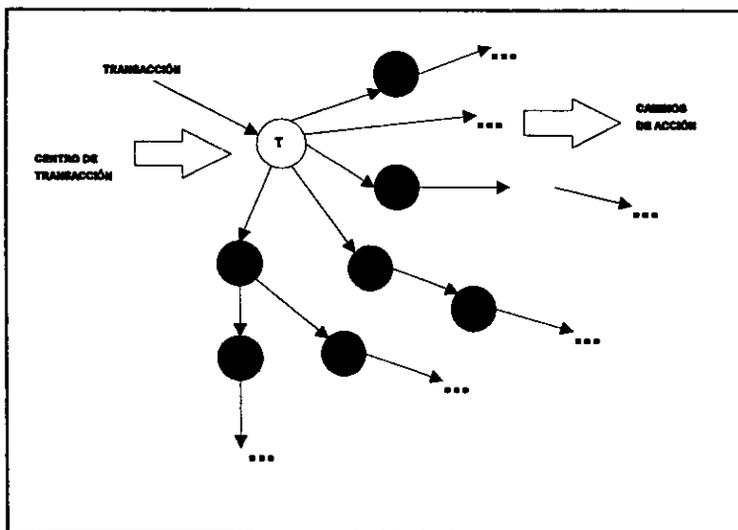


Figura 2.23 Representación de un flujo de transacción.

Hay que tener en cuenta que en el DFD de un gran sistema pueden estar presentes los dos tipos de flujos, y hacer un análisis tanto de transacción como de transformación. Por ejemplo, dentro de un flujo de transacción, el flujo de la información a través de un camino de acción puede tener características de flujo de transformación. La figura 2.24 muestra el método de diseño paso a paso. Sin embargo pueden existir variaciones y adaptaciones.

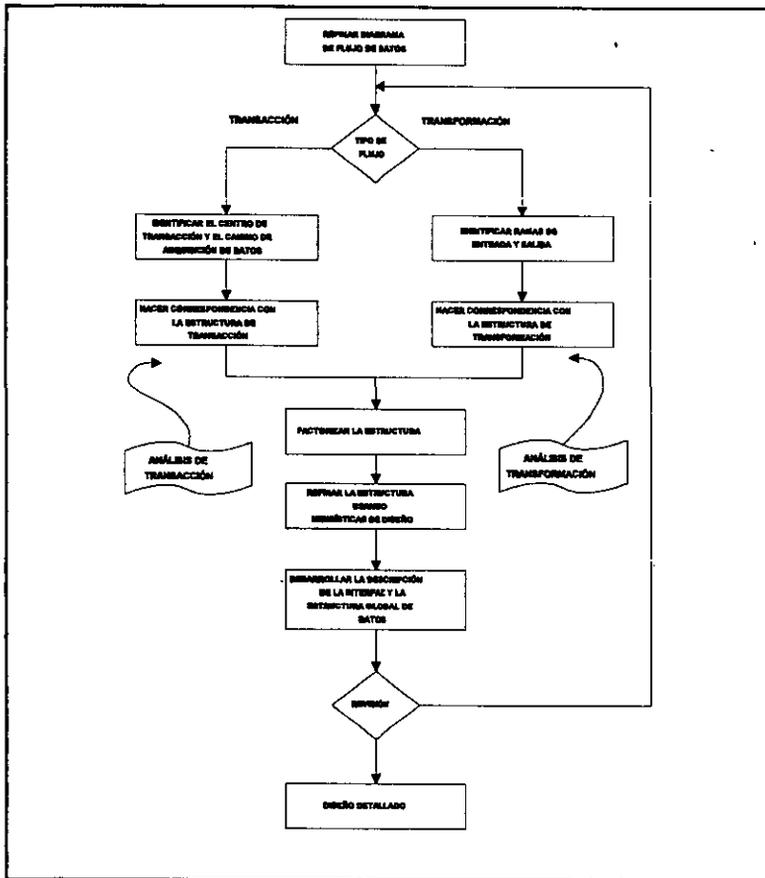


Figura 2.24 Diseño orientado al flujo de datos.

El análisis de transformación es un conjunto de pasos de diseño que permiten convertir un DFD, con características de flujo de transformación, en una plantilla para la estructura del programa.

En muchas aplicaciones del software, un elemento de datos determina uno o más flujos de información, que afectan a la función de acuerdo al elemento de transacción y sus correspondientes características de flujo. A esto se le llama análisis de transacción.

La semántica tradicional puede derivarse de la siguiente interpretación literal: una transacción es una acción (usualmente involucradas dos partes) que transforma el estado del sistema. Más de una vez, el concepto de modularización nos ayuda a obtener la estructura que necesitamos para implementar el estado de transacción. En los años setentas Larry Bjork y Charles Davies pudieron formalizar este concepto, especificando las bases para el sistema. Para 1980 la atención se enfocó en estas técnicas, creando un modelo en la estructura y flujo de cualquier diseño. Un ejemplo específico en esta categoría es la ingeniería de bases de datos; los creadores de bases de datos encontraron que los modelos utilizados eran insuficientes, pues existían numerosas propuestas de crear o diseñar una base de datos con un modelo de transacciones, pero hasta que no encontraron una dinámica para el diseño de archivos en bases de datos no pudieron implementar este modelo, y aplicarlo para la construcción del sistema.

Es fácil implementarlo, pero contiene muchas restricciones en algunos casos, sobre todo para desarrollos que tienen vida larga. Para estos casos probablemente no sea una buena solución implementar el modelo de transacción, pues algunos accesos que hayan sido utilizados pueden ser modificados durante la vida del desarrollo y convertir las decisiones ya tomadas en erróneas u obsoletas.

Los modelos de transacciones tratan de darnos una forma adecuada, para asegurar el diseño de una aplicación, mucho más que otros modelos que tratan de reflejar la estructura de la propia aplicación; los diferentes modelos de transacción se distinguen siempre por el aspecto de dar soporte al desarrollo de la misma aplicación, ya que al tomar múltiples niveles de transacciones, cualquier cambio se facilita en el sistema.

#### Pasos del análisis de transacción y transformación

- 1) Revisión del modelo fundamental del sistema. El modelo fundamental del sistema engloba el DFD de nivel 0 y la información complementaria. El diseño comienza con una evaluación de la especificación de requerimientos y del software.
- 2) Revisión y refinamiento de los diagramas de flujo de datos del software. Con el fin de conseguir un mayor detalle, se refina la información contenida en los modelos de análisis de la especificación de requisitos del software.
- 3) Determinar si el DFD tiene características de transformación o de transacción. En general, el flujo de la información de un sistema podrá representarse como una transformación. Desde luego, cuando se encuentra una característica de transacción, se recomienda una estructura de diseño diferente.
  - 4a) Aislar el centro de transformación especificando los límites de los flujos entrante y saliente.
  - 4b) Identificar el centro de transacción y las características del flujo de cada camino de acción.

- 5) Transformar el DFD en una estructura de software adecuada al procesamiento de transacciones o transformaciones (primer nivel de factorización). La factorización da como resultado una estructura del programa, en la que los módulos de nivel superior toman las decisiones de ejecución y los módulos de nivel inferior ejecutan la mayoría del trabajo de entrada y salida. Los módulos de nivel intermedio ejecutan algún control.
  
- 6) Factorizar y refinar la estructura de transacciones o transformaciones (segundo nivel de factorización). El segundo nivel de factorización se realiza mediante la conversión de las transformaciones individuales (burbujas) de un DFD, en los módulos correspondientes de la estructura del programa. A su vez cada camino de acción del diagrama de flujo de datos tiene sus propias características de flujo de información.
  
- 7) Refinar la estructura inicial del software usando medidas de diseño para mejorar la calidad (utilizar el DER). La estructura del programa siempre puede refinarse aplicando los conceptos de independencia de módulos. Al proponer modificaciones de la estructura se debe considerar la facilidad de mantenimiento.

La aplicación correcta del análisis de transformación o de transacción se complementa con la documentación requerida como parte del diseño arquitectónico. Después de desarrollar y refinar la estructura del programa, deben realizarse las siguientes tareas:

- Desarrollar, para cada módulo, un texto explicativo del procesamiento.
- Dar, para cada módulo, una descripción de la interfaz.
- Definir las estructuras de datos locales y globales.
- Anotar todas las restricciones/limitaciones del diseño.
- Llevar a cabo una revisión del diseño preliminar.
- Considerar la posibilidad de optimizar.

Esta última tarea, si se requiere y se justifica, debe obtener el menor número de módulos pero manteniendo una modularidad efectiva y la menor complejidad de las estructuras de datos que sirva adecuadamente a los requisitos de la información. Aún cuando nuestros procesos no son con tiempos críticos, esta metodología es aplicable también a este tipo de procesos. Para éstos es razonable proponer el siguiente método:

1. Desarrollar y refinar la estructura del programa sin preocuparse del tiempo.
2. Utilizar herramientas CASE para simular el funcionamiento de ejecución.
3. Desarrollar cuidadosamente procedimientos (algoritmos) efectivos en tiempo.
4. Codificar en un lenguaje de programación de alto nivel.
5. Instrumentar el software para aislar los módulos que se lleven al procesador.

Las técnicas presentadas conducen a una descripción del diseño preliminar del software. Se definen los módulos, se establecen las interfaces y se desarrollan las estructuras de datos. Estas representaciones del diseño forman la base de todo el trabajo posterior de desarrollo.

## **CAPÍTULO III**

# **A N Á L I S I S Y D I S E Ñ O**

### **III.1 INTRODUCCIÓN Y METODOLOGÍA**

### **III.2 ANÁLISIS DE LOS PROCEDIMIENTOS**

### **III.3 REQUERIMIENTOS DEL SISTEMA**

### **III.4 BASES Y TAREAS ADMINISTRATIVAS DEL SISTEMA**

### **III.5 DISEÑO DEL SISTEMA**

III.5.a Listas de Archivos y Campos

III.5.b Conexión de Archivos

III.5.c Catálogo de Causas

III.5.d Diagramas de Flujo de Datos

### III.1 INTRODUCCIÓN Y METODOLOGÍA

Todas las organizaciones requieren desarrollar sistemas que les permitan obtener información necesaria para la toma de decisiones. El desarrollo de sistemas, se compone de tres partes: el análisis de sistemas, el diseño de sistemas y la implementación de sistemas. Esto se traduce en un proceso en el que se examina la situación de una empresa con el propósito de mejorarla con métodos más adecuados.

El análisis de sistemas consiste en clasificar e interpretar los hechos, diagnosticar los problemas y emplear la información para recomendar mejoras al sistema. En otras palabras, el examen crítico de las partes que lo integran con el fin de detallar de una manera precisa sus deficiencias o fallas, ya sea para mejorar el sistema existente o diseñar uno nuevo con alto grado de eficiencia, economía y capacidad práctica.

El análisis del sistema es necesario por varias razones, entre las que se cuentan las siguientes:

- **Solución de problemas.**- Cuando un sistema no funciona como se esperaba, se corrigen todas las anomalías.
- **Nuevas necesidades.**- Independientemente de la causa que origine la necesidad, el análisis del sistema identificará las modificaciones o adiciones que deben hacerse al sistema de información con el fin de que la empresa satisfaga tal necesidad.
- **Mejoramiento general de los sistemas.**- Cuando se tiene simplemente la intención de encontrar el modo de hacer mejor lo que ya se está haciendo (reducción de costos, información más rápida, etcétera).

Los cambios en un sistema existente por lo general consisten en modificaciones de un procedimiento o de varios métodos en los mismos procedimientos. Por otra parte, se puede hacer el análisis para implantar un sistema más amplio.

Partiendo de lo que se tiene dentro del sistema actual de Soporte y Control de la producción, derivamos un diagnóstico, para su mejoramiento y rediseño, comprendiendo las siguientes etapas:

- 1.- Análisis de los procedimientos. Este análisis consiste en un examen crítico de cada uno de los pasos que se siguen en la ejecución de los procedimientos que integran el sistema. Realizando una serie de preguntas para cada uno de los pasos que componen el procedimiento, se detectarán las anomalías.
- 2.- Análisis de la documentación del sistema. Esto se refiere a los documentos que se manejan actualmente y sus destinos; es decir, la eficiencia de las formas usadas, lo cual tiene que ver con la forma de archivar, clasificar, reproducir y verificar los datos y documentos que conforman las salidas.
- 3.- Identificación de requerimientos del sistema. Esto consiste en un estudio para saber cómo opera y en dónde es necesario introducirle mejoras. El estudiar el sistema da como resultado una evaluación de la forma en que se están utilizando los métodos actuales y si existe la necesidad o posibilidad de efectuar ajustes.

Por otro lado, el diseño del sistema es la fase de la síntesis (reconstrucción) del trabajo para el mejoramiento del sistema. Hay que estar convencidos de que un nuevo sistema funcionará mejor que el actual y, sobre todo, se deben eliminar las circunstancias que perjudican al sistema existente.

Para la fase de diseño se consideran todas las deficiencias detectadas en la fase del análisis, así como los requerimientos funcionales y de información del usuario.

El diseño de un sistema implicará efectuar: simplificaciones, eliminaciones, adiciones, sustituciones, combinaciones y reacomodos de las partes que lo componen y sus relaciones.

El trabajo del diseño de nuestro sistema actual lo dividimos en tres etapas:

- 1.- Bases y tareas administrativas del sistema. Establecer los objetivos, definir las restricciones, y establecer las premisas del sistema, son las partes que componen esta etapa.
- 2.- Diseño del sistema. Considerar la metodología usada, decidir sobre el grado de mecanización y evaluar las preferencias personales.
- 3.- Evaluación del sistema. Esta etapa consiste en repasar el arreglo general y aprobar el sistema propuesto.

Estas acciones sólo se aplican al diseño de un nuevo sistema. Si el problema es la modificación de un producto que ya existe, sólo se incluyen las actividades que se consideran necesarias. De esta manera nuestro principal objetivo es el mejoramiento o rediseño del sistema actual. Ejemplo: supóngase que se modificará un sistema X, pero las restricciones y los objetivos de ese sistema existente van a permanecer constantes; bien, pues simplemente se excluyen estas consideraciones en el rediseño de las etapas.

## III.2 ANÁLISIS DE LOS PROCEDIMIENTOS

Para realizar este análisis comenzaremos por contestar las siguientes preguntas que nosotros mismos nos hicimos; con esto podremos clasificar las posibles deficiencias que puedan presentar los procedimientos.

¿ Qué se hace o se quiere hacer ?

R: El objetivo principal es rediseñar o ampliar el Sistema de Administración de Problemas (SAP), para poder dar un soporte y un control más completo de la producción, cumpliendo con los requerimientos que manifestaron cada una de las áreas comprometidas a dar este servicio, de tal manera que el sistema proporcione más herramientas para atacar los distintos problemas que se tienen con los otros sistemas de información.

¿ Por qué se hace ?

R: Actualmente el sistema funciona con los objetivos principales que se plantearon, pero aún así, existen pequeñas anomalías que implican: duplicidad de trabajos, retraso de la información, trabajos innecesarios, métodos inadecuados de trabajo, etcétera. Por lo tanto se requiere realizar este rediseño, para que se pueda contar con un sistema que proporcione la información exacta y precisa de cada uno de los problemas que se tienen con los sistemas de información ya liberados, de tal manera que se puedan atacar y disminuir estos problemas. A su vez afinar cada uno de los sistemas de información que existen en el ambiente productivo, para que día a día existan menos cancelaciones o problemas y la información que el usuario requiere sea confiable, manejable y rápida de aplicar o consultar.

¿ Dónde se hace y por qué se hace ahí ?

R: El lugar donde se realiza no es tan importante, como el objetivo del porqué se hace, pero es conveniente estar siempre cerca de las instalaciones de la compañía, para su mayor contacto con el sistema que se diseñó en un principio, y así poder analizar y observar de cerca el comportamiento y las anomalías o deficiencias que presenta actualmente, para después poder atacar éstas con mayor fluidez.

¿ Cuándo se hace y por qué en ese momento ?

R: El cuándo o el tiempo al que nos referimos en esta pregunta no es exactamente una fecha precisa, porque es en estos momentos cuando se está realizando el análisis y diseño de la aplicación, además de que esto puede llegar a implantarse o no, y no exactamente en el sentido de que las mejoras al sistema sean aceptadas, sino sólo como una primera propuesta al querer mejorar y satisfacer los requerimientos de la persona a la que le sirve esta información y así hacer una proyección a futuro de que el sistema se tiene que ajustar. Es por ésta y muchas otras razones por las cuales se tiene que realizar tarde o temprano y no importa el momento exacto en el que se realice.

¿ Quién lo hace y por qué lo hace ?

R: Si comentáramos en estos momentos que estamos realizando esto, es en cierta manera falso y verdadero; el porqué sí lo estamos haciendo se puede ver claro, ya que se ha recopilado información, nos hemos avocado a pensar cómo se puede mejorar, hemos partido a ver cómo se puede hacer el desarrollo del análisis y rediseño de la aplicación, en fin un sin número de pasos que se tienen que seguir para proponer el sistema que pueda ser más útil para la compañía, con el menor número de deficiencias posible. Pero en realidad se necesita no sólo un profesional, sino un grupo especializado para realizar cada una de las tareas que se proponen para el mejoramiento del sistema. De esta

manera se pueden distribuir una serie de responsabilidades y deberes en distintos niveles, y no saturar a un solo elemento, y que este profesional tarde o temprano no pueda completar o argumentar el diseño de la aplicación. El porqué se hace es muy claro, ya que el objetivo del sistema es dar control y soporte; entonces lo que se puede ver es que hay ocasiones donde el sistema puede ser obsoleto para la administración de los sistemas de información, causando grandes fallas en la compañía, es decir, que no se tengan las herramientas o la información necesaria para atacar todo tipo de anomalía con la producción.

¿ Cómo se hace y por qué se hace de este modo ?

R: Ésta es para nosotros una de las preguntas más importantes, porque, para que el sistema funcione correctamente necesita tener las bases bien definidas, los requerimientos bien estipulados y la metodología con la que se va a desarrollar el sistema. Es simplemente por esta razón que se debe comenzar con el pie derecho, si no el día de mañana tendremos los mismos problemas o deficiencias y posiblemente hasta más; de tal manera que dedicamos un capítulo exclusivamente de la metodología a seguir, también presentamos una serie de pasos a seguir en el análisis y diseño del sistema para llegar a concluir que lo que se propuso es lo más adecuado para mejorar la administración de problemas que se tengan con los distintos sistemas de información de la compañía y que esta solución se desarrolló de manera adecuada, para promover buenas cualidades en los sistemas.

Además de aplicar esta serie de preguntas en el principio del análisis como esbozo general, se deben aplicar algunas a cada uno de los requerimientos del usuario, para así poder definir cómo se van a atacar las deficiencias que se tienen.

Comenzaremos por la parte de consulta:

**Catálogo de causas.** Esta parte es sencilla porque se cuenta con una serie de problemas constantes con la misma causa. Para esto se necesita unificar cierto tipo de cancelaciones que tengan relación entre sí pero con diferente código de error, como por ejemplo: si una aplicación trabaja con IDMS y ésta se cancela por un área en uso o es atrapada, tendrá un código S0C1-0966, en cambio si la aplicación trabaja en ADABAS y sucede lo mismo, el código será 0004-NAT3148. Como podemos ver el código es distinto pero se trata de la misma causa de error. A esto nos referimos con unificar el tipo de problema y así hacer del catálogo una referencia fácil para el área de Administración de Problemas, y que no tenga todo un manual de cancelaciones, ya que a ellos no les corresponde saber el código en sí, sino el porqué. Para ciertos casos es difícil, ya que pueden ser nuevas cancelaciones, que el operador no sepa la verdadera causa; para estos casos se tiene que proponer una causa alterna o sin referencia.

Más adelante definiremos todo un catálogo de posibles causas, desglosando cada una de ellas en base al historial de cancelaciones que se tienen dentro de la compañía. Sabemos que la implementación de un catálogo de este estilo no es tarea sencilla, pero estamos convencidos de la racionalidad que respalda el diseño, por lo que creemos que atacado este problema por personal capacitado podrá ser solucionado.

**Historico de Soluciones.** Es difícil atacar este problema ya que se tienen pocas herramientas para conocer a fondo el sistema de SAP, el cual trabaja bajo ambiente Paradox, pero se mostrará más con ilustraciones que se puede poner dentro del sistema, es decir, creando un icono más dentro del sistema para consultas referentes a esto. Es fácil ver que siempre se tiene una solución, ya sea temporal o permanente. Por esta razón lo que ilustraremos serán únicamente ejemplos de cómo se puede consultar este tipo de requerimientos.

**Relaciones de Tiempo-Solución.** Este punto es muy parecido al anterior, ya que en base a la alimentación de información que se le de al sistema obtendremos esta consulta, y podríamos colocarlo dentro del mismo icono que el anterior. También será únicamente ilustrado.

Para continuar tenemos la parte de captura:

**Campos de captura.** En este punto es difícil especificar por tema el requerimiento, ya que lo único que se requiere son campos que denoten quién hizo la petición y de dónde viene ésta, es decir, qué área lo solicita. Lo único que se hará es poner uno o dos campos en la pantalla de peticiones especiales, la cual ya fue creada, y que los datos que se capturen queden guardados en la misma base de datos del sistema. Esto también se hará en forma ilustrada.

**Alta automática de folios para las cancelaciones por lote.** Éste es el punto o requerimiento más importante del sistema, ya que utilizaremos toda una metodología para realizarlo. Haremos énfasis especial en este punto, además de que será el primero en ser analizado, ya que es todo un procedimiento el que se planea hacer, desde dónde obtener cada cancelación, guardarla en un archivo secuencial (el cual vaya creando un histórico), crear un proceso por lote con una serie de programas en un lenguaje determinado y utilizar herramientas para poder dejar la información desglosada al sistema de SAP, y que el área de Administración de Problemas tenga el folio levantado por cada problema que se presentó en el día con la producción por lote, y no se tenga un descuadre en base a los folios que se pudieron capturar y las cancelaciones del día. Se propone hacerlo automáticamente para que no se tengan fallas manuales. Cabe agregar que el folio será levantado con la información más completa porque viene de la salida de la bitácora del rastreador de trabajos JOBTRAC, por el cual se manda toda la producción por lote, y es difícil que algo dentro de la producción se pida por fuera de esta herramienta. Una de las referencias de

la compañía para ver el número de cancelaciones por año, mes y día es una aplicación que saca las estadísticas de todas las cancelaciones de la producción por lote; esta aplicación se basa en la misma salida de la bitácora del rastreador de trabajos, por eso es que la información con la que se levanten los folios será completamente consistente con lo sucedido, y coincidirá con todas las áreas usuarias a ambos paquetes (o recursos, como se le quiera llamar).

**Bases de datos en distintos servidores.** Este punto será abordado de manera urgente, ya que el problema causado por esto es enorme, pues se tiene desfase con la información y no cuadra el consecutivo de los folios. Por lo tanto manejaremos el diseño de implementar todo en una sola base de datos y así evitar drenar información a otro lado.

Se ha hablado únicamente de algunas preguntas al análisis del sistema del porqué y el cómo se arreglarán cada uno de los requerimientos del usuario. Faltaría ver dónde, cuándo, y quién va a realizar esto, pero esa parte la dejamos para el momento que se implante el rediseño, asignando responsabilidades y tareas a distintas personas y áreas; ellos podrán saber, por ejemplo, para cuándo lo tienen y dónde lo van a realizar.

### III.3 REQUERIMIENTOS DEL SISTEMA

Los requerimientos del sistema se refieren a las características que deben incluirse en éste, y entre ellas pueden contarse: producción de información, inclusión de una forma específica para capturar o procesar información, control de una actividad en particular, etcétera.

De esta manera, la determinación de requerimientos establece la relación del estudio de los sistemas con los detalles relacionados con los mismos requerimientos. Para poder determinar estos requerimientos realizamos una tabla, la cual contendrá:

ÁREA	CONSULTA	ALTA Y CAPTURA
MESA DE AYUDA	CATÁLOGO DE CAUSAS	CAMPOS DE CAPTURA
		BASE DE DATOS ÚNICA
CONTROL DE LA PRODUCCIÓN	HISTÓRICO DE SOLUCIONES	ALTA AUTOMÁTICA DE FOLIOS
	RELACIÓN DE TIEMPO-SOLUCIÓN	
SOPORTE A LA PRODUCCIÓN	HISTÓRICO DE SOLUCIONES	ALTA AUTOMÁTICA DE FOLIOS
	RELACIÓN DE TIEMPO-SOLUCIÓN	CAMPOS DE CAPTURA
ADMINISTRACIÓN DE PROBLEMAS		CAMPOS DE CAPTURA
		BASE DE DATOS ÚNICA

Una vez obtenidos y bien clasificados estos datos se analizan para determinar las especificaciones de los requerimientos, es decir, la descripción de las características del nuevo sistema, y a su vez pasar a la siguiente etapa del análisis e investigación de la documentación del sistema actual.

### III.4 BASES Y TAREAS ADMINISTRATIVAS DEL SISTEMA

Como primera etapa para introducirnos al diseño, hemos mencionado el definir tareas y bases administrativas del sistema, lo cual está dividido en tres partes:

- A) **Establecer los objetivos.** Al iniciar el diseño de un sistema, las identificaciones de objetivos necesariamente deben estar expresadas en términos amplios y generales. A medida que el diseño progresa, se crean objetivos más concretos. La búsqueda exhaustiva de la información necesaria, por quién, cuándo y en qué forma, descubre objetivos específicos de la información que ayudan al trabajo creativo de los sistemas. Un enunciado sencillo sería: "*Proporcionar cualquier información que sea necesaria al receptor, con el fin de que satisfaga las necesidades de expedición y servicio*".

Deseamos crear un sistema para administrar los problemas que existen por cada uno de los sistemas de información que componen el funcionamiento de la producción de la compañía, capaz de soportar las fallas que tenga el usuario al aplicar sus movimientos y controlar él mismo estos problemas, disminuyendo al paso del tiempo el número de percances que tenga esa aplicación y a su vez afinando la misma. Éste es uno de los objetivos principales por los que se creó el sistema SAP, al cual no tocaremos, ya que en cierta forma se cumplió. Es por esto que no podemos plantear objetivos ya definidos, sino buscarlos conforme al rediseño de la aplicación, en base a los requerimientos ya planteados por el usuario. En este caso nuestros objetivos son:

- Alta de folios automática para los problemas que se tengan por lote, para llevar correctamente la administración de problemas, conteniendo la información exacta de cada uno extraída de la bitácora de producción, para que el día de mañana no se presente un problema que no fue tratado con el mismo nivel de solución, como otros que sí fueron reportados y atendidos por cualquier nivel de soporte, como debe de ser. De cierta

manera, que no exista la información descuadrada para las áreas que dan soporte a la producción, ya que esto puede causar un mal mantenimiento de la información procesada por el usuario, o duplicidad de problemas, teniendo constantemente una falla que puede ocasionar retraso de actividades, para cualquier sector de la compañía.

- Cumplir con todos los requerimientos de consulta, para facilitar el soporte de cada problema que se tiene, o de la aplicación en general, y así cumplir con el área que lo solicita, para que ésta pueda administrar cada una de sus actividades. Cabe aclarar que sólo se ilustrará la forma en que el usuario podrá consultar estos documentos y no desarrollarlos en la aplicación en sí, como se mencionó en el análisis del sistema.

- Mostrar cómo quedarán los campos que se pide agregar en las peticiones especiales, para que el área de Administración de Problemas pueda distinguir de dónde viene y quién hizo el requerimiento.

Después de establecer estos objetivos, generalizando en pocas palabras los puntos a tratar, podemos avocarnos a otra etapa de este procedimiento.

B) **Definir las restricciones.** Deben establecerse fronteras aún cuando sea en forma tentativa, para definir el ámbito del estudio y los posibles límites del sistema o de quien desarrolla el sistema. Este punto es bueno tocarlo, aunque ya se habló algo al respecto anteriormente. Se refiere a aquellos requerimientos que no se van a desarrollar por completo o que de plano no se tocarán, como por ejemplo:

- Dado que la metodología que utilizamos en este proyecto es estructurada, se atacarán aquellos puntos susceptibles de ser analizados bajo este enfoque. En los casos en que haya incompatibilidad entre lo ya desarrollado y lo que estamos proponiendo, procederemos a describir la interfaz entre ambos módulos.

- La unificación de la base de datos en un solo servidor. Este punto vale la pena mencionarlo, ya que la urgencia de contar con una base de datos consistente y congruente para el área de Administración de Problemas es crítica. Por ello se tiene que hacer prioritariamente, y partir de aquí para diseñar y modificar el resto del sistema y establecer las etapas de implementación, de acuerdo a las prioridades de la empresa.

Tanto las restricciones como los objetivos suelen ser incompletos al inicio del diseño del sistema, pero se hacen suposiciones y se toman las mejores decisiones posibles, teniendo en cuenta que pueden ser revisadas y refutadas a medida que progresa el trabajo de diseño.

C) **Establecer las premisas del sistema.** Los pasos previos en el diseño del sistema proporcionan información sobre las normas de calidad y el volumen de trabajo que ha prevalecido, por lo que es sencillo proyectar y fijar a éstos como los niveles sobre los cuales debe diseñarse el sistema. Las herramientas que se utilizan en cada uno de los pasos le dan forma al sistema y proporcionan flexibilidad en el desarrollo del mismo.

Para comenzar fijando los niveles de solución o diseño ordenaremos los requerimientos de la siguiente forma:

- 1º) Unificación de la Base de Datos
- 2º) Creación del Catálogo de Causas
- 3º) Creación del Histórico de Soluciones
- 4º) Creación de Relaciones Tiempo Solución
- 5º) Inclusión de Campos de Captura
- 6º) Diseño de Alta Automática de Folios para las Cancelaciones por Lote

Con este orden tocaremos cada punto del rediseño de la aplicación, pensando desde luego en desglosar y trabajar cada uno de los requerimientos del usuario.

### III.5 DISEÑO DEL SISTEMA

Para iniciar con el diseño de la aplicación misma o sistema se proponen las siguientes listas de archivos con sus respectivos campos y descripción:

#### III.5.a Listas de Archivos y Campos

<b>REPORTES</b>	
<b>CAMPO</b>	<b>DESCRIPCIÓN</b>
Número de Folio	Consecutivo
Status	Abierto o Cerrado
Reportado por:	Nombre del usuario que reporta
Teléfono	Teléfono o extensión del usuario que reporta el problema
Hora del reporte	Hora en que se recibió el problema
Fecha del reporte	Fecha en que se recibió el problema
Recibido por (solucionador) :	Operador o persona a la que se le asignó el problema para solucionarlo
Categoría (Plataforma en la que está desarrollada la aplicación)	- Aplicaciones Equipo Central - Hardware Equipo Central - Software Equipo Central - Aplicaciones Computadoras Personales - Micros - Vax
Producto/aplicación	En qué aplicación o producto se tiene el problema
Descripción o tipo de problema	Es una descripción breve del problema

Canalizado	Bandera que indica si fue canalizado el problema a otra persona
Canaliza a:	Persona a la que se le canalizó el problema
Causa	Causa del problema
Fecha de solución	Fecha en que se solucionó el problema
Hora de solución	Hora en que se solucionó el problema
Tiempo de solución	Tiempo total transcurrido desde que se reportó el problema hasta que se solucionó
Vo.Bo. del usuario	Bandera que indica si ya se le notificó al usuario de que el problema ha sido solucionado, y dio su Visto Bueno
Fecha del Vo.Bo.	Fecha en que el usuario dio Visto Bueno
Hora del Vo.Bo.	Hora en que el usuario dio Visto Bueno
Para el caso de los problemas por Lote se incluyen los siguientes datos	
Proceso	Nombre del proceso cancelado
Número de Tarea	Número de Tarea del proceso cancelado
Número de paso	Paso en el que se canceló
Programa	Programa que se estaba ejecutando
Fecha de cancelación	Fecha en la que ocurrió la cancelación
Hora de cancelación	Hora en la que ocurrió la cancelación
Error	Código de cancelación según cve. de error

<b>SOLUCIONADORES</b>	
<b>CAMPO</b>	<b>DESCRIPCIÓN</b>
Clave	Clave del empleado o externo
Nombre	Nombre de la persona que soluciona
Nivel	Nivel en el que se encuentra esta persona
Subdirector	Subdirector o Supervisor en el cargo
Área	Área en la que se encuentra
Teléfono (extensión)	Teléfono o extensión a donde se localiza

<b>APLICACIONES</b>	
<b>CAMPO</b>	<b>DESCRIPCIÓN</b>
Plataforma	Equipo Central, PC, VAX, etcétera
Clave	Clave o siglas de la aplicación
Descripción	Nombre de la aplicación
UEN	Unidad estratégica de negocio
Experto	Experto que lleva a su cargo la aplicación

<b>CATÁLOGO DE CAUSAS</b>	
<b>CAMPO</b>	<b>DESCRIPCIÓN</b>
Clave	Clave de error
Descripción	Breve descripción del error

<b>SOLUCIONES</b> (sólo cuando se canaliza)	
<b>CAMPO</b>	<b>DESCRIPCIÓN</b>
Número de Folio	Consecutivo
Solucionador	Nombre de la persona/área que le está dando solución
Fecha y Hora de Canalización	Fecha y Hora en que se asignó el reporte
Fecha y Hora de cierre	Fecha y Hora en que se le dio solución o canalizó a otro nivel
Solución	Solución aplicada
Tiempo de solución	Tiempo en el que se resolvió el problema

<b>USUARIOS</b>	
<b>CAMPO</b>	<b>DESCRIPCIÓN</b>
Nombre	Nombre del usuario que reporta
Clave de empleado	Clave del empleado o externo
RFC	Registro Federal de Causantes
Puesto	Puesto o actividad que desempeña
Ubicación Física	Lugar donde se encuentra físicamente
Área	Área en la que se encuentra
Teléfono (extensión)	Teléfono o extensión a donde se localiza
Clave de Correo	Clave de correo que tiene asignado
Datos del equipo	Marca, modelo, número de serie

<b>SOLICITUDES ESPECIALES</b>	
<b>CAMPO</b>	<b>DESCRIPCIÓN</b>
Número de Folio	Número de folio asignado
Estatus	Estatus del folio
Fecha y Hora de recepción en Help Desk	Fecha y hora de la petición
Solicitado por :	Nombre del usuario que hace la petición
Teléfono o Extensión	Teléfono o extensión donde se encuentra
Recibido por :	Nombre de la persona que atendió
Plataforma	Equipo Central, PC, VAX, etc...
Aplicación	Aplicación o sistema para liberar petición
Descripción	Breve descripción del requerimiento
Fecha de compromiso	Fecha en la que esté completamente desarrollado y liberado a producción

<b>UEN's (unidades estratégicas de negocio)</b>	
<b>CAMPO</b>	<b>DESCRIPCIÓN</b>
Clave	Clave o siglas de la aplicación
Descripción	Descripción breve de la aplicación

### III.5.b Conexión de Archivos

Por consiguiente la conexión de estos archivos de la base de datos se da como los diagramas siguientes lo muestran (figura 3.1, 3.2 y 3.3), tanto para alta de folios como consulta.

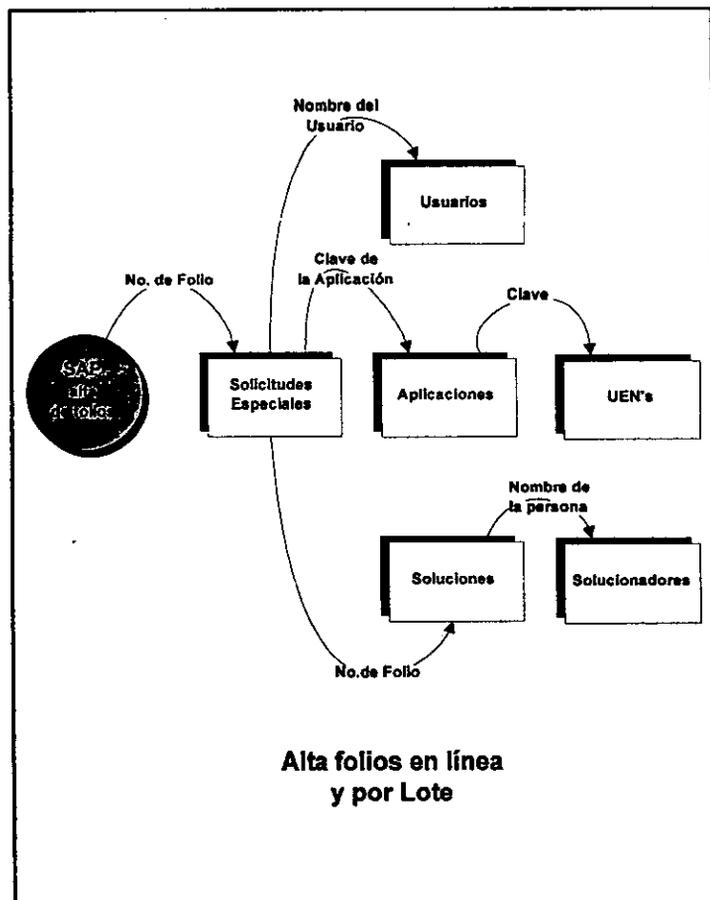


Figura 3.1 Alta de folios en Línea y por Lote.

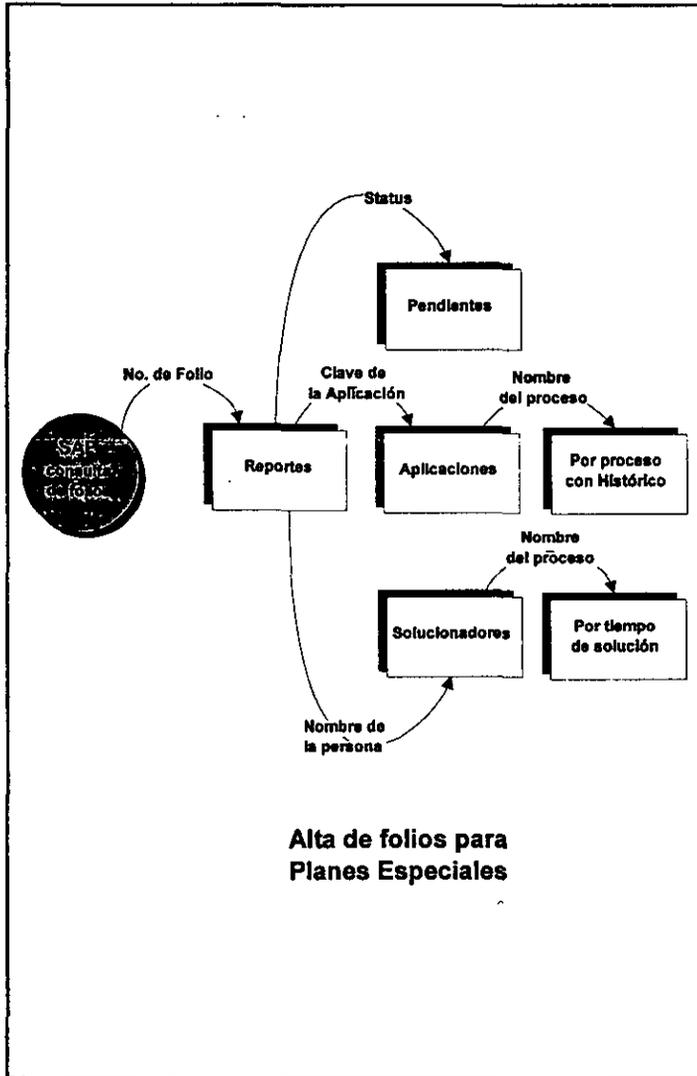


Figura 3.2 Alta de folios para Planes Especiales.

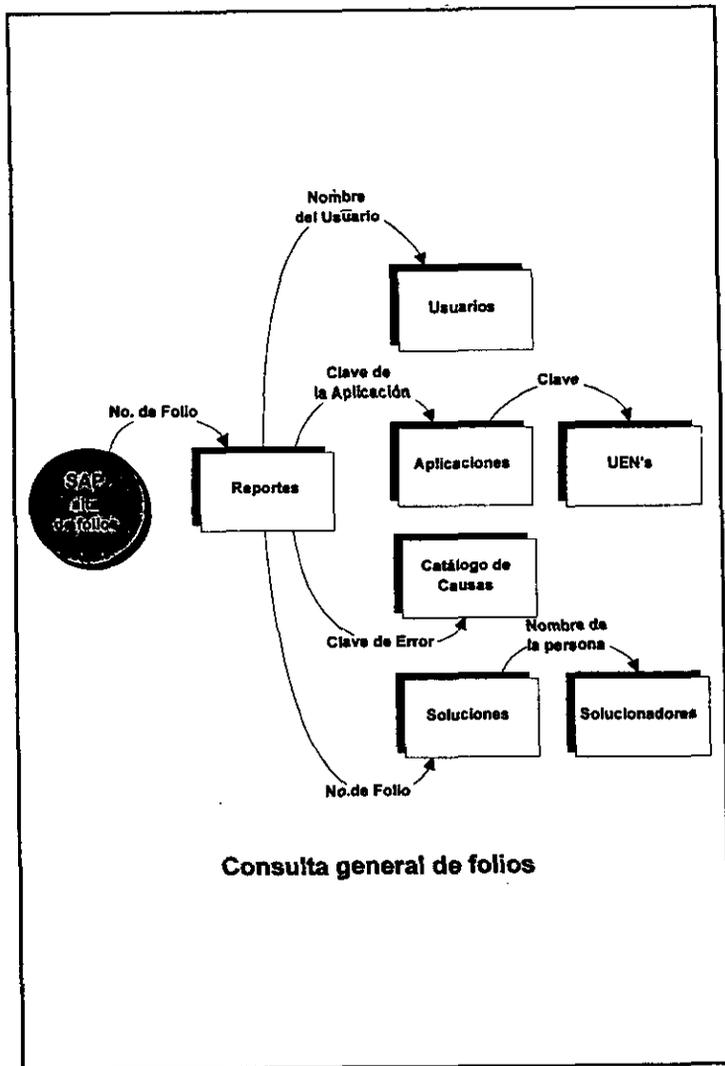


Figura 3.3 Consulta de folios.

### III.5.c Catálogo de Causas

A continuación se incluye una tabla previamente definida con ciertos códigos de cancelación (tipificados por la gente de Administración de Problemas en coordinación con Soporte a la Producción), para el catálogo de causas, donde se separa clave y descripción.

Tabla con códigos de Cancelación. (Sopprod)

#### A.- OPERACIÓN

- A001 - Cancelación por dependencia.
- A002 - Error al contestar requerimiento.
- A003 - Cancelado por solicitud del operador.
- A004 - Cancelado por parámetros erróneos del usuario.
- A005 - Cancelado por parámetros erróneos del armador.
- A006 - Por estar en espera de un requerimiento.
- A007 - Solicitud de cinta en unidad de cartucho.
- A008 - Solicitud de cartucho en unidad de cinta.
- A009 - Solicitud de la base de datos para respaldar.
- A010 - No se atendió un requerimiento de sistema.
- A011 - Por ipl ( *initial program load* ).

#### B.- RECURSOS

- B001 - Disco en mal estado ( *i/o error* ).
- B002 - Por problemas con un controlador.
- B003 - Unidad de cinta con problemas.
- B004 - Unidad de cartucho con problemas.
- B005 - Falta espacio en cintas o cartuchos.
- B006 - Cinta en mal estado.
- B007 - Falta parámetro en gdg ( *grupo de generaciones* ) de dcb ( *data control block* ).
- B008 - Región o memoria insuficiente.
- B009 - Acceso no autorizado.
- B010 - Falta espacio en disco.
- B011 - Falta espacio en un elemento PO ( *partitional organization* ).
- B012 - No hay espacio en archivo VSAM.
- B013 - Error al transferir un archivo de un ambiente a otro.

### **C.- POR JCL**

- C001 - El procedimiento tiene errores de jcl.
- C002 - EL programa tiene incongruencias con el jcl.
- C003 - Definición de espacio insuficiente en jcl.
- C004 - Falta DDNAME.
- C005 - Falta DCB ( *data control block* ).
- C006 - Error en la dcb ( *data control block* ).
- C007 - Incongruencia en la dcb entre el programa y el jcl ( *job control lenguaje* ).
- C008 - Archivo no encontrado ( *data set not found* ).
- C009 - Procedimiento no encontrado.
- C010 - Proceso mal armado en cadena.
- C011 - Parámetro simbólico no encontrado en el procedimiento.
- C012 - Parámetro UNIT incorrecto.
- C013 - Por códigos de condición.

### **D.- POR PROGRAMA**

- D001 - Error en datos, el programa no contempla algunos datos.
- D002 - Error en datos, los datos fueron mal generados.
- D003 - Error al encontrar una división entre 0.
- D004 - Cancelación por estar ciclado ( *loop* ) el programa.
- D005 - EL programa está mal compilado.
- D006 - El programa contiene errores de lógica.

### **E.- BASE DE DATOS**

- E001 - Por áreas atrapadas.
- E002 - El status de las áreas no es el requerido por el programa.
- E003 - Error al dar BIND/MODIFY/UPDATE/WRITE.
- E004 - Error al dar FIND/OBTAIN/GET/READ.
- E005 - Por falta de espacio en las áreas de base de datos.
- E006 - Por registro duplicado.
- E007 - Por registro no encontrado.
- E008 - Por correr en versión central sin estar activa la Base de Datos.
- E009 - Por no encontrar fin de paquete.
- E010 - Por problemas o incongruencias con el subesquema.

## **F.- REPROCESOS**

- F001 - Por pérdida de reportes.
- F002 - Porque el proceso no generó reportes.
- F003 - Por resultados erróneos.
- F004 - A solicitud del usuario.

## **G.- ARCHIVOS**

- G001 - Archivo duplicado.
- G002 - Archivo de entrada vacío.
- G003 - No encontró fin de archivo.
- G004 - Longitud errónea de registro.
- G005 - Acceso a archivos de CICS activo.
- G006 - Definición errónea del archivo VSAM.
- G007 - Llave duplicada en archivo VSAM.
- G008 - Error de *i/o* en archivo VSAM.
- G009 - Archivo VSAM mal cerrado.
- G010 - Llave no encontrada en archivo VSAM.
- G011 - No existe el elemento en la biblioteca particionada.
- G012 - Módulo no encontrado.

## **H.- USUARIO**

- H001 - El proceso se modificó en directo o con *override*.
- H002 - No está el paso de donde se quiere reiniciar.
- H003 - No se dio la razón de reproceso.
- H004 - No se modificó el tipo de reproceso.
- H005 - Falta de dar de alta uno o varios gdgs ( *grupos de generaciones* ).

Con esta lista de códigos de cancelación generalizados se podrá identificar fácilmente el problema y a su vez solucionar éste, de tal manera que todas las áreas reconozcan la cancelación de primera instancia y no estar repitiendo o mal informando el problema que se tuvo. Otra solución que se presenta es extraer información de cierto periodo en base a una misma cancelación y atacar de lleno, solicitando exactamente los recursos necesarios para corregir la falla y resolverla en el menor tiempo posible.

### III.5.d Diagramas de Flujo de Datos

A continuación presentaremos como una comparación la situación actual del sistema y la situación propuesta, para así denotar las modificaciones que se tienen y el proceso de mejoras que nos proporciona la reingeniería de la aplicación.

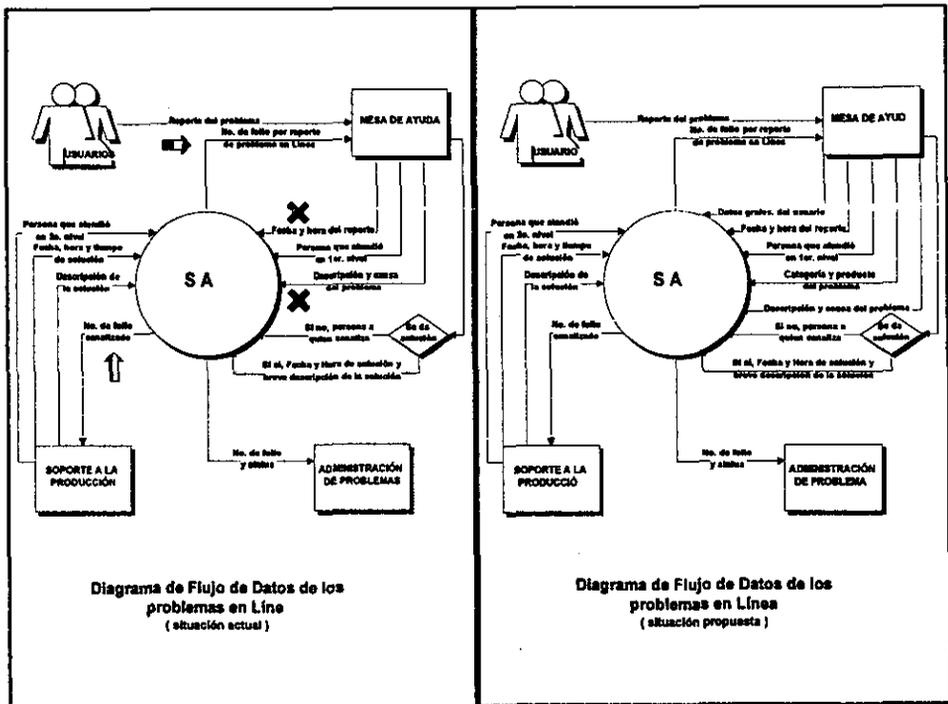


Figura 3.4 Diagramas de flujo de datos de los problemas en línea.

En los diagramas anteriores (figura 3.4) quedan marcados con una cruz los nuevos campos de captura, con los cuales se podrá identificar fácilmente a la persona que reportó el problema y el producto o aplicación a la que pertenece este reporte. En la descripción y causa quedará un código de la lista anterior.

Por último podemos ver unas flechas en los campos que actualmente se distinguen entre los folios de producción (denotados con una P) y los folios mesa de ayuda (denotados con una H), para su diferencia entre las distintas bases de datos; en la situación propuesta no existe esta diferencia, teniendo un solo folio para una base de datos y evitando duplicidad.

En el siguiente diagrama (figura 3.5) de los problemas por lote, se ven claramente las mismas diferencias y adicionalmente existe una flecha donde se describen los datos generales del proceso cancelado y no únicamente el nombre del proceso, de tal manera que se pueda ver claramente el paso, programa, error, número de trabajo.

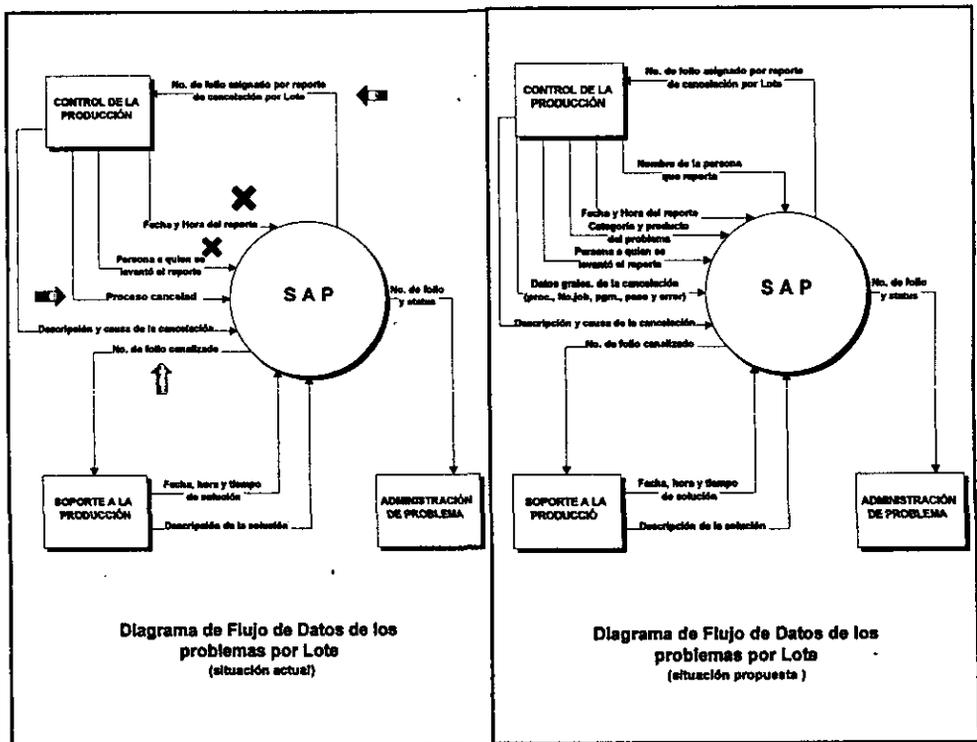


Figura 3.5 Diagramas de flujo de datos de los problemas por lote.

Tanto para los problemas en línea como los problemas por lote se tiene un diagrama, pero los nuevos desarrollos o solicitudes especiales no contemplan un diagrama en específico, sino que se utiliza el mismo diagrama para éstos, dependiendo la necesidad del usuario.

Una vez que se comentó anteriormente que en un nuevo desarrollo dentro de la misma aplicación ya liberada se sigue el mismo procedimiento de reporte, vamos a poner la situación propuesta como un tercer diagrama, ya que esta vez sí tendrá un formato nuevo de solicitud. Para ello supondremos en una forma de solicitud, tomando en cuenta las posibles diferencias como una visión gráfica del nuevo procedimiento (figura 3.6).

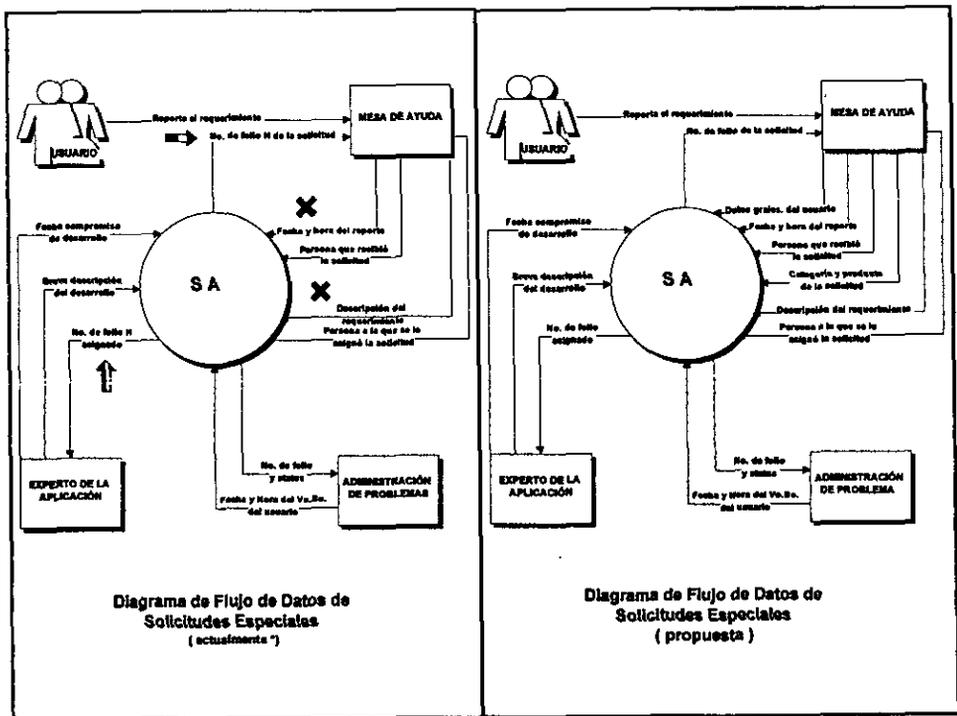


Figura 3.6 Diagramas de flujo de datos de solicitudes especiales.

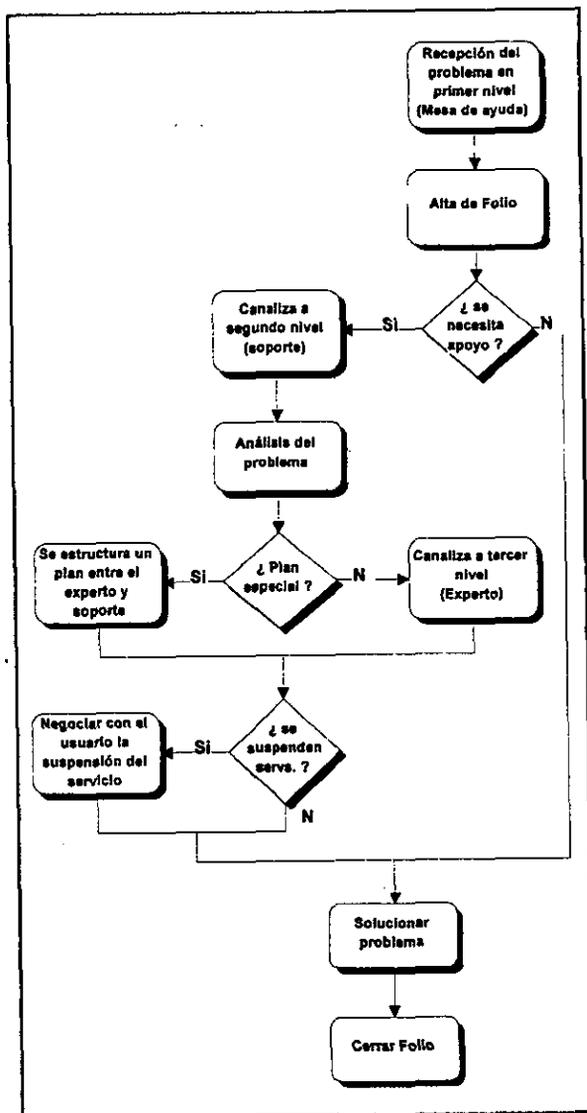
Claramente se puede ver que estos folios son los que se levantan en la segunda base de datos, denotándose con una H. En el diagrama propuesto ya no existe esta diferencia, el (\*) simplemente es porque nosotros utilizamos el modelo propuesto para llegar al modelo actual, es decir, cómo sería el reporte de solicitudes especiales si existiera en el sistema SAP que se utiliza en estos momentos.

Ya nos ubicamos en el procedimiento que sigue el sistema para reportar un problema o solicitar una petición especial, ahora nos resta ver el nuevo procedimiento de atención de folios, tomando en cuenta también las diferencias entre la propuesta y el sistema actual.

Como primer elemento, en el diagrama (figura 3.7) a continuación se puede ver el seguimiento de los folios levantados por problemas en línea, en el cual se propone continuar con la misma dinámica.

El procedimiento que sí cambia es para levantar folios con problemas por lote (figura 3.8), ya que la situación propuesta lo hace automáticamente, y el procedimiento actual es primero abrir folio y reportar a segundo nivel, luego analizar y corregir la falla; segundo nivel proporciona causa-solución a Control de la Producción y por último éste hace el cierre del folio.

Para el procedimiento propuesto de levantamiento de folios con cancelaciones por lote (figura 3.9), se identifica y reporta a segundo nivel, como primera instancia, posteriormente se analiza y corrige el problema, informando causa-solución; seguido de esto se ejecuta un proceso automáticamente todos los días a las 6:00 a.m., el cual extrae la información de las cancelaciones obtenidas durante el día, únicamente resta consultar los folios abiertos para su cierre con la causa-solución obtenida por segundo o tercer nivel.



**Figura 3.7** Folios levantados para problemas en línea.

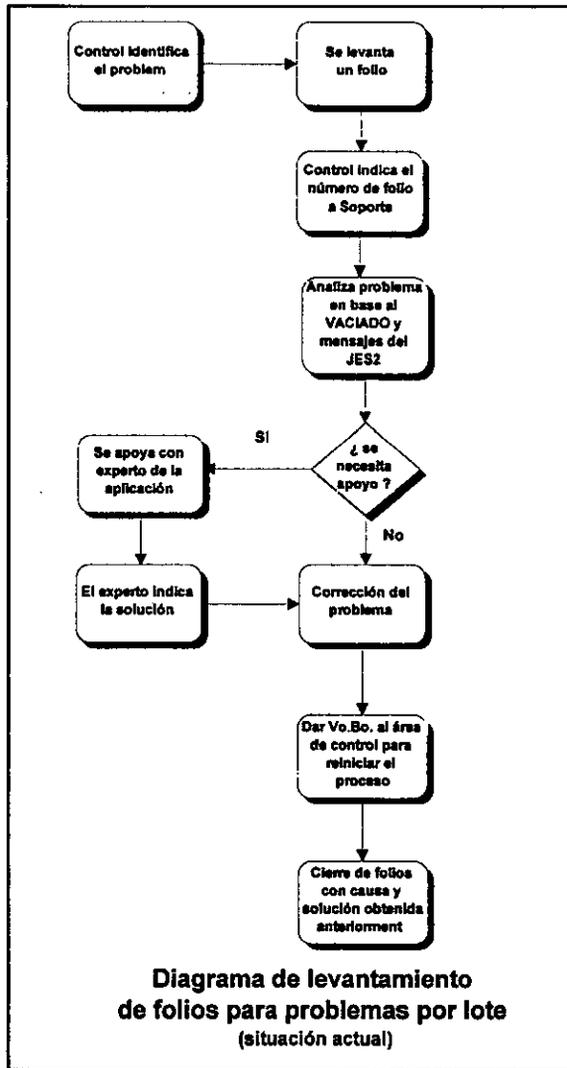


Figura 3.8 Folios levantados para problemas por lote (situación actual).

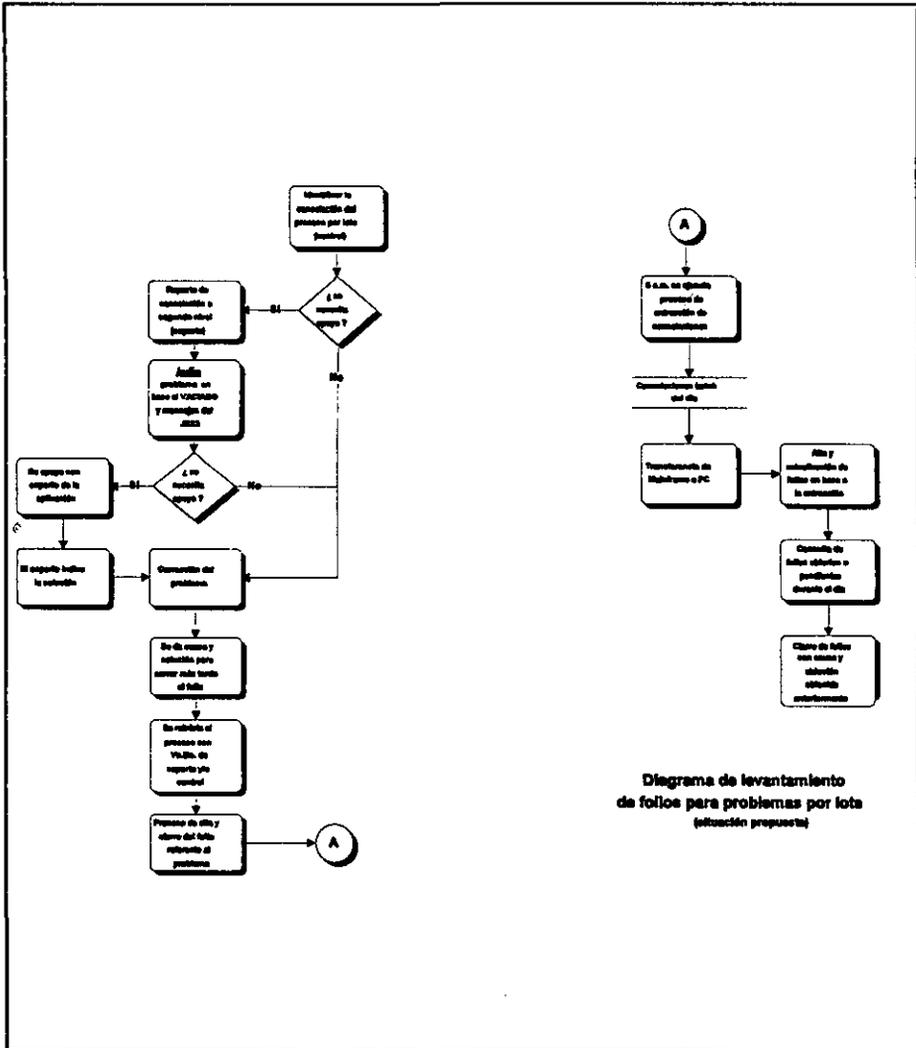


Figura 3.9 Folios levantados para problemas por lote (situación propuesta).

Como un diagrama más (figura 3.10) para levantamiento de folios con solicitudes especiales sólo planteamos la situación propuesta, por lo mismo que no existe actualmente procedimiento para peticiones de este tipo.

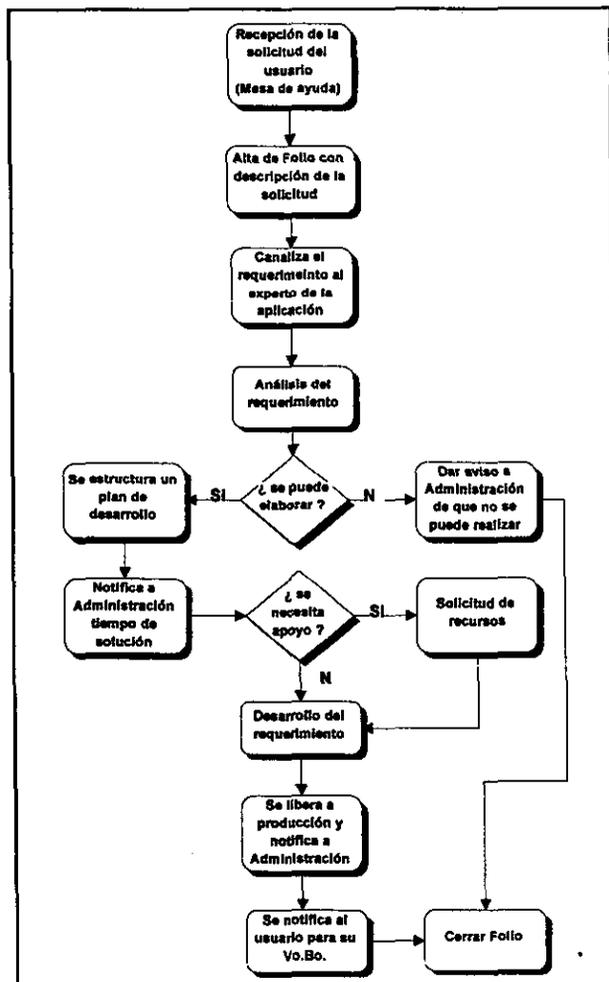


Figura 3.10 Folios levantados para solicitudes especiales.

Como plano general de consulta veremos a continuación los diagramas siguientes (figura 3.11 y figura 3.12) donde claramente se identifica con una cruz en la situación actual las dos nuevas consultas que se proponen por requerimiento del usuario, las cuales son por proceso con su historia de cancelaciones y por tiempo-solución que se le dio a cierto proceso.

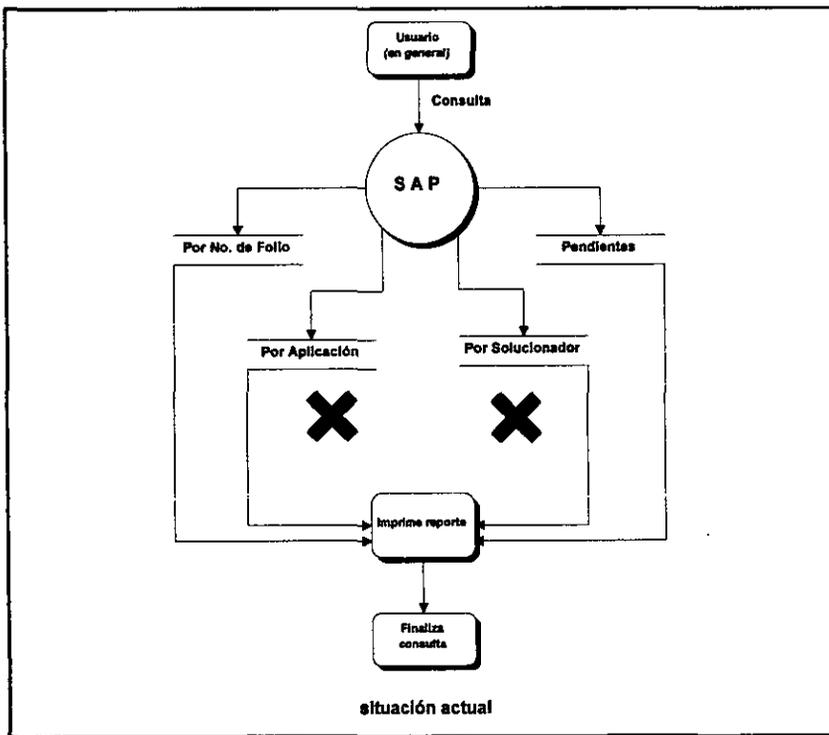
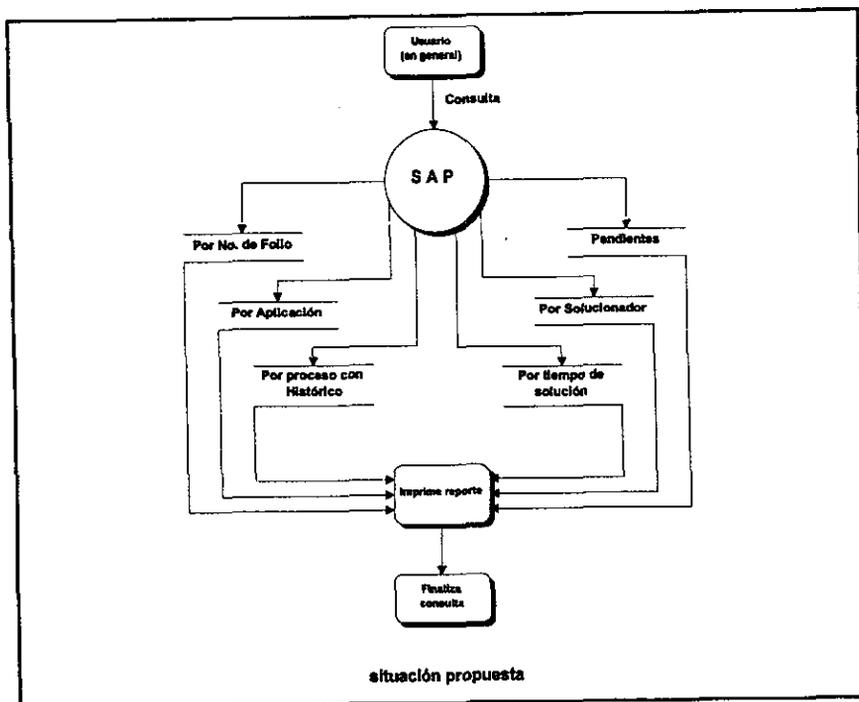


Figura 3.11 Diagrama de consultas (situación actual).



**Figura 3.12** Diagrama de consultas (situación propuesta).

Por último evaluaremos el sistema respondiendo a varios objetivos que nos planteamos. La aplicación que se rediseñó satisface por completo los requerimientos del usuario, como fueron la nueva creación de una base de datos para no perder consecutivos y no navegar en una variación de datos proporcionados, al mismo tiempo que evitar duplicidad de folios. Cabe aclarar que dentro de esta nueva base de datos nos avocamos a otro requerimiento, que fue incluir nuevos campos de captura para saber quién reportó el problema y a qué aplicación corresponde; se ven claramente en la comparación de diagramas estos dos nuevos campos, en los cuales queda bien especificado quien proporcionará los datos. Se creó un catálogo de causas como otro requisito, para identificar y unificar la distinta problemática que se tiene dentro de la producción por lote y en línea, tomando en cuenta una nueva lista de códigos, con la cual estén familiarizadas todas y cada una de las áreas de soporte.

Otros dos puntos o requerimientos fue la creación de dos nuevas consultas, la primera de un Histórico de Soluciones, la cual podrá proporcionar datos del porqué se canceló dicha aplicación o proceso de tal manera que si se presentara otra vez la misma problemática únicamente nos avocaríamos a repetir la solución y posteriormente aplicar la solución de fondo, para que no se vuelva a presentar. Otra consulta sería la relación tiempo-solución, que servirá sobre todo para sacar estadísticamente qué cancelaciones son más fáciles de atacar e ir disminuyendo esta serie de problemas; esta consulta nos dará un enfoque de qué problemas se llevan más tiempo y recursos, de tal manera que se pongan con luz roja, para tomarlos en cuenta en base a su grado de prioridad.

El último requerimiento, y que por cierto se dio como propuesta principal, fue el alta automática de folios para las cancelaciones por lote. Este requerimiento se propuso para estar a nivel con todos los problemas que se tienen durante el día, y no estar recapturando éstos por la simple razón de que no fue levantado el folio y sí atendido el problema. Para este punto se tuvo que cambiar totalmente el desarrollo del alta de folios, por lo que se diagramó en dos partes para ver detalladamente su procedimiento y eliminar cualquier duda que se tuviera y de cierta forma que la persona que hubiese reportado la cancelación no se viese presionado a levantar al mismo tiempo el folio y de esta manera seguir contando con una solución más rápida. Ayudará a la gente de Administración de problemas, ya que únicamente consultarán la lista de cancelaciones por la mañana, y como se dijo anteriormente no tendrán que repetir el procedimiento de levantamiento de folios.

Al operar el sistema de esta forma, será mucho más eficiente y precisa la información que se tenga en las distintas áreas relacionadas con control y soporte a la producción. También se tendrá mayor fluidez y rapidez de los datos proporcionados por las soluciones que se den a cada problema, de tal manera que el usuario se encuentre completamente satisfecho y desempeñe su trabajo con el menor número de reportes o problemas.

El sistema de alguna forma servirá para afinar cada una de las aplicaciones que se tienen dentro de la compañía. Con esto, el usuario final y la persona que desarrolló el sistema estarán de acuerdo en incorporar una retroalimentación y control adecuado al mismo sistema o a otras nuevas aplicaciones que se desarrollen en un futuro.

Después de haber realizado el presente estudio, llegamos a las siguientes conclusiones:

- Todas las empresas, independientemente de su tamaño e importancia, tienen la necesidad de sistematizar sus operaciones para que éstas se realicen con mayor fluidez y eficacia, lo que mejorará la situación tanto económica como administrativa de la compañía.
- Es conveniente que los sistemas sean implantados por analistas expertos, ya que las grandes empresas deben tener un departamento de sistemas, que esté subdividido en distintas áreas, para reducir el trabajo y especializarlo en sus distintas ramas. Esto es con el objeto de que el funcionamiento del sistema quede lo mejor posible.
- Para la realización del sistema es necesario conocer primero la situación actual en que se encuentra la empresa, cosa que se dio en este proyecto, y después, buscar la solución adecuada a ella, tomando en cuenta los requerimientos del usuario. Con esto se debe presentar un diseño básico, que será puesto a consideración de los supervisores de cada departamento o área. La presentación de dicho proyecto se hará para evitar un rediseño más al sistema definitivo, haciendo las correcciones necesarias en el primer diseño.
- La implantación de un sistema como éste puede hacer que una empresa con volumen de datos considerable, o con la información sobre ellos requerida, acceda a ellos con mucha rapidez y estos accesos sean mucho más sencillos.
- Una vez implantado el sistema, debe prepararse un programa de mantenimiento y control, para observar que las operaciones se realicen como fueron diseñadas y realizar las mejoras pertinentes para conservar el sistema vigente.

- A lo largo del proyecto se pudieron observar las ventajas que otorgará la utilización del sistema como tal, así como la gran importancia que tiene el realizar un control y soporte a las distintas aplicaciones que se tienen dentro de la compañía.
- Se utilizaron herramientas específicas para facilitar el diseño del sistema, valorando en gran medida su importancia en el desarrollo del proyecto.
- Por la experiencia adquirida a través del desarrollo del sistema, se puede decir que toda empresa o departamento de sistemas debe tener un área en específica que dé el control y soporte de la producción, ya que de esta manera se puede garantizar al usuario final un porcentaje alto de confiabilidad para el desempeño correcto de cada aplicación.
- Por lo general, en cualquier caso dado, existen muchos sistemas posibles que pueden ser diseñados para ajustarse a las necesidades, pudiendo diferir mucho en cuanto a su conveniencia y costo.
- En todas partes del sistema se debe poner énfasis en la necesidad de la información que se proporciona y qué propósito tiene. Deben reconocerse las limitaciones que se tienen durante el desarrollo del diseño y, sobre todo, el sistema propuesto debe ser totalmente claro y práctico.
- En gran parte, el diseño de sistemas consiste en una gran cantidad de trabajo pesado, el apego a las "reglas básicas", el conocimiento completo de los hechos y el mantenerse constantemente en el diseño hasta llegar a una solución adecuada.

- Después de haber desarrollado todo nuestro análisis y diseño de la forma estructurada, es bueno mencionar el porque no se utilizó un análisis orientado a objetos (AOO), el cual se encarga propiamente de los objetos y atributos, clases y miembros, y no se ocupa demasiado en el seguimiento del flujo de datos durante el análisis estructurado, causa principal por la cual el proyecto se realizó con el método formal estructurado; por esta razón y el respeto al producto que fue desarrollado desde un principio en la forma estructurada, se tomo en cuenta continuar con la misma metodología. Por último es bueno mencionar que para la modificación de altas automáticas, se tomo en gran parte el flujo de los datos, por lo tanto fue importante proponer el modelo usual de análisis y diseño estructurado, y no comenzar estableciendo el modelo de AOO.

- El diseño de un sistema toma tiempo, no debe ser apresurado. Se ensayan varias ideas, algunas se retienen, muchas son descartadas. El proceso es repetitivo y tentativo a la vez, se desarrollan y afinan una serie de operaciones. Hacia la meta deseada, desde lo que se requiere al principio, y desde luego a través de los varios pasos para su resultado lógico.

# ÍNDICE GRÁFICO

## CAPÍTULO I

<b>Figura I.1</b>	Niveles de atención para problemas recibidos	7
<b>Figura 1.2</b>	Atención de problemas reportados a Mesa de Ayuda	9
<b>Figura 1.3</b>	Atención de problemas reportados por Producción	11

## CAPÍTULO II

<b>Figura 2.1</b>	Elementos del sistema	18
<b>Figura 2.2</b>	Esquema con tipos de usuarios	19
<b>Figura 2.3</b>	Ciclo de vida clásico	21
<b>Figura 2.4</b>	Ingeniería de software	21
<b>Figura 2.5</b>	Notación básica de un diagrama de flujo de datos	23
<b>Figura 2.6</b>	El modelo en espiral	25
<b>Figura 2.7</b>	Ejemplos de procesos	27
<b>Figura 2.8</b>	Representación de un flujo	27
<b>Figura 2.9</b>	Representación de un DFD con sus respectivos flujos	27
<b>Figura 2.10</b>	Ejemplos de almacenes	28
<b>Figura 2.11</b>	Representación de un almacén implementado	28
<b>Figura 2.12</b>	Representación de una entidad externa	29
<b>Figura 2.13</b>	Representación completa de un diagrama de flujo de datos	29
<b>Figura 2.14</b>	Representación completa de un diagrama entidad-relación	32

<b>Figura 2.15</b>	Representación de un tipo de objeto	34
<b>Figura 2.16</b>	Representación gráfica de relaciones	34
<b>Figura 2.17</b>	Múltiples relaciones entre objetos	35
<b>Figura 2.18</b>	Múltiples relaciones entre múltiples objetos	36
<b>Figura 2.19</b>	Representación de relaciones uno a uno	36
<b>Figura 2.20</b>	Representación gráfica de indicadores asociativos de objetos	37
<b>Figura 2.21</b>	Representación gráfica de indicadores de Supertipo/Subtipo	38
<b>Figura 2.22</b>	Representación de un flujo de transformación	41
<b>Figura 2.23</b>	Representación de un flujo de transacción	42
<b>Figura 2.24</b>	Diseño orientado al flujo de datos	43

### **CAPÍTULO III**

<b>Figura 3.1</b>	Alta de folios en línea y por lote	67
<b>Figura 3.2</b>	Alta de folios para planes especiales	68
<b>Figura 3.3</b>	Consulta de folios	69
<b>Figura 3.4</b>	Diagramas de flujo de datos de los problemas en línea	73
<b>Figura 3.5</b>	Diagramas de flujo de datos de los problemas por lote	74
<b>Figura 3.6</b>	Diagramas de flujo de datos de solicitudes especiales	75
<b>Figura 3.7</b>	Folios levantados para problemas en línea	77
<b>Figura 3.8</b>	Folios levantados para problemas por lote (situación actual)	78
<b>Figura 3.9</b>	Folios levantados para problemas por lote (situación propuesta)	79
<b>Figura 3.10</b>	Folios levantados para solicitudes especiales	80
<b>Figura 3.11</b>	Diagrama de consultas (situación actual)	81
<b>Figura 3.12</b>	Diagrama de consultas (situación propuesta)	82

**BOCCHINO, A. William.**

Sistemas de Información para la Administración.

Trillas, 1992.

**CARBAJAL, Díaz del Campo Eduardo.**

Manual de Organización, Soporte a Sistemas de Información.

México, 1996.

**Centro Nacional de Productividad.**

Manual de Diseño de Sistemas.

México, 1974.

**JOHANSEN, Bertoglio Oscar.**

Introducción a la Teoría general de Sistemas.

Limusa, 1985.

**JONES, W. Gregory.**

Software Engineering.

John Wiley and Sons, 1990.

**METTEHIES, Leslie.**

Recursos humanos en el diseño de Sistemas Administrativos.

Limusa, 1982.

**PIATTINI, Velthuis Mario G.**

Elementos y Herramientas en el desarrollo de Sistemas de Información.

Addison-Wesley Iberoamericana, 1995.

**PRESSMAN, S. Roger.**

Ingeniería del Software, un enfoque práctico.

Mc Graw Hill, 1995.

**QUEZADA, Patricia.**

Reglamento interno de Mesa de Ayuda.

México, 1996.

**SOMMERVILLE, Ian.**

Ingeniería de Software.

Addison-Wesley Iberoamericana, 1988.

**SUAREZ, Alvarado Eduardo.**

Manual de Procedimientos, Soporte a Sistemas de Información.

México, 1996.

**TSAI, Y. H. Alice.**

Sistemas de Base de Datos: Administración y Uso.

Prentice Hall, 1990.

**YOURDON, Edward.**

Modern Structured Analysis.

Prentice Hall, 1989.