**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**

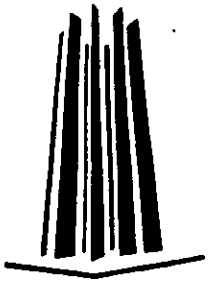
ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
"CAMPUS ARAGON"

# EL PAPEL DE UN SISTEMA ADMINISTRADOR DE APLICACIONES COMO ESTRATEGIA DE ESTANDARIZACION E INTEGRACION DE SISTEMAS

## T E S I S

QUE PARA OBTENER EL TITULO DE
**INGENIERO EN COMPUTACION**
P R E S E N T A :

ARTURO PONCE MORENO



ASESOR: ING. ROBERTO BLANCO BAUTISTA

MÉXICO 1998

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ARAGON

TEMA DE TESIS:

# EL PAPEL DE UN SISTEMA ADMINISTRADOR DE APLICACIONES COMO ESTRATEGIA DE ESTANDARIZACION E INTEGRACION DE SISTEMAS

ING. ARTURO PONCE MORENO (1985-1989)

# INTRODUCCION

Actualmente las organizaciones tienen a su alcance el creciente uso de la computación; Los modernos sistemas de comunicación, las cada vez más poderosas computadoras, las novedosas y accesibles tecnologías de información, la conectividad entre redes de diferentes plataformas (mainframes, minis, micros); ha motivado que las organizaciones deban automatizar cada vez mas aplicaciones de misión critica, almacenando en sus equipos un gran volumen de información confidencial, este hecho pone en riesgo la seguridad operacional de las organizaciones sobre todo cuando esta depende del buen funcionamiento de sus aplicaciones, ya que una vez almacenada la información queda expuesta ha ser manipulada por cualquier persona que tenga acceso ha esta, o que se logre violar la seguridad del sistema.

El presente trabajo pretende proporcionar una solución integral a la problemática de estandarización de presentación y manejo de los sistemas que se tienen en ejecución dentro de un equipo de computo o una red, hacer notar los beneficios que se obtienen si se cuenta con un sistema de seguridad y control de las aplicaciones, determinando el entorno de trabajo que establezca una serie de sencillas reglas a seguir y ofreciendo un medio ambiente de trabajo sencillo y amigable, analizar los aspectos fundamentales a considerar para lograr sistema administrador de aplicaciones eficiente y productivo. Esta solución permite obtener una mejor organización y un mejor control de todas nuestras aplicaciones. Ver figura 1.


**Fig. 1.** Control de Nuestras Aplicaciones.

Todas las personas que intervienen en un proyecto de desarrollo de aplicaciones deben invertir tiempo para resolver problemas que en realidad no se derivan de los requerimientos propios de la aplicación, si no que son comunes para cualquier tipo de sistema. Por ejemplo: control de acceso, autorización y prohibiciones de ejecución a usuarios, documentación en línea, integridad de Base de Datos y muchos otros tópicos que irremediablemente se tienen que resolver para que una aplicación funcione adecuadamente. Dependiendo entonces de la habilidad y experiencia de los analistas que integran un proyecto, será el nivel de la solución que se dé a dichos problemas, pero nunca será la misma para diferentes sistemas desarrollados por diferentes personas.

Los problemas más comunes que se presentan en una instalación, cuando no se cuenta con Sistema Administrador de Aplicaciones, son que cada una de ellas normalmente resuelve de forma diferente situaciones que son comunes, por ejemplo: control de acceso, autorización y prohibiciones de ejecución a usuarios, documentación en línea, integridad de base de datos, presentación de menús, y muchos otros.

La experiencia por mas de diez años en el área de desarrollo de sistemas, y por haber observado por todo este tiempo la forma tan poco orientada acerca de la importancia del control y la seguridad de la información, me han motivado el desarrollo de este trabajo. Es evidente que se ha hablado mucho acerca de seguridad, pero de seguridad física, y en cuanto a la seguridad lógica la mayor parte se ha dejado en manos de los fabricantes de equipos, de sistemas operativos y de bases de datos, además nos hemos encontrado que existe bastante información documental acerca de la administración y de la seguridad lógica. Esto nos ha motivado aun más para desarrollar este trabajo.

El concepto de Sistema Administrador de Aplicaciones intenta dar una solución estándar a ese tipo de problemas comunes que tienen que resolverse para cualquier aplicación,

partiendo de la premisa de invertir recursos de diseño y desarrollo, una sola vez y dando como resultado un producto de uso generalizado, que brinde una plataforma de desarrollo ordenado y de presentación homogénea, para todas las aplicaciones. Este último punto en el ámbito de usuario es el más importante ya que el modo y el método de acceso son únicos facilitando con ello la capacitación de los usuarios en el manejo de una o varias aplicaciones.

Un Sistema Administrador de Aplicaciones no es solamente una aplicación que se encarga de administrar otras aplicaciones, si no que es una serie de herramientas estándares que forman la base del desarrollo de aplicaciones y que permiten encausar los esfuerzos del equipo de trabajo a dar soluciones a los problemas propios de los usuarios y no a los problemas técnicos de estructura de los sistemas.

Actualmente los administradores de aplicaciones se están presentado en las diferentes plataformas de computadoras. Estos administradores o manejadores iniciaron cumpliendo las necesidades inmediatas de los usuarios de un ambiente común de trabajo, pero ahora son más completos, llegando a ser muy complejos. Esta complejidad esta dada por los requerimientos que, día con día, el usuario adopta en la necesidad tener y ofrecer un tiempo de respuesta muchos más rápido que lo que le ofrecían las aplicaciones fuera de un administrador o manejador.

En las empresas que se inicia en el ámbito de las computadoras como herramienta de trabajo para sus tareas administrativas de control o manufactura, etc. (ver figura 2); es muy común que las áreas correspondientes sean dirigidas por personas o profesionistas que no tienen o tienen muy poca experiencia. Normalmente, estas personas, con el fin de dar los mejores resultados posibles en su nuevo cargo, empiezan por averiguar quienes le pueden ofrecer software de acuerdo a sus necesidades. Es muy probable, que muchas

casas de software les ofrezcan sus servicios para venderles aplicaciones ya existentes o desarrollarles aplicaciones de acuerdo a sus necesidades. Pero hay muy pocas casas de software que cuenten con administrador o manejador de aplicaciones. Lo ideal sería que en el inicio de una empresa en el ámbito de las computadoras contara con un administrador de aplicaciones, ya que esto le permitiría obtener mejores resultados que los obtenidos por empresas que cuentan con aplicaciones que trabajan en forma independiente y cada una en su propio ambiente.

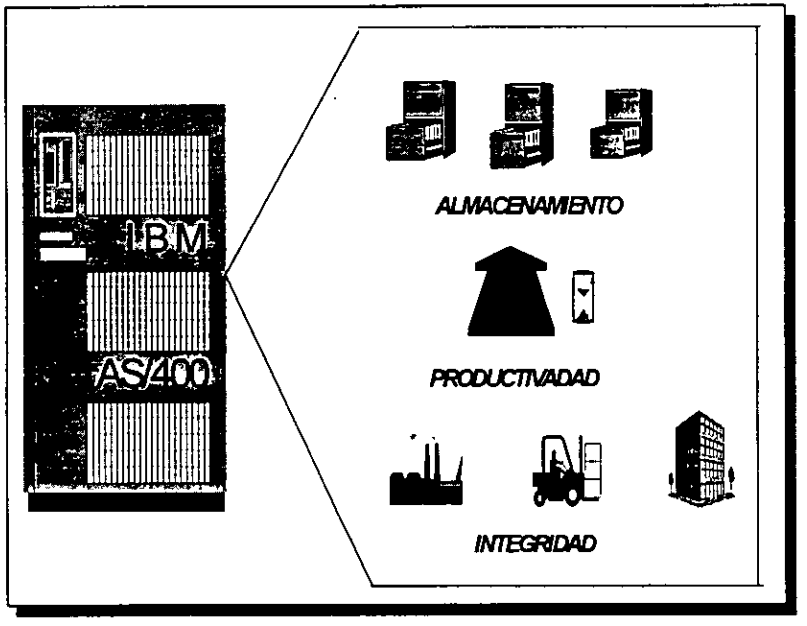



**Fig. 2.** Computadora: Herramienta de Tareas Administrativas y de Producción

En un esquema tradicional de trabajo en una computadora existe redundancia de trabajo y de información, tanto para el usuario que explota una o varias aplicaciones como para el responsable del buen funcionamiento de estas aplicaciones. Para el usuario implica memorizar los diferentes ambientes y modos de acceso, las diferentes contraseñas, teclas de función usadas, modo de obtención de reportes, niveles de menú para llegar al

programa final deseado, restricciones de programas, diferentes pantallas de captura, etc. Para el responsable del funcionamiento de la aplicación (conocido como "líder de proyecto") implica parametrización y mantenimiento para cada una de las aplicaciones en: control de acceso, autoridad a los usuarios, seguridad, definición de formatos de impresión, definición de presentación.

Toda organización o persona que automatiza sus operaciones, debe implantar un **sistema de control y seguridad total, conjugando tanto aspectos administrativos como técnicos y de procedimiento**. Ya que este le brindara un clima de tranquilidad en el resguardo del recurso información, vital para sus operaciones automatizadas. Ya que actualmente en México no existe una cultura de *seguridad informática*, la cual es vital que se desarrolle ya que nuestro país esta entrando de lleno a la globalización junto con otros países que cuentan con un enfoque más amplio acerca de la *seguridad informática*.

Para entender los alcances que puede tener un sistema de este tipo, en el primer capítulo: **Antecedentes,** se muestra una perspectiva de los diferentes tipos de Sistemas Administradores de Aplicaciones, sus principales características, sus objetivos, sus componentes, los diferentes nombres con los que han dado a conocer en el mercado, así como las marcas de computadoras en los que corren.

Una de las principales características de un Sistema Administrador de Aplicaciones es la **Seguridad en los sistemas**, por lo que en el segundo capítulo se hace hincapié en los principales aspectos de seguridad que un sistema debe tener, para mantener la integridad de las aplicaciones que lo componen.

Tratar de seguir una serie de reglas que le permitan al diseñador de sistemas que sus aplicaciones a desarrollar sean entendibles y fácilmente usadas por el usuario final, en el capítulo tres, se describe la importancia de los **Estándares**, así como las principales partes de un sistema donde se deben aplicar para lograr un máximo de legibilidad y de orden de los sistemas.

Aun cuando no existe un procedimiento adecuado que nos permita mantener todas nuestras aplicaciones para verlas como un solo sistema, en el capítulo 4, se describen algunos principios de **Integración de sistemas**, que pueden ayudar al diseñador de sistema a establecer una ruta homogénea para todas las nuestras aplicaciones, que le permitan al usuario final usar de manera productiva y sencilla todos los recursos disponibles, además de administrarlos en forma adecuada.

La mejor manera de demostrar cada uno de los temas comprendidos en este documento, en el capítulo 5, se muestra un ejemplo del uso y la **Aplicación de un Sistema Administrador de Aplicaciones**, el cual se puede seguirse como modelo con los requerimientos mínimos. Este modelo fue realizado a partir de cero, es decir, que no pertenece a ninguna compañía o institución. Sólo es un diseño particular para el entendimiento de este documento. Este capítulo servirá como referencia de los principales temas tratados a los largo de los capítulos anteriores.

# INDICE

A mis maestros:

A mis maestros, que durante mi vida de estudiante, me proporcionaron todos los conocimientos necesarios para tener todos los elementos suficientes para ir avanzando en cada uno de los niveles escolares, hasta llegar al nivel profesional, que me permitirá seguir adelante confiando en que cuento con una buena base para lograr cualquier meta que me proponga en la vida.

A mi esposa:

A mi esposa, por su amor como pareja, por su apoyo en los momentos más difíciles y por su admirable paciencia mientras yo le quitaba algo de su tiempo para finalizar este documento.

A mis hermanos:

A mis hermanos, por que siempre compartieron conmigo los ratos tristes y los ratos felices, por el sacrificio que hicieron al dejar de estudiar para poder trabajar y aportar al sustento de nuestra familia, y por el apoyo que me dieron para salir adelante.

A mi padre:

A mi padre, por su cariño y su atención, por sus grandes consejos para triunfar en la vida como ser humano, por confiar en mi y sobre todo por su gran apoyo en mi formación como profesionista.

A mi madre:

Dedico esta obra a mi madre en especial (que en paz descanse), por el amor y el cuidado que me dio y por el esfuerzo que realizo para que nunca me faltara nada. Sé que donde quiera que se encuentre, seguirá compartiendo conmigo mis logros, seguirá enviándome su bendición y sobre todo seguirá estando presente en mi corazón.

# CAPITULO I

# ANTECEDENTES

Dentro de la gran problemática de aplicaciones que se ejecutan dentro de un mismo equipo de computo, está la falta de control de acceso, la diversidad de ambientes de ejecución y la variedad de presentación de ambientes interactivos y reportes en papel. Como consecuencia de esta problemática, el analista de sistemas tiene que invertir un tiempo considerable en la capacitación del uso de las diferentes aplicaciones a todos sus usuarios.

En una empresa donde la rotación de personal dedicado a la programación de las aplicaciones es considerable, resulta una gran inversión de tiempo y dinero, cuando en las aplicaciones que se están desarrollando o se les esta dando mantenimiento, no tienen una organización adecuada y mucho más importante, cuando no tienen predefinido una estandarización.

Por todo ello, en algunas empresas, algunos diseñadores de sistemas se han tomado la tarea de elaborar Sistemas Administradores de Aplicaciones tal es el caso de Grupo TEA y de la Compañía Nestlé. La intensión de este capítulo es dar a conocer cierto tipo de administradores de aplicaciones existentes en el mercado o propios de algunas empresas, sus características, medio ambiente, su estructura, etc.; así como la experiencia que han tenido estas empresas en el uso de los mismos. Por si solos, cada unos de los Administradores de Aplicaciones aquí mencionados, son un ejemplo claro. Se notará que sus objetivos son comunes y que únicamente varia la forma en que se obtienen y la forma en que se desarrollan.

*En el capítulo "5. 1. Objetivos." se describen los objetivos básicos para cualquier Administrador de Aplicaciones independientemente de la plataforma en que se desarrollen.*

**1. 1. Manejador de Aplicaciones de Grupo TEA.**

Grupo TEA es una empresa de servicio de Informática dedicada a la venta de equipo de computo así como al diseño de software; uno de sus últimos diseños es un software modular denominado ADAM.

**1. 1. 2. El concepto de ADAM.**

El nuevo concepto de Aplicaciones Administrativas de GRUPO TEA para el sistema AS/400 de I.B.M. se denomina ADAM. ADAM es una familia completa de Aplicaciones Administrativas que cubren las necesidades de administración y control de empresas Industriales, Comerciales, Bancarias y de Servicio.

ADAM es una familia de aplicaciones estandarizadas, modulares y totalmente integradas, que incorporan elementos innovadores de diseño a las más ambiciosas tecnologías de aplicación de Sistemas de administración. Los módulos de ADAM están estructurados con todas las funciones y todo el poder, para constituirse en herramientas administrativas de alto nivel, al mismo tiempo que brindan al usuario todas las facilidades para obtener toda la información, memorizando un solo dato; su contraseña.

El concepto ADAM consiste en una estructura basándose en plataformas en la que la adaptabilidad, la estandarización y el crecimiento gradual se hacen posibles (ver figura 1.1.). En el nivel inferior de la estructura ADAM esta el concepto de Administrador de Aplicaciones, que constituye una base estándar para el desarrollo y operación de aplicaciones. El Manejador de Aplicaciones ofrece un acceso único al usuario a todas las aplicaciones instaladas en una computadora (AS/400), sea de la familia ADAM o no, al tiempo que da las capacidades para facilitar el uso del sistema operativo (OS/400). Sobre el Manejador de Aplicaciones se basa la familia de soluciones universales ADAM que comprende 31 módulos totalmente integrados, orientados a la administración de los recursos financieros, humanos y materiales de toda organización. Los módulos ADAM, diseñados para obtener todo el provecho de la tecnología AS/400, están preparados para adaptarse a las necesidades de cualquier organización, sin que esto implique modificar la programación. A través de las facilidades del Manejador de Aplicaciones y de la apertura de los módulos ADAM, es fácil desarrollar e integrar soluciones especializadas para toda la industria. Las soluciones especializadas pueden integrarse con facilidad, tanto con el Manejador de Aplicaciones, como los módulos de la familia ADAM, utilizando un acceso común uniforme y compartiendo e intercambiando información sin limitaciones.

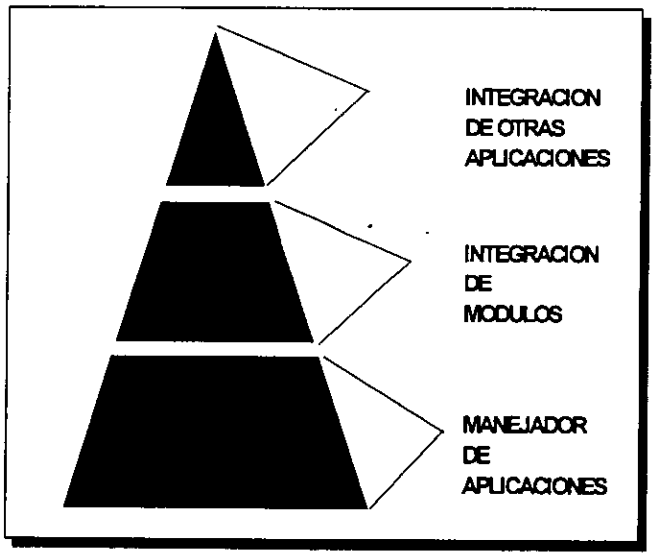



**Fig. 1.1.** El Concepto ADAM

### 1. 1. 3. ADAM, AS/400 Y S.A.A.

La utilidad de las computadoras radica en su habilidad para ser accesibles las técnicas que mejoran la manera en que el hombre realiza su trabajo; es por lo que en ADAM, con su concepto innovador de software de aplicación, se ha cuidado en mantener y expandir todas las facilidades y poder que han hecho populares y han dado prestigio internacional a las aplicaciones de software para minicomputadoras y microcomputadoras.

A través de la asociación de los conceptos de ADAM, con las técnicas modernas de Ingeniería de Software, con las tecnologías de punta implementadas en la arquitectura del Sistema AS/400 y con los estándares y filosofía de S.A.A. (System Applications Architecture) de I.B.M., se ha logrado producir un software que reúne los requerimientos de flexibilidad y crecimiento de los clientes de Grupo TEA, porque permite a los usuarios aprovechar las funciones de los módulos de ADAM en la medida que son necesitados. ADAM hace un óptimo aprovechamiento de los conceptos del AS/400, que constituye uno de los avances tecnológicos más significativos de nuestros tiempos porque incorporan las tecnologías más recientes como son:

- El Manejador de Base de Datos Relacional integrando al hardware y al sistema operativo.
- El Lenguaje de Datos S.Q.L.

AS/400 y ADAM incorporan los nuevos estándares detallados de la Arquitectura de Aplicación de Sistemas (S.A.A.), proporcionando al usuario final un ambiente de trabajo

único y transparente entre el sistema operativo OS/400, y las sesiones de trabajo de ADAM. S.A.A. se compone de tres elementos significativos que son:

- El acceso común de usuario.
- La interfaz de programación común.
- El soporte común de comunicaciones.

Estos componentes de S.A.A. regulan las interfaces de software, protocolos y convenciones para la interacción hombre-màquina con aplicaciones tales como ADAM, las soluciones especializadas de cada industria, a los servicios del sistema operativo; los mecanismos de comunicación que interconectan los sistemas, y las interfaces de programación para el desarrollo de programas.

### 1. 1. 4. Módulos ADAM.

Los módulos de aplicaciones universales que integran ADAM, son soluciones generales que permiten el máximo de flexibilidad y ofrecen resultados sobresalientes en términos de tiempo y costo. Ver figura 1.2.


**Fig. 1.2.** Módulos de las Aplicaciones que Integran ADAM.

Algunas de las características más sobresalientes de los módulos de ADAM son:

- *Integridad*: La base de datos única para todos los módulos de ADAM está diseñada con el máximo de integración, tomando en cuenta las formas de normalización aplicadas a las relaciones, lo que permite evitar las posibles anomalías al actualizar los datos, así como eliminar la redundancia y aumentar la confiabilidad.

- *Interactividad*: Las operaciones de trabajo de ADAM están estructuradas de tal manera que el usuario pueda realizar todas las actualizaciones y consultas en línea

y en una sola sesión.

- *Adaptabilidad y parametrizaciòn*: Los módulos de ADAM son totalmente adaptables a los requerimientos de las empresas usuarias, a través de parámetros que se registran en tablas relaciónales, sin necesidad de modificar internamente los programas, lo que permite al usuario final configurar los módulos para cubrir sus necesidades actuales y futuras.

- *Manejo de múltiples compañías*: Los módulos de ADAM permiten manejar múltiples compañías en función de la estructura de cada grupo.

- *Alta retroalimentaciòn*: El usuario de ADAM requiere conocer únicamente su contraseña para extraer toda la información del sistema, ya que ADAM siempre proporcionará o dará acceso a la lista de valores posibles para cualquier dato requerido, incrementando la utilidad de las aplicaciones y la productividad del usuario.

- *Facilidad de operación y ayudas en líneas*: Las sesiones de trabajo están totalmente estandarizadas y diseñadas de manera que el usuario pueda utilizarlas con facilidad, contando en todo momento con ayudas en línea. La facilidad de ayuda provee información contextual o extendida por cada pantalla o sesión de trabajo de ADAM.

- *Acceso a información selectiva*: El usuario de ADAM obtiene sólo la información que necesita, ya que el diseño de la base de datos relacional de ADAM permite aprovechar de manera integral los datos almacenados, ofreciendo la posibilidad de consultar interactivamente los datos o imprimir un reporte previa selección de su contenido.

- *Total integración con AS/400*: ADAM está desarrollado tomando en consideración la totalidad de recursos que ofrece el sistema, logrando con esto una familia de aplicaciones verdaderamente nativas de AS/400. Las figuras 1.3. y 1.4. son ejemplos del mantenimiento de la instalación de módulos que integran ADAM.

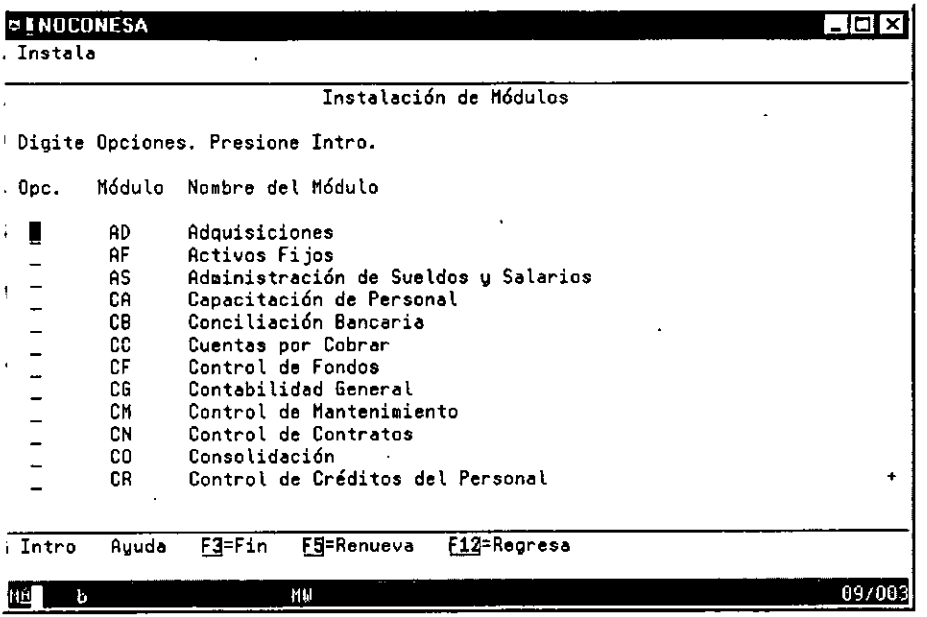



**Fig. 1.3.** Pantalla Mantenimiento a Instalación de Módulos en ADAM.

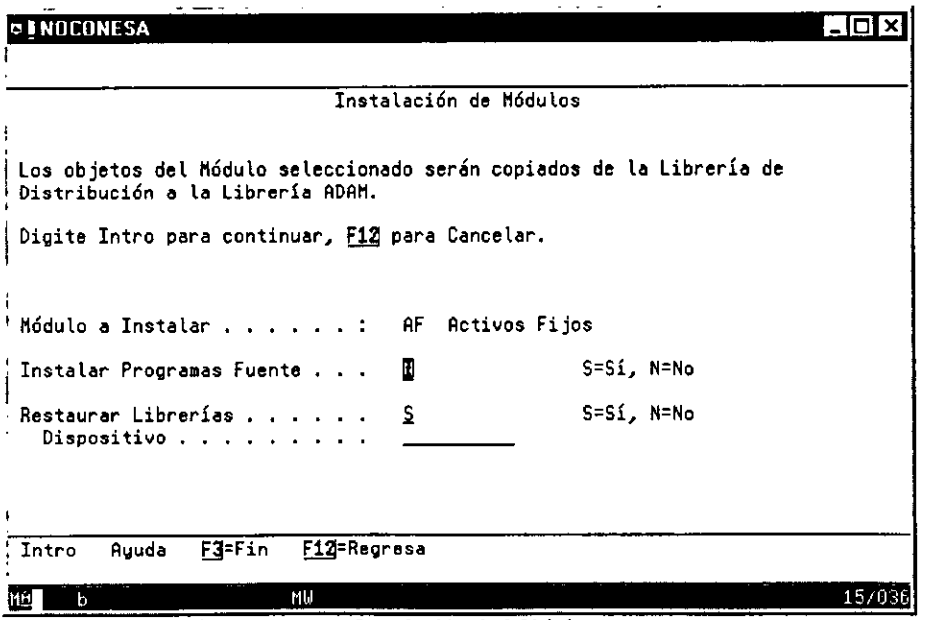



**Fig. 1.4.** Pantalla de captura para Instalación de Módulos.

### 1. 1. 5. Modulo MA de ADAM.

El módulo Manejador de Aplicaciones (MA) de ADAM, es la base estándar para el desarrollo y operación de las Aplicaciones que permite:

- Facilidad de uso y acceso a la información.
- Máxima seguridad de datos.
- Crecimiento gradual.
- Estandarización.

El Manejador de Aplicaciones de ADAM, controla la ejecución de sesiones de trabajo en un solo nivel, evitando la navegación a través de menús; tiene la capacidad para manejar múltiples tipos de llamadas tanto de operaciones de ADAM, como de comandos del sistema operativo OS/400.

Entre sus principales funciones se encuentra la del manejo de seguridad que, a través de la definición del perfil de usuario, se encarga de regular el acceso a las operaciones o sesiones de trabajo y las bases de datos de la compañía a las que el usuario tiene autorización. Como Manejador de aplicaciones permite configurar los módulos, funciones y operaciones bajo su control; facilita la integración de los módulos, funciones y operaciones con soluciones especializadas de la empresa, así como la instalación de módulos de aplicación de usuario final de ADAM. Adicionalmente, proporciona facilidades para el acceso a servicios de administración de base de datos de los módulos, tales como el respaldo y restauración de archivos; la creación, inicialización y copia de información, etc.; así como servicios de exportación e importación de datos en formatos especializados. En la figura 1.5. se muestran las principales operaciones de

mantenimiento y control del módulo MA de ADAM.

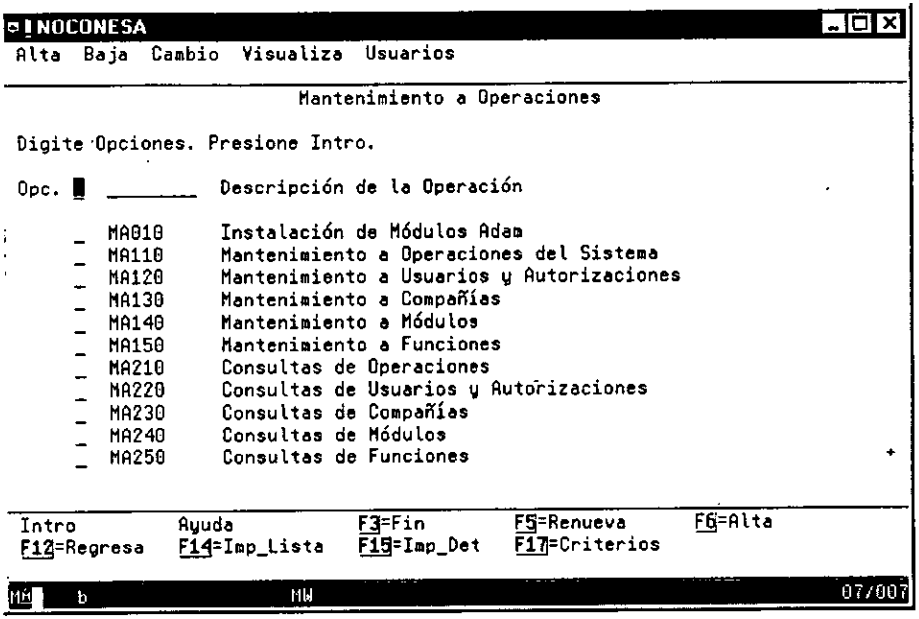



**Fig. 1.5.** Principales operaciones de mantenimiento del módulo MA de ADAM.

### 1. 1. 6. Estandarización de ADAM.

En las aplicaciones actuales, el seguimiento de estándares detallados es crítico para garantizar el óptimo aprovechamiento del Sistema a través de su comprensión tanto por usuarios no especializados, como por especialistas en Sistemas.

Algunos de los elementos de estandarización han sido suministrados precisamente por

los que I.B.M. ha desarrollado e integrado a la Arquitectura de Sistemas de Aplicación (S.A.A.) y han sido adoptados en el desarrollo de ADAM para darle consistencia a la aplicación con otras desarrolladas por I.B.M. u otros proveedores. Algunos de los aspectos de estandarización que se han utilizado en ADAM son:

| **CONCEPTO** | **ESTANDAR** |
|---|---|
| Diseño de pantallas | SAA |
| Técnicas de interacción | SAA |
| Mecanismo de dialogo y ayuda | SAA |
| Uso de teclas de función | SAA |
| Estructura de programación | ADAM |
| Nomenclatura de archivos y campos | ADAM |
| Nomenclatura de programas y objetos | ADAM |
| Formatos de impresión de reportes | ADAM |
| Utilización de colores | SAA |
| Nomenclatura de librerías | ADAM |
| Uso de variables de trabajo e indicadores | ADAM |

Las partes importantes de un Sistema Administrador de Aplicaciones que se recomienda que tenga una estandarización bien definida se describen el capitulo 5. Ver tema "5. 4. Estándares."

## 1. 2. AMS/400 (Application Manager System/400) de Nestlé.

Debido a la introducción del sistema AS/400 en oficinas centrales de la Compañía Nestlé México y derivado de la imposibilidad de conversión directa de programas desde 4381, fue necesario implantar un plan para el rediseño, programación e instalación de nuevas aplicaciones centrales que funcionarán aprovechando al máximo las características y filosofía del nuevo equipo. Las aplicaciones existentes en 4381 mostraban cierto grado de obsolencia y su conversión al AS/400, según pruebas realizadas, costaría un esfuerzo considerable, el cual daría como resultado tener sistemas que no aprovecharían las facilidades y características del nuevo equipo.

La Compañía Nestlé S.A. de C.V., ha diseñado un Sistema Administrador de Aplicaciones denominado: AMS/400 (Application Manager System). Este Sistema Administrador de Aplicaciones fue diseñado desde 1990, para satisfacer la necesidad de integración y estandarización de las Aplicaciones que hasta entonces, cada una de ellas se ejecutaba bajo su propio ambiente de trabajo (ver figura 1.3.). Los objetivos para los bajo los cuales fue diseñado el AMS/400 los siguientes:

- Resolver problemas comunes de operación, administración y control en todas las aplicaciones.
- Proveer una interface de usuario estándar para todas las aplicaciones.
- Controlar eficientemente un ambiente multi-compañías/ multi-tareas/multi-idiomas/ multi-usuarios.
- Suministra herramientas generales para lograr contar con aplicaciones de usuario completamente autónomas y sin tareas técnicas de operación.
- Alcanzar un alto grado de parametrización.

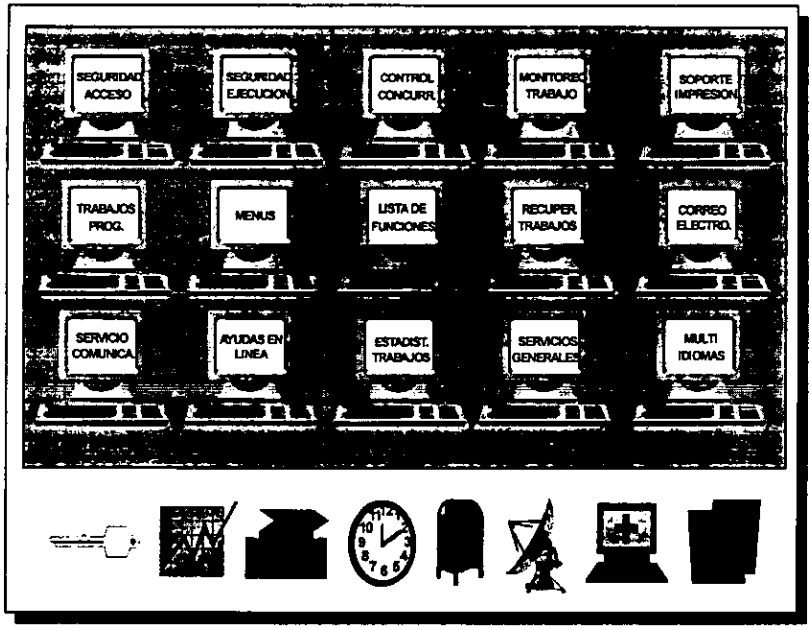



**Fig.1.3.** Medio Ambiente de AMS/400.

### 1. 2. 1. Configuración de AMS/400.

El Sistema Administrador de Aplicaciones AMS/400 permite las siguientes funciones de configuración:

- Funciones.
    - Status (activa o suspendida).
    - Tipo de autorización (especifica o libre).

- Programas a ejecutar.
- Código de reidentificación.
- Código de control de ejecuciones concurrentes.
- Nivel de control de concurrencias.
- Código de recuperación requerida.
- Código de control de emisión de reportes.
- Código de servicio generales.

- Menús.
  - Estilo (una o dos columnas).
  - Comentarios a incluir por idioma.
  - Funciones a ejecutar.
- Concurrencias de ejecución prohibidas.
- Parámetros de impresión reportes.
  - Descripción de programa emisor.
  - Parámetros a diferentes niveles de detalle.
  - Uso de constantes o palabras claves.
  - Definición de atributos de reporte.
- Textos de ayuda en línea.
  - Soporte multi-idomas.
  - Por función formatos de pantalla.
  - Definición de atributos de despliegue de pantalla.
- Áreas/Unidades.

- Perfiles de Usuario.
    - Tipo de Operador (normal, administrador).
    - Área/unidad.
    - Parámetros iniciales de sesión
        - Localidad inicial.
        - Menú inicial.
    - Autorización a multi-sesión.
    - Autorización de ambientes de trabajo.
    - Nivel de uso de listas de funciones.
    - Autorización de correo electrónico.
    - Autorización ejecución por número de función.
    - Autorización de ejecución de funciones libres.
    - Idioma inicial a utilizar.
    - Lista de terminales autorizadas.
- Menús autorizados.
    - Hasta 50 menús por usuario normal.
- Funciones autorizadas para ejecución.
    - Funciones normales de asistencia.
    - Por localidad.
    - Lista de subfunciones prohibidas (opcional).
- Información adicional para correo electrónico.
- Información de usuarios grupales.

El AMS/400 se crea por lo tanto como una solución integral de las aplicaciones explotadas en una gran Red de equipos AS/400 de I.B.M. (52 equipos) con que cuenta la Compañía Nestlé S. A de C. V. Ver figura 1.4.

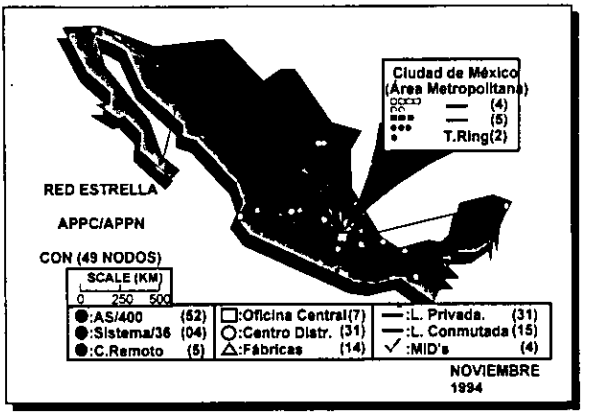



**Fig.1.4.** Nestlé - Red de Comunicaciones.

### 1. 2. 2. Personalización de ambiente de ejecución para cada usuario.

AMS/400 proporciona una serie de funciones que permiten ingresar y dar mantenimiento a diferentes parámetros, que en su conjunto definen y delimitan las características y modalidades de acceso y ejecución de cada usuario para cada aplicación en cada localidad. Ver figura 1.5.


**Fig. 1.5.** Personalización de Usuarios.

Las características principales que se pueden personalizar para cada usuario mediante mantenimientos son:

- Tipo de Operador: AMS/400 soporta dos tipos de operadores: 1 = operador de tipo usuario. 2 = operador tipo administrador. El tipo de operador Usuario es un operador normal, al cual se le deberán dar autorización sobre los Menús y Funciones que puede ejecutar. El tipo de Administrador es un operador que no tiene limitación alguna sobre el uso de las Aplicaciones, Funciones y Menús existentes. El tipo administrador debe usarse para pruebas del personal de Informática y se debe vigilar muy cerca su asignación.

- Modalidad de Ingreso: Define si el usuario sólo esta autorizado a usar el ambiente de menús o el ambiente de menús y de lista de funciones.

- Usuario de firma múltiple: Define si el usuario puede estar firmado en una terminal solamente o en varias al mismo tiempo.

- Limite de Espera: Define la cantidad de minutos que AMS/400 esperara una acción de operación desde los dos ambientes, antes de dar de baja la sesión en forma automática. También se puede especificar una espera permanente (sin baja automática).

- Uso de función: Define si el usuario estará autorizado a ejecutar funciones digitando directamente el número de función deseado, o si debe restringirse a la ejecución de tareas mediante opciones de Menú.

- Localidad inicial: Parámetros de Compañía/Localidad y Sublocalidad para el inicio de la sesión del usuario. El mismo usuario, una vez en sesión, podrá cambiar de localidad tantas veces como sea necesario.

- Cola de salida: Cola de impresión por defecto asignada al usuario.

- Cola principal: Parámetros que define cual de las colas disponibles para el usuario tendrá prioridad de uso como valor predeterminado al momento de la sesión. Las posibilidades son que use la cola asignada al dispositivo como predeterminado. En un ambiente de múltiples impresoras asignadas a terminales, es posible que los reportes que se generen "persigan" al usuario codificándole al mismo la clave de cola principal.

- Descripción de trabajo: Descripción de trabajo asignado al usuario para la sumisión de trabajo batch.

- Menú inicial: Parámetro opcional que define le menú que AMS/400 presentará cuando inicie la sesión para el usuario.

- Habilitación de correo: Define si el usuario participará en el módulo de Correo Electrónico que viene integrado con el AMS/400.

- Permiso de búsqueda: Define si el usuario podrá armar lista de funciones, eligiendo si se incluyen todas las funciones existentes o sólo las autorizadas o si el sistema limitará el armado de listas a trabajar o únicamente con las funciones autorizadas para ejecución.

- Permiso de uso de funciones libres: Define la modalidad de autorización general que tendrá el usuario antes las funciones de tipo "libre" o que no requieren autorización de ejecución expresa.

- Lista de menús: Define hasta 50 menús habilitados para uso del usuario con un potencial máximo de 1,500 funciones autorizadas. Esta lista de menús estará disponible durante la sesión y el usuario puede cambiarse a cualquiera de sus menús mediante la navegación directa o búsqueda visual desde una lista. Lo anterior es aplicable únicamente a operadores de tipo usuario, ya que los administradores pueden ejecutar cualquier menú de los existentes. (Máximo: 99,999)

*Los Puntos descritos anteriormente proporcionan un ambiente común de acceso al usuario definido por ciertas reglas y parámetros. Este ambiente permite que el usuario únicamente tenga que conocer una sola vista de acceso y no múltiples de estas, por cada aplicación.* ***En el capítulo "5. 2 Estructura" se describe las características básicas de un ambiente de trabajo común de un sistema administrador de aplicaciones.***

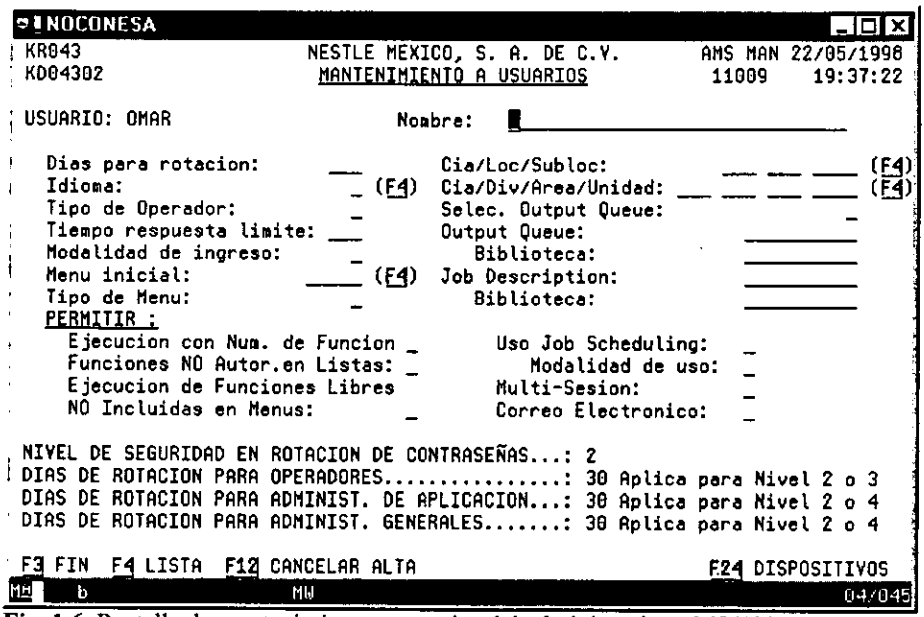


**Fig. 1.6.** Pantalla de mantenimiento a usuarios del administrador AMS/400.

### 1. 2. 3. Control y seguridad de acceso.

AMS/400 tiene su propio nivel de identificación y contraseña, el cual se exige al inicio de la sesión y para dar seguimiento a todas las tareas que el usuario realice. Basándose en esta contraseña, AMS/400 determinará las autorizaciones de ambientes y funciones para el usuario y registrará en bitácoras los trabajos efectuados. El usuario sólo puede acceder a las aplicaciones y a los comandos del sistema operativo mediante AMS/400, no permitiendo alguna otra entrada. Aun cuando algún programador intente crear un perfil de usuario mediante comandos del sistema operativo, éste no podrá tener acceso a ninguna de las aplicaciones si no se ha dado (tal perfil de usuario) desde el mantenimiento a usuarios del manejador AMS/400. Cuando un usuario se firma al sistema, el AMS/400 no sólo verifica que el usuario exista en el archivo de usuarios de AMS/400, si no que también checa que exista en el propio sistema; si esto no se cumple, el sistema niega el acceso.

En la figura siguiente se muestra la pantalla de entrada principal de bienvenida al AMS/400. Aun cuando se muestran más campos de captura (propios del sistema operativo OS/400) adicionales al "Usuario" y "Contraseña" , estos están debidamente controlados para que no tomen efecto al iniciar la sesión, es decir, el AMS/400 toman control del acceso y la presentación de los menús y funciones autorizadas al usuario.

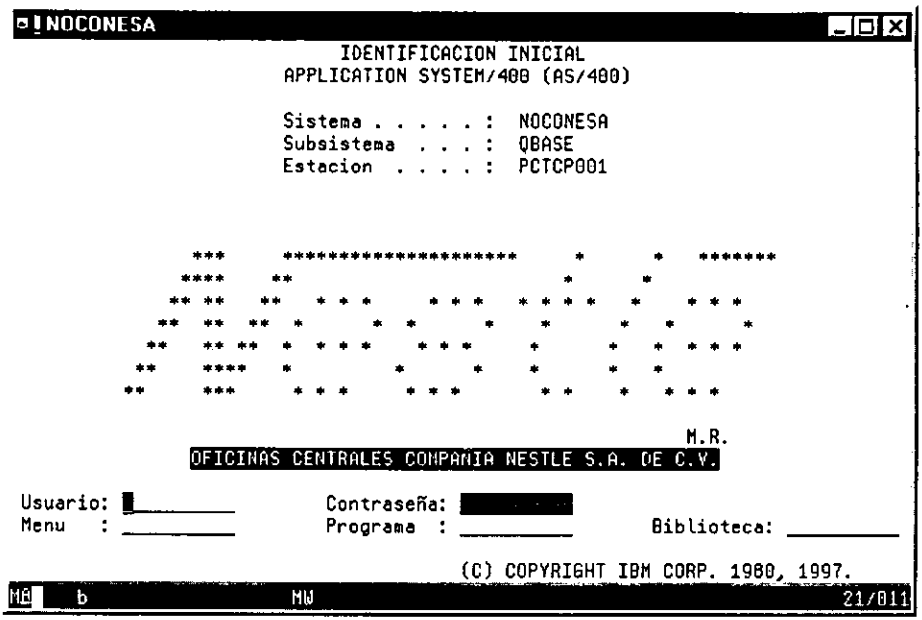



**Fig. 1.7.** Pantalla de identificación y bienvenida al AMS/400.

El AMS/400 además de verificar la existencia del usuario, verifica que el dispositivo que se esta utilizando para entrar al sistema esté dado de alta en el archivo de terminales o dispositivos autorizados. También, como medida de control, el AMS/400 puede ser posible que verifique que el usuario no esté firmado en más de una terminal o dispositivo, siempre y cuando se haya definido en el perfil de usuario.

Una vez que el usuario a entrado al sistema y ejecuta una función, es posible que el sistema le solicite que se reidentifique para ejecutar tal función, sí así se ha definido en uno de los parámetros de la función. Esta medida de seguridad permite dar la pausa para ejecutar una función de cierta importancia, además de asegurar que el usuario que está ejecutando la función es el mismo que se firmó en el sistema y no un intruso que intenta tomar ventaja de que alguien ha dejado una terminal activa en el sistema.

Para garantizar que las contraseñas de los usuarios nunca podrán ser vistas o conocidas, estás encriptadas en el sistema. Por lo que, aun un programador muy especializado podrá obtenerlas. Cuan un usuario olvida cual es su contraseña, no será posible obtener cual era su contraseña, por lo que sólo será posible darle una nueva contraseña para su identificación de usuario.

### 1. 2. 4. Estructuración y definición de funciones.

Una función es una tarea de tipo individual que se puede ejecutar por medio de una función de menú: Las funciones pueden correr programas de usuario o comandos que se habiliten. Todos los trabajos que se efectuarán las aplicaciones serán funciones definidas bajo AMS/400.

El sistema soporta hasta 99,000 funciones, las cuales se arman en estructuras de aplicación y módulos. Adicionalmente se definen los conceptos de tópicos y subtópicos para indicar que tipo de función es. Esto es muy útil para el manejo de lista de funciones.

A continuación se relacionan los datos principales necesarios para definir una función bajo AMS/400:

- Aplicación/Módulo: Clave que definirá a qué aplicación y módulo pertenece la función. Estos datos son requeridos y se utilizan fundamentalmente para ayudas y

para criterios de armado de lista de funciones en el segundo ambiente del Manejador.

- Tópico/SubTópico: Definen qué tipo de función es de acuerdo a una clasificación previamente estipulada por los diseñadores. La utilidad de estos parámetros radica fundamentalmente en el ambiente de lista de funciones en el cual se podría solicitar que el sistema localice y arme una lista, por ejemplo, de todas las consultas existentes o las de una sola aplicación, o las consulta a maestros, etc.

- Descripciones: Textos explicativos de la función a usar cuando Seguridad arme los menús antes de su presentación, o en las ayudas disponibles para el usuario.

- Subfunciones: Lista de subfunciones que la función va a contener, es decir, subtareas que estarán disponibles en la función, por ejemplo: en una función de mantenimiento las Altas, Bajas, Cambios o Consultas.

- Programa/Biblioteca de ejecución: Programa y biblioteca que ejecutará la función (de usuario).

- Programa/Biblioteca de recuperación: Programa que ejecutará la recuperación en caso de corte por falla externa.

- Programa/Biblioteca de parámetros: Programa interactivo para ingreso de parámetros de usuario a pasar a la función que se ejecutara en batch. Este programa sólo se podrá especificar para funciones de tipo Batch.

- Clave de recuperación: Especifica si esta función tendrá una recuperación individual

en caso de corte por falla externa. Para funciones interactivas, si se especifica que si, se deberá suministrar el programa correspondiente. Para funciones batch, si se especifica que si, AMS/400 resumitirà el trabajo desde el punto donde se quedo. Si la función batch tenía un punto donde se cortó, el sistema correrá (en caso de haberlo especificado), el programa recuperador específico.

- Clave Bloqueo: Especifica si es necesario efectuar algún bloqueo de terminales cuando se esté corriendo por recuperación específica de la función.

- Reidentificaciòn: Especifica si AMS/400 solicitará al usuario una refirma de confirmación antes de permitir ejecutar esta función. Esta modalidad permite establecer un nivel de control de acceso adicional para funciones de tipo confidencial o de uso restringido.

- Control de Tareas: Especifica si AMS/400 debe verificar qué tareas están en ejecución antes de dar acceso a esta función, a efectos de efectuar alguna prohibición de ejecución simultánea y notificarla.

- Autorización de ejecución: Especifica si los usuarios deberán ser autorizados expresamente para ejecutar la función o si esta es de tipo "libre", es decir, que esta abierta para todos los usuarios.

- Ambiente de ejecución: Define el ambiente de ejecución de la función. Las posibilidades son Interactivas o Batch.

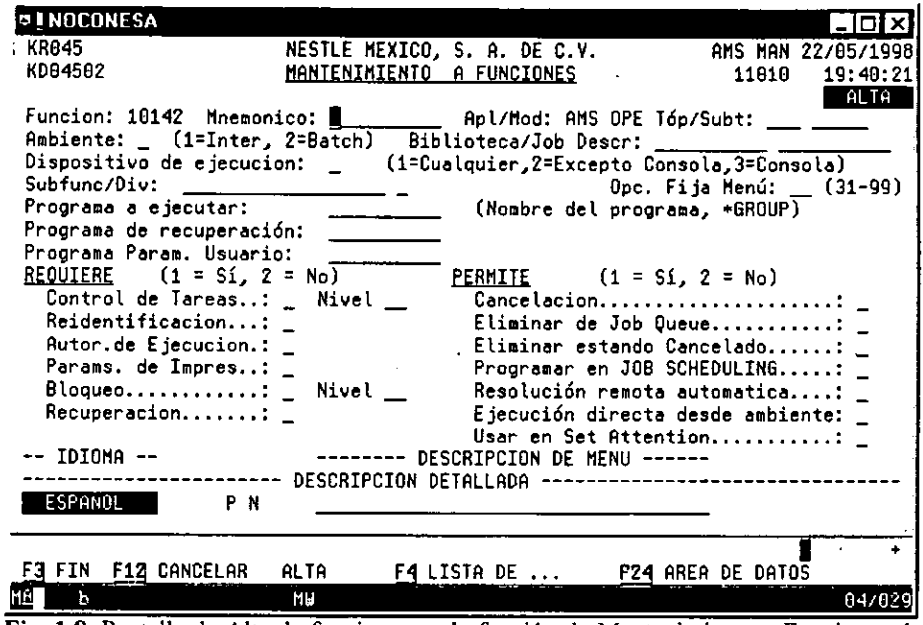



**Fig. 1.8.** Pantalla de Alta de funciones en la función de Mantenimiento a Funciones de AMS/400.

### 1. 2. 5. Control de ejecución.

Autorización sobre uso de funciones para cada operador de tipo usuario. Control de permisos de ejecución antes de iniciar las tareas desde cualquiera de los ambientes de ejecución de AMS/400. Rearmado dinámico de menús al inicio de sesión, al cambio de menú a ejecutar y al cambio de localidad a utilizar. Lo anterior significa que AMS/400 arma los menús que presenta al usuario, basándose en las funciones que el usuario tiene autorizadas para ejecutar bajo la localidad donde se encuentra. Las estructuras de menús que se cargan como parámetros en seguridad sólo sirven como base, pero el armado real del menú a presentar se realiza al momento de solicitar su ejecución.

En la figura siguiente se muestra el menú inicial (1) de un usuario tipo "Administrador". Como se menciono antes, este es un tipo de menú dinámico y cuando el usuario hace un cambio de menú, se actualiza únicamente la parte correspondiente a las funciones autorizadas en determinada compañía.

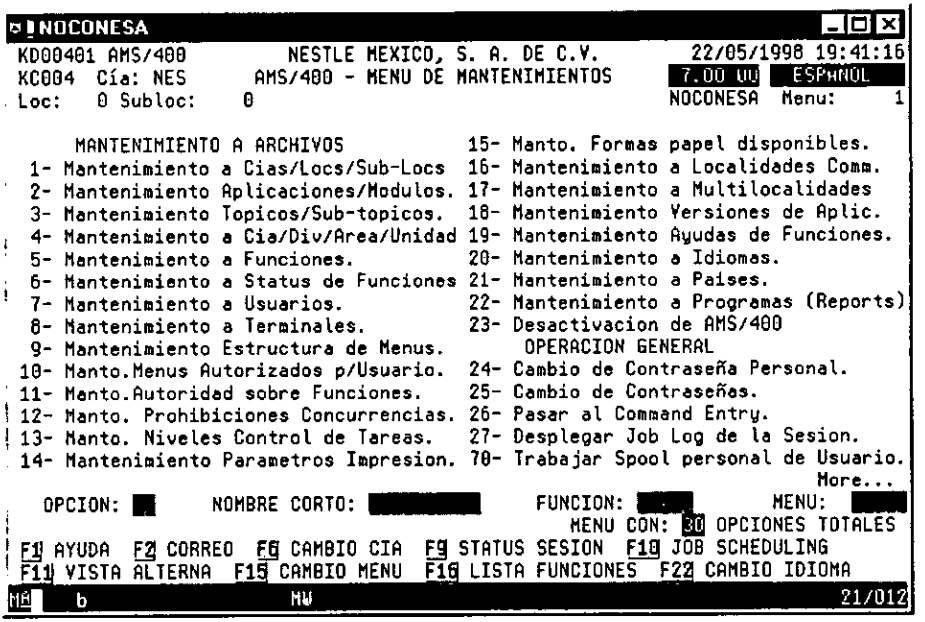



**Fig. 1.9.** Menú principal de mantenimiento de AMS/400

### 1. 2. 6. Control y monitoreo de uso de aplicaciones.

Localización y seguimiento de tareas interactivas o batch en ejecución o terminadas bajo el sistema. Funciones de consulta e inclusive recuperación de tareas interactivas o batch ejecutadas en el equipo del solicitante o desde cualquier equipo. Información explicativa

completa sobre responsables, funciones en ejecución, situación actual, fechas y horarios relacionados, aplicaciones en uso, dispositivos utilizados, menús en uso, localidad en uso, perfiles en uso, etc. Ver figura 1.10. y figura 1.11.

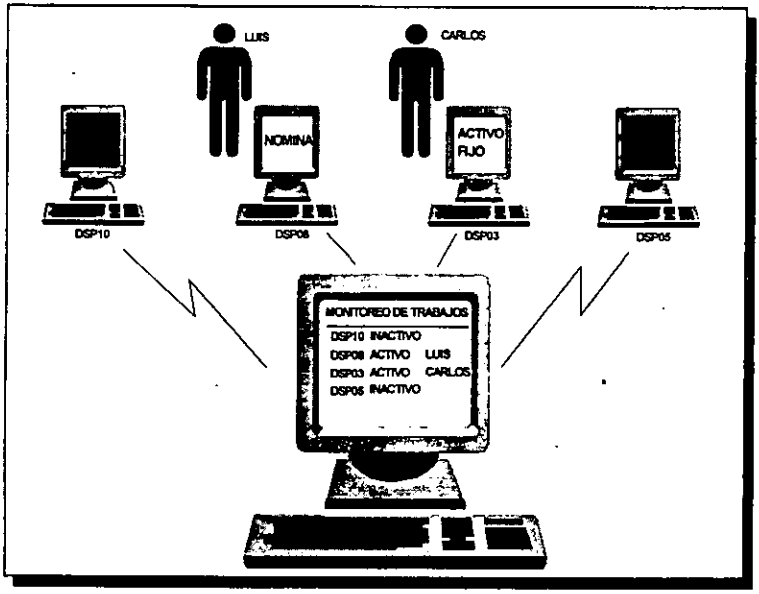



**Fig. 1.10.** Monitoreo de Uso de Aplicaciones.

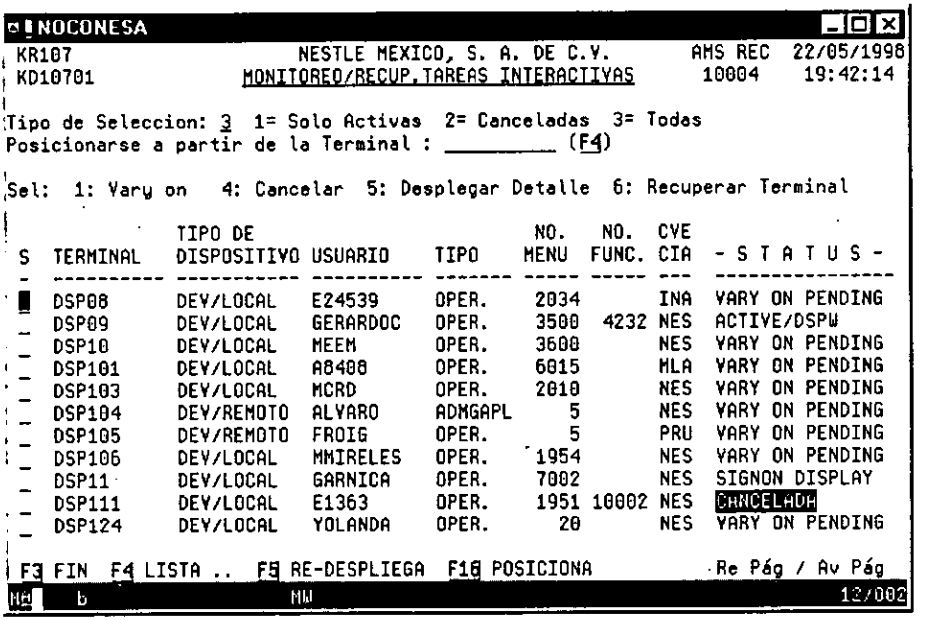



**Fig. 1.11.** Pantalla de Control y Mantenimiento de trabajos interactivos de AMS/400.

### 1. 2. 7. Control y administración de tareas concurrentes.

AMS/400 proporción un mantenimiento para indicar las funciones que no pueden ejecutarse en forma concurrente, y al momento de ejecución, realiza la verificación de concurrencias prohibidas de acuerdo a la estructura de niveles de control de tareas definidos. En caso de encontrar alguna situación de concurrencia prohibida, tanto para funciones interactivas como batch, notifica detalladamente al usuario la imposición en ejecución que provoca la restricción, así como de quien la está usando, desde donde y desde qué hora está corriendo (batch).

A continuación se explica la mecánica de funcionamiento del sistema AMS/400 con respecto a los niveles y control de tareas concurrentes, esto es, se proporciona la lógica que utiliza el manejador para verificar y rechazar casos de concurrencias prohibidas.

Los niveles de control de tareas son códigos predefinidos desde AMS/400 que sirven para determinar el ámbito o frontera de búsqueda dentro del cual, AMS/400 realizará el chequeo de funciones activas dentro de las solicitudes de ejecución para determinar si se acepta la solicitud o se rechaza por detectar una concurrencia prohibida.

Este control de concurrencia es una de las funciones principales de AMS/400 y permite, mediante la captura de parámetros, poder establecer y mantener la estrategia de control de concurrencias de tareas sin tener que incluir rutinas fijas en programas.

Dentro de AMS/400, se suministran una serie de códigos de niveles de control de tareas que deseen respetarse y mantenerse intactos. Estos códigos se dividen en tres grupos que son:

- *Especiales*. Son dos niveles (10 y 20), que provocan que AMS/400 no realice las tareas de control de concurrencias prohibidas cuando se solicita una función. Estos códigos, como se explica mas adelante, son de uso restringido y en general no se usan para aplicaciones de usuario.

- *Individuales*. Este grupo esta formado por los niveles entre 50 y 53 y corresponden a las funciones a las cuales se les cargará la lista de concurrencias prohibidas, o sea la lista de otras funciones contra las cuales no puede ejecutarse simultáneamente. Este grupo es el más usado para las funciones que conforman una aplicación de usuario.

Casi todas las funciones de aplicaciones deben usar niveles incluidos en este grupo.

- *Generales.* Este grupo esta formado por los niveles comprendidos entre los siguientes rangos: 70 a 73, 80 a 83 y 90 a 93. Las funciones con estos niveles no tendrán cargada una lista de concurrencias prohibidas (inclusive el sistema no permite capturarla), sino que AMS/400 asume que ninguna función, dentro del ámbito correspondiente al nivel, puede ejecutarse en forma concurrente a esta función. Más adelante se ejemplificará esta situación y el tipo de algoritmo que sigue AMS/400 para efectuar la verificación de concurrencias entre funciones con niveles de diferentes grupos y niveles.

A continuación se proporciona la lista de los códigos disponibles y una explicación de cada uno:

*Especiales.*

10 Sin control de tareas ni actualización de terminales. Para funciones con este nivel no se realizará ningún chequeo de concurrencias no tampoco actualizará el archivo de terminales de la sesión. Este nivel no debe usarse más que para funciones muy especiales como es el terminar la sesión o despliegue de mensajes, o sea funciones que en cualquier momento puedan ejecutarse y que no usen ningún objeto que resida en las bibliotecas de las aplicaciones de usuario.

20 Sin control de tareas pero con actualización de terminales. Este código es similar al anterior pero si deja huella de que se está usando la función en el archivo de terminales. Al igual que para el nivel 10, el uso de este nivel debe estar restringido a tareas especiales que no usen objetos residentes en bibliotecas de aplicación.

*Individuales.*

50 Control individual. Ámbito: Compañía, Localidad y Sublocalidad. Para prohibir

concurrencia a las funciones de este nivel es necesario cargar la lista de concurrencias prohibidas. Al momento de solicitar ejecutar una función con este nivel, AMS/400 verificará si existe activa alguna de las funciones cargadas como prohibidas de concurrencia para la función que se está solicitando. Si es así, procede a comparar los parámetros de ejecución de la solicitud y de la función activa en los campos: Compañía, Localidad y Sublocalidad. Si estos tres campos coinciden se procede a rechazar la solicitud de ejecución.

51 Control individual. Ámbito: Compañía y Localidad. Para prohibir concurrencia a las funciones de este nivel es necesario cargar la lista de concurrencias prohibidas. Al momento de solicitar ejecutar una función con este nivel, AMS/400 verificará si existe activa alguna de las funciones cargadas como prohibidas de concurrencia para la función que se esta solicitando. Si es así, procede a comparar los parámetros de ejecución de la solicitud y de la función activa en los campos: Compañía y Localidad. Si estos dos campos coinciden se procede a rechazar la solicitud de ejecución.

52 Control individual. Ámbito: Compañía. Para prohibir concurrencia a las funciones de este nivel es necesario cargar la lista de concurrencias prohibidas. Al momento de solicitar ejecutar una función con este nivel, AMS/400 verificará si existe activa alguna de las funciones cargadas como prohibidas de concurrencia para la función que se está solicitando. Si es así, procede a comparar los parámetros de ejecución de la solicitud de la función activa en el campo: Compañía.

53 Control individual. Ámbito: Sistema. Para prohibir concurrencia a las funciones de este nivel es necesario cargar la lista de concurrencias prohibidas. AL momento de solicitar ejecutar una función con este nivel, AMS/400 verificará si existe activa alguna de las funciones cargadas como prohibidas de concurrencia para la función que se esta solicitando. Si es así, y sin necesidad de comparar ningún parámetro para este nivel, el sistema rechazará la solicitud de ejecución de esta función.

*Generales.*

70 Tarea general. No activos en ámbito: Compañía, Localidad, Sublocalidad, Aplicación y Módulo. A las funciones con este nivel no se le podrán definir funciones específicas prohibidas de concurrencia. AMS/400 verificará que no exista ninguna función activa que esté corriendo con los mismos valores de: Compañía, Localidad, Sublocalidad, Aplicación y Módulo que la función que se está solicitando. Si encuentra una función activa que coincida en los parámetros mencionados con la solicitud, la función solicitada es rechazada.

71 Tarea general. No activos en ámbito: Compañía, Localidad, Aplicación y Módulo. A las funciones con este nivel no se le podrán definir funciones específicas prohibidas de concurrencia. AMS/400 verificará que no exista ninguna función activa que esté corriendo con los mismos valores de: Compañía, Localidad, Aplicación y Módulo que la función que se está solicitando. Si encuentra una función activa que coincida con los parámetros mencionados con la solicitud, la función solicitada es rechazada.

72 Tarea general. No activos en ámbito: Compañía, Aplicación y Módulo. A las funciones con este nivel no se le podrán definir funcione específicas prohibidas de concurrencia. AMS/400 verificará que no exista ninguna función activa que esté corriendo con los mismos valores de: Compañía, Aplicación y Módulo que la función que se está solicitando. Si encuentra una función activa que coincida con los parámetros mencionados con la solicitud, la función solicitada es rechazada.

73 Tarea general. No activos en ámbito: Aplicación y Módulo. A las funciones con este nivel no se le podrán definir funciones específicas prohibidas de concurrencia. AMS/400 verificará que no exista ninguna función activa que esté corriendo con los mismos valores de: Aplicación y Módulo que la función que se está solicitando. Si encuentra una función activa que coincida en los parámetros mencionados con la solicitud, la función solicitada es rechazada.

80 Tarea general. No activos en ámbito: Compañía, Localidad, Sublocalidad y Aplicación. A las aplicaciones con este nivel no se le podrán definir funciones especificas prohibidas de concurrencia. AMS/400 verificará que no exista ninguna función activa que esté corriendo con los mismos valores de: Compañía, Localidad, Sublocalidad y Aplicación que la función que se está solicitando. Si encuentra una función activa que coincida con los parámetros mencionados con la solicitud, la función solicitada es rechazada.

81 Tarea general. No activos en ámbito: Compañía, Localidad y Aplicación. A las funciones con este novel no se le podrán definir funciones específicas prohibidas de concurrencia. AMS/400 verificará que no exista ninguna función activa que esté corriendo con los mismos valores de: Compañía, Localidad y Aplicación que la que se está solicitando. Si encuentra una función activa que coincida en los parámetros mencionados con la solicitud, la función solicitada es rechazada.

82 Tareas general. No activos en ámbito: Compañía y Aplicación. A las funciones con este nivel no se le podrán definir funciones específicas prohibidas de concurrencia. AMS/400 verificará que no exista ninguna función activa que esté corriendo con los mismos valores de: Compañía y Aplicación que la función que se esta solicitando. Si encuentra una función activa que coincida con los parámetros mencionados con la solicitud, la función solicitada es rechazada.

83 Tarea general. No activos en ámbito: Aplicación. A las funciones con este nivel no se le podrán definir funciones específicas de concurrencia. AMS/400 verificará que no exista ninguna función activa que esté corriendo con los mismos valores de: Aplicación. Si encuentra una función activa que coincida en el parámetro mencionado con la solicitud, la función solicitada es rechazada.

90 Tarea general. No activos en ámbito. Compañía, Localidad y sublocalidad. A las funciones con este nivel no se le podrán definir funciones específicas de concurrencia.

AMS/400 verificará que no exista ninguna función activa que esté corriendo con los mismos valores de: Compañía, Localidad y Sublocalidad que la función que se está solicitando. Si encuentra una función activa que coincida con los parámetros mencionados con la solicitud, la función solicitada es rechazada.

91 Tarea general. No activos en ámbito. Compañía y Localidad. A las funciones con este nivel no se le podrán definir funciones especificas de concurrencia. AMS/400 verificará que no exista ninguna función activa que esté corriendo con los mismos valores de: Compañía y Localidad que la función que se está solicitando. Si encuentra una función activa que coincida con los parámetros mencionados con la solicitud, la función solicitada es rechazada.

92 Tarea general. No activos en ámbito: Compañía. A las funcione con este nivel no se le podrán definir funciones específicas de concurrencia. AMS/400 verificará que no exista ninguna función activa que esté corriendo con los mismos valores de: Compañía que la función que está solicitando. Si encuentra una función activa que coincida con el parámetro mencionado con la solicitud, la función solicitada es rechazada.

93 Tarea general. No activos en ámbito: Sistema. A las funciones con este nivel no se le podrán definir funciones específicas de concurrencia. AMS/400 verificará que no exista ninguna función activa que esté corriendo bajo AMS/400 sin importar con que parámetros lo está haciendo. Este nivel sirve para funciones que necesiten todos los recursos del ambiente de AMS/400 en forma dedicada. Si encuentra una función activa bajo AMS/400, la función solicitada es rechazada.

### 1. 2. 8. Manejo y control de multi-compañias/multi-localidades.

Uno de los elementos o parámetros básicos a través de todo el sistema de AMS/400, es la estructuración o delimitación del ámbito en que se ejecutan las aplicaciones. A tal efecto AMS/400 soporta una estructura a tres niveles: Compañía, Localidad y Sublodalidad. Para efectos explicativos, el conjunto de estos tres parámetros se denomina genéricamente localidad. Estas localidades son parámetros a definir por los diseñadores de las aplicaciones, que servirán para establecer estructuras basadas en subdivisiones geográficas o lógicas (ver figura 1.12.), de donde correrán y bajo las cuales AMS/400 controlará las aplicaciones. Las localidades de tipo geográfico serían las definiciones de diferentes códigos para diferentes compañías y sus sucursales, almacenes u oficinas. Las localidades de tipo lógico serían subdivisiones que no corresponderían a una realidad de tipo geográfico (por ejemplo: definir localidades que se encarguen de manejar una aplicación desde el mismo lugar, pero con separación de tareas y datos por criterios, no de ubicación física sino de tipo - una localidad significa por ejemplo empleados de tipo quincenal y otra empleados de tipo semanal). Quien diseñe e implante las aplicaciones tendrá la obligación de crear estas localidades, a efecto de que posteriormente se establezcan las autorizaciones específicas sobre las funciones para cada una de las localidades, así como los niveles de control de tareas. Se recomienda que la asignación de éstas localidades se haga en forma genérica para intentar que sea aplicable a cualquier aplicación. Con una buena definición al respecto y respetando y aplicando estándares de diseño de aplicaciones, se puede lograr que múltiples localidades, ejecuten y controlen las mismas aplicaciones mediante los mismos programas con información almacenada en bases de datos únicas, sin interferencia ni riesgos de intromisiones por otros usuarios. Adicionalmente AMS/400 proporciona soporte para cambiarse de localidad desde la misma sesión del usuario, por medio de teclas de función.

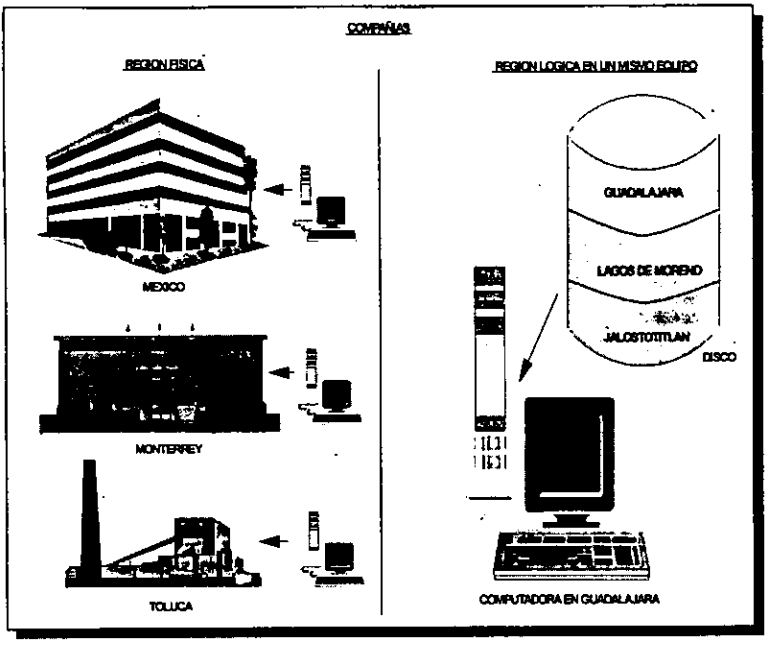



**Fig. 1.12.** Compañía: Región Física o Región Lógica en el Mismo Equipo.

### 1. 2. 9. Manejo de multi-aplicaciones.

Como ya se menciono, AMS/400 fue diseñado para ejecutar y controlar múltiples aplicaciones. La aplicación AMS/40 es un desarrollo que está completamente independiente de cualquier aplicación de usuario y no tiene ningún tipo de constante o restricción que ate a otras aplicaciones. Por lo tanto, cualquier aplicación puede ser diseñada o adaptada a que corra bajo el ambiente de AMS/400, sin tener que realizar más que mantenimientos a parámetros del sistema. En la figura 1.13. a) y b) se muestra un ejemplo de las aplicaciones existentes en compañía Nestlé y que se explotan mediante AMS/400.


**Fig. 1.13.** Aplicaciones en Compañía Nestlé (a).

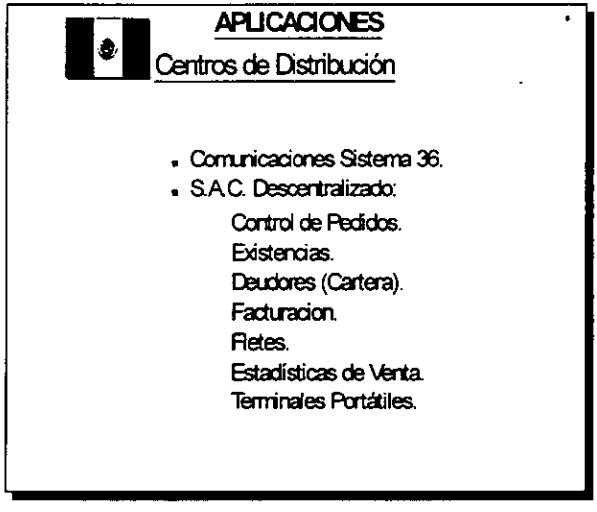



**Fig. 1.13.** Aplicaciones en Compañía Nestlé (b).

Todas las aplicaciones de usuario, sus módulos, funciones que lo componen y los menús

que las agrupan, se definen a un sólo nivel en Seguridad, con lo cual se logra (si se desea), armar un ambiente donde las funciones de diferentes aplicaciones se mezclan sin restricción y sin problemas con respecto a su ubicación en la máquina (biblioteca) o sus condiciones especiales de ejecución. Por ejemplo un menú para uno o más usuarios puede amarse con opciones provenientes de diferentes aplicaciones, que ejecuten programas de diferentes bibliotecas y que inclusive corren en diferentes ambientes y todo esto ser completamente transparente para el usuario, sin incluir ninguna constante o tabla fija en los programas de AMS/400.

En todos los ambientes del sistema, AMS/400 proporciona información completa sobre las características de las funciones y su origen (aplicación y módulo a los que pertenecen) y para el caso de los operadores Administradores, indican los nombres de programas y bibliotecas que ejecutan las tareas. Lo anterior permite que la presentación y estructuración de las aplicaciones sean homogéneas a través de toda la instalación o por lo menos de todas las aplicaciones que decida manejar por medio de AMS/400.

### 1. 2. 10. Capacidad de ayuda y documentación en línea.

AMS/400 proporciona el soporte completo para el ingreso y uso de Ayudas a usar por todas las aplicaciones. El esquema seguido es el de tener ayudas a nivel pantalla. Para cada pantalla de cada función de puede cargar una ayuda de hasta 380 renglones, la cual se accederá al memento de ejecución mediante el uso de la tecla de ayuda, Estas ayudas se almacenan en archivos de AMS/400 y su mantenimiento se efectúa mediante funciones de AMS/400. Adicionalmente se proporciona una rutina de uso común que llama a un programa de acceso y despliegue de ayudas a incluir en cada función de las aplicaciones. Con lo anterior se hace autónomo el programa de aplicación y sus ayudas,

permitiendo darles mantenimiento en forma independiente y accediendolas en forma inmediata.

Además de las ayudas por pantalla, AMS/400 proporciona para cualquier función, pantalla de ayudas generales que muestran todos los datos referentes a las definiciones que las funciones en el sistema y la relación de localidades desde donde se pueden ejecutar. Lo anterior está disponible en los dos ambientes de ejecución y se complementa con funciones de consulta y reportes de aplicaciones/módulos, funciones definidas y tópicos y subtópicos.

### 1. 3. Application Program Driver/400.

En las empresas organizadas de hoy, la gente se enfrenta a un número creciente de aplicaciones. Las aplicaciones no sólo se diferencia en su funcionalidad, si no también en sus interfaces, Algunas veces es difícil para el usuario poder recordar como se accede y se navega a través de ellas.

El sistema Applicaction Program Driver/400 de I.B.M., que en lo siguiente nos referiremos como APD/400, permite la integración de aplicaciones nativas del equipo AS/400, dentro de un ambiente de procesamiento común. El diseñador de sistemas puede integrar cualquier aplicación de I.B.M., aplicaciones diseñadas por terceros o sus propias aplicaciones. APD/400 además provee una interfase estandarizada, donde las funciones comunes de las aplicaciones integradas pueden ser usadas de la misma

manera. Para los usuarios, no pareciera que están usando diferentes aplicaciones, sino una sola aplicación con una infinidad de funciones. Como se explica más adelante, APD/400 no solo ofrece facilidades al usuario final, sino que también a los operadores de sistemas, administradores de aplicaciones y a los desarrolladores de aplicaciones.

Para facilitar el trabajo de los usuarios finales, operadores de sistemas, administradores de aplicaciones y desarrolladores de aplicaciones, APD/400 ofrece las siguientes características y beneficios:

### 1. 3. 1. Manejador de menús.

Todas las funciones se trabajan mediante menús controlados. Los usuarios pueden decidir el contenido del menú inicial después de haberse firmado en el sistema. Así mismos pueden crear el contenido de su menú personal, conteniendo las funciones de las aplicaciones más comunes. Las funciones pueden ser mostradas en menú de barras, menús pull-down, menús pop-up o a través de código experto. En caso de código experto, se muestra una lista para que el usuario pueda seleccionar el código habilitado.

### 1. 3. 2. Soporte Multi-ídioma.

APD/400 ofrece el soporte para poder correr las aplicaciones en diferentes idiomas. Esto significa, que los diferentes usuarios pueden trabajar con APD/400 e integrar sus aplicaciones en diferentes idiomas al mismo tiempo. El primer idioma en la lista es el idioma seleccionado. APD/400 y la aplicación usada aparecen en el idioma de más alta prioridad en la lista, Sin embargo, el usuario puede cambiar de idioma en cualquier momento.

### 1. 3. 3. Control de Acceso.

El administrador puede permitir o prohibir las autorizaciones de acceso interactivamente. APD/400 soporta el control de acceso para las diferentes instalaciones de una aplicación, menús, menú de barra, menú pull-down, etc., Para facilitar la administración la autorización de acceso, el administrador de la aplicación no tiene que asignar autorización a cada usuario en forma individual. En cambio, los usuarios con los mismos derechos de acceso son agrupados dentro de un grupal de usuario. Entonces, las autorizaciones son definidas para un grupal como un todo.

### 1. 3. 4. Manejo de Conflictos.

APD/400 ofrece un manejo de conflictos para tareas y objetos del sistema operativo (OS/400). Un conflicto ocurre cuando tareas incompatibles son ejecutadas simultáneamente, Por ejemplo una operación de respaldo no se puede ejecutar mientras que los archivos a ser respaldados están siendo actualizados o, un objeto no puede ser cambiado por dos trabajos al mismo tiempo. APD/400 implementa el manejo de conflictos a través de exclusiones y listas de exclusiones. Nuevamente, las exclusiones no tendrán que ser definidas en forma individual. Para reducir el esfuerzo administrativo, las tareas y los objetos con la misma exclusión pueden ser agrupados dentro de una lista de exclusiones.

### 1. 3. 5. Auditoria de Actividad.

Para la auditoria de actividad se provee información estática acerca de la actividad en el sistema del uso de APD/400. Para cada aplicación, el administrador de la aplicación puede seleccionar funciones para ser auditada. Entonces, cada evento es registrado en archivo. Un evento es, por ejemplo, cuando un usuario inicia y termina una tarea. Se ofrecen simples consultas los cuales pueden ser cambiados o se pueden crear nuevas consultas dinámicas donde se puede seleccionar la información solicitando que cuando se usaron determinadas funciones de determinada aplicación.

*La información proporcionadas por los reportes de control pueden ayudar a visualizar de manera clara todo aquello que podemos auditar. Desde el mismo instante que consultamos esto reporte estamos auditando, es decir, estamos determinando como es el entorno de cada uno de los usuarios que entran a nuestro sistema. En el tema:"*5. 5. 3. Reportes de Control generados por S.I.A.*" se dan ejemplos de reportes de control con los que debe contar un Sistema Administrador de Aplicaciones.*

### 1. 3. 6. Supervisión de recuperación.

Para cada tarea de aplicación, el administrador puede especificar que pasa una tarea falla, por ejemplo, por una terminación anormal:

- El usuario es notificado acerca de la falla de la función.
- El usuario no es notificado acerca de la falla de la función.
- Una recuperación es obligatoria.

Cuando un usuario se firma nuevamente al sistema, APD/400 reinicia la última opción de menú usada.

### 1. 3. 7. Respaldo y Restauración.

La herramienta de respaldo y restauración permite definir lo siguiente:

- Intervalos de respaldo.
- Número de generación de respaldos.
- Secuencia de restauración.
- Identificación de volúmenes de los respaldos.
- Dependencias de respaldos.

Los respaldos pueden ser ejecutados, ya sea programados para determinada fecha y hora, o en forma individual inmediata. Se pueden mantener hasta nueve generaciones de respaldos para las bibliotecas y folders. Después de cada respaldo se almacena la historia de éste, en un archivo de cinta histórico. APD/400 restaura en forma automática los respaldos. El usuario puede seleccionar si se restauran todas las bibliotecas o en forma individual cada una de las bibliotecas y folders.

### 1. 3. 8. Instalación de Aplicaciones.

Mediante APD/400, se realiza la instalación de las aplicaciones interactivamente. El administrador es ayudado a través de una serie de pantallas que explican los pasos requeridos para instalar la aplicación seleccionada desde una cinta. También se tiene la liberta de poder instalar una aplicación con los propios procedimientos del administrador. Las aplicaciones pueden ser instaladas varias veces. Si la aplicación es correctamente codificada, además, se puede especificar cierta de base de datos a ser usada.

### 1. 3. 9. Interfase de estandarización.

APD/400 ofrece una interfase de estandarización para todas las aplicaciones integradas que no requiere de un conocimiento especial de AS/400 del usuario final. Todas aplicaciones y funciones que contienen son llamas de la misma forma. Para los usuarios les aparece como si estuvieran usando una sola aplicación con una variedad de funciones. Una vez familiarizados con la interfase, los usuarios pueden trabajar fácilmente con cualquier aplicación instalada bajo APD/400. Esto reduce la curva de aprendizaje y la posibilidad de error de los usuarios, salva tiempo, y así incrementa la productividad.

### 1. 3. 10. Incremento de Productividad.

La productividad no sólo se incrementa a través del uso de la interfase de estandarización. Existen más atributos que ayudan a salvar tiempo:

- Los usuarios pueden fácilmente cambiarse entre las aplicaciones o las diferentes instalaciones de la misma aplicación. No es necesario dejar una aplicación, iniciar una nueva, moverse a cualquier pantalla, ejecutar una tarea, dejar la aplicación y finalmente reiniciar la primera aplicación. Sólo dejar la sesión de la aplicación activa y cambiarse a otra nueva.

- Se puede además entre las diferentes instalaciones de la aplicación. Puede ser, por ejemplo, instalaciones para producción, prueba, educación, etc. Se puede ejecutar una prueba en el ambiente de pruebas, y después cambiarse al ambiente de producción e implantar los que se están probando.

- Hay varios atajos que el usuario puede usar. Si se tienen varias instaladas bajo APD/400, el menú inicial provee una selección restringida para las aplicaciones más usadas. El menú personal que los usuarios pueden definir en forma individual, contiene las tareas más usadas de una o más aplicaciones que normalmente el usuario trabaja.

- Dentro de una aplicación, se pueden invocar funciones mediante menús o mediante códigos expertos. Los códigos expertos además proveen la ventaja, ya que se llama a la función en forma directa, sin haber navegado a través de varios menús.

### 1. 3. 11. Esfuerzo de Desarrollo Reducido.

Desarrollando aplicaciones bajo APD/400 se ahorra tiempo y esfuerzo porque, para muchas funciones, se puede usar las facilidades y servicios de APD/400 en lugar la programación manual de las mismas. El usuario puede definir interactivamente el flujo del menú para una aplicación, grupo de pantallas, o crear el contenido de todos los menús en cascada requeridos para ejecutar una tarea. Además, puede utilizar las funciones para el chequeo de autorización, control de exclusión, soporte de instalación, etc., o usar su propio implemento de funciones.

### 1. 3. 12. Conjunto de Datos Separados.

Dentro de una aplicación, el usuario puede trabajar con diferentes conjuntos de datos para separar claramente los datos procesados por diferentes clientes, o para distinguir entre los diferentes proyectos o departamentos.

### 1. 3. 13. Personalización Simplificada.

Las siguientes áreas pueden ser mejoradas y adaptadas de acuerdo a las necesidades:

- Datos de Texto: El usuario puede crear o cambiar datos de texto como ayudas o mensajes.

- Menús y Ventanas: El usuario puede especificar como serán presentadas las pantallas completas de menús y ventanas. Pueden ser definidos los colores estilos de varios componentes de pantalla y ventanas, incluyendo menú de barra, área de títulos, borde de ventanas, áreas de opciones de menús y línea de comandos. Un ilimitado número de ventanas puede ser presentado en forma de cascada.

# CAPITULO II

# SEGURIDAD EN LOS SISTEMAS

El acceso no autorizado a los datos almacenados en una computadora implica una violación a la privacidad ya sea de una empresa o de una persona, dicha privacidad requiere la imposición de estrictas normas técnicas acerca de como entrar, recuperar y almacenar información, si se carece de dichas políticas acerca del uso de los recursos computacionales de la organización, las personas pueden y asumirán que tienen derecho a utilizar los sistemas y la información contenida cuando así lo consideren.

La mayoría de las organizaciones ponen atención exagerada en la seguridad física, y dejan en un segundo plano y en ocasiones sin importancia los aspectos técnicos y de procedimiento, no contando con un enfoque de seguridad total, y sufren las consecuencias cuando dejan de operar por largos periodos por no haber prevenido el desastre, y tiene que aplicar medidas correctivas, cuando pudieron haber implantado medidas preventivas, con una ambiente de seguridad total, acorde a sus necesidades.

La protección de la información que rodea a un sistema se puede conseguir por diversos medios: psicológicos, legales, físicos, organizativos y programados. Los programas y datos deben estar protegidos contra el daño accidental y contra el acceso a copiado no autorizado. La prevención de posibles daños a programas o datos se puede conseguir siguiendo los procedimientos adecuados; concretamente, haciendo una copia de todos los programas nuevos recibidos y creando copias de reserva de los nuevos datos. La protección física de los medios de almacenamiento, discos y cintas por ejemplo, se consigue almacenándolos de la forma apropiada. La protección de los programas y datos contra el acceso no autorizado es una tarea más difícil, pero se puede lograr. En este capítulo se describirá el papel que juega la seguridad como protección de la información así como su integridad.

## 2. 1. Medidas de protección.

Uno de los métodos más simples y efectivos cara a la seguridad es tener cerrado bajo llave. Se puede controlar el acceso a la sala de la computadora, si es necesario mediante un cuaderno de control de entradas/salidas. Como precaución adicional se puede equipar la computadora propiamente dicho con una cerradura y llave. Paralelamente, se puede substituir el conmutador de alimentación de la mayoría de las terminales por una cerradura con llave. También constituye una buena medida de precaución guardar todos los programas y datos esenciales en armarios con llaves.

Naturalmente, tales medidas drásticas limitan el número de usuarios autorizados y pueden quitar las ganas de usar la computadora. En vista de las inconveniencias de tales soluciones, se pueden usar técnicas de protección alternativas, menos severas. Concretamente, se han diseñado numerosos sistemas operativos y programas de control de acceso que dan acceso mediante una contraseña, y protegen así los programas y los datos; el usuario autorizado tendrá que teclear la contraseña correcta antes de leer un archivo específico o ejecutar un programa determinado. Para que el sistema conserve su eficacia es preciso cambiar la contraseña con cierta frecuencia.

Si se necesitara seguridad adicional, se puede recurrir a programas que registran el tiempo de trabajo de la computadora y el nombre de usuario. Hay complejos sistemas operativos y programas de tratamiento de archivos que permiten a los usuarios proteger selectivamente sus archivos haciéndolos sólo leíbles, sólo ejecutables, o completamente inaccesibles excepto para el propietario del archivo. Estas facultades no pueden ser añadidos por el usuario; son parte del sistema de archivos, es decir, parte del sistema operativo generalmente.

### 2. 1. 1. Cifrado como protección.

Si se precisa seguridad adicional para la protección del contenido de los archivos de lecturas no autorizadas, un método sencillo de proteger los archivos es cifrar los archivos de datos. Con ello, resultan ilegibles para el usuario sin el correspondiente código o programa descifrador. El programa cifrador tomará un determinado archivo y convertirá la secuencia de caracteres del archivo en otra secuencia de caracteres determinada por el algoritmo cifrador. El material resultante es totalmente ilegible para las personas y, para otro usuario, aparecerá que carece de significado. Mediante el programar descifrador adecuado se puede volver a convertir a su forma original.

La desventaja de esta solución radica en el tiempo que se pierde en cifrar y descifrar los archivos. Sin embargo, es una herramienta muy efectiva, pues desanima a copiar información que nos es directamente identificable. Naturalmente, la etiqueta del disco o la cinta no ha de manifestar explícitamente su contenido, o quedará parcialmente derogado el propósito del algoritmo cifrador, pues ha sido revelada la naturaleza de su contenido.

### 2. 1. 2. Ocultación de la información.

Un principio de seguridad adoptado por las organizaciones militares es el de la "necesidad de conocer". Sólo a los individuos que necesitan conocer un parte particular de información para efectuar sus deberes se les proporciona esa información. La

información que no tiene una relación directa con su trabajo se oculta. Al programar debe adoptarse un principio análogo para controlar el acceso de las unidades programa a los datos del sistema.

En un principio a cada unidad de programa sólo se le debe permitir el acceso a aquellos objetos de programa que necesite para realizar su función. El acceso a otros objetos que la unidad no necesita debe de ser negado por las reglas de alcance del lenguaje de programación que ocultan la existencia de esos objetos. A esto se le llama "ocultación de la información". La ventaja de ocultar la información innecesaria, es que no hay manera de que la información oculta corrompida por una unidad de programa que se supone no debe utilizarla. Esto significa que los programas son mas seguros y, en algunas circunstancias, pueden proporcionar independencia de datos. Se puede cambiar la representación de los datos sin cambiar las unidades de programa que los emplean. Además, si se declaran los objetos cerca de donde se van a usar, se mejora la legibilidad del programa. El lector no necesita revisar varias páginas de listado del programa para encontrar la definición de un objeto.

En los lenguajes sin construcción de ocultación se necesita un enfoque muy desiplinado de la programación, si se van a simular tales construcciones. El enfoque que se debe adoptar es la definición de procedimientos de propósito especial para tener acceso a tipos determinados. Se debe evitar el acceso a los objetos de esos tipos de datos en cualquier otra forma.

### 2. 1. 3. Contra medidas técnicas.

Existen muchas contra medidas técnicas. Entre ellas se incluyen los programas de control de acceso, que automatizan y controlan e acceso a los programas y a los archivos de datos. Estos programas de control de acceso suelen usar contraseñas y otros códigos, y restringen el acceso a los programas y archivos a sólo los usuarios específicamente autorizados. También pueden registrar e informar automáticamente acerca de los accesos al sistema o archivos confidenciales del sistema. Hay, además, programas de control complejo que sólo autorizan funciones específicas de usuario. Concretamente, puede estar automatizado el tipo de acceso a cada archivo así como la lista de personas autorizadas.

El cifrado o revoltijo de la información puede servir también para proteger archivos, tanto almacenados en medios magnéticos como cuando son transmitidos por línea telefónica.

## 2. 2. Métodos de seguridad.

Propiamente dicho, no existen métodos de seguridad establecidos. Sin embargo, cada una de las plataformas y los diferentes sistemas operativos han tratado de adoptar su propia forma de seguridad, desde la explotación de archivos o base de datos hasta el uso de programas. No obstante que cada sistema operativo cuenta con su propio nivel de seguridad, no siempre es satisfactorio. Es por eso que los analistas y diseñadores de

sistemas se han dado ha la tarea de crear nuevos y más convincentes métodos de seguridad, usando lenguajes de cuarta generación o lenguajes que les permite trabajar en la computadora que están trabajando. El analista y el diseñador realizan su máximo esfuerzo en conseguir un nivel de seguridad que satisfaga sus necesidades, ya que muchas veces tienen que "nadar contra la corriente", ya que el sistema operativo por si solo fija sus limitaciones o los lenguajes de programación no son tan poderosos.

### 2. 2. 1. Ejemplo de seguridad para equipos AS/400 de I.B.M.

Seguridad del Sistema para los equipos AS/400 de I.B.M. esta basado en el Sistema Operativo OS/400 y consiste en el medio de seguridad que el usuario construye dentro del sistema para ayudarse a alcanzar control sobre quien pueden usar sus dispositivos, datos y programas y para prevenir cambios accidentales o intencionales, o la destrucción de los recursos del sistema.

La seguridad ayuda a prevenir acceso a objetos por los usuarios que no tienen autoridad, y ayuda a proteger la integridad de los datos en el sistema. El termino OBJETO, es cualquier cosa que existe y ocupa espacio dentro de la memoria y en el cual se pueden ejecutar operaciones, tal es el caso de los programas, archivos, librerías y folders.

En muchas aplicaciones, el control de la integridad de los datos es más critico que el control de la seguridad de los datos. Para ayudar a alcanzar la integridad de la aplicación, el usuario desearía asegurarse de que:

- Los cambios a los datos de aplicaron puedan hacerlo solamente por usuarios que deben tener autoridad.
- Los cambios a los datos de aplicación son hechos solamente a través de programas aprobados.

La seguridad puede ayudar a controlar:

- Accesos al sistema mediante la requisición al usuario de una identificación a través de un nombre de perfil y una palabra clave (PASSWORD) cuando se firma en el sistema, o a través del uso de descripciones de trabajo para correr trabajos en lotes.
- Accesos al sistema requiriéndole a un usuario de un sistema remoto un nombre de perfil y una palabra clave que arranque un trabajo de comunicaciones.
- Recursos usados por el sistema, tal como las estaciones de trabajo e impresoras, mediante la variación de autoridad a usuarios para usarlos.
- Datos en el sistema requiriéndole a los usuarios la autoridad para usar objetos específicos, tales como archivos, programas, comandos y dispositivos.
- Funciones en el sistema, tales como adiciones de usuarios al sistema, salvar y restaurar el sistema operativo, ejecución de funciones de servicio y el control de trabajos de otros usuarios requiriéndole a los usuarios autoridades especiales para usar los comandos y programas.

Una forma de definir la seguridad en el sistema es mediante la definición de Niveles de Usuarios que ingresan al sistema. Cuando se identifica a un usuario en el sistema, el administrador del sistema puede definir la clase o nivel de usuario en el perfil de usuario. La clasificación del usuario administrador especifica que operaciones de control del sistema y que opciones de menú pueden usar el usuario. Esto no necesariamente limita el uso de comandos. Cada clase de usuario tiene un conjunto de autoridades especiales dadas. Las clases de usuarios son:

- Oficial de seguridad (*SECOFR).
- Administrador de Seguridad (*SECADM).
- Programador (*PGMR).
- Operador del Sistema (*SYSOPR).
- Usuario (*USER):

Adicionalmente a los niveles de usuario, se pueden acondicionar definiendo un tipo de autoridad especial, adicional a la asignada a la hora de definir el nivel de usuario. Estas autoridades especiales son:

- *ALLOBJ Autoridad a todos los objetos es permitida para el usuario. El usuario tiene autoridad a todos los objetos en el sistema.
- *AUDIT Autoridad de Auditar es permitida al usuario. EL usuario tiene la autoridad de ejecutar la función de Auditar. Las funciones de Audición incluyen habilitar o deshabitar la función de Auditar el sistema y controlar los niveles de audición en un objeto o un usuario.

- *JOBCTL Autoridad de control de trabajos es permitida al usuario. El usuario tiene la autorización de cambiar, desplegar, detener, liberar, cancelar y limpiar todos los trabajos. El usuario además tiene la autoridad pata arrancar el sistema y arrancar los escritores y para subsistemas activos.

- *SAVSYS Autoridad para respaldar el sistema es permitido al usuario. Esta autoridad le permite al usuario: respaldar, restaurar y liberar memoria para todos los objetos en el sistema.

- *SECADM Autoridad de Administrador de Seguridad es permitida al usuario. El usuario puede crear o cambiar perfiles de usuario en el sistema si el usuario es autorizado a los comandos de crear y cambiar perfiles de usuario. Esta autoridad no le permite dar autoridades especiales a otro usuario que este usuario no tenga.

- *SERVICE Autoridad de servicio es permitida al usuario. El usuario puede ejecutar funciones de servicio al sistema.

- *SPLCTL Autoridad de control de lista archivos de impresión en espera, es permitida al usuario. El usuario puede eliminar, cancelar, detener, liberar, cambiar o enviar cualquier archivo de impresión de cualquier lista de espera de cualquier usuario.

### 2. 2. 2. Auditoria como método de seguridad.

En los sistemas en línea, a diferencia de los ambientes en lotes, puede no haber copias de los documentos fuente de entrada para regresarlos si se presentan fallas del sistema durante el procesamiento. Asimismo, es posible para los usuarios en línea entrar al sistema, alterar los datos almacenados en los archivos y anotar la salida nuevamente sin dejar una posible vista de lo que sucedió. A lo menos que analista desarrolle procedimientos de auditoría, no existe protección en los sistemas en línea y distribuidos.

Una ruta de auditoría se diseña para buscar cualquier registro de entrada o proceso desarrollado en un sistema para que vuelva a su fuente original. Una forma para realizarlo es mantener automáticamente una bitácora de las transacciones. Los detalles de cada transacción se registran en un archivo de transacciones que está separado dentro del sistema. El almacenamiento de estos detalles es automático e invisible para el usuario; la información de éste también debe almacenarse de manera que sea claro quien realizó la transacción. Si el sistema tiene un reloj interno, a cada transacción también se le asigna la hora para decir cuándo ocurrió. Si existe la necesidad de auditar un registro particular en un archivo, es relativamente fácil determinar quien realizó la transacción, cuándo se presentó, que datos contenía y como se modifico el registro del archivo maestro. En otras palabras existe una pista completa de la transacción en su totalidad y de su efecto en el sistema.

Hoy en día, la mayoría de los Sistemas Operativos cuentan por los menos con un módulo de auditoria que permite monitorear cierto tipo de tareas especiales que requieren de autorización para ser ejecutadas. Quizá, no resuelva el problema de impedir que un intruso entre al sistema o ejecute una tarea especial, con una contraseña que tal vez no le pertenezca. Pero este tipo de herramienta ayuda a determinar los posibles responsables de cualquier atentado en un centro de computo. Al mismo tiempo, ayuda a reducir la probabilidad de que un intruso pueda cometer fechorías. Regularmente, este tipo de módulos ofrece ciertos históricos de los procesos que se desean auditar, proporcionando datos como: fecha, hora, usuario, aplicación usada, módulo de la aplicación, código de la función o tarea ejecutada y pantalla o terminal desde donde se ejecuto la tarea o función.

No sólo las tareas o funciones especiales pueden ser auditadas, también pueden ser auditadas las bases de datos, las contraseñas usadas, las pantallas o terminales o cualquier otro tipo de objetos considerados importantes. Estos módulos de auditoria, además de registrar la información antes mencionada, permiten activar un mensaje que interactivamente y en línea avisa al responsable y/o al administrador de seguridad que se

esta accediendo un objeto de importancia, manteniéndolos en alerta. Inclusive, algunos módulos de auditoria, proceden a desactivar al usuario o la pantalla que esta violando las reglas de seguridad, evitando que el proceso continúe y que el posible desastre se concluya.

Aun cuando se requiere de más recursos para poder almacenar la infamación que un módulo de auditoria ofrece, vale la pena mantener presente, que puede resultar más barato que tratar de recuperarse de un desastre informático o un fraude.

## 2. 3. Riesgos en la seguridad.

La gran variedad de los sistemas y software de uso comercial determinan que cada instalación imponga diferentes limitaciones y requisitos, que además, evolucionan con el tiempo. Es importante hacer regularmente una evaluación de los riesgos que acompañan a los métodos específicos puestos en vigor.

Uno de los mayores riesgos que se corren en la instalación de una computadora es la confianza ciega. En la mayoría de los casos, el director y los consultores no saben lo suficiente acerca del funcionamiento interno del sistema, y lo dejan en manos de una persona de la organización. Tal hecho se puede dar a diferentes niveles: el director ejecutivo puede confiar en el financiero, y éste en el director de informática, hasta llegar al operador de estrada de datos y al técnico de mantenimiento a cargo del hardware y del software, o incluso hasta el personal de mantenimiento del edificio o de la persona de la

conserjería. No se debe olvidar que cada una de estas personas, si saben lo que están haciendo, están en actitud de llevar acabo acciones no autorizadas. Se deben utilizar controles sistemáticos para comprobar esas posibilidades.

Por desgracia, muchas compañías que llevan estrictos controles de seguridad cuando utilizaban sistemas contables manuales, desde que cambiaron al uso de la computadora parecen confiar ciegamente en las manifestaciones de la computadora.

El programador que haya diseñado un sistema es unos de los riesgos caros a la seguridad potencial de ese sistema. La razón estriba en que, aparte del programador, nadie más conocerá nunca a la perfección todas las características del programa en uso. En consecuencia el programador ha de ser honesto y toda la confianza, pues tiene capacidad para realizar numerosos actos delictivos o no autorizados que escapan a la detección.

Las trampas son posibilidades abiertas a acciones no autorizadas o no previstas, dejadas en el programa más por accidente que por diseño. Como un programador no puede garantizar nunca que un programa está totalmente libre de errores, puede haber muchos errores accidentales que permitan que usuarios no autorizados accedan al sistema y lleven acabo acciones no autorizadas. Tal situación es especialmente posible en el caso de grandes sistemas conectados a una red o que utilicen líneas de telecomunicación. En tales situaciones, se ha dado caso de que usuarios remotos han podido interrumpir en las grandes bases de datos y llevar acabo acciones no autorizadas, desde la modificación de las notas escolares hasta el acceso a archivos de gobierno y de compañías de alta seguridad. Una vez más, la revisión, el control de acceso y la auditoría, adecuadas, son las únicas formas de limitar riesgos.

La consecuencia a sacar de estos riesgos especiales es prohibir a los programadores todo acceso a su software una vez en uso en una computadora. Además, se les debe negar de forma rotunda el acceso a cualquier archivo comercial. Por desgracia, es muy difícil hacer cumplir esta restricción en las organizaciones pequeñas, en las que el programador externo o el "genio" informático de la casa es tan amigo de todos. Si se salta esta regla, se debe ser consciente al menos de las consecuencias potenciales.

## 2. 4. Procedimiento de seguridad.

No se debe de subestimar la importancia del componente psicológico a la hora de salvaguardar un sistema. En los sistemas pequeños, y siempre que sea posible, ha de haber una persona que tenga bajo su responsabilidad la integridad de los programas y datos de instalación, que esté suficientemente motivada para demostrar su interés de alguna manera visible. Si se sabe que hay una persona responsable de la seguridad y de la integridad del sistema, es menos probable que haya infracciones casi con independencia de las medidas específicas que se hayan tomado.

Se puede recurrir a muchos procedimientos de seguridad, Cada uno de ellos tiene, por si solo, una eficacia limitada, pero en conjunto componen un adecuado obstáculo a las interferencias, planeadas o accidentales, que afecten a la seguridad del sistema.

## 2. 5. Problemas de seguridad.

Las computadoras han tenido muchos problemas de seguridad. Los intentos aproximados por arreglar esos problemas han sido analizados durante mucho tiempo. Muchos problemas no son resueltos o los métodos para resolver han sido fallidos. Esto es, ha sido claro demostrar que la seguridad no es una cualidad a adicionar, pero que debe de ser considerada antes del diseño del hardware y el software de las computadoras. En lugar de tratar de arreglar el problema, puede ser mejor sacar cada cosa a flote y empezar por marcarlas. Esto incluiría:

- reducir la corriente de las computadoras,
- diseñar el hardware de las computadoras para apoyar en seguridad tanto como sea posible usando técnicas de encriptado,
- evitar "puertas" de seguridad (como la identificación) pero autenticar y verificar cada comando,
- evitar software basado en seguridad,
- y finalmente comprometerse en seguridad sólo cuando no halla solución conocida o cuando el resultado de la computadora sea completamente imparcial.

En otras palabras, puede ser mejor redefinir que la computadora es tal que no puede ser suficientemente protegida. Una pregunta sería si, es permisible cambiar la meta de los investigadores. Ciertamente modificando la definición de la computadora, ellos pueden asegurarse, esto parece ser nada mas que una excusa. Sin embargo, puede ser mejor tener en un tiempo de quince años algo seguro en hardware y software que tener versiones actualizadas de computadoras actuales que se vuelven inefectivas por el incremento en

fraude de computadoras.

De acuerdo a la definición de criptografía, parece ser la disciplina de ayuda más potencial, pero uno no debe excluir la influencia de otras áreas. Uno no debe olvidar que los temas más investigados son influenciados por otras áreas.

## 2. 6. Administración de seguridad.

Una buena administración del uso y resguardo de los activos computacionales implica un manejo seguro y adecuado de los recursos tecnológicos del proceso de datos. Las perdidas o daños a la información pueden resultar desastrosos, sobre todo cuando el sistema abarca aplicaciones de alto riesgo.

Los fabricantes de equipo, de sistemas operativos y de bases de datos, integran cada vez mejores herramientas de protección, pero hoy en día no existe una herramienta que pueda considerarse cien por ciento segura.

Esto se debe básicamente al crecimiento de las redes las cuales logran conectar equipos

de diferentes plataformas, hoy en día la tendencia de la computación esta totalmente orientada al uso de estas, ya no tan solo por las compañías sino también por las personas que tienen en sus hogares computadoras personales este hecho pone de manifiesto la importancia que hoy en día tiene la información, y por lo tanto de ahí su gran importancia que tiene la seguridad en los equipos de computo.

Hoy en día esa vulnerabilidad se esta convirtiendo en algo cada vez mas serio debido a la proliferación de las redes de computadoras. La tendencia en ese periodo ha sido asignar a las computadoras un número cada vez mayor de tareas de administración de nuestras actividades personales y de negocios. Las computadoras manejan de manera rutinaria la correspondencia más confidencial. Los sistemas electrónicos de transferencia de fondos hacen circular nuestro dinero en corrientes de bits. Los sistemas de prevención de colisiones de tráfico aéreo se han hecho cargo de gran parte de las tareas que antes eran dirigidas por los controladores de trafico. Los microprocesadores han sido incorporados a una gran variedad de dispositivos para proveer un control inteligente.

Conforme los sistemas de computación se han vuelto más complejos y sus aplicaciones se han difundido mas, también ha crecido la necesidad de proteger su integridad. En un principio, la protección se concibió como una añadidura a los sistemas operativos de multiprogramación para que varios usuarios, en los que no se podía confiar, pudieran compartir con seguridad un mismo espacio lógico, como un directorio de archivos o un mismo espacio fisico, como la memoria. Se ha evolucionado hasta los modernos conceptos de protección para aumentar la seguridad de cualquier sistema complejo que utilice recursos compartidos.

Por lo tanto, la confiabilidad de un sistema es directamente proporcional a la seguridad que este implementada en este; es decir, entre mas y mejores mecanismos de protección existan, se obtendrá un sistema con mayor grado de confiabilidad. Ahora bien esta confiabilidad de la que se habla no únicamente depende de estos procedimientos de seguridad, sino que también esta soportada en puntos de recuperación y reconstrucción de información en caso de un desastre de cualquier índole (físico, o lógico).

La confiabilidad de un sistema debe descansar en los siguientes puntos:

- *Confidencialidad (Privacidad)*
- *Disponibilidad (Perdida de datos)*
- *Integridad*

Es importante implantar medidas de seguridad para garantizar el funcionamiento ininterrumpido y libre de esos sistemas, ya que una interrupción podría causar grandes perdida económica o humanas, sobre todo cuando la organización depende de sus sistemas.

*Una empresa de tamaño mediano, que hace uso extenso de computadoras, fue puesta en liquidación como consecuencia de un accidente en el que un avión choco contra su centro de computo. La gran*

*dependencia en la actividad computarizada y la perdida de toda la información así almacenada, dejó a la institución en la imposibilidad de continuar realizando sus actividades comerciales'*[1]

## 2. 6. 1. Confidencialidad

Un aspecto importante cuando se aborda el tema de la seguridad es la privacidad o confidencialidad. Proteger a los usuarios de un mal uso de la información referente a una persona o una organización, trae consigo implicaciones de tipo moral y legal, ya que no debemos olvidar que una vez que se almacena la información en una computadora esta expuesta a ser explotada por cualquiera que tenga acceso al sistema o que logre penetrar al sistema en forma ilegal. Cuando se conocen datos confidenciales de una persona quien conoce dicha información si no maneja esta en forma adecuada puede ocasionar daños al propietario de los datos.

*A fin de estimar el costo operativo para el próximo mes en Consolidated Iron Works, se escribe un programa que utiliza los datos de personal para explotar costos a partir de salarios, horas normales de trabajo y días de vacaciones. Lucia Bermúdez, quien prepara el informe, no debería tener acceso a los antecedentes psiquiátricos del personal, que se conservan en línea junto con los datos de personal.*[2]

[1] Seguridad en Centros de Computo, Leonard H. Fine, pagina 13

[2] Diseño de Bases de Datos, Gio Wiederhold, pagina 684

Al igual cuando hablamos de acceder datos de una organización, el daño puede no tan solo ser de tipo, moral sino hasta económico.

*Un gran usuario de computadoras produjo análisis de ventas altamente confidenciales, los cuales mostraban la partición de ganancias, las acciones y otros datos competitivos. Para el procesamiento de estos datos se tuvo una precaución considerable y se empacaron los informes una vez, completos, para transmitirse a los usuarios. Se empleo personal de bajo novel para recolectar la información, uno de ellos no pudo resistir la tentación cuando lo enviaron a buscar informes y cierta información fue fotocopiada. Sólo por casualidad, la institución se entero de que algunos datos se habian vendido a un gran competidor.*[3]

En los años de 60's en los Estados Unidos, se tuvo el primer reconocimiento oficial acerca de la privacidad. En 1972 también en los Estados Unidos, la Academia Nacional de Ciencias reconoció el estudio sobre los sistemas computacionales. En 1973 en los Estados Unidos, el informe del Departamento de Salud sobre educación y bienestar propone acciones que ayudan a la privacidad del individuo. Este informe llamado "Principios Fundamentales de la Información Practica" propone cinco puntos como guía para el desarrollo de normas y leyes concernientes a la privacidad:

- *Los sistemas y sus registros serán secretos*
- *Como será usada la información de los registros*
- *Prevenir el acceso de la información ya que podría obtenerse para un propósito y ser usada para otro diferente*

[3] Seguridad en Centros de Computo, Leonard H. Fine, pagina 71

- *Solo la persona indicada podrá corregir algún dato*
- *Cualquier organización que usa y disemina la información del personal asegurara la rehabilitación de los datos, tomando precauciones para prevenir el mal uso de estos.*

Con el informe se distinguió dos tipos de sistemas: primero, el sistema automatizado de personal y el segundo, para la investigación y reporte de estadísticas. Con esto se promulgo una legislación estableciéndose un 'Código de Información Practica' para todos los sistemas automatizados, como sigue:

- *El código deberá definir requerimientos específicos de salvaguarda para la información*
- *El código deberá prohibir la violación de cualquier requerimiento de salvaguarda*
- *El código deberá prevenir la información contra faltas criminales y civiles*
- *El código deberá prevenir la violación de cualquier requerimiento de seguridad*
- *El código deberá proveer normas para tener correcta la información y recobrarla en caso de algún daño a esta*

Consideremos como ejemplo el conocimiento de la información de algún paciente, en el cual el medico la estudiara para diagnosticar su enfermedad. La existencia de un libre acceso de datos representa un gran riesgo y puede ser mal usado, pero su conocimiento podría salvar una vida. Por lo tanto mientras haya un acceso libre de datos el factor

permanente de privacidad podrá ser roto con el mal uso o destrucción de estos.

Debemos mencionar que actualmente en México la legislación en materia de informática es casi nula, esta se menciona en algún rubro de los derechos del autor, pero el delito informático no cuenta como tal en nuestras leyes, en el capítulo ocho se abordara este tema mas ampliamente.

*Concepto.*- Podemos definir a la privacidad como el derecho que tienen las personas y las organizaciones a decidir por si mismos cuando, como y porque se permite el conocimiento de su información a terceros.

Debemos tener en cuenta los siguientes puntos importantes:

- *¿Cómo y qué información deberá ser almacenada?*
- *¿Cómo y porqué será usada?*
- *¿Cómo podrá ser accedida, modificada y procesada?*

El concepto de la privacidad es aplicable por las siguientes razones:

- *La utilización de sistemas de información en línea*

- *El uso de redes de comunicación*
- *La aplicación de bases de datos*
- *Los sistemas de procesos distribuidos*
- *El impacto organizacional ocasionado por la violación de la información*
- *Los aspectos económicos con respecto al valor de la información*

La disponibilidad de las computadoras personales ha traído el gusto por las computadoras, al publico en general. Millones de personas han tomado cursos de computación en las universidades, han leído libros de computación, han comprado computadoras personales o han usado sistemas computación en su trabajo. No podemos manjar sistemas de información sin reconocer que gran parte de la fuerza laboral de la informática tiene una profunda vinculación con el manojo de las PC's. Las PC's permiten a los oficinistas trabajar para beneficio propio. Estas personas realizan actividades como, la composición de resúmenes, correspondencia para mejorar la imagen publica y personal, la autocapacitaciòn etc.

La privacidad personal dicta la imposición de estrictas normas acerca de entrar, recuperar y almacenar información. La memoria privada de una persona debe estar protegida, y el individuo debe borrar toda esta información cuando abandone el empleo.

En la era de la PC la protección y preservación de la privacidad de los archivos en una computadora es cada vez más difícil, debido a que existen muchas herramientas que permiten acceder datos en forma muy sencilla, aunado a esto nos enfrentamos a la gran proliferación de las redes, conectando sin mayor problema equipos de diferentes plataformas.

Hoy en día se incrementa en forma acelerada el trafico de datos por lo que los expertos llaman el *ciberespacio*. Actualmente es muy sencillo y relativamente barato que cualquier persona se pueda conectar a una red publica de tipo satelital, teniendo a su alcance un sinnúmero de aplicaciones, juegos, conferencias, documentales y todo tipo de información que nos podamos imaginar. Basta con tener una PC, con multimedia, un módem y pagar una suscripción y mensualidad a la RED que se seleccione, una vez contratado esto el propio usuario se da de alta con el nombre que a él le parezca mejor, esta tecnología da una combinación de privacidad y anonimato.

El anonimato puede ser libertador y divertido cuando hablamos de elecciones, o de una fiesta de disfraces, pero el anonimato no es tan divertido cuando hablamos de robo o fraude. Es importante no confundir el anonimato con la privacidad.

*En una caricatura del NEW YORKER, dos perros están sentados ante una computadora personal. Uno de dice al otro, "Lo bueno de estar en la Internet es que no saben que eres un perro".*

### 2. 6. 2. Integridad

En este rubro es importante que exista una supervisión para no permitir que la información y las Aplicaciones sean transgredidas, en otras palabras se debe tener la seguridad que cuando accedemos al sistema, tanto las aplicaciones como la información son totalmente confiables. La información en la computadora debe estar protegida contra destrucción o alteración con fines indebidos, y la introducción accidental de inconsistencia.

La perdida accidental de la consistencia de los datos puede deberse a:

- *Caídas durante el procesamiento de las transacciones*
- *Anomalías por acceso concurrente a la base de datos*
- *Anomalías que resultan de la distribución de datos entre varias computadoras*
- *Un error lógico que viola la suposición de que las transacciones respetan las*

*limitantes de consistencia de la base de datos.*

Es más fácil prevenir la perdida accidental de consistencia de los datos que prevenir el acceso mal intencionado. Algunas formas de acceso indebido son:

- *La lectura de datos sin autorización*
- *Modificación no autorizada de datos*
- *Destrucción no autorizada de datos.*

Para proteger la base de datos es necesario adoptar medidas de seguridad en varios niveles:

***Físico.-*** La localidad o localidades que contienen a los sistemas de computo deben protegerse físicamente contra la penetración de intrusos.

***Humano.-*** Debe tenerse mucho cuidado al conceder autorización a los usuarios para reducir la probabilidad de que un usuario autorizado permita el acceso a un intruso a cambio de sobornos o favores.

***Sistema Operativo.-*** Aunque el sistema de dase de datos este bien protegido, si no se protege en forma adecuada al sistema operativo este puede servir para obtener acceso a una porción limitada de la base de datos. Esto es posible también que algunos usuarios se les permita consultas, pero se les prohibe modificar la base de datos. El sistema de base de datos tiene la responsabilidad de garantizar que no violen estas restricciones.

Los valores almacenados en la computadora deben satisfacer ciertos tipos de limites de consistencia. Por ejemplo, el saldo de una cuenta no debe bajar de un mínimo previamente especificada, o el número de horas que puede trabajar un empleado no debe exceder un numero especifico. Es vital que estos limites no sean violados.

### 2. 6. 3. Disponibilidad

Es determinante tener la información correcta en el momento que se requiera, es por eso que se deben prever cambios accidentales o destrucción de la información. Los datos hoy en día son considerados como parte del activo de las organizaciones, de ahí que haya crecido su importancia, considerando el crecimiento tan acelerado de los sistemas de redes y los equipos multi-usuarios con una gran facilidad de conectarse entre sí sin importar su plataforma (Minis, Mainframes, PC), él tráfico de datos transitando por el llamado *ciberespacio*, hace que el riesgo de perder datos se incremente de forma inimaginable.

Cuando hablamos de perdida de datos debemos analizar las siguientes causas:

- *Fuerza mayor*
- *Hardware y/o Software*
- *Humanos*
- *Errores*
- *Negligencia*

**Fuerza mayor.-** Podemos perder datos debido a la presencia de fenómenos que no están a nuestro alcance controlar, tales como; terremotos, guerras, inundaciones, plagas, etc.

**Hardware y/o Software.-** Perdida de datos ocasionada por el mal funcionamiento de la unidad central de proceso, por error en discos, disquetes, cintas, virus informáticos. O perdida ocasionada por problemas ocultos en los procesos (Software), montando cintas equivocadas, liberando software no probado adecuadamente (Rutinas deficientes ocultas o no identificadas).

**Humanos.-** Es bastante común que el ser humano ocasione perdida de los datos, y estos pueden ser ocasionados por errores o por negligencia.

Cuando se están captando datos se puede desconocer la forma correcta de ingreso de los datos, o por descuido se puede utilizar cintas discos o disquetes equivocados, por falta de atención se pueden omitir datos. Cuando el personal se siente molesto con la organización nos puede causar daños a los datos, con causa de conocimiento.

*Un operador agraviado alteró de manera deliberada las cintas magnéticas con un imán mientras desmontaba las copias de seguridad de la unidad de cintas, tenia la intención de dañar la instalación de forma irreparable por cierto tiempo. Afortunadamente, un pedido no esperado de copias de seguridad revelo que estaban alteradas y dejo sus acciones al descubierto.*

La mayor parte de perdida de datos ocasionados por estas causas pueden ser controladas estableciendo un adecuado sistema de respaldos, dentro y fuera de sitio, contando con procedimientos adecuados de control, soportados por los manuales del sistema.

### 2. 6. 4. Intrusos en el sistema

Un problema en la seguridad de los datos la representan los intrusos en el sistema, estos se presentan en dos grandes grupos a reconocer:

- *Los Intrusos Pasivos, los cuales solo desean acceder información a la cual no tiene autorización*

- *Los intrusos activos, estos son más maliciosos ya que estos buscan poder alterar los datos no autorizados.*

Cuando hablamos de identificar a los intrusos en el sistema, debemos primeramente determinar que tipo de intruso es el que buscamos, dependiendo de este serán los recursos que debamos emplear en nuestra búsqueda algunos de los tipos más comunes son:

- *Curiosidad de usuarios no técnicos.-* Los sistemas de tiempo compartido, así como las redes llámense esta LAN (Red Local), WAN (ejemplo: Internet), o las nuevas redes llamadas VIRTUALES, asignan terminales a los usuarios del sistema, (Estas terminales pueden ser PC's o terminales de un equipo multi-usuario), considerando la naturaleza humana de querer saber: "de que se trata?", intentan acceder información a la cual no tienen autoridad, y muchas de las veces lo logran sobre todo cuando el sistema no ha colocado obstáculos que impidan que esto suceda. No debemos olvidar la gran proliferación de los juegos en las computadoras personales, esto motiva aun más la curiosidad de estos usuarios.

- *Curiosidad de personas enteradas.-* Los estudiantes, los programadores, operadores, y de mas personal técnico, además hoy en día los usuarios finales que no trabajan directamente en el área están cada vez mas capacitados técnicamente, esto motiva a estas personas a considerar un reto el violar la seguridad de los sistemas.

- *Intento decidido de ganar dinero o perjudicar a la organización.-* Algunas personas capacitadas técnicamente, intentan violar la seguridad de los sistemas, con el firme propósito de ganar dinero.

*Un experto en programación, una gran empresa, manejaba programas y archivos a través de contactos. De esta manera, no solo cometió fraude contra la compañía por cuantiosas sumas de dinero, sino que uso cantidades considerables de tiempo de computadora no autorizado.*[4]

- *Espionaje Comercial o Militar.-* Se refiere a un intento serio y bien fundado por parte del competidor o país, por robar programas secretos de mercado, patentes, tecnología, etc.

Con frecuencia este intento implicará la intercepción de líneas telefónicas o hasta la instalación de antenas dirigidas a las computadoras para captar su radiación electromagnética. Los recursos que las organizaciones inviertan en la seguridad de sus sistemas, deben ser enfocados a los tipos de intrusos que de desea frenar, no es lo mismo un intruso que solo deseo enviar un mensaje de buenos días a otros usuario, que un intruso que busca obtener datos de la cartera de los clientes para poder venderla a la competencia, o aun mas cuando se habla de robo de tecnología.

[4] Seguridad en Centros de Computo, Leonard H. Fine, pagina 14

En la actualidad, y principalmente en las computadoras personales, nos encontramos con un intruso oculto llamado *Virus Informático*, este normalmente tiene intenciones dañinas, y en el menor de los casos es solo afán de hacerse notar.

*En una compañía multinacional un empleado elaboró un programa que automáticamente entraba al correo electrónico internacional y dejaba un mensaje de felicidades. Al momento en que la persona que recibia el mensaje entraba a su correo electrónico (para lo que tenia que teclear su llave de seguridad) encontraba un mensaje de felicitación por la Navidad. Automáticamente el programa tomaba el directorio del usuario, enviaba mensajes idénticos a todas las personas que se encontraban en el directorio. Esto, que aparentemente tuvo buenas intenciones, genero mensajes en forma exponencial bloqueando toda la red internacional de la compañía.*[5]

El crecimiento de los fraudes por computadora han hecho patente que la potencialidad de los crímenes crezca en forma más rápida que los sistemas de seguridad, los motivos de los delitos por computadora normalmente son; Beneficio personal, Beneficios para la organización, Síndrome de Robin Hood (por beneficiar a otras personas), jugando a jugar, fácil de desfalcar, odio a la organización, mentalidad turbada, etc.

José Antonio Echenique, en su libro *Auditoria en Informática*, establece cuatro factores que han permitido el incremento en los crímenes por computadora y son:

[5] Innnnnn

- *El aumento del número de personas con conocimientos informáticos*
- *El aumento del número de empleados que tiene acceso a una computadora*
- *La facilidad en el uso de los equipos de computo*
- *El incremento en la concentración del numero de aplicaciones y, consecuentemente de la información.*

Estos factores, aunque es objetivos de todo centro de computo, también constituyen una posibilidad de uso con fines delictivos. Actualmente el 95 % de los delitos cometidos por computadora se han descubierto por accidente y la gran mayoría no han sido divulgados para evitar dar ideas a personas mal intencionadas.

*Entre los crímenes mas conocidos están, el del Banco Wells Fargo Co. (21.3 millones de dólares), en el cual se evidencio que la protección de los archivos es todavía inadecuada, y la publicada el 17 de septiembre de 1987 en la que dos alemanes entraron a los archivos confidenciales de la NASA*[6]

*Un ejemplo de destrucción de la información de la compañía USPA & IRA de Fort Worth; cuando despidieron a un programador en 1985, este dejo una rutina que destruía mensualmente la información de las ventas. Este incidente provocó el primer juicio en los Estados Unidos contra una persona por sabotaje a la computadora*[7].

[6] Auditoria en Informática, José Antonio Echenique, pagina 102

[7] Auditoria en Informática, José Antonio Echenique, pagina 103

*David LaMàcchia, un estudiante de 20 años del MIT, fue levado a los tribunales por utilizar los servidores de la Internet de la universidad para distribuir programas protegidos por la ley de propiedad literaria. LaMacchia hace encara a cargos de conspiración para cometer el fraude. Si queda convicto recibirá tantos como cinco años de prisión federal, mas multas que van mas allá del alcance de los créditos escolares ordinarios. El punto de defensa es que LaMacchia no recibió dinero por sus esfuerzos y no cargo o descargo por teleproceso ninguno de los programas protegidos. Tan solo operaba un tablero de boletines que hacia que tales actividades fueran posibles. El caso de LaMacchia esta pendiente.* [8]

## 2. 6. 5. Concepto de Seguridad Total

El uso cada vez más frecuente de los sistemas de computo, motivado por el crecimiento acelerado de la tecnología esta permitiendo, poner al alcance ya no solo de las organizaciones pequeñas sino hasta de las personas mismas el simplificar las cargas de trabajo en las mas variadas áreas donde se aplican, lo cual en consecuencia aumenta los riesgos de abuso en el manejo de las computadoras, o bien incrementa el riesgo de fraude, robo, sabotaje o interrupción en las actividades de computo.

Según Leonar H. Fine en su libro *Seguridad en Centro de Computo*, existen factores que han modificado el contexto dentro del cual se usan las computadoras y han aumentado el

[8] Computerword Mayo de 1995, pagina C-14

nivel de seguridad que se requiere y son:

- *Concentración del procesamiento y aplicaciones más grandes y de mayor complejidad, las cuales son parte integral de la institución.*
- *Dependencia en el personal clave.*
- *Desaparición de los controles tradicionales.*
- *Terrorismo urbano e inestabilidad social*
- *Mayor conciencia de los proveedores de computadoras.*

si a esto anexamos los siguientes puntos:

- *Crecimiento acelerado en el uso de las computadoras personales*
- *Crecimiento en las redes computacionales*
- *Tener interconectividad entre equipos de plataformas diferentes.*

Estos tres últimos puntos han incrementada considerablemente los riesgos de fraude, robo, sabotaje, o uso inadecuado de los datos confidenciales referentes a una persona o de una organización.

### 2. 6. 5. 1. Enfoque Tradicional

La seguridad tradicionalmente es tratada considerando solo las áreas donde los resultados se pueden ver, medir y cuantificar, de tal modo que las áreas que en este rubro destacan y que son las que en su mayoría aun hoy día son tomadas en cuenta, son:

- *La seguridad física, que incluye la seguridad de acceso y contra incendios*
- *La seguridad de los datos y los archivos.*

Esto trae como consecuencia, que actualmente en muchas de las organizaciones se requiere de procedimientos completos de control de acceso, los cuales incluyen personal de seguridad, sistemas de control de tarjetas de acceso, reconocimiento de voz, etc. La actividad de salvaguardar los archivos y ubicarlos en lugares distantes para su almacenamiento se hace de forma cuidadosa, pero todas estas actividades hacen creer a la compañía que cuenta con un sistemas adecuado y eficiente en el renglón de la seguridad.

*Una gran institución de defensa contaba con un nivel muy alto de seguridad para sus computadoras. El perímetro de la instalación estaba protegido por una doble barda de seguridad de mas de tres metros de altura. El área comprendida entre el perímetro y la instalación se encontraba vigilada. Sin embargo, dentro de la instalación misma, los 250 empleados - considerados de confianza - gozaban del acceso*

*prácticamente ilimitado a todas las instancias. Aunque la confianza hacia el personal es desde luego un elemento clave, la dependencia ciega en él resulta inadecuada.*[9]

### 2. 6. 5. 2. Enfoque Amplio de la Seguridad

El punto de vista tradicionista concede mayor atención a los aspectos de seguridad visible, pero una visión mas objetiva de la seguridad en computación exige considerar una amplio numero aspectos para que sea realmente efectiva (ver figura 2.1.).

Según Leonard H. Fine en su libro *Seguridad en Centros de Computo*, existen dos grandes áreas que se deben incorporar a tal enfoque y son:

- *Aspectos administrativos*
- *Aspectos técnicos y de procedimiento.*

Los aspectos claves, los resume de la siguiente manera:

[9] Seguridad en Centros de Computo, Leonard H. Fine, pagina 16

***Elementos administrativos***:

- *Política definida en la organización sobre seguridad*
- *División de las responsabilidades*
- *Seguridad física y contra incendios*
- *Políticas hacia el personal*
- *Seguros*

***Elementos Técnicos y de Procedimiento***

- *Seguridad de los sistemas (equipo y programación, redes y sistemas terminales)*
- *Seguridad de las aplicaciones, incluyendo la seguridad de los datos y los archivos*
- *Estándares de programación y operación de los sistemas*
- *Función de la auditoria*
- *Plan de simulacro para desastres*

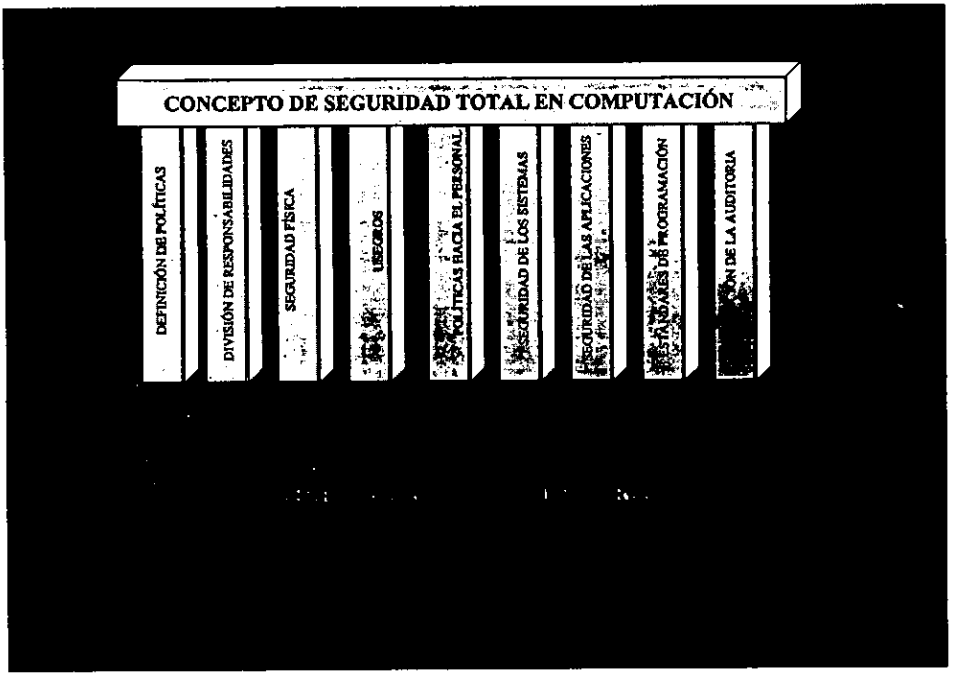



**Fig. 2.1.** Concepto de Seguridad Total

Todos los elementos mencionados, por si solos no tienen una importancia exclusiva, esto depende de cada instalación especifica. Es decir según la instalación se dará mayor peso a algunos de los puntos antes mencionados, pero si se excluye alguno se estarán dejando vacíos o puertas falsas en el manejo y control de la seguridad.

El presente trabajo no pretende hacer mucho énfasis en los aspectos donde se refiera a la seguridad física ya que esta ha sido tratada extensamente sobre todo con el enfoque

tradicional, por lo tanto se podrá especial interés en aquellos puntos donde la seguridad no es tan visible y mucho menos tangible, pero que son aspectos que hoy en día merecen todo el interés y la importancia ya que si se decidan como ha sido hasta ahora estaremos en peligro constante de sufrir largos periodos inhabilitados en nuestra actividad de computo, y esto puede ocasionar grandes perdidas.

*Seguridad de los Sistemas.*- Esta se refiere básicamente a la seguridad del equipo de computo, incluye, aspectos como el equipo fisicamente, los programas de uso general, las redes, las líneas de comunicaciones, las terminales y los programas generales directamente asociados.

Las puertas falsas en los programas representan una amenaza substancial y fundamental para la seguridad del sistema. Actualmente se han desarrollado herramientas de ayuda para la programación (QUERYES, Etc.), las cuales constituyen una amenaza para la seguridad. Así como la programación esta constituida en una área muy complicada donde solo un pequeño grupo de especialistas maneja los conocimientos correspondientes. El mayor riesgo en las redes es el acceso no autorizado a estas, con el propósito de obtener información confidencial o de hacer uso indebido de las instalaciones de procesamiento la parte esencial del sistema de seguridad es el análisis detallado del desempeño de los sistemas.

Algunos proveedores de equipos ofrecen programas completos de seguimiento. También

existen bastantes monitores de equipo y programación. Todos ellos facilitan el Monitoreo del desempeño del equipo y los sistemas.

La seguridad de los sistemas constituye una parte muy técnica y compleja de la seguridad. Se necesita un enfoque metódico para identificar las debilidades, definir los controles y garantizar que estos se lleven a cabo y se vigilen. Esta revisión debe incluir el equipo y los programas de la computadora principal, de redes y de terminales. El seguimiento del desempeño de la computadora central, las redes y las terminales es una función administrativa y de seguridad importante. Se deben considerar cuidadosamente los recursos disponibles. Además se requiere seguimiento de todo intento de acceso indebido a los programas o archivos.

*La seguridad de las aplicaciones*.- El alcance de la seguridad de los datos y los archivos de una aplicación incluye tanto el trabajo de la computadora como otras labores. Se necesita considerar en forma cuidadosa la relación de las actividades de computo con las que no lo son. Los controles del usuario no se pueden delegar al departamento de computo. El control que se mantiene en el centro de computo es, en general, muy detallado, en especial respecto a los archivos y programas. La revisión constante de los controles de la aplicación es una parte importante de la función de auditoria.

*Estándares de programación y operación de sistemas*.- Los estándares de sistemas,

programación y operación, así como la documentación, tienen efectos de suma importancia en la seguridad. Los requisitos de seguridad se deben revisar en forma periódica como parte del proceso de planeación computacional a largo plazo, así como también el desarrollo y la realización de las aplicaciones individuales. La existencia de métodos de trabajo efectivos mejora la seguridad y ofrece la documentación adecuada como un derivado (CASES). Se debe considerar de manera cuidadosa el acceso a esta documentación, a fin de reforzar la división de las responsabilidades.

*Función de los auditores.*- Las funciones de auditoria cumplen un papel importante en la seguridad en computación. Los principales aspectos que se deben considerar son:

- *El alcance de la auditoria interna y externa*
- *La relación entre auditoria interna y externa*
- *La función de la auditoria interna dentro de las fases de desarrollo y operativa de los sistemas*

Durante mucho tiempo se ha reconocido la necesidad de contar con la participación extensa de la auditoria interna en los sistemas computacionales. En la practica muy pocas instituciones han logrado éxito. Probablemente el obstáculo más grande se el compromiso de la gerencia, el personal de computo y los auditores para aplicar los procedimientos.

### 2. 6. 5. 3. La calidad total y la seguridad

El mundo actual vive un proceso acelerado de cambio y transformación, que fomenta una mayor interdependencia en el ámbito mundial. Se abren posibilidades a los países (empresas) con la capacidad de adaptación a estos cambios y que puedan insertarse eficientemente a las nuevas corrientes. El factor que determinara esta capacidad será el de *Competitividad.*

Actualmente todo el mundo habla de calidad como la llave mágica que nos va a resolver todos nuestros problemas, en nuestros días las compañías están intentando cambiar sus formas de trabajo orientadas por esta nueva tendencia, se preocupan por establecer este concepto en toda su organización.

Previamente a la época industrial, los juicios emitidos acerca de la calidad total versaban, entre otras cosas, sobre el aspecto estético y los gustos de la época. Cuando con la revolución industrial, muchos de los pequeños talleres pasaron a convertirse en pequeñas fabricas, esto dio inicio a la evolución de los procedimientos para valorar y entender la calidad. A partir de entonces, dichos procedimientos han evolucionado en tres diferentes formas; Calidad por inspección, control estadístico de la calidad y el aseguramiento de la calidad, y calidad total como estrategia competitiva.

En fin todos coinciden en la importancia de trabajar muy cerca con el proceso de producción y reducir su variabilidad, de adaptar las posibilidades del equipo hacia las especificaciones del cliente, de trabajar en equipo para la resolución de problemas y acciones de mejora. Hoy en día, el rumbo hacia la calidad total compromete un profundo y significativo cambio de la forma de pensar, de trabajar y de administrar.

Esta filosofía acepta que el ser humano es el origen y el final en la actividad de toda organización, reconoce al trabajador como persona pensante, define que una mayor productividad, mejores resultados, y mejor calidad de vida en el trabajo se logra mediante el involucramiento y la participación personal, al trabajar en equipo se fomenta la integración y el sentido de pertenencia, además acepta el respeto y consideración humanos, esta filosofía acepta que no es solo la calidad del producto, sino que calidad significa también:

- Calidad personal
- Calidad en proceso
- Calidad en instalaciones
- Calidad en relaciones interpersonales
- Calidad de vida

Las organizaciones para asegurar el éxito están implementando los siguientes aspectos:

- Determinar claramente los objetivos
- Apoyo y soporte de la dirección
- Participación de acuerdo a la estructura natural de la organización
- Entrenamiento del personal
- Acceso a la información
- Motivación y reconocimiento

Es bastante evidente que uno de los aspectos clave en esta filosofía descansa en el personal. Todos estos aspectos mencionados en el concepto de *Calidad Total*, son por demás aceptables, y consideramos que esta filosofía no es una moda sino más bien es el resultado obligado del cambio de paradigmas de nuestra actualidad. Ahora bien cabe mencionar que en esta apertura de las compañías hace que el personal cuente con capacidad para tomar decisiones lo que han llamado *empowerment*, implica el poder tener acceso a mas información clave de la compañía, y en este concepto no se menciona con claridad el aspecto de la *privacidad* de la información.

Hemos analizado estos conceptos, y nos damos cuenta que son polos opuestos, pero consideramos que puede ser más peligroso el tratamiento que se le da a la información en el enfoque de alto desempeño. Imaginemos que no podrá hacer un empleado mal intencionado conociendo demasiados datos confidenciales de la organización. Debemos aceptar que aun no existe una *Cultura de Seguridad Informática*, y que a pesar de los avances de la tecnología, quienes manejan esta son las personas y que por lo tanto no

sirve de mucho tanto avance tecnológico, tantos enfoques de calidad hacia los productos, tanto preocuparse de la calidad de vida de los empleados, si no nos procuramos por hacer una cultura de seguridad informática en todo nuestro personal, estaremos en peligro ser saboteados o robados.

### 2. 6. 5. 4 Cultura de seguridad Informática

Consideramos que tanto en las organizaciones, las escuelas, y también en nuestros hogares debemos crear un cambio en la forma de pensar, debemos sensibilizar a nuestro personal, a nuestros alumnos y a nuestros hijos acerca de la *Privacidad* de la información referente a una persona u organización, consideremos que nos encontramos de lleno en la era informática, debemos implantar cursos, platicas, boletines, circulares, y todos aquellos elementos que estén a nuestro alcance para empezar a formar una *Cultura de Seguridad Informática*. Las compañías y las escuelas deben establecer planes de capacitación a su personal buscando rescatar valores como:

- Honestidad
- Cooperación
- Seguimiento de las normas y políticas de seguridad que establezca la organización

Las escuelas deben incorporar en sus planes de estudio aspectos como:

- La seguridad en sistemas computacionales
- La privacidad y sus implicaciones legales
- La privacidad y sus implicaciones morales

Estos conceptos deben ser impartidos a todas las carreras y no sólo a las afines a la informática, ya que el objetivo principal es crear una mentalidad sana y profesional en este aspecto.

Por otra parte las organizaciones deben incorporar acciones que den la importancia a la seguridad de la información a la que los empleados tengan acceso, acciones como:

- La seguridad informática en las platicas de inducción
- La seguridad informática en el reglamento interno de trabajo
- Implantar procedimientos de revisión continua.

Es vital para lograr esto que el empleado el alumno, reconozcan la importancia y el valor funcional y económico que tienen los datos que le son encomendados, debe aceptar el empleado que es su responsabilidad ante la organización el buen uso y resguardo de

estos datos y que la divulgación mal intencionada de estos debe hacerlo acreedor a una sanción preestablecida.

Debemos formar esta cultura de *Seguridad Informática*, para evitar que sigan cometiéndose fraudes, sabotajes, robo de software, divulgación mal intencionada de información confidencial. No debemos conformarnos y dejar la seguridad en manos de los avances tecnológicos ya que estos son creados y manipulados por las personas, y no importa que tan segura se crea la tecnología siempre existen personas con la firme intención de violar esta seguridad. Por los tanto debemos aceptar que la seguridad debe radicar en el pensamiento y en los principios de la humanidad, si logramos esto podremos aceptar que contamos con sistemas seguros.

# CAPITULO III

# ESTANDARES EN LOS SISTEMAS

La necesidad que existe sobre nuestros sistemas de: innovar, mantener, actualizar y conocerlos mejor; ha obligado al analista de sistemas a desarrollar con sistemas que les facilite lograr todos estos objetivos. Para ello hace falta establecer una serie de reglas o normas que sean acatadas por todos aquellos programadores que intervengan en el desarrollo de un proyecto. Es por eso que este capítulo intenta dar una idea de lo que implican los estándares en una aplicación y como es que estos nos ayudan a contar con sistemas que pueden llegar alcanzar un alto nivel de eficiencia y vida útil.

## 3. 1. Calidad en la programación de un sistema.

La calidad de los programas es una preocupación primordial de los ingenieros de programación, las características importantes de la calidad dependerán, obviamente, del producto en particular. En algunos casos, la transportabilidad del producto entre diversas máquinas podrá ser un atributo de importancia capital, mientras que en otras ocasiones el uso eficiente de la memoria puede ser lo fundamental; por otro lado, existen algunas características de calidad que son fundamentales en todo producto de programación; entre ella están la utilidad, claridad, confiabilidad, eficiencia y economía.

El factor más importante de la calidad de un producto es su utilidad, es decir, que el producto de programación satisfaga las necesidades del usuario. Esto podrá parecer obvio, pero muchos sistemas entregados a los usuarios con frecuencia no desempeñan las funciones esperadas; este problema es síntoma de la pobre comunicación existente entre el cliente, los usuarios y los ingenieros de programación. La planeación cuidadosa, el análisis la participación del cliente son obligatorias para el desarrollo de sistemas útiles.

Los sistemas deben estar escritos con claridad y ser fáciles de entender. Como se notará las pruebas y las actividades de mantenimiento consumen gran cantidad del presupuesto del sistema. La clave para realizar un sistema fácil de probar y mantener radica en hacerlos comprensibles, de aquí la sugerencia de que un Administrador de Aplicaciones se diseñe y se desarrolle con una estandarización y parametrización adecuada.

La obtención de una buena calidad de un sistema esta basado también en el nivel tecnológico. El nivel tecnológico utilizado en un proyecto de programación incluye aspectos como selección del lenguaje, ambiente computacional, prácticas de programación y herramientas disponibles. Los lenguajes de programación modernos proveen características mejoradas para la definición y manejo de datos, estructuras de construcción mejoradas para la definición del flujo de control, mejores facilidades de modularización, manejo eficiente de condiciones y facilidades para la programación concurrente.

El ambiente computacional se refiere al conjunto de características del equipo y los programas disponibles para el desarrollo, uso y mantenimiento del sistema. La estabilidad y disponibilidad del ambiente computacional influye notablemente en la productividad y la calidad del sistema.

Las técnicas modernas de programación comprenden el uso de un análisis sistemático y técnicas de diseño, nomenclatura apropiada, estandarización, programación estructurada, técnicas de depuración y estudio de documentos y código fuente y pruebas sistemáticas. Las herramientas de programación van desde las herramientas más elementales como ensambladores y depuradores sencillos hasta ambientes totales de programación que incorporan las herramientas para la administración y el control del desarrollo del proceso.

## 3. 2. Conceptos técnicos de estándares.

De acuerdo con el diccionario, un estándar es "aquello que se ha fijado y establecido por una autoridad, por la costumbre o el consentimiento general, como una regla para la medición de una cantidad, de una extensión, de un valor, o de una calidad". Los estándares se expresan generalmente en forma escrita o gráfica, pero también pueden consistir en modelos u objetos.

Si el estándar está expresado en el lenguaje de las ciencias físicas o naturales, se le llama un estándar técnico. En la profesión de la ingeniería se emplean diferentes clases de estándares técnicos. La definición de términos, de símbolos y de unidades les es muy familiar. Los estándares dimensionales son convenios sobre la forma, el tamaño y el número de estilos de los artículos comerciales. La calidad de un producto manufacturado es un argumento de utilidad, o sea de especificación de dicho producto. Análogamente se tienen estándares para el servicio, como en el caso de servicio de mensajes telefónicos, se cuenta con el estándar de eficiencia.

Los ingenieros de sistemas están relacionados principalmente con los estándares técnicos de eficiencia para los productos o servicios. Estos estándares establecen en términos cuantitativos (donde sea posible) las características que deben tener los sistemas para satisfacer sus objetivos. Los estándares son un convenio entre cada uno de los que toman parte en la planeación, en el desarrollo y operación del sistema, y finalmente con el cliente. En todo estándar se debe incluir distribución estadística que represente los parámetros que se puedan medir para la calidad del producto o servicio.

La simple formulación y la unidad de medida de la calidad comprensible, constituyen la mayor parte del trabajo del ingeniero de sistemas. También puede recomendar los niveles para los diferentes estándares técnicos, y debe estar capacitado para responder las preguntas que le haga la gerencia sobre la utilidad de los cambios de los estándares. La determinación final de los estándares de calidad es la responsabilidad de la gerencia. Los gerentes deben aceptar las sugestiones de los usuarios, puesto que éstos disfrutarán del servicio o producto, únicamente en la extensión que a ellos les plazca y que consideren que sea una ventaja. Por lo tanto, cualquier objetivo estándar tiene que estar basado sobre la determinación subjetiva del proceder humano y su preferencia. El trabajo de la ingeniería de sistemas, los estándares técnicos sirven a veces como planes permanentes, simplificando el trabajo de desarrollo de requisitos para los sistemas. Los estándares técnicos se pueden emplear como objetivos para nuevos sistemas. Por los tanto, los métodos para establecer los estándares técnicos están intimamente relacionados con los métodos para el establecimiento de los objetivos, puesto que ambos comprenden temas de utilidad y de medición subjetiva.

### 3. 3. Estándares y directrices.

Se pueden desarrollar varios tipos de estándares y principios generales para mejorar el mantenimiento del software. Los formatos estándar para los documentos de requisitos y las especificaciones de diseño, las convenciones de codificación estructurada, y los formatos estandarizados para los documentos de apoyo como el plan de prueba, los principios de operación, el manual de instalación y el del usuario contribuyen a la comprensión y por lo tanto al mantenimiento del software. El grupo de control de calidad puede tener la responsabilidad de desarrollar y hacer cumplir los distintos estándares y principios generales durante el desarrollo del software. Los administradores pueden asegurar que las marcas de logros se estén cumpliendo, y que los documentos se estén desarrollando a tiempo, junto con las especificaciones de diseño y el código fuente.

Los estándares de codificación son especificaciones para un estilo de codificación preferido. Dada la situación de elegir los caminos para lograr un efecto, se especifica un camino preferido. Los estándares de codificación a menudo son vistos por los programadores como mecanismos para restringir y devaluar las habilidades para resolver problemas creativos de los programadores. Este argumento es normalmente esgrimido por los programadores que no entienden el espíritu o la intención de un buen estilo de codificación. La creatividad siempre ocurre dentro de un marco de trabajo básico de estándares. Los artistas siguen los principios básicos de estructura y composición, los poetas se apegan al ritmo y métrica del lenguaje, y los músicos utilizan progresiones de un acorde fundamental. Sin marco de trabajo de estándares que guíen y canalicen una actividad, la creatividad se vuelve un caos sin sentido.

Así, es deseable que todos los programadores de un proyecto adopten un estilo de codificación similar, de modo que se produzca un código de calidad uniforme. Esto significa que todos los programadores deben pensar igual, o que deben instrumentar servilmente todos los algoritmos en la misma forma. En realidad, el estilo individual de cada programador en un proyecto puede identificarse aun cuando se observe un apego rígido a los estándares de estilo de programación.

Los estándares establecido a seguir deben quedar bien documentados y con frecuencia deben ser revisados y actualizados sí es necesario, procurando que estos cumplan con las expectativas del administrador de los lideres de proyecto. Así se logra tener un mejor control y una buena administración de nuestras aplicaciones. En el tema: "5. 4. Estándares." Se muestra un ejemplo de las directrices especificadas en Sistema Administrador de Aplicaciones.

### 3. 3. 1. Estándares de programación.

Un estándar de programación debe especificar cosas como:

1. No se utilizan proposiciones GOTO.
2. La profundidad de anidamiento de las construcciones de un programa no excederá cinco niveles.
3. La longitud de las subrutinas no excederá de 30 líneas.

Un principio general ordena estas especificaciones en la siguiente manera:

1. El uso de proposiciones GOTO debe evitarse debe evitarse en circunstancias normales.
2. La profundidad de anidamiento de las construcciones de un programa debe ser cinco o menos en circunstancias normales.
3. Apartarse de las circunstancias normales requiera aprobación del jefe del proyecto.
4. Cualquier desviación de las circunstancias normales requiere la aprobación del jefe del proyecto.

Algunas personas categorizan los estándares como aquellas características que pueden revisarse por medio de la máquina (GOTO hacia atrás, sangrado apropiado, tamaño de las rutinas) y los principios generales como aquellos aspectos que deben verificarse por

medio de seres humanos (uso de nombres significativos para identificadores, apego a los criterios de modularización específicos, etc.).

Se requieren varias condiciones para obtener un apego voluntario a los principios generales de la programación, y para garantizar que de hecho se sigan estos principios. Primero, los programadores deben comprender el valor de los principios generales de la programación. Segundo, los programadores deben tener la oportunidad de participar al establecer los principios. Tercero, los principios generales deben sujetarse a examen y revisión cuando se vuelven opresivos. Cuarto, debe haber un mecanismo para permitir violaciones de los principios generales en circunstancias especiales. Quinto, se deben usar herramientas automatizadas para verificar el apego a los principios generales.

Como regla general, no es posible inspeccionar manualmente el apego del código a los principios. Si no se hacen exámenes automatizados, los principios generales serán inútiles. Bajo las condiciones antes delineadas pueden desarrollarse los principios generales de la programación para mejorar la calidad de los productos de la programación. Con tacto y diplomacia apropiados, los programadores verán a los principios generales de la programación como aspectos positivos del medio ambiente de trabajo.

Existen una serie de puntos que constituyen lo que podemos denominar el "estilo" del software de un programador o de un equipo de programación. Dichos puntos deben fijarse previamente al comienzo del diseño y seguir a rajatabla en todo el proceso de desarrollo de una aplicación. Entre los más importantes podemos fijar los siguientes:

- 1) Estilo de menús y pantallas uniformes.
- 2) Reglas para fijar en la escritura del código fuente (poner los títulos y comentarios siempre en la misma forma, seguir un mismo criterio para los comentarios, escribir los comandos y las teclas de función con mayúsculas y las variables y otros símbolos con minúsculas, etc.)
- 3) El usuario debe de estar siempre en la mente del programador. La escritura de cada procedimiento ha de hacerse de la forma siguiente:
    - Planear de forma teórica el problema y solucionarlo también de forma teórica.
    - Escribir el código tal como se ha planteado en la solución teórica. No dejarse llevar por los inconvenientes que surjan. Lo normal es que tengan solución dedicándoles algo más de tiempo.
    - Someterlo siempre a la critica de un usuario y reescribir según sus consejos.
- 4) Racionalización del formato de los procedimientos. Es decir, hace que sigan siempre un esquema similar los diferentes procesos que se hacen en los mismos: entrada, apertura de archivos y áreas de trabajo, operaciones de control de flujo, salida, etc.
- 5) Parametrización de todos aquellos aspectos que puedan ser interesantes de cambiar en cualquier momento sin necesidad de volver a compilar la aplicación, de aquellos otros que permitirán la movilidad de nuestro programa a un sistema distinto, etc. Entre los más importantes datos a paremetrizar hay que destacar:
    - a) Los nombres de directorios y bibliotecas.
    - b) Los códigos de impresión.
    - c) Los colores de la pantalla.
    - b) Los nombres de empresas, particulares, etc., De modo que nuestro software pueda ser lo más estandarizado posible.
- 6) Estudio exhaustivo del tiempo de ejecución. Nuestra aplicación ha de fijarse

límites en este aspecto y comprobarlos luego exhaustivamente con un volumen representativo de datos. Por ejemplo, fijar que el equipo máximo deseable para la obtención de una estadística mensual es de 20 minutos y comprobar después con un volumen de datos equiparable al máximo posible fijado por el usuario.

- 7) Cuidado con la excesiva proliferación de constantes. Las aplicaciones cargadas de constantes saturan la memoria de trabajo, lo que hace aconsejable en ocasiones que las tratemos en archivos en disco y no como bloques de texto dentro de nuestros programas.

- 8) Planificación rigurosa de la fiabilidad de la formación. Este es uno de los puntos más difíciles de conseguir, ya que para lograr una aplicación viable, el único método es el de probarla exhaustivamente.

- 9) Determinación de protocolo a seguir para guardar fuera de todo riesgo la seguridad e integración de datos.

- 10) El software desarrollado no es un producto aislado. Se relaciona siempre con su entorno.

- 11) Las ayudas de usuario deben ser minuciosamente detalladas. Ningún imprevisto debe sucederle a la persona que se sienta enfrente al teclado sin que tenga a mano una ayuda que le sirva para tomar una decisión más acertada.

Al establecer estándares para el desarrollo de las aplicaciones, se obliga al analista de sistemas a contemplarlos en lo posible, no obstante se ha mencionado que aplicaciones hechas con otra filosofía o inclusive paquetes de software se pueden adaptar con más o menos facilidad. Sin embargo, lo que se pretende es uniformar los nuevos desarrollos e ir integrando todas las aplicaciones existentes.

Se podría hablar de estandarizar todas aquellas actividades que se hacen cuando se

diseña, esto implica un esfuerzo adicional y de poco beneficio y causa una baja en la creatividad del analista. No obstante, es necesario establecer ciertas reglas para mejorar la técnica de las aplicaciones así como simplificar el proceso de capacitación de usuarios. Debido a lo anterior es preciso mencionar los estándares propios de la operación de las aplicaciones como son: nomenclaturas de objetos, diseño de pantallas, teclas de función y formatos de reportes.

Uno de los objetivos de estandarizar la presentación de las pantallas es proporcionar al usuario una forma de trabajar única para todas las aplicaciones que maneje.

Al igual que la estandarización de las pantallas, la estandarización de reportes tiene como objetivo proporcionar al usuario una forma de presentación única de los listados en todas las aplicaciones que maneje.

La necesidad de la documentación en línea como apoyo al usuario para poder entender la funcionalidad de un sistema, ha obligado al analista a integrar en todos sus sistemas lo que se conoce como "Ayuda". Es entonces la ayuda un concepto más que requiere del uso de estándares. Las reglas establecidas por estos estándares permiten que el usuario se familiarice con el método a seguir para resolver sus dudas acerca de la funcionalidad del sistema en cuestión o acerca de algunos puntos del mismo. Los estándares deben establece los pasos a seguir, desde la tecla de función que activa la ayuda, hasta la forma de búsqueda de alguna palabra utilizada dentro del sistema este donde este ubicado.

### 3. 3. 2. Estándares en la documentación de un sistema.

Todos los sistemas, con independencia de su aplicación, tienen una cantidad enorme de documentación asociada. Esta puede clasificarse como documentación del usuario o del sistema. La documentación del usuario se compone de aquellos documentos relacionados con las funciones del sistema, sin referirse a las formas de aplicarlas. La documentación, por otra parte, describe todos los aspectos del diseño, implantación y pruebas del sistema.

La información proporcionada junto con el sistema debe satisfacer varios requisitos tiene que describir:

- Como usar el sistema.
- Como instalar y operar el sistema.
- Los requisitos y diseños de todo el sistema.
- La aplicación del sistema y los procedimientos de prueba, para poderle dar mantenimiento.

Los estándares para la documentación tienen que describir con exactitud lo que debe incluir la documentación y las notaciones a utilizar en ella. Dentro una organización, es útil establecer un "formato estándar de la casa" y solicitar que todos los documentos se ajusten a él. Dicho estándar puede incluir la descripción de un formato para la cubierta frontal para hacer adoptados en todos los documentos, las convenciones para la numeración y anotación de página, los métodos para hacer referencia a otros documentos y la numeración de los títulos y subtítulos.

Todos los tipos de documentos necesitan índices efectivos. Un buen índice que permita al usuario encontrar la información que necesita es quizá la característica más útil que se puede proporcionar pero, por desgracia, suele ser la parte más olvidada de la producción de documentos. Un índice comprensible puede hacer que un documento mal escrito sea utilizable; sin índice, aun la prosa mejor escrita es inadecuada para convencer al usuario de que el documento es efectivo.

### 3. 4. Documentación de estándares.

Siempre que un grupo de personas está unido en un esfuerzo común, hace falta establecer normas eficientes, las cuales deben ser respetadas por cada uno de los miembros del grupo. Se imponen estas normas como "estándares de realización".

El uso de estándares ha sido completamente aceptado en fabricación, donde con frecuencia sirven de base para la compensación o promoción. Los estándares de producción son el medio por el cual la administración puede estar segura de todo el trabajo se realiza eficientemente y de que el producto final sea de buena calidad.

En el diseño de un sistema comercial por computadora se ha reconocido también la importancia de los estándares. Los dirigentes de la tarea de diseño han sentido la necesidad de que todos los analistas y programadores de un proyecto sigan las mismas reglas. Si todos los analistas siguen el mismo curso general de acción al diseñar un sistema y elaboran documentación muy similar, podrán trabajar en grupos y compartir o

intercambiar partes de la carga de trabajo. También si todos los programadores siguen ciertos enfoques básicos y emplean técnicas aceptadas, será más fácil que el mismo programador, u otro diferente, dé mantenimiento u optimice un programa en fecha posterior.

Por estas razones, la mayoría de las instalaciones para procesar información establecen un conjunto de técnicas de documentación, que deben ser observadas por todos los analistas y programadores. Generalmente se describen en un manual de estándares de documentación y se entrega una copia a todo el personal técnico. Una base para medir la habilidad de un analista o de un programador es la aplicación que él o ella hagan de los estándares.

Casi siempre, una empresa establece sus estándares sin conocer bien los utilizados en otras firmas. Por supuesto, hay ciertas normas más utilizadas que otras, como los símbolos del diagrama de flujo y lógico, pero no son estándares de la industria. Las prácticas que adopta una empresa son las que considera mejores para cumplir sus requerimientos, y pueden ser totalmente diferentes a las fijadas en cualquier otra. El punto importante es que usen estándares - no que sean iguales en todas las compañías -. Sin lugar a dudas, a medida que se perfeccione la tarea de diseño en la industria de la computadora, habrá mayor aceptación de estándares de documentación comunes, Sin embargo, nunca llegará el día que en que todas las empresas registren sus sistemas de una manera idéntica. Existe mucha variación en los requerimientos de las compañías.

### 3. 5. Razones del uso de estándares.

Los siguientes puntos permiten darse cuenta de la necesidad del uso de los estándares, básicamente considerados como problemas de los sistemas:

- Cuestan mucho.
- Toman mucho tiempo para clomplementarse.
- Son costosos en su mantenimiento.
- No son entendibles.
- No son modificables.
- No son confiables (fallan).
- No cumplen con los requisitos de los usuarios.
- Su complejidad excede la capacidad humana.
- Depuración no es eficiente.
- Son costosos en su operación.

Por tales razones es deseable que habiendo utilizado estándares podremos obtener los siguientes objetivos para un sistema:

- Tener un sistema confiable:

- Su documentación es completa y bien estructurada.
- Sus programas son legibles.
- Su lógica es sencilla y clara.
- Sus archivos tienen información ordenada y lógica.

- Tener un sistema modificable:
  - El manejo manual del sistema es flexible.
  - Los programas que manejan la información son flexibles.
  - Los archivos que contienen información son flexibles.
  - Las necesidades de los usuarios son dinámicas y el sistema esta preparado para esos cambios.

- Tener un sistema confiable:
  - Tener el mínimo número de fallas.
  - Manejar en forma apropiada información válida e inválida.
  - Detectar errores en su diseño.
  - Fácil de rastrear.

**3. 6. Los estándares en la implantación.**

El apego a los estándares y principios generales de la implantación por parte de todos los programadores de un proyecto da como resultado un producto de calidad uniforme. Los estándares se definen como aquellos aspectos que pueden ser revisados por una herramienta automatizada, mientras que determinar el apego a un principio general requiere de interpretación humana. Se deben observar varias condiciones para obtener un apego voluntario a los estándares y principios generales. Los documentos de apoyo para la fase de implantación incluyen todos los productos de trabajo estandarizados de las fases de análisis y diseño, así como los cuadernos de notas de cada unidad. La marca de logro principal para la implantación del producto es la integración con éxito de los componentes del código fuente en un sistema que funcione.

# CAPITULO IV

# INTEGRACION DE SISTEMAS

La suma de las dificultades que presentan cada una de las aplicaciones existentes en un equipo de computo, representa la magnitud de la problemática que tendrá el usuario para poder manejar, comprender, operar y mantener las mismas. Cuando cada una de estas aplicaciones es diseñada y desarrollada por diferentes casas de software y después es instalada en un equipo de computo, es muy común que las diferencias se hagan notar desde el primer día de su uso, ya que, por lo general cada una contiene su propio control de acceso, seguridad de base de datos, navegación de menús para la ejecución de tareas, estándares en su presentación de pantallas y reportes, uso particular de las teclas de función o tal vez alguna de ellas no contenga nada de lo antes mencionado. Por todo ello, el usuario se ve en la ardua tarea de entender el ambiente de cada una de sus aplicaciones y tratar de no confundirse para evitar cometer errores irreversibles.

La intención de este capítulo es mostrar la importancia de mantener nuestros sistemas integrados en medio ambiente común de trabajo y control, que nos permita facilitar el trabajo del diseñador de sistemas y del usuario final, agilizar cualquier mantenimiento y sobre todo realizar una explotación adecuada del concepto de base de datos. Obviamente, un Sistema Administrador de Aplicaciones sería la herramienta más adecuada para éste fin.

Dentro de la integración de sistemas, un Sistema Administrador de Aplicaciones nos permite:

- Definir un solo nivel de acceso para todas nuestras aplicaciones, donde se tenga mantenimiento de los usuarios que podrán ingresar al sistema, las funciones a las que estarán autorizados, las compañías (bases de datos) sobre las que podrán trabajar y en si las limitantes que deberán contemplar.
- Definir bases de datos de uso común para todas nuestras aplicaciones, evitando redundancia de información que implique ocupar espacio en los medios de

almacenamiento y que proporcione información confiable.

- Poder contar con un solo control de seguridad de nuestros sistemas, contemplando uso restringido de base de datos, ejecución de tareas especiales o particulares, obtención de reportes o consultas de información altamente confidencial, etc.
- Modular las aplicaciones para facilitar el mantenimiento de las mismas mejorando el tiempo de respuesta de alguna falla o solicitud de cualquier usuario.
- Mantener centralizada la información de la parametrización de cada una de las aplicaciones (sí estas lo requieren).
- Lograr un ambiente común de trabajo independientemente de la aplicación con la que se este trabajando, facilitando el aprendizaje de los usuarios en el uso de las aplicaciones.

**4. 1. Estructura de las aplicaciones.**

El Diagrama que se presenta en la figura no. 4.1. se muestra la ubicación de los diferentes componentes que intervienen en el proceso de computo bajo la filosofía de un Sistema Administrador de Aplicaciones. En los niveles inferiores (C y D) se encuentra el equipo físico (Hardware) y el Sistema Operativo. Estos dos elementos son la base del proceso siendo los encargados de interpretar y realizar las funciones ordenadas desde las aplicaciones en ejecución.

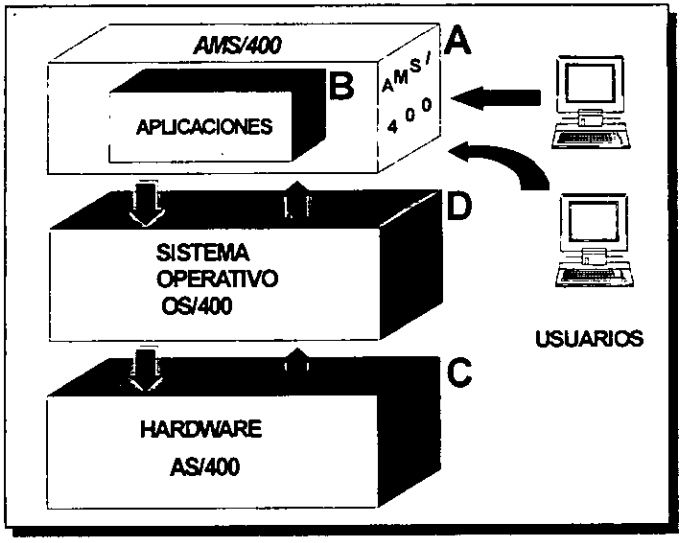



**Fig. 4.1.** Estructura del Control de Acceso de AMS/400

El nivel B (aplicaciones de usuario) son los sistemas de aplicación que realizan los trabajos de las diferentes áreas usuarias. Este nivel es formado por los programas que el lenguaje comprensible para el equipo ordenan las funciones que cubren los requerimientos de usuarios para diferentes aplicaciones. El nivel A es el Sistema Administrador de Aplicaciones y como se ve en el diagrama cubre o encapsula a las aplicaciones de usuarios encargándose de la comunicación o interfase con el usuario final y del intercambio de información y órdenes con el Sistema Operativo.

Un Sistema Administrador de Aplicaciones sería el responsable de la comunicación inicial de todos los usuarios con sus aplicaciones y del control y supervisión de las tareas que se soliciten.

**4. 2. Principio de integración.**

La mejor forma de visualizar el sistema de negocios es describiendo el flujo del proceso, analizando cada segmento e investigando las relaciones y contribuciones de las partes con el todo. En esta forma es posible dirigir la atención y estudiar aquellos segmentos que no cumplen al elevar el máximo sus contribuciones al sistema total. Integración significa: hacer un todo o totalizar, traer partes y juntarlas dentro de un todo. Algunas teorías o razonamientos proponen los siguientes puntos:

- 1. - El todo es primario y las partes son secundarias.
- 2. - La integración es la condición de la interrelación de las muchas partes dentro de una.
- 3. - Las partes así constituidas forma un todo indisoluble en el cual ninguna parte puede ser afectada sin afectar todas las otras partes.
- 4. - El papel que juegan las partes depende del propósito para el cual existe el todo.
- 5. - La naturaleza de la parte y su función se derivan de su posición dentro del todo y su conducta es regulada por la relación de todo a la parte.
- 6. - El todo es cualquier sistema o complejo o configuración de energía y se conduce como una pieza única, no importando qué tan compleja sea.
- 7. - La totalidad debe empezar como una premisa y las partes, así como sus relaciones deberán evolucionar a partir del todo.

El todo se renueva a sí mismo constantemente a través de un proceso de transposición; la

identidad del todo y su unidad se preserva, pero las parte cambian. Este proceso continúa indefinidamente, algunas veces es planeado y observado, en tanto que otras ocurren sin notarlo; a menudo es alentado, pero otras veces se le resiste. Una empresa de negocios es un todo integrado en donde cada sistema, subsistema y subsistema de apoyo están relacionados con la operación total. Su estructura por lo tanto, es creada por cientos de sistemas arreglados en orden jerárquico. La salida del más pequeño de los sistemas resulta la variable de entrada para el próximo sistema mayor, que a su vez proporciona la variable de entrada para un nivel superior.

### 4. 3. Configuración de los sistemas para su Integración.

Un sistema por si solo, representa un conjunto de funciones que permiten al usuario usar los recursos que un equipo de computo ofrece. Cada sistema tiene un objetivo global y cada función un objetivo específico. Sin embargo, cuando un sistema se integra a otros sistemas, a pesar de pasar a ser una parte de un todo, éste sigue cumpliendo con sus objetivos, pero además toma fuerza e importancia al compartir recursos con el resto de los sistemas. Para un sistema administrador, cada unos de los sistemas que en un principio tienen su propio ambiente de trabajo, se convierte en una aplicación del sistema integral, denominàndosele así, Sistema Administrador de Aplicaciones.

A continuación se proporciona el orden de actividades sugeridas para habilitar o dar de alta cada una de las aplicaciones dentro de un Sistema Administrador de Aplicaciones. Se debe sobre entender que el sistema, por si solo, controla que los mantenimientos se hagan en cierto orden lógico, de acuerdo a sus validaciones.

Secuencia de actividades:

- 1. - Determinación de estructura y nomenclatura de la aplicación a integrar. Se codificará la clave que identificará a la Aplicación y se determinará de qué módulos se compone, asignándole a estos su código de identificación y una descripción correspondiente.

- 2. - Determinación de estructura de localidades bajo las cuales va a correr la aplicación. Este punto es muy importante y definirá, de acuerdo a las características de la aplicación, bajo que estructura de localidades correrá la aplicación. Se dan dos reglas a seguir al respecto:

    - A - Intentar que la estructura a utilizar sea la de más bajo nivel de acuerdo a los requerimientos. Esto significa que si la aplicación puede correr sólo bajo compañía, no definir otro nivel (por ejemplo: localidad y sublocalidad).

    - B - Si se definen otros niveles (localidades), revisar la estructura de los niveles ya existentes en el sistema (definidas previamente por otras aplicaciones), y si es posible se adaptan, usar algunas de las vigentes, La idea sería que todas las aplicaciones corrieran bajo la misma estructura.

- 3. - Determinar los niveles de concurrencia que deberá utilizar la aplicación. Es preciso definir la concurrencia que tendrán cada uno de las funciones para evitar que la información pierda su integridad y los procesos especiales se bloqueen entre sí.

- 4. - Integrar los objetos ejecutables para cada función de la aplicación. El Sistema Administrador de Aplicaciones deberá contar con esqueletos predefinidos para los diferentes tipos de tareas (batch o interactivos). Estos esqueletos deben contener los parámetros estándarizados que comunican a cada uno de los procesos con el corazón del sistema.

- 5. - Asignación de números de funciones a todas las tareas de la aplicación. Esta asignación se puede hacer en masa si se tienen bien definidas las tareas que conformarán la aplicación, o de manera progresiva a medida que las tareas se vayan definiendo. Se recomienda que se defina un rango de número de funciones para cada

aplicación e inclusive cuando la estructura de módulos de la misma sea completa, establecer subrangos para las funciones de cada módulo.

- 6. - Mediante el mantenimiento correspondiente, dar de alta cada una de las funciones.

- 7. - Dar de alta la concurrencia de las funciones de la aplicación.

- 8. - Una vez definidas las funciones de nuestra aplicación, definir la estructura de nuestros menús. Se recomienda agrupar las funciones por objetivos para determinar cada uno de los menús que compondrán la aplicación; así los menús estarán compuestos de funciones homogéneas.

- 9. - Abrir usuarios. Abrir las identificaciones necesarias para los usuarios que utilizarán la aplicación, si es que estos no existen previamente. Si los usuarios de la nueva aplicación ya están dados de alta en el sistema, sólo bastará con autorizarle las nuevas funciones y menús de la nueva aplicación. Se recomienda no abrir más de un perfil por usuario, sino mantener una sola con todo el acceso y el permiso necesario para ejecutar todas las funciones, de todas las aplicaciones a que tenga derecho en un sólo nivel de identificación.

- 10. - Autorización de funciones. Mediante el mantenimiento correspondiente se dará autorización de las funciones que los usuarios, ya existentes, usarán de la aplicación. Se recomienda dar autorización a aquellos usuarios que cumplan con cierto perfil, el cual luego será usado como patrón, para poder copiar a otros usuarios con el mismo perfil.

- 11. - Liberación. Cumpliendo todos los pasos anteriores (o los necesarios), se está en posibilidades de liberar en firme o para pruebas, los ambientes de ejecución de la nueva aplicación. Cabe acotar que en cualquier momento se podrán modificar, mediante los mantenimientos mencionados, algunas o todas las definiciones explicadas en los puntos anteriores.

# CAPITULO V

# APLICACION DE UN SISTEMA ADMINISTRADOR DE APLICACIONES

En este capítulo daremos una muestra clara de lo que es un Sistema Administrador de Aplicaciones. Como ya se menciono antes, un Sistema Administrador de Aplicaciones debe ofrecer soluciones comunes a problemas comunes.

Este capítulo tiene como objetivo ejemplificar el papel de un Sistema Administrador de Aplicaciones. Con el propósito de darle un nombre y debido a que uno de los propósitos de esta aplicación de control es integrar aplicaciones; nos referiremos a esta aplicación con las siglas: S.I.A. (Sistema Integrador de Aplicaciones) con el fin de diferenciar el propio tema de Tesis y esta aplicación, sin que dejen de tener relación.

## 5. 1. Objetivos.

Los objetivos que a continuación se listan son los de un Sistema Administrador de Aplicaciones, independientemente del nombre que se le asigne:

- Liberar recursos de desarrollo para atender la implantación de Aplicaciones de usuario y no solucionar problemas de ambientes de ejecución o estructuración y presentación de aplicaciones.
- Proporcionar una base uniforme, que solucione problemas de operación y control de tareas comunes a todas las aplicaciones que sean completamente independientes de las aplicaciones de usuario.
- Alcanzar un nivel elevado de estandarización en cuanto a estructura de aplicaciones, su presentación y su operación, proporcionando un nivel de entrada común para todos

los sistemas.

- Proporcionar ambientes de ejecución de tareas amigables y con diferentes niveles de ayuda y navegación que permitan auxiliar a usuarios novatos y experimentados en forma sencilla y selectiva.

- Controlar eficientemente un ambiente de multi-aplicaciones sin agregar tareas de operación a los usuarios.

- Brindar un nivel de autonomía casi total a los usuarios eliminando casi por completo la intervención de un analista en la ejecución y control de las aplicaciones.

- Proporcionar documentación y ayudas en línea que faciliten la utilización de las aplicaciones por los usuarios.

- Garantizar la integridad y seguridad de la información mediante el control de acceso a los sistemas y a cada una de sus funciones asignadas a los usuarios que puedan usar esta aplicación.

- Proporcionar un ambiente multi-compañías que permita al usuario agrupar y organizar su información en base de datos en caso de manejar más de una compañía como región física o lógica.

## 5. 2 Estructura

De acuerdo a los principios ya mencionados para un Sistema Administrador de Aplicaciones, el usuario contará con un ambiente común de trabajo, por lo que, no deberá poder trabajar fuera de este ambiente. De tal forma el usuario al ingresar a su maquina tendrá activo este ambiente (S.I.A.).

La organización de las bases de datos existentes estará dada por el concepto de multi-compañías, es decir, para cada base de datos se debe de asignar un nombre de compañía que tenga relación con la información que se almacenará. Se podrá determinar un nombre de compañía en caso de ser una región física o bien si es una región lógica.

Las diferentes aplicaciones que se vallan a integrar deberán residir en el mismo lugar donde resida la aplicación principal de control (S.I.A.). El usuario, en forma dinámica y sin salir del ambiente podrá navegar a todas las aplicaciones existentes y a las autorizadas según sea el caso.

La unidad básica bajo la cual el usuario podrá ejecutar cada una de las tareas para las aplicaciones existentes, se denominará "función". Una función deberá estar relacionada con una agrupación de tópico. Este tópico es una agrupación con el fin de que al usuario le sea más fácil localizar a cualquier función por el tipo de tarea que ejecuta. La función también deberá estar asociada a una sola aplicación.

El ambiente común de trabajo estará formado por menús dinámicos formados por todas aquellas funciones que estén autorizadas en determinada compañía para determinado usuario. Por lo que, cuando un usuario se cambie de compañía (presionando la tecla correspondiente) puede que no tenga las mismas funciones que en la compañía anterior. Así entonces, el usuario podrá armar sus menús dinámicos seleccionando la aplicación y la compañía con la que quiere trabajar.

Los cambios de compañía y de aplicación se podrán hacer activando un submenu que

siempre estará disponible (éste parecerá en la parte superior de la pantalla). Con este submenu el usuario también podrá ejecutar una función en forma directa, (siempre y cuando esté autorizado a la función) digitando el número de función, sin necesidad de cambiarse de aplicación.

Al ejecutar una función (de una aplicación que se integra y no que se crea siguiendo los estándares de S.I.A.) tal vez cambie el ambiente interactivo de trabajo, pero al terminar de correr volverá al mismo ambiente.

S.I.A. deberá contar con todas las funciones necesarias para sustituir el uso directo de comandos del sistema operativo (copiar archivos, renombrar archivos, listar un directorio, etc.) eliminando así los riesgos de ingresar comandos indebidos (eliminar archivos, formatear disco duro, etc.).

**5. 3. Filosofía.**

Todas las tareas que estén disponibles dentro de una aplicación, serán ejecutadas exclusivamente por usuarios definidos y autorizados bajo S.I.A. Bajo este concepto estarán todas las tareas que conforman las aplicaciones, y el sistema se encargará de su control y seguimiento.

Todo usuario definido en S.I.A. deberá tener asignada una compañía por defecto y se determinará a cuales de las compañías existentes podrá tener autorización.

Como aplicaciones existirán dos grandes tipos que serán ejecutadas bajo S.I.A., las primeras son las que son desarrolladas íntegramente pensando en aprovechar todos los beneficios y herramientas que le proporciona el administrador de aplicaciones; las segundas son las aplicaciones que fueron desarrolladas sin pensar en S.I.A. o aplicaciones que son consideradas como paquetes. Estas aplicaciones requieren un esfuerzo adicional de las áreas de desarrollo de sistemas para su implantación bajo S.I.A., lo cual no siempre alcanza el 100% de integración lo cual provoca que algunas de las facilidades que se tienen en las primeras aplicaciones no se tengan en las segundas.

Toda aquella tarea a ejecutar estará definida mediante una función, la cual será única para cada una de estas tareas. Cada una de estas funciones deberá tener relación por lo menos con una Aplicación y un Tópico. Las funciones que sean creadas bajo los estándares y el ambiente de S.I.A. deberá proporcionar una ayuda en línea; por lo que se contara con una función que permita dar mantenimiento a cada una de estas ayudas. Será responsabilidad del analista de sistemas que las aplicaciones ya existentes, proporcionen una ayuda en línea, por lo tanto, esta situación quedara fuera del alcance de S.I.A.

**5. 4. Estándares.**

A continuación se presenta una serie de reglas y estándares que hacen que el diseñador

desarrolle aplicaciones con alto grado de eficiencia y primordialmente aplicaciones pensadas para estar bajo el ambiente de S.I.A. aprovechando todas las ventajas que este ofrece.

### 5. 4. 1. Bibliotecas o Subdirectorios de archivos ejecutables.

# UUVLIB

**UU** Variable de dos posiciones que identifican la Aplicación. Se recomienda que proporcione una simple idea del nombre de la aplicación que se va a desarrollar y que va a contener todos los archivos ejecutables de la misma.

**V** Variable de una posición que identifica el Módulo consecutivo de la Aplicación. En ocasiones, se hace necesario dividir en varias bibliotecas la aplicación a desarrollar, por lo que esta posición servirá para identificar el módulo dentro del sistema. Por ejemplo: el Sistema de Administración Comercial (SA) se dividiría en módulos como: Cuentas por Cobrar (SACLIB), Inventarios (SAILIB), Facturación (SAFLIB), etc.

**LIB** Constante de tres posiciones en la nomenclatura del nombre de la biblioteca para identificar a una biblioteca de archivos ejecutables.

### 5. 4. 2. Bibliotecas o Subdirectorios de archivos fuentes.

# UUVSRC

**UUV** Variable de tres posiciones que identifican la biblioteca de la aplicación a la cual pertenece, de acuerdo a la nomenclatura de biblioteca de archivos ejecutables.

**SRC** Constante de tres posiciones en la nomenclatura del nombre de biblioteca para identificar a una biblioteca o subdirectorio de archivos de fuentes.

### 5. 4. 3. Programas y Archivos de datos.

# UUVWXXYY

**UUV** Variable de tres posiciones que identifican la biblioteca de la aplicación a la cual pertenece el programa o el archivo.

**W** Variable de una posición para identificar el archivo de datos o programa ("**A**" archivo de datos, o "**P**" archivo de programa, "**D**" estructura de archivo, "**I**" índice de

archivo y "**R**" rutinas).

**XX** Variable de dos posiciones para identificar el consecutivo del nombre del archivo o programa dentro de la misma biblioteca o subdirectorio.

**YY** Consecutivo de un archivo de índices. Esta posición tendrá uso únicamente cuando se defina un archivo de índices para identificar un consecutivo del mismo.

### 5. 4. 4. Archivo de mensajes.

# UUVMSG

**UUV** Variable de tres posiciones que identifican la biblioteca de la aplicación a la que se esta haciendo referencia.

**MSG** Constante de tres posiciones que define que se trata de un archivo de mensajes.

### 5. 4. 5. Diseño de pantallas.

El diseño de una pantalla constará de tres partes principales:

- 1. - La cabecera de la pantalla; donde se definirá la compañía y el titulo de la función que se esta ejecutando.
- 2. - El cuerpo de la pantalla; será responsabilidad del diseñador de equilibrar la estética con la funcionalidad para brindar al usuario aplicaciones amigables.
- 3. - En una ventana en la parte inferior izquierda se mostrarán las teclas de función permisibles para su ejecución tomando diferentes acciones al presionarlas. Cabe señalar que la tecla ENTER se sobreentiende por lo que no se deberá especificar como función permitida y algo muy importante es que no se deberán presentar teclas de función que no tengan ningún efecto en ese momento.
- 4. - Para aquellas funciones creadas bajo los estándares y el ambiente de S.I.A. se usará una área definida para solicitar los datos a capturar así como los datos que se deseen presentar (Altas, Bajas, Cambios y Consultas).

Para identificar la función que se esta ejecutando es necesario auxiliarnos de algunos parámetros que la aplicación S.I.A. nos proporcionará.

- SIA_NOCIA Es una variable alfanumérica de 40 caracteres que contendrá el nombre de la compañía de la sesión.
- SIA_APL Es una variable alfanumérica de 3 caracteres que nos indica la aplicación a la que pertenece la función.
- SIA_NOFUN Es una variable alfanumérica de 35 caracteres que contiene la

descripción de la función que es mostrada en los menús.

- SIA_FUN Es una variable numérica de 5 caracteres con cero decimales que nos proporcionará el número de la función.

El diseño de la pantalla estará representado como se muestra en la figura 5.1. En donde:

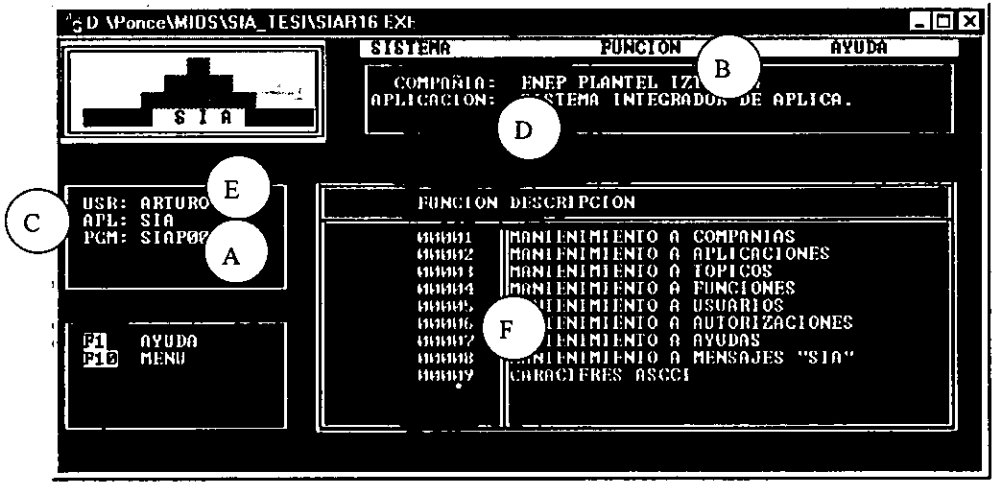



**Fig. 5.1.** Diseño de pantalla.

- A - Nombre del programa (alfanumérico de 7 posiciones). Podrá ser constante o variable.

- B - Nombre de la compañía (alfanumérico de 40 posiciones). Parámetro SIA_NOCIA.

- C - Aplicación (alfanumérico de 3 posiciones). Parámetro SIA_APL.

- D - Descripción de la función que se esta ejecutando. Parámetro SIA_MOFUN.
- E – Nombre del usuario activo en la sesión. Parámetro SIA_USU.
- F - Contenido. Esta es la parte donde el analista deberá darle la mejor presentación al usuario buscando siempre la funcionalidad.

En un cuadro en la parte inferior izquierda se presentara los mandatos o funciones que estarán permisibles en esa pantalla. Donde :

ESC CANCELAR Regresa a una pantalla anterior en una secuencia dentro de un mismo programa o función compuesta por varios subprogramas. Podrá tener la función de cancelar si así lo desea el programador.

FXX Será la función deseada con atributos especiales con el fin de que estos puedan ser fácilmente identificables por el usuario.

XX.... Será la descripción o significado de la tecla de función según la siguiente tabla:

- 1 AYUDA.

Invocar ayuda de la pantalla que ese momento se está presentando.

- 3 FIN.

Termina el programa de ejecución (regreso a ambientes de ejecución).

- 4 LISTA DE.....

Presentación de una lista de valores existentes o posible de un campo.

- 10 GRABAR, CONFIRMAR

Ejecuta o confirma un proceso en su ultimo nivel, por ejemplo: en un programa de

mantenimiento a un archivo maestro que nos permita dar Altas, Bajas, Cambios y Consultas; esta tecla nos permitirá grabar, confirmar baja o confirmar cambio a un registro.

- 13 IMPRIMIR.

  Dar salida a impresión a través de un reporte formateado de información que se esta mostrando en ese momento en la pantalla.

### 5. 4. 6. Diseño de Reportes.

El diseño de un reporte consta de 3 partes principales:

- 1. - La cabeza del reporte, que identifica la función que se esta ejecutando.
- 2. - El cuerpo del reporte, responsabilidad del diseñador de equilibrar la estética con la funcionalidad para brindar al usuario aplicaciones amigables.
- 3. - El pie de reporte o los totales.

Para poder identificar la función que se esta ejecutando y que genera el reporte es necesario auxiliarnos de algunos parámetros que el Sistema Administrador de Aplicaciones proporcionará.

El diseño del reporte estaría representado por la figura 5.2.

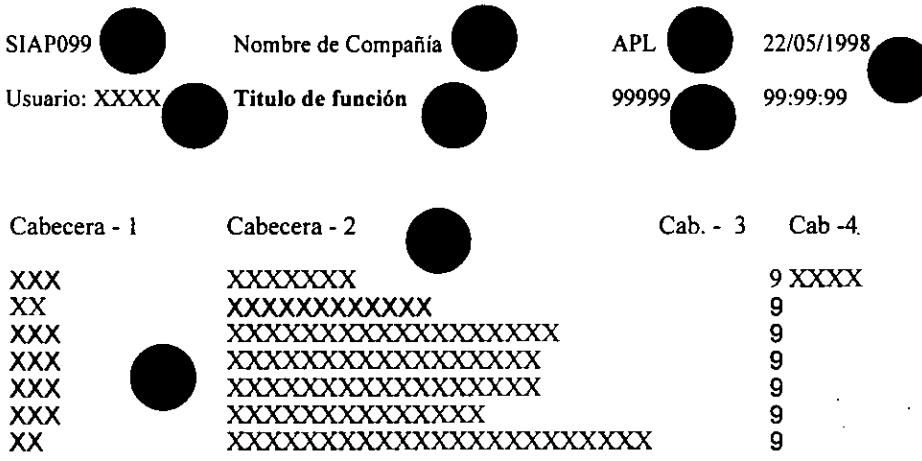



**Fig. 5.2.** Diseño de reporte.

- SIA_NOCIA Es una variable alfanumérica de 40 caracteres que contendrá el nombre de la compañía de la sesión.
- SIA_APL Es una variable alfanumérica de 3 caracteres que nos indica la aplicación a la que pertenece la función.
- SIA_NOFUN Es una variable alfanumérica de 35 caracteres que contiene la descripción de la función que es mostrada en los menús.
- SIA_FUN Es una variable numérica de 5 caracteres con cero decimales que nos proporcionará el número de la función.
- SIA_DSP Es una variable alfanumérica de 10 caracteres que contiene la

identificación del equipo desde donde se ejecuta la función.

- SIA_USER Es una variable alfanumérica de 10 caracteres que contiene la identificación del operador que hizo la petición.

A continuación se describen los datos especificados en el diseño de reporte de la figura 5.2.

- A - Nombre del programa (alfanumérico de 7 posiciones). Podrá ser constante o variable.
- B - Nombre de la compañía (alfanumérico de 40 posiciones). Parámetro SIA_NOCIA.
- C - Aplicación (alfanumérico de 3 posiciones). Parámetro SIA_APL.
- D – Fecha y hora en que se genero el reporte.
- E - Descripción de función que se ejecuto. Parámetro SIA_NOFUN.
- F - Número de la función que se ejecuto. Parámetro SIA_FUN.
- G - Usuario que ejecuto la función. Parámetro SIA_USER.
- H - Títulos adicionales que complementen la descripción del reporte.
- I - Contenido. Esta es la parte donde el analista deberá darle la mejor presentación al usuario buscando siempre la funcionalidad.

## 5. 5. Diseño de S.I.A.

Como ya se embozó en los objetivos, el concepto de S.I.A. esta basado en el hecho de controlar, estandarizar y adecuar a un ambiente común a todas las tareas que pertenecen a una variedad de aplicaciones que pueden existir en una empresa.

La idea parte poder crear un ambiente amigable y común, en donde un variado número de aplicaciones puedan explotarse como si fuera una sola aplicación. Controlar, administrar e integrar diferentes aplicaciones son objetivos que conseguirán respetando los estándares definidos previamente, así como las siguientes reglas.

### 5. 5. 1. Reglas para el diseño.

- Para el manejo de mensajes a desplegar en cada función a ejecutar, se deberá usar un archivo de mensajes, indexado por una clave de mensaje. Será responsabilidad de programador identificar la siguiente clave consecutiva del archivo de mensajes que se vaya a utilizar en el programa que en ese momento se este desarrollando; de lo contrario deberá consultar los mensajes existentes y seleccionar el que a juicio se apropie más al mensaje que se desee desplegar. Con esto se intenta tener un archivo de mensajes por cada aplicación, evitando tener mensajes repetidos y redundantes, además de evitar el uso de constantes en programas.

- Con el fin de optimizar el uso de la memoria principal, así como el rendimiento de

nuestro equipo de computo; los programas deberán evitar el uso de constantes en lo posible. Uno de los métodos para poder lograr éste objetivo, es el uso de las estructuras de archivos en los programas de mantenimiento (llevarán una "D" en su nombre), es decir, aquellos programas que permitan altas, bajas, cambios y consultas de estos archivos.

- Todas las rutinas a utilizar llevaran una "R" (de acuerdo a los estándares) en el nombre de las mismas.

- Cada uno de los programas creados bajo el ambiente de S.I.A., deberá ir documentado internamente, es decir, en el inicio de la codificación de cada programa fuente a nivel de comentario deberá indicarse: el objetivo del programa, procedimiento, función o rutina; fecha y nombre del autor.

- La ayuda en línea ofrecida en cada una de las tareas de S.I.A. estará controlada por una rutina, que manejara un archivo único por cada aplicación, en donde se registraran y se concentraran los textos de ayudas identificados por una clave y el número de función a la que se relaciona.

- Por conveniencia se deberán declarar todas las variables utilizadas en los programas a nivel global, con el fin de evitar variables indefinidas en procedimientos.

### 5. 5. 2. Procesos de control que integran S.I.A.

El control y la seguridad lograda por un Sistema Administrador de Aplicaciones se detallan en los principales procesos que lo componen. Adicionalmente a los archivos de control y a su estructura, es importante el camino a seguir durante el flujo de los procedimientos. En la figura 5.3. se mencionan los principales procesos que componen S.I.A.

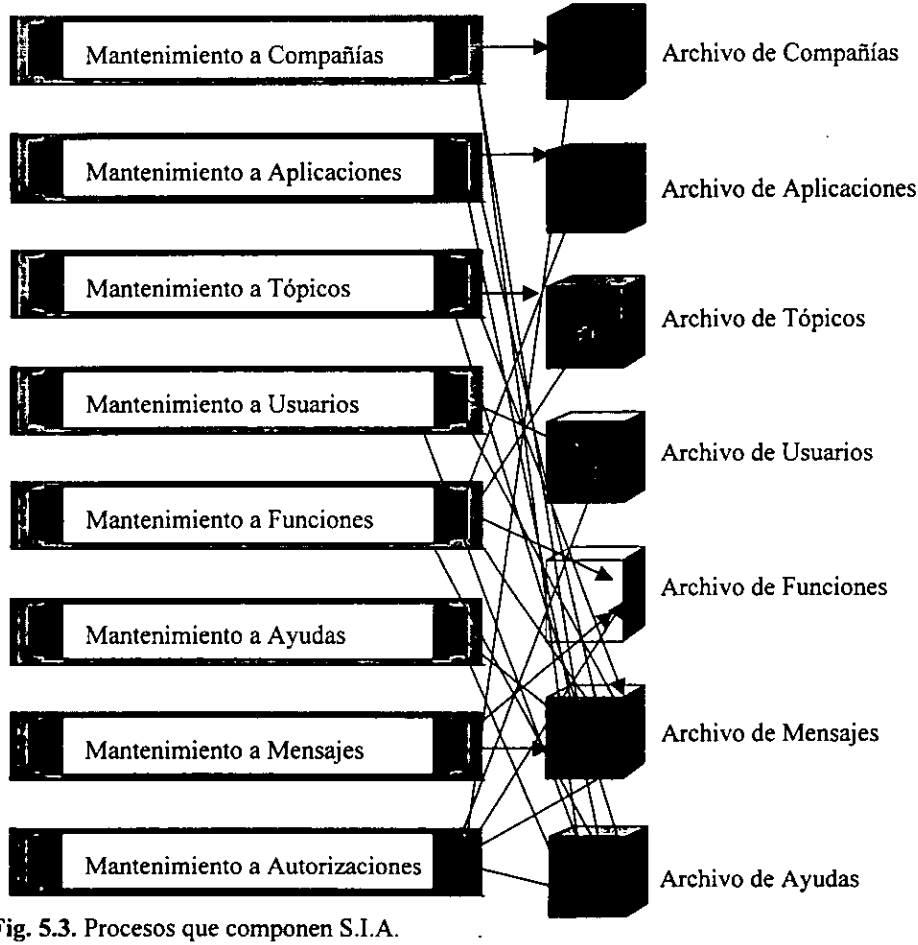



**Fig. 5.3.** Procesos que componen S.I.A.

*Nivel inicial de S.I.A.*

Debido a las limitaciones del lenguaje utilizado para desarrollar esta aplicación, se creó este pequeño proceso inicial que consiste en la activación por bloque de dos procesos principales para S.I.A. El primero es la solicitud de identificación del usuario y el segundo es el control maestro del ambiente de S.I.A. Estos dos procesos se detallan en este mismo capítulo. En la figura Fig..5.4. se presenta el seudocódigo que hace posible este proceso. En la figura 5.5. se muestra el control de flujo de este mismo proceso. Cabe señalar que para poder pasar al ambiente del sistema operativo DOS, S.I.A. solicitará una contraseña la cual únicamente la deberá saber el usuario que de defina como administrador del sistema. Para que un usuario pueda entrar al ambiente de S.I.A. deberá tener una identificación y una contraseña de lo contrario nunca podrá acceder al ambiente.

1. *Solicitar identificación de entrada al usuario*
2. *Restaurar los parámetros iniciales de atributos para S.I.A.*
3. *Ejecutar control maestro del ambiente de S.I.A. mientras no se quiera salir al sistema operativo DOS.*
4. *Si se requirió ejecutar una función, restaurar nuevamente parámetros de atributos para S.I.A..*
5. *Correr programa que ejecuta la función solicitada.*

**Fig. 5.4.** Seudocódigo para nivel inicial de S.I.A.

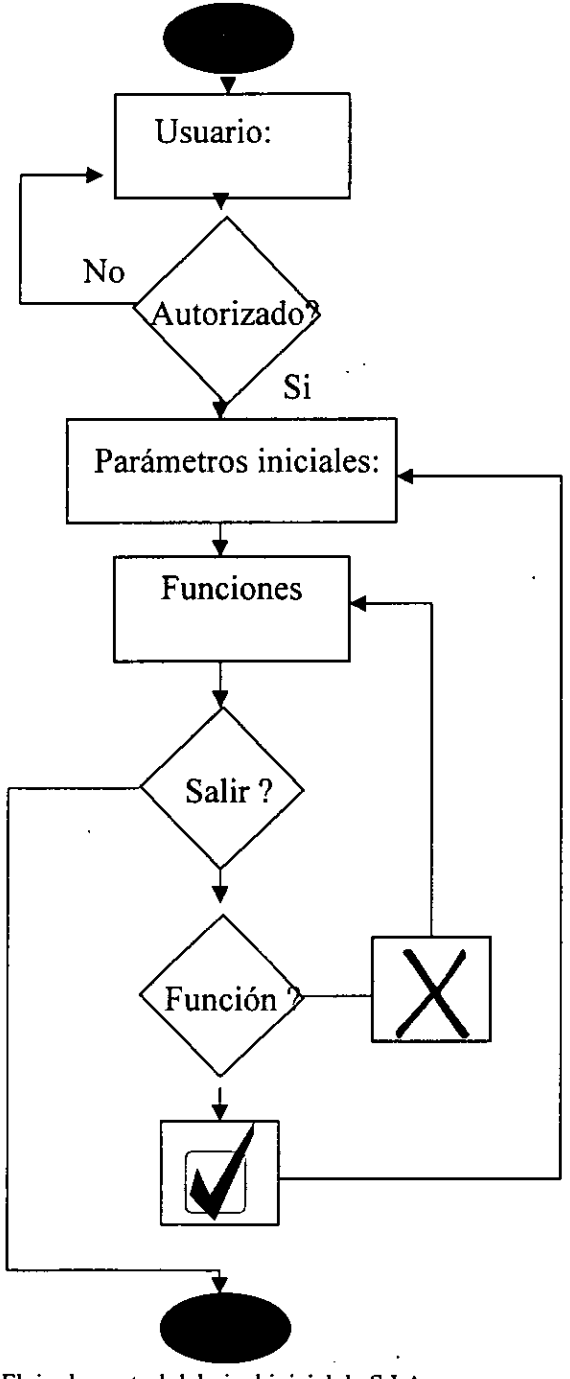



**Fig. 5.5.** Flujo de control del nivel inicial de S.I.A.

*Solicitud de identificación.*

Este proceso se encarga de controlar el acceso de los usuarios al ambiente de S.I.A., solicitando una identificación y una contraseña. Mientras el usuario no dimite un usuario y contraseña correcta no podrá entrar al sistema y se mantendrá en este paso. Una vez digitada la intensificación y contraseña correcta, este proceso verificará el estado de su ultima finalización de esta aplicación. En caso de que la última finalización de esta aplicación haya sido "anormal", se presentará una pantalla al usuario informándole lo propio e indicándole cual fue la última compañía accedida, cual fue la última aplicación accesada y cual fue la última función solicitada. En la figura Fig. 5.6. se muestra el seudocódigo que hace posible este proceso.

---

1. *Presentar la pantalla de solicitud de identificación y contraseña mientras no sean correctos los datos.*

   *1.1. Verificar usuario y contraseña en archivo de usuarios en el sistema.*

   *1.2. 2. Verificar ultima finalización del sistema para el usuario.*

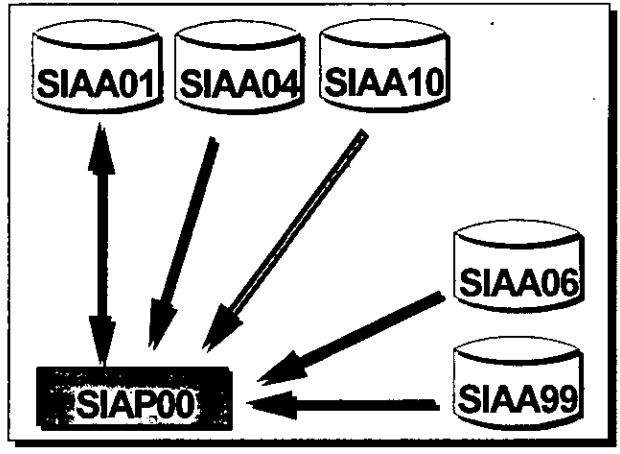
2. *En caso de finalización anormal, mostrar pantalla informativa de última finalización.*

---

**Fig. 5.6.** Seudocódigo para solicitud de identificación.

*Control maestro del ambiente de S.I.A.*

Este proceso es el principal, bajo cual se controla el ambiente de S.I.A. Es aquí donde el usuario decide con compañía y aplicación quiere trabajar. Al inicio el usuario tendrá una compañía y una aplicación por defecto. Una vez seleccionada la compañía y la aplicación bajo la cual quiere trabajar, se le presentará una lista de todas las funciones autorizadas para tal selección. Para cambiar de compañía o aplicación, basta activar un menú que se presenta en forma de barra en la parte superior derecha. Además de poder cambiar de compañía o aplicación, el usuario puede ejecutar directamente una función a la cual tiene autorización, sin necesidad de estar ubicado en la lista adecuada. Cada vez que ejecuta una función y ésta termina, el control siempre regresa a este ambiente de S.I.A. Ver figura 5.7. y figura 5.8.




**SIAA01** – Archivo de Compañías.

**SIAA04** – Archivo de Usuarios

**SIAA06** – Archivo de Mensajes

**SIAA10** – Archivo de funciones autorizadas

**SIAA99** – Archivo de Ayudas.

**Fig. 5.7.** Flujo de Archivos en proceso de control maestro.


**Fig. 5.8.** Esquema del proceso principal de S.I.A.

1. *Identificar usuario firmado al sistema.*
2. *Recuperar compañía y aplicación inicial para el usuario.*
3. *Presentar lista de funciones autorizadas por compañía y aplicación.*
4. *Mantener en forma activa el menú que permite cambiar de compañía y aplicación o ejecutar una función en forma directa o por lista de funciones.*
5. *Si ejecuto función:*
6. *Ejecutar función. Al finalizar función regresar al punto 3.*
7. *Si solicito cambio de compañía o aplicación:*
8. *Presentar menú de barra y seleccionar la compañía o la aplicación a la que se desea cambiar. Pasar al punto 3.*

**Fig. 5.9.** Seudocódigo de flujo maestro de S.I.A.

Con la descripción archivos descritos en el apéndice "A" (las figuras A.1. y A.2. ayudan a comprender la relación de entidades de las tablas de S.I.A.) y las funciones de mantenimiento descritas en el apéndice "B", se pude puede comprender la funcionalidad de los procesos que comprenden el S.I.A.,. Sin embargo en la figura 5.10. (a) y (b) se da un ejemplo de la secuencia de pantallas de mantenimiento al sistema S.I.A.

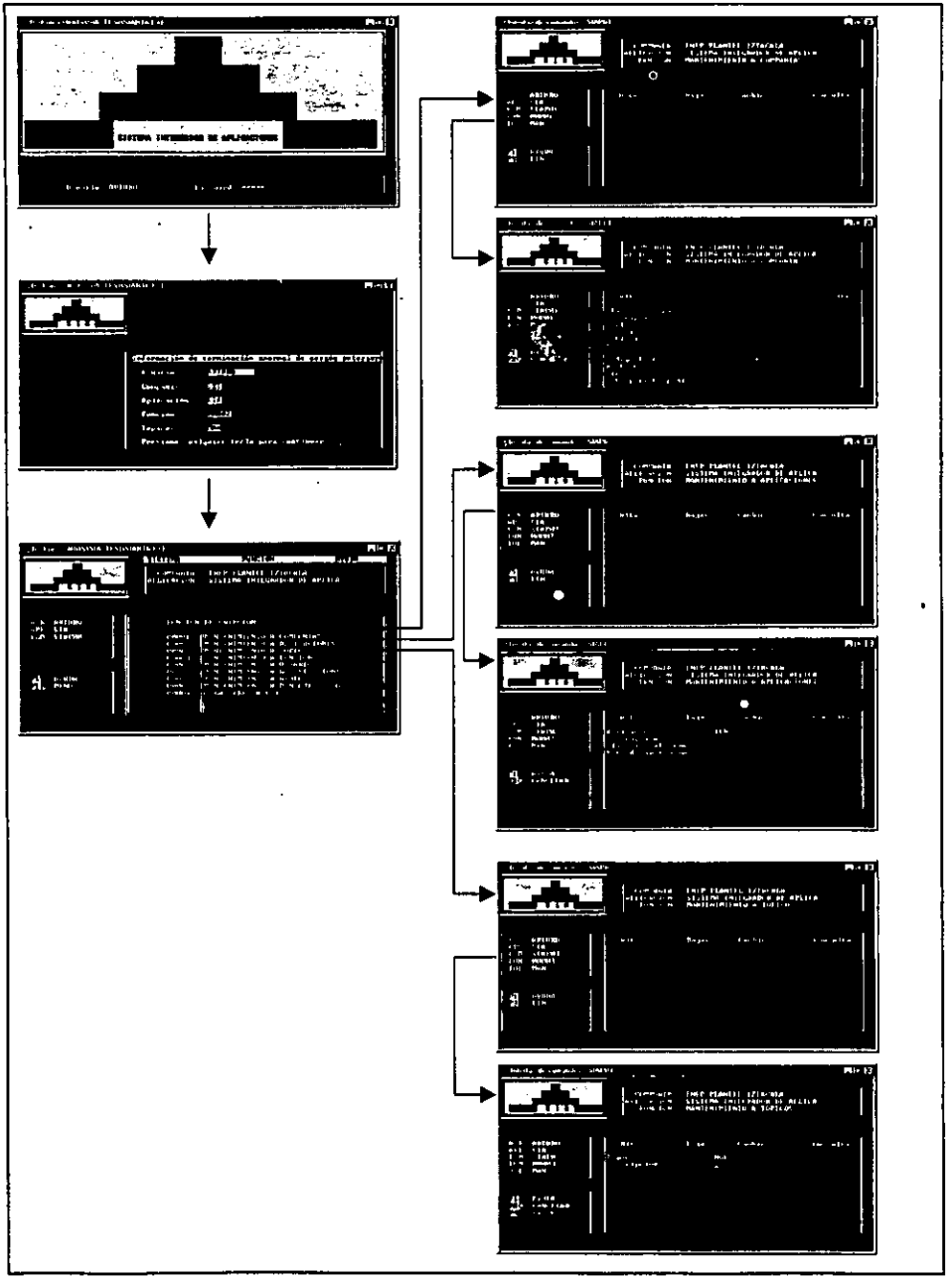


**Fig. 5.10. (a)** Secuencia de pantallas de los mantenimientos a S.I.A.

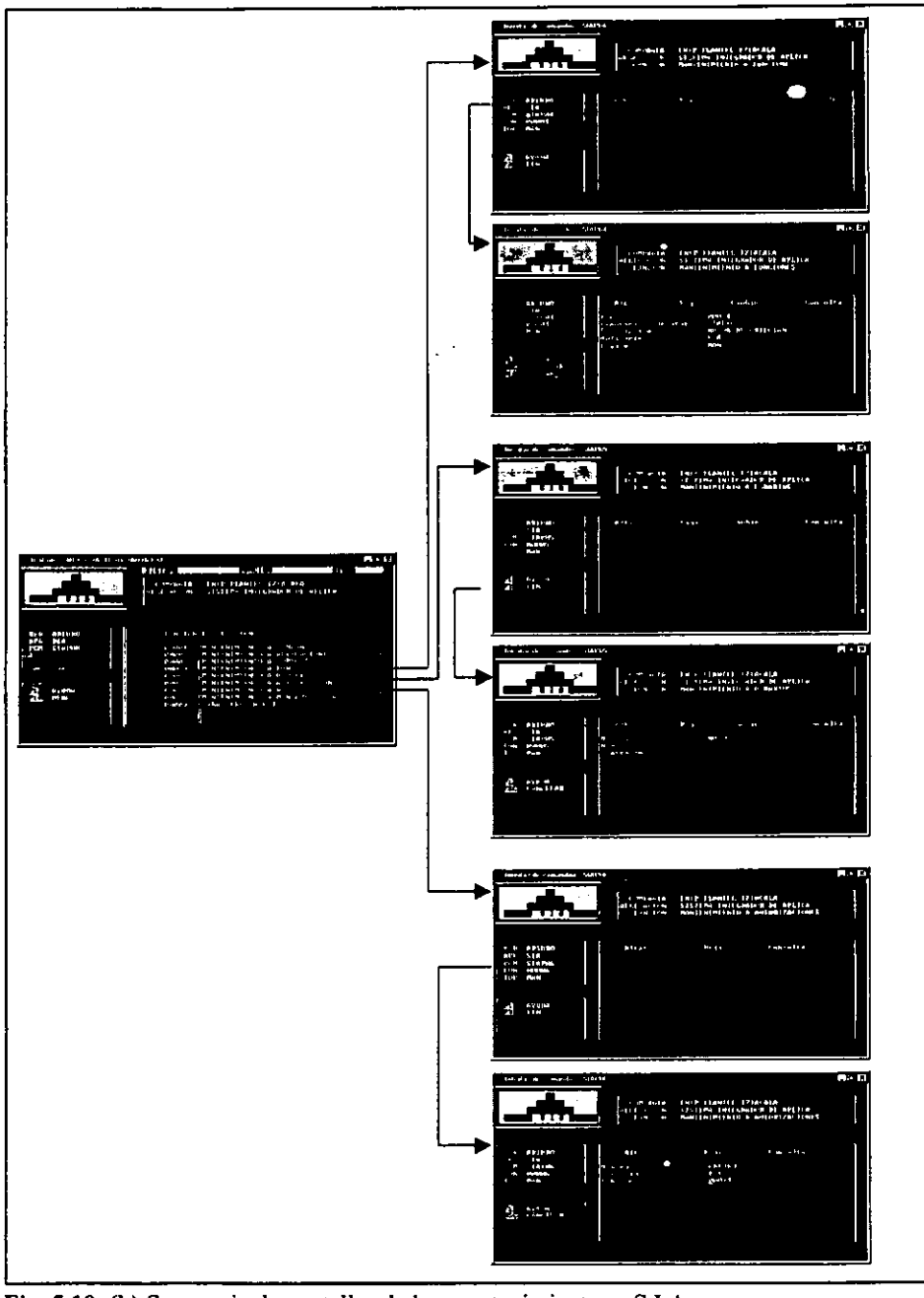


**Fig. 5.10.** (b) Secuencia de pantallas de los mantenimientos a S.I.A

### 5. 5. 3. Reportes de Control generados por S.I.A.

Con la intención de que el administrador de sistema logre tener un buen control, es indispensable contar con la información en línea que le permita visualizar en forma clara como esta organizado su entorno. Para este fin a continuación se presentan algunos de los principales reportes de control generados por S.I.A. Estos reportes permiten al usuario tener una relación de cada uno de los componentes de S.I.A., es decir, nos permiten conocer los usuarios definidos en el sistema y sus características, las compañías definidas, las aplicaciones habilitadas, las funciones por aplicación y sus rangos reservados, las funciones autorizadas por usuario, ultimo día y hora en que inicio y terminó su sesión, etc. Todos los reportes aquí presentados siguen los estándares definidos anteriormente en este mismo capítulo.

| | | | |
|---|---|---|---|
| SIAP065 | Compañía de Prueba | SIA | 22/05/1998 |
| Usuario: Admin | **Reporte de Aplicaciones de SIA** | | 15:00:42 |

| Clave de Aplicacción | Descripción | Clave de Instalación | Ruta de Acceso |
|---|---|---|---|
| ACF | ACTIVO FIJO | 1 | C:\ACF |
| ACF | FUNCIONES GENERALES ACTIVO FIJ | 2 | |
| ACF | PROCESOS DE OPERACION ACTIVO F | 2 | |
| ACF | ACTIVO FIJO PRESUPUESTOS | 2 | |
| ACF | ACTIVO FIJO PROGRAMACION | 2 | |
| ACF | TABLAS DE ACTIVO FIJO | 2 | |
| SIA | SISTEMA INTEGRADOR DE APLIACIONES | 1 | C:\SIA |
| SIA | SIA - BITACORAS | 1 | C:\SIA |
| SIA | SIA - COMUNICACIONES | 1 | C:\SIA |
| SIA | SIA - CONSULTAS | 1 | C:\SIA |
| SIA | SIA - CORREO ELECTRONICO | 1 | C:\SIA |
| SIA | SIA -DIRECTORIO TELEFONICO | 1 | C:\SIA |
| SIA | SIA - MANTENIMIENTOS | 1 | C:\SIA |
| SIA | SIA - OPERACION GENERAL | 1 | C:\SIA |
| SIA | SIA - EMISION DE REPORTES | 1 | C:\SIA |

SIAP065 Compañía de Prueba SIA 22/05/1998

Usuario: Admin **Reporte de Funciones de SIA** 15:00:42

| Función | Descripción | Aplicación | Tópico | Programa |
|---|---|---|---|---|
| 1 | Captura Movimientos de Activo Fijo | ACF | 10 | ACFC001 |
| 2 | Correcciones Captura de Movimiento. | ACF | 10 | ACFC002 |
| 3 | Tablas de C¢digos Relacionados | ACF | 20 | ACFC003 |
| 4 | Tablas de C¢digos Relacionados | ACF | 30 | ACFC004 |
| 5 | Tablas de Depreciaciones | ACF | 20 | ACFC005 |
| 6 | Tablas de Depreciaciones | ACF | 30 | ACFC006 |
| 7 | Indices Nacionales de Precios | ACF | 20 | ACFC007 |
| 8 | Indices Nacionales de Precios | ACF | 30 | ACFC008 |
| 9 | Registro De Planta Por Llave Gral. | ACF | 10 | ACFC009 |
| 10 | Reg. de Planta por Centro de Costo | ACF | 10 | ACFC010 |

SIAP065 Compañía de Prueba SIA 22/05/1998

Usuario: Admin **Reporte de Usuarios de SIA** 15:00:42

| Usuario | Nombre | Aplicación por defecto | Compañía por defecto |
|---|---|---|---|
| AALVAREZ | AZUCENA JIMENEZ CARRION | ACF | MLA |
| AARCEO | ALBERTO ARCEO CHABLE | SIA | NES |
| ABM | ARTURO BENITEZ MALDONADO | CON | NES |
| ACALDERON | ALBERTO CALDERON QUEZADA | ACF | NES |
| ADALBERTO | ADALBERTO MIRON MORENO | CON | NES |
| ADELA | ADELA CANO GANDARA | NOM | NES |
| ADIGR | ADRIANA GUTIERREZ RODRIGUEZ | SIA | NES |
| ADMIN | USUARIO TIPO ADMINISTRADOR | NOM | NES |
| ADOLF | ADOLFO DIAZ ANAYA | ACF | NES |
| ADOLFO | ADOLFO GUTIERREZ TORRES | NOM | NES |
| ADPE01 | ROBERTO BELTRAMI PARENTI | SIA | NES |
| ADPE02 | ALMA RODRIGUEZ LAZCANO | NOM | SIL |
| ADPE03 | EVELIA VALENCIA | NOM | NES |

SIAP065 Compañía de Prueba SIA 22/05/1998

Usuario: Admin **Reporte de Funciones Atorizadas** 15:00:42

| Usuario | Aplicación | Función | Compañía |
|---|---|---|---|
| AALVAREZ | AMS | 17001 | NES |
| | | 12104 | ALF |
| | | 10002 | NES |
| | | 10003 | PRU |
| | | 10016 | NES |
| | | 10036 | NES |
| | | 10078 | NES |
| ABM | AMS | 17001 | ALF |
| | | 17002 | ALF |
| | | 17003 | NES |
| | | 17005 | NES |
| | | 17007 | NES |
| AARCEO | AMS | 12104 | ALF |
| | | 10002 | ALF |
| | | 10003 | ALF |
| | | 10016 | ALF |
| | | 10078 | ALF |
| ADELA | AMS | 17027 | NES |
| | | 13001 | NES |
| | | 13004 | NES |
| | | 12104 | NES |
| | | 10002 | NES |
| | | 10003 | NES |
| | | 10016 | NES |
| | | 10036 | NES |
| | | 10078 | NES |
| | | 10131 | NES |

# CONCLUSIONES

No se puede decir que el conocimiento de lo ya existente y las ideas que se están plasmando en este documento, se lleven a cabo como aquí se sugiere. Cada uno de los analistas de sistemas y diseñadores de software desarrollan sus productos logrando obtener sus objetivos principales, pero no deja de ser importante que consideren situaciones que implican que sus productos puedan ser mejores.

Quizás, en el mercado ya exista un sin fin de productos de computo que ofrezcan una amplia gama de facilidades que permitan tener un mejor control y una eficiente seguridad de nuestras aplicaciones, pero pensar que nuestras necesidades se vuelven especiales y que los productos existentes son insuficientes, nos lleva a la tarea de invertir tiempo en nuestro propio diseño de un sistema de control y administración de aplicaciones. De aquí que, este documento puede resultar beneficioso y de gran ayuda, no sólo para el principiante en rama de la informática, si no para el analista y diseñador experimentado, que día con día, se preocupa por mantener sus sistemas en optimas condiciones y sobre todo, actualizados de acuerdo a la innovación de hardware y la demanda de software por parte de las empresas.

Aun cuando aquí se mencionaron varios ejemplos de administradores de aplicaciones, actualmente en el mercado han ido apareciendo sistemas que ya cumplen con estas características. Todavía más importante, es el hecho de que hoy, estamos entrando al gran mundo de las aplicaciones para las redes publicas (Internet) y para las redes privadas (Intranet) en el ámbito mundial, donde resulta super indispensable contar con una seguridad, un buen control acceso y confiable la estandarización.

# GLOSARIO

**ADAM**: Sistema de Aplicaciones Administrativas de la compañía TEA. No es una abreviación es un nombre al azar.

**Ambiente de ejecución**: Ambiente desde donde se puede ejecutar una función: Menús o Lista de funciones.

**AMS/400**: Application Manager System. Sistema Administrador de Aplicaciones diseñado por la Compañía Nestlè en México.

**APD/400**: Applicaction Program Driver/400. Sistema Administrador de Aplicaciones de I.B.M.

**AS/400**: Application System/400. Mini-computaroras marca I.B.M.

**Biblioteca**: Tipo de objeto en un equipo AS/400 que sirve para almacenar cualquiera de los tipos de objetos del sistema.

**Cifrado**: Conversión de secuencia de datos en otra secuencia diferente para generar un código no legible.

**Clave bloqueo**: Clave para definir la manera en la que una función podrá bloquear su

**Documentación en línea**: Servicio de ayuda en cada una de las funciones habilitadas en Sistema Administrador de Aplicaciones.

**Encriptaciòn**: Método de conversión de un código estándar visible a un código no visible o legible.

**Estándar**: Norma o regla a seguir en el diseño de sistemas.

**Función**: Unidad única de identificación de un trabajo o una tarea interactiva o batch.

**I.B.M.**: International Bussiness Machines. Marca de computadoras alemana.

**Intrusos activos**: son los intrusos más maliciosos, ya que estos buscan poder alterar los datos no autorizados.

**Intrusos Pasivos**: son los intrusos que sólo desean acceder información a la cual no tienen autorización

**LAN**: Local Area Network. Red de Area Local.

ejecución en caso de que otra función tenga contraposición con esta.

**Códigos expertos**: Códigos especiales para ocultar la información temporalmente.

**Cola de salida**: Listas de espera en donde se depositan los archivos de impresión generados por trabajos en el sistema.

**Control de Tareas**: Especifica si AMS/400 debe verificar qué tareas están en ejecución antes de dar acceso a una función, a efectos de efectuar alguna prohibición de ejecución simultánea y notificarla.

**Control de tareas concurrentes**: Control del AMS/400 indicar las funciones que no pueden ejecutarse en forma concurrente, y al momento de ejecución, realiza la verificación de concurrencias prohibidas de acuerdo a la estructura de niveles de control de tareas definidos.

**Correo electrónico**: Aplicación de mensajería de los usuarios entre los diferentes equipos que componen una red WAN.

**Descripción de trabajo**: Conjunto de valores que definen los recursos que utilizara un trabajo en un equipo AS/400.

**MA**: Manejador de Aplicaciones. Sistema Administrador de Aplicaciones de la compañía Grupo TEA.

**Mnemónico de función**: Código alfanumérico para identificar una función dentro del AMS/400.

**Multi-aplicaciones**: Termino que se le da un sistema que nos permite trabar con más de una aplicación al mismo tiempo.

**Multi-compañias**: Termino para definir a una aplicación que permite trabajar con la información de más de una compañía desde de la misma aplicación.

**Multi-idiomas**: Termino para definir que un sistema puede presentarse en diferentes idiomas.

**Multi-localidades**: Termino para definir que una aplicación puede acceder la información de más de una localidad en el mismo equipo.

**Multi-tareas**: Termino para definir que un equipo puede realizar más de una tarea el mismo tiempo.

**Mùlti-usuarios**: Termino para definir que un equipo puede dar servicio de procesador a más de un usuario al mismo tiempo.

**Multi-sesión**: Termino para definir que un dispositivo físico o estación de trabajo puede ofrecer más de una sesión con el mismo dispositivo.

**Objeto**: Cualquier cosa: programa, comando, biblioteca, archivo, etc., dentro de un equipo AS/400.

**OS/400**: Operating System/400. Sistema operativo del AS/400.

**Reidentificaciòn**: Confirmación de firma en una sesión para asegurar que el usuario que entro al sistema es el mismo que esta ejecutando una función de gran importancia.

**S.A.A.**: System Application Arquitecture

**S.I.A.**: Sistema Integrador de Aplicaciones.

**S.Q.L.**: System Query Language

**Sesión**: Conexión virtual de un dispositivo físico hacia un equipo multi-usuario.

Trabajo interactivo entre una terminal o PC y un equipo multi-usuario.

**Subsistema**: Definición de un conjunto de recursos y especificaciones del sistema que atiende a una serie de trabajos interactivos y batches.

**Tarea batch**: Proceso en lotes que ejecuta un equipo en forma desatendida.

**Tarea interactiva**: Proceso en línea que ejecuta un equipo en forma atendida.

**TEA**: Tecnología en Administración e Informática

**Usuario de firma múltiple**: Usuario que puede firmarse al sistema en más de una sesión al mismo tiempo.

**Virus Informático**: Programa, comúnmente, destructor de información en equipos de computo. Este tipo de programas se puede filtrar a las computaras insertando disquetes contaminados o mediante la una red.

**WAN**: Wide Area Network. Red de Area Remota.

# APENDICE "A"

# DEFINICIONES DE ARCHIVOS DE S.I.A.

Los nombres de los archivos que usa S.I.A. están formados siguiendo los estándares establecidos para esta aplicación. Los nombres de los campos no deben exceder de 6 caracteres. En todo lo posible se deberá respetar que todos los campos dentro de un solo archivo contengan en su nombre los tres primeros caracteres iguales, es decir, un prefijo que haga referencia al contenido del archivo.

Las abreviaciones utilizadas para definir tipo de campo son:

C=Caracter,

N=Númerico y

M=Memo.

- Archivo de Compañías (SIAA01).

| Campo | Tipo | Longitud | Texto |
|---|---|---|---|
| CIACVE | C | 3 | Clave de compañía |
| CIADES | C | 30 | Descripción de la compañía |
| CIACAL | C | 30 | Calle |
| CIACOL | C | 30 | Colonia |
| CIAPOB | C | 30 | Población |
| CIAEST | C | 30 | Estado |
| CIAPAI | C | 30 | País |
| CIACPO | N | 5 | Código Postal |
| CIARFC | C | 15 | Registro Fiscal |
| CIAGIR | C | 30 | Giro o Actividad de la Compañía |
| CIARUT | C | 30 | Ruta de acceso para la compañía |

- Archivo de Aplicaciones (SIAA02).

| Campo | Tipo | Longitud | Texto |
|---|---|---|---|
| APLCVE | C | 3 | Clave de la Aplicación |
| APLDES | C | 30 | Descripción de la Aplicación |
| APLINS | C | 1 | Clave de estado de instalación. |
| APLRUT | C | 30 | Ruta de acceso de la Aplicación |

- Archivo de Tópicos (SIAA03).

| Campo | Tipo | Longitud | Texto |
|---|---|---|---|
| TOPCVE | C | 3 | Clave de Tópico |
| TOPDES | C | 30 | Descripción de Tópico |

- Archivo de Funciones (SIAA04).

| Campo | Tipo | Longitud | Texto |
|---|---|---|---|
| FUNCVE | N | 5,0 | Clave de la Función |
| FUNPGM | C | 10 | Programa a ejecutar por la Función |
| FUNDES | C | 30 | Descripción de la Función |
| APLCVE | C | 3 | Clave de Aplicación relacionada |
| TOPCVE | C | 3 | Clave de Tópico relacionado |

- Archivo de Usuarios (SIAA05).

| Campo | Tipo | Longitud | Texto |
|---|---|---|---|
| USUCVE | C | 10 | Clave de Usuario |
| USUNOM | C | 30 | Nombre de Usuario |
| USUCON | C | 6 | Contraseña de Usuario |
| CIACVE | C | 3 | Clave de Compañía por defecto |
| APLCVE | C | 3 | Clave de Aplicación por defecto |

- Archivo de Mensajes (SIAA06).

| Campo | Tipo | Longitud | Texto |
|---|---|---|---|
| MSGCOD | C | 7 | Código de mensaje |
| MSGDP1 | C | 30 | Primera parte de mensaje |
| MSGDP2 | C | 30 | Segunda parte de mensaje |

- Archivo de Ayudas (SIAA99).

| Campo | Tipo | Longitud | Texto |
|---|---|---|---|
| FUNCVE | N | 5 | Clave de función relacionada |
| AYUTIT | C | 30 | Titulo de la ayuda |
| AYUTEX | C | 30 | Texto de la ayuda |

La siguiente definición de archivo nos permitirá formar las diferentes estructuras de las pantallas que necesitarán los programas de mantenimiento para cada uno de los archivos mencionados anteriormente. La ventaja de este tipo de archivos radica en la facilidad de cambiar el texto de una leyenda para la captura de un campo y para variar las posiciones de este campo dentro de la pantalla, sin necesidad de compilar el programa en el que se usa o se usan.

- Archivo de Estructura.

| Campo | Tipo | Longitud | Texto |
|---|---|---|---|
| CAMPO | C | 8 | Nombre del campo relacionado |
| TIPO | C | 1 | Tipo del campo relacionado |
| LON | N | 2 | Longitud del campo relacionado |
| FILA | N | 2 | Número de línea en la pantalla |
| COL | N | 2 | Número de columna en la pantalla |
| LEYENDA | C | 20 | Leyenda del campo relacionado |
| MASCARA | C | 10 | Mascara de edición en la pantalla |

Los archivos de estructura usados en S.I.A. de a cuerdo a la definición anterior son:

- SIAD01 = Archivo de estructura para la pantalla en mantenimiento a compañías.

- SIAD02 = Archivo de estructura para la pantalla en mantenimiento a aplicaciones.

- SIAD03 = Archivo de estructura para la pantalla en mantenimiento a tópicos.

- SIAD04 = Archivo de estructura para la pantalla en mantenimiento a funciones.

- SIAD05 = Archivo de estructura para la pantalla en mantenimiento a usuarios.

- SIAD06 = Archivo de estructura para la pantalla en mantenimiento a mensajes.

- SIAD10 = Archivo de estructura para la pantalla en mantenimiento a autorización de funciones a usuarios.

- SIAD99 = Archivo de estructura para la pantalla en mantenimiento a ayudas del sistema.

Las siguientes figuras muestran la relación de entidades de las bases de datos de S.I.A.

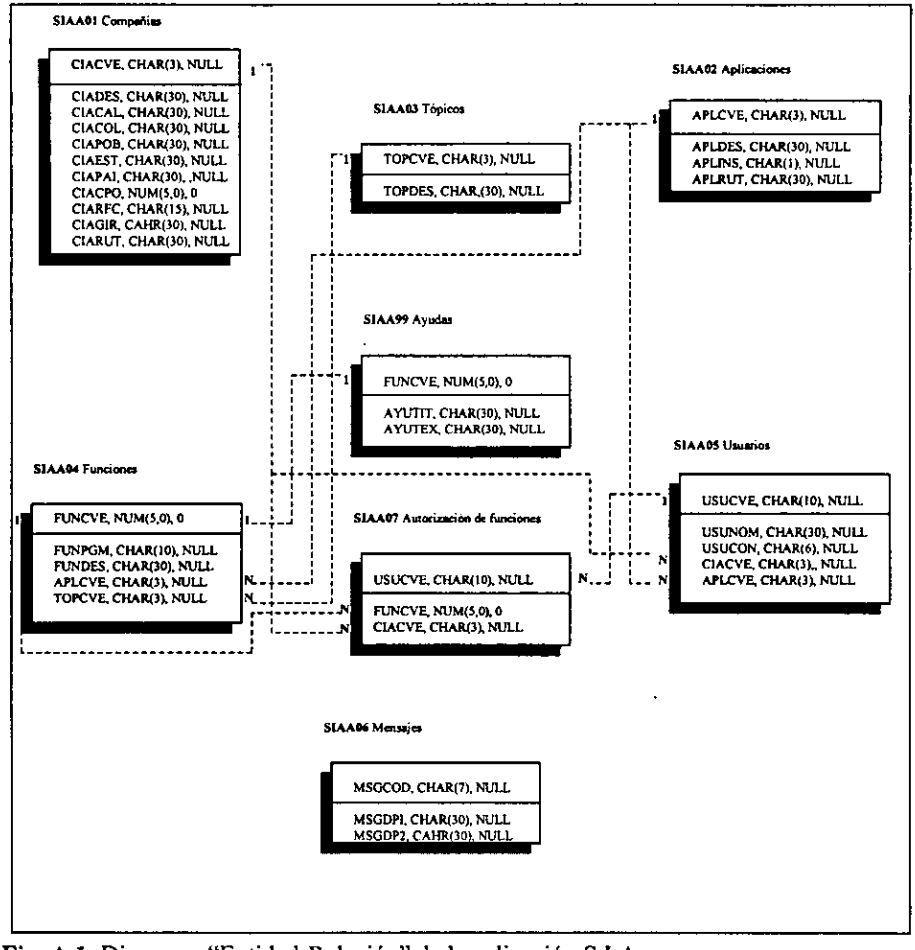



**Fig. A.1.** Diagrama "Entidad-Relación" de la aplicación S.I.A.

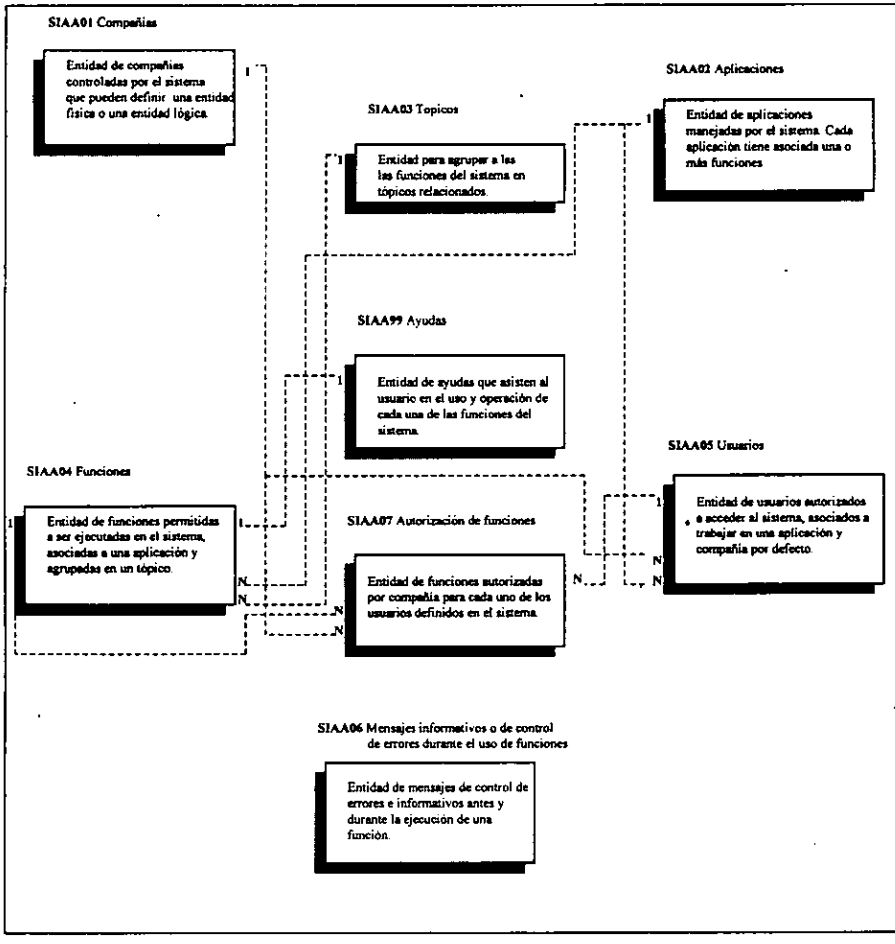



**Fig. A.2.** Diagrama "Entidad-Relación" de la aplicación S.I.A. (Conceptual).

# APENDICE "B"

# PROCESOS DE MANTENIMIENTO QUE INTEGRAN S.I.A.

*Mantenimiento a Compañías.*

Altas, Bajas, Cambios y Consultas al archivo de Compañías. Está compuesto de una clave de compañía, descripción de compañía y dirección de compañía (colonia, población, estado, país, código postal, registro de la compañía y giro o actividad de la compañía). Ver figura B.1.

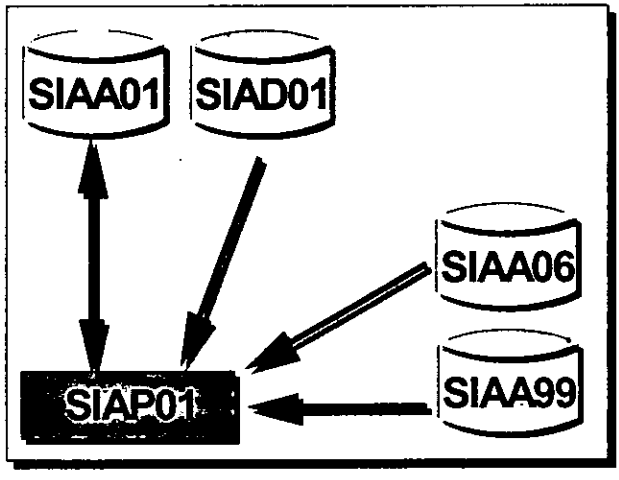



**Fig. B.1.** Flujo de Mantenimiento a Compañías.

*Mantenimiento a Aplicaciones.*

Altas, Bajas, Cambios y Consultas a archivo de Aplicaciones/Módulos. Los parámetros a definir son: clave de la aplicación, descripción de la aplicación y una clave que define si la aplicación esta instalada o no. Ver figura B.2.

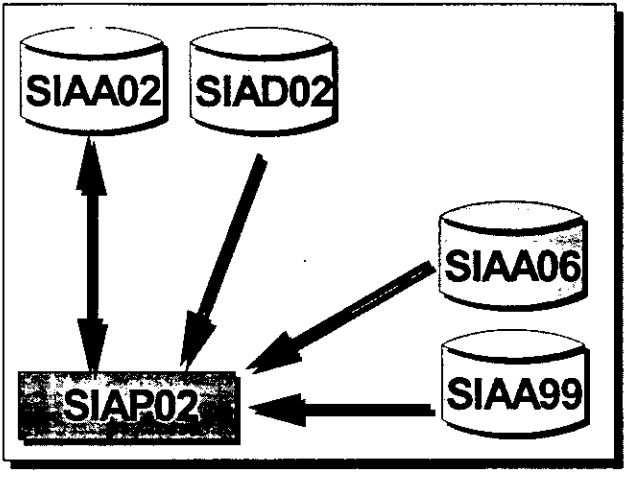



**Fig. B.2.** Flujo de Mantenimiento a Aplicaciones.

*Mantenimiento a Tópicos.*

Altas, Bajas, Cambios y Consultas al archivo de Tópicos. Los parámetros a definir son: clave del tópico, y descripción del tópico. Ver figura B.3.

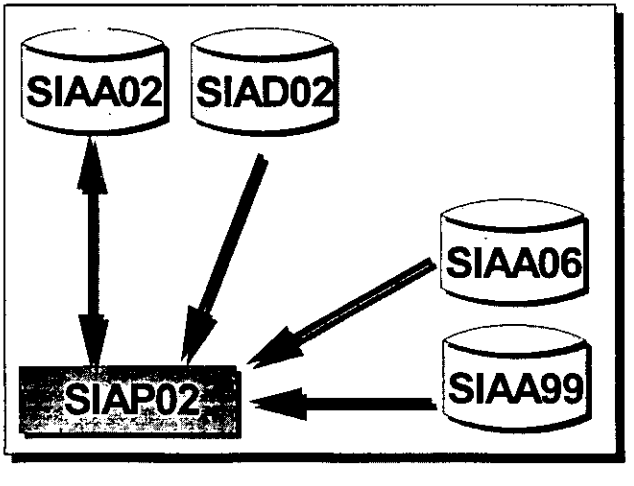



**Fig. B.3.** Flujo de Mantenimiento a Tópicos.

*Mantenimiento a Funciones.*

Altas, Bajas, Cambios y Consultas al archivo de Funciones. Los parámetros a definir son: clave de la función, descripción de la función, programa a ejecutar por la función, clave del modulo a la que se relaciona y la aplicación a la que se relaciona. Ver figura B.4.

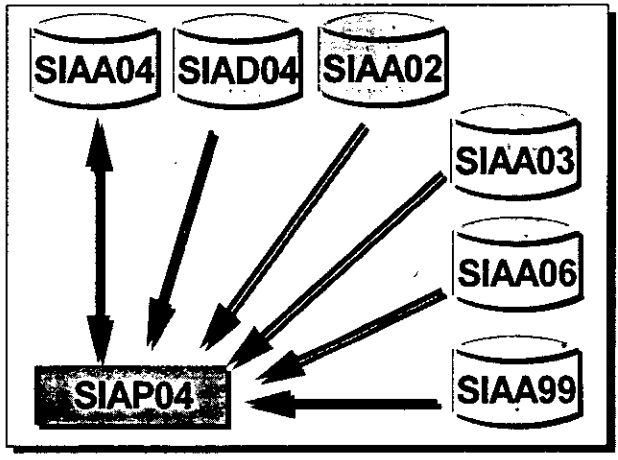



**Fig. B.4.** Flujo de Mantenimiento a Funciones.

*Mantenimiento a Usuarios.*

Altas, Bajas, Cambios y Consultas al archivo de Usuarios. Los parámetros a definir son: Clave del usuario, nombre del usuario, contraseña no visible y clave de la compañía por defecto. Ver figura B.5.

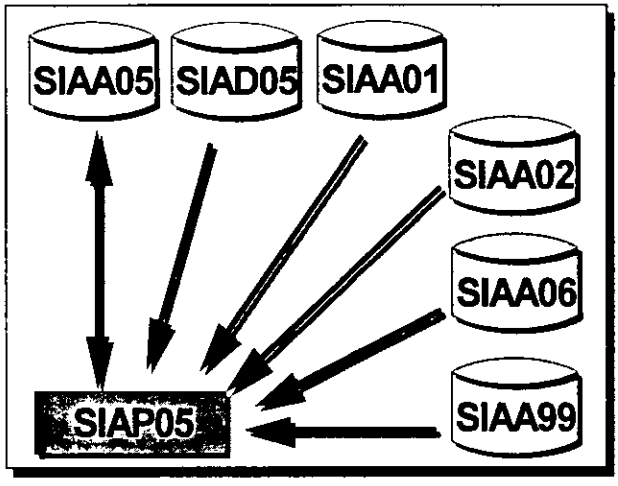



**Fig. B.5.** Flujo de Mantenimiento a Usuarios.

*Mantenimiento a Mensajes.*

Altas Bajas, Cambios y Consultas al archivo de mensajes. Los parámetros a definir son: clave identificador de mensaje y texto de mensaje. Ver figura 5.6.

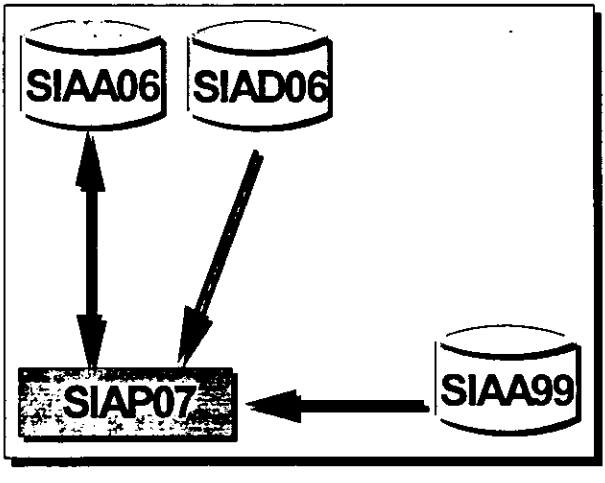



**Fig. B.6.** Flujo de Mantenimiento a Mensajes del Sistema.

*Mantenimiento a Ayudas del Sistema.*

Altas, Bajas, Cambios y Bajas al archivo del Ayudas. Los parámetros a definir son: clave de la función a la que hace relación la ayuda, identificación de secuencia de la ayuda dentro de la función, texto de la ayuda y titulo desplegar para la ayuda. Ver figura B.7.

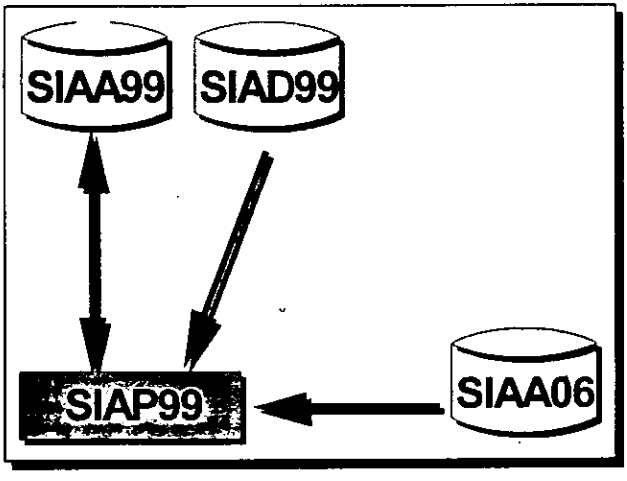



**Fig. B.7.** Flujo de Mantenimiento a Ayudas del Sistema.

*Mantenimiento a Autorizaciones.*

Aquí se determinará las funciones a las cuales el usuario tiene autorización. Se podrá dar Altas y Bajas. Se deberá definir bajo que compañía y que aplicación se estará autorizando la función o las funciones a determinado usuario. Ver figura B.8.

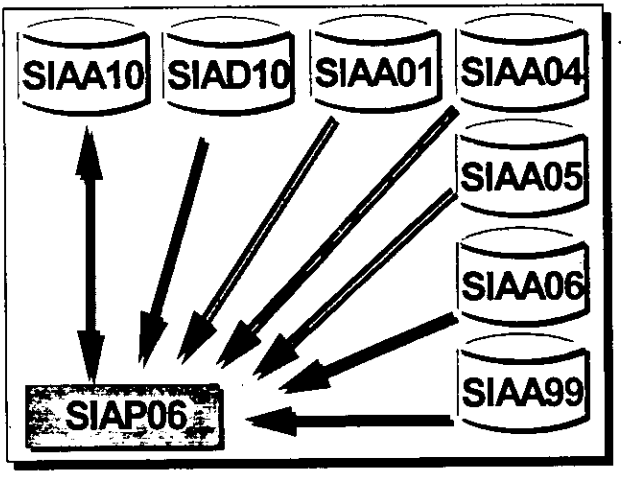



**Fig. B.8.** Flujo de Mantenimiento a Autorización de Funciones.

# BIBLIOGRAFIA

- Aldrete Bernal Fernando

***SEGURIDAD EN CENTROS DE COMPUTO***

Editorial: Trillas

Edición: 1988

- Anderassen Karl

***COMPUTER CRYPTOLOGY BEYOND DECODER RINGS***

Editorial: Prentice Hall

Edición: 1988

- Echenique José Antonio

***AUDITORIA EN INFORMÁTICA***

Editorial: McGRAW-HILL

Edición: 1990

- Fairley Richard

***INGENIERIA DE SOFTWARE***

Editorial: McGRAW-HILL

Edición: 1989.

- Fine Leonard H.

***SEGURIDAD EN CENTROS DE COMPUTO, POLÍTICAS Y PROCEDIMIENTOS***

Editorial: Trillas

Edición: 1994

- Frizgerald Jeny

***CONTROLES INTERNOS PARA SISTEMAS DE COMPUTACIÓN***

Editorial: Limusa

Edición: 1993.

Año 15 No. 429 27 de marzo de 1995
Año 15 No. 431 24 de abril de 1995
Año 16 No. 433 15 de mayo de 1995

- James A. Senn

***ANALISIS Y DISEÑO DE SISTEMAS DE INFORMACION***
Editorial: McGRAW-HILL
Edición: 1988.

- Rodríguez Luis Angel

***SEGURIDAD EN LA INFORMACION EN SISTEMAS DE COMPUTO***
Editorial: Ventura
Edición: 1995

- Servicios Editoriales SAYROLS S.A. de C.V.

***PERSONAL COMPUTING MÉXICO***
México Ejemplar JUNIO de 1995

- Sommerville Ian

***INGENIERIA DE SOFTWARE***
Editorial: SITESA
Edición: 1988.

- Wiederhold Gio

***DISEÑO DE BASE DE DATOS***
Editorial: McGRAW-HILL
Edición: 1991.

- Grupo INFOSOL S.A de C.V.

***TECNO MUNDO*** (Boletín Semanal en informática y Telecomunicaciones)
México D.F.
Ejemplares consultados:
Año IV No. 251 20 de Enero de 1995
Año IV No. 253 03 de Febrero de 1995
Año IV No. 255 17 de Febrero de 1995
Año IV No. 258 10 de Marzo de 1995
Año IV No. 259 17 de Marzo de 1995
Año IV No. 260 24 de Marzo de 1995
Año IV No. 263 14 de Abril de 1995
Año IV No. 267 12 de Mayo de 1995
Año IV No. 271 09 de Junio d 1995
Año IV No. 273 23 de Junio de 1995
Año IV No. 279 04 de agosto de 1995

- Harvey M. Deitel

***INTRODUCCION A LOS SISTENAS OPERATIVOS***
Editorial: Iberoamericana
Edición: Primera edición.

- IDG Comunicaciones S.A. de C.V.

***COMPUTERWORLD MÉXICO***
México D. F.
Ejemplares consultados:
Año 15 No. 406 18 de julio de 1994
Año 15 No. 415 17 de octubre de 1994
Año 15 No. 425 13 de febrero de 1995
Año 15 No. 426 27 de marzo de 1995
Año 15 No. 427 06 de marzo de 1995
Año 15 No. 428 20 de marzo de 1995