

59



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

JEFATURA DE AREA DE INGENIERIA MECANICA ELECTRICA CAMPUS ARAGON

“DESARROLLO DE UN SISTEMA DE COMUNICACION ENTRE PROCESOS REMOTOS EN AS/400”

25-2-

T E S I S

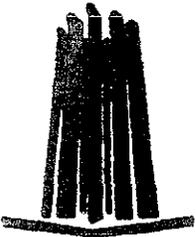
QUE PARA OBTENER EL TITULO DE:

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A :

EDUARDO MARTINEZ SALINAS

ASESOR DE LA TESIS: ING. RAUL BARRON VERA



SAN JUAN DE ARAGON. EDO. DE MEX. MEXICO



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

A mi madre

Por darme la vida y ser siempre un ejemplo de bondad y amor, por ayudarme en mis primeros pasos por la vida e impulsarme en el camino del estudio.

A mi padre

Por predicar siempre con el ejemplo, por inculcar siempre el amor al trabajo, por ser la persona más congruente consigo mismo que he conocido y por haber tenido el coraje, la voluntad y la inteligencia necesaria para guiar a su familia por un buen camino.

A Laura

Por ser esa persona maravillosa que me impulsa a superarme, por llenar de alegría mi vida, por el su amor y cariño, el cual espero ser merecedor siempre, por su apoyo y ayuda en el presente trabajo, y sobre todo, por permitirme compartir con ella tristezas y alegrías.

A mi asesor

Por su excelente guía en la realización de este trabajo, por su disponibilidad, apoyo y sugerencias.

INTRODUCCIÓN

CAPÍTULO I SISTEMA AS / 400

| | | |
|-------|---|----|
| I.1 | Qué es un sistema AS/400..... | 2 |
| I.2 | Opciones de conectividad..... | 3 |
| I.3 | Arquitectura del sistema AS/400..... | 4 |
| I.4 | Sistema operativo..... | 8 |
| I.4.1 | Archivo físico..... | 9 |
| I.4.2 | Archivo lógico..... | 10 |
| I.4.3 | Pantallas..... | 10 |
| I.4.4 | Programas RPG..... | 11 |
| I.4.5 | Programas CL..... | 11 |
| I.4.6 | Perfiles de usuario..... | 11 |
| I.4.7 | Áreas de datos..... | 12 |
| I.4.8 | Utilerías..... | 13 |
| I.5 | Necesidad de compartir información con un sistema remoto..... | 23 |
| I.5.1 | Directorio de Distribución..... | 23 |
| I.5.2 | Colas de distribución..... | 24 |
| I.6 | Opciones para compartir información en el AS/400..... | 25 |
| I.6.1 | Administrador de Distribución de Datos (DDM)..... | 25 |
| I.6.2 | Protocolo de Transferencia de Archivos (FTP)..... | 30 |
| I.6.3 | Espejeo de Datos..... | 32 |
| I.6.4 | Transferencia con comandos del AS/400..... | 33 |
| I.7 | Hardware Básico..... | 39 |

CAPITULO II LENGUAJES DE PROGRAMACIÓN.

| | | |
|--------|--|----|
| II.1 | Lenguajes de Control..... | 44 |
| II.2 | Comandos que operan en los objetos OS/400..... | 45 |
| II.3 | Sintaxis del lenguaje de control..... | 47 |
| II.3.1 | Etiqueta del comando..... | 47 |
| II.3.2 | Nombre del comando..... | 49 |
| II.3.3 | Parámetros del comando..... | 49 |
| II.3.4 | Sintaxis de los comandos..... | 51 |
| II.3.5 | Delimitadores del comando..... | 52 |
| II.3.6 | Caracteres especiales..... | 53 |
| II.4 | Lenguaje RPG..... | 54 |
| II.4.1 | Control..... | 55 |
| II.4.2 | Descripción de archivos..... | 56 |
| II.4.3 | Extensión..... | 57 |
| II.4.4 | Contador de líneas..... | 57 |
| II.4.5 | Entrada..... | 57 |

| | |
|--|----|
| II.4.6 Cálculo..... | 57 |
| II.4.7 Salida..... | 58 |
| II.5 Códigos de operación..... | 58 |
| II.5.1 Operaciones aritméticas..... | 59 |
| II.5.2 Operaciones de mover..... | 61 |
| II.5.3 Operaciones de comparación..... | 62 |
| II.5.4 Operaciones declarativas..... | 63 |
| II.5.5 Operaciones de inicialización..... | 64 |
| II.5.6 Operaciones con series de caracteres..... | 64 |
| II.5.7 Operaciones de subrutinas..... | 65 |
| II.5.8 Operaciones de programación estructurada..... | 65 |
| II.5.9 Operaciones con archivos..... | 66 |
| II.5.10 Operaciones de llamado de programas..... | 67 |
| II.5.11 Operaciones con arreglos..... | 67 |
| II.5.12 Operaciones con áreas de datos..... | 68 |

CAPITULO III DESARROLLO DE UN SISTEMA.

| | |
|---|----|
| III.1 Sistema..... | 70 |
| III.2 Metodología de implementación..... | 71 |
| III.2.1 Análisis..... | 73 |
| III.2.1.1 Análisis preliminar..... | 75 |
| III.2.1.2 Análisis detallado..... | 77 |
| III.2.2 Diseño..... | 79 |
| III.2.2.1 Diseño funcional..... | 82 |
| III.2.2.2 Diseño técnico..... | 83 |
| III.2.3 Desarrollo..... | 84 |
| III.2.3.1 Creación de la base de datos..... | 86 |
| III.2.3.2 Codificación..... | 87 |
| III.2.3.3 Pruebas unitarias..... | 88 |

CAPÍTULO IV DESARROLLO DEL SISTEMA PARA COMUNICACIÓN ENTRE PROGRAMAS EN AS/400

| | |
|---|----|
| IV.1 Análisis..... | 90 |
| IV.1.1 Análisis preliminar..... | 90 |
| IV.1.1.1 Antecedentes..... | 90 |
| IV.1.1.2 Definición de objetivos..... | 93 |
| IV.1.2 Análisis detallado..... | 93 |
| IV.1.2.1 Definición de los alcances..... | 94 |
| IV.1.2.2 Características generales del sistema..... | 95 |
| IV.1.2.3 Descripción de términos especiales..... | 95 |
| IV.1.2.4 Identificación de usuarios..... | 97 |
| IV.1.2.5 Flujo actual de información..... | 97 |
| IV.1.2.6 Requerimientos funcionales..... | 99 |
| IV.1.2.7 Requerimientos no funcionales..... | 99 |

| | |
|--|-----|
| IV 2 Diseño..... | 99 |
| IV.2.1 Diseño funcional..... | 99 |
| IV.2.1.1 Diagrama conceptual de usuario..... | 100 |
| IV.2.1.2 Diagrama jerárquico del sistema..... | 101 |
| IV.2.1.3 Modelo conceptual de datos..... | 102 |
| IV.2.2 Diseño técnico..... | 103 |
| IV.2.2 1 Definición de estándares..... | 103 |
| IV.2.2 2 Modelo físico de datos..... | 105 |
| IV 2.2.3 Diagrama de flujo de datos..... | 106 |
| IV 2.2.4 Especificación de procesos..... | 108 |
| IV.3 Desarrollo..... | 112 |
| IV.3.1 Mantenimiento a las Bases de Datos del Sistema..... | 112 |
| IV.3.1.1 Definición del área de almacenamiento..... | 112 |
| IV.3.1 2 Creación de fuentes..... | 112 |
| IV.3.1 3 Generación de archivos..... | 115 |
| IV.3.2 Codificación..... | 116 |
| IV.3.2 1 Codificación de pantallas y reportes..... | 116 |
| IV 3.2.2 Codificación de programas..... | 126 |
| IV 4 Mantenimiento a las bases de datos del sistema..... | 163 |
| Conclusiones | 171 |
| Bibliografía | 175 |

INTRODUCCIÓN

La plataforma informática AS/400 (Sistema de Aplicación 400) es la máquina computacional más exitosa de la historia. Su éxito lo debe a las ventajas que ofrece su arquitectura en el manejo de negocios de las organizaciones que lo utilizan.

Debido al volumen de información que manejan algunas organizaciones es necesaria la utilización de más de una máquina para el almacenamiento y proceso de datos. Al hacer uso, una organización de más de una máquina, los procesos e información de las diferentes áreas que la integran quedan divididas en las máquinas. Sin embargo y debido a la interrelación de las áreas, es necesario compartir información y, por lo tanto, mantener comunicación entre máquinas.

La existencia de las organizaciones está basada en su productividad, la cual a su vez está apoyada por la rapidez y eficiencia de sus sistemas de información. Cuando por necesidad se utilizan más de una máquina para manejar toda la información y procesos de una empresa, la eficiencia en la transmisión de datos juega un papel muy importante.

En la presente investigación se estudia el computador AS/400, su arquitectura, características, lenguajes de programación principales, sus diferentes opciones para la transmisión de información y se desarrolla una aplicación para mejorar la rapidez de ésta.

La presente investigación, en su capítulo uno, describe el concepto de AS/400, su arquitectura, sus características principales, sus funciones en el ambiente general de negocios y sus opciones de conectividad. Se presentan también las principales formas de compartir información entre máquinas AS/400, sus ventajas y desventajas. Se presentan las principales utilerías del AS/400 empleadas en el desarrollo de sistemas (PDM, SDA), los cuales constan de editores para la codificación de los programas y de una herramienta que permite diseñar pantallas, colocar campos, dar presentación y genera automáticamente el código fuente de ésta.

Los principales lenguajes que se utilizan para el desarrollo de aplicaciones en el AS/400 son el RPG y el CL los cuales son tratados en el segundo capítulo. Se da una explicación de cada una de las “hojas” utilizadas en RPG y el orden de las mismas dentro de un programa, sus instrucciones principales y su utilización dentro de las aplicaciones, así como de la utilización de comandos en el lenguaje CL y la sintaxis de éstos.

El procedimiento para la realización de un sistema, desde el estudio previo que se debe hacer hasta el desarrollo final es el tema principal que conforma al capítulo tres de la presente investigación. Tomando en cuenta diferentes teorías se obtiene una metodología para la realización de sistemas de información, la cual nos indica los pasos a seguir para tener un buen resultado en la realización del sistema.

En el capítulo cuatro se desarrolla un sistema para la optimización de las transferencias de información entre procesos remotos mediante el uso de comandos de transferencia de archivos propios del AS/400 realizando los pasos de la metodología del capítulo tres (análisis, diseño y desarrollo). Se presentan también todos los archivos fuente de los programas, pantallas, archivos físicos y lógicos, así como las principales pantallas de mantenimiento de datos del sistema y la forma de alimentar a éstos. El aumento en la rapidez de transferencia de información que proporciona la aplicación del sistema desarrollado en el último capítulo, se traduce en un mejor desempeño de los sistemas de información que requieren datos de diferentes máquinas. Este aumento en la rapidez de la transferencia de información se debe al estudio de las comunicaciones entre equipos AS/400, el cual, permitió descubrir la forma más eficiente de realizar dichas transferencias entre los mismos y aplicarla en el desarrollo del sistema.

CAPÍTULO I

SISTEMA AS / 400

CAPÍTULO I SISTEMA AS/400

I.1 QUÉ ES UN SISTEMA AS/400

El AS/400 (Sistema de Aplicación 400) es un computador de IBM que gracias a su arquitectura es una máquina computacional optimizada para el ambiente general de negocios donde se manejan grandes volúmenes de información. Esta arquitectura le proporciona a la vez integridad, consistencia, precisión, y seguridad de su base de datos. El sistema AS/400 se utiliza para el manejo de la información de las diferentes áreas que integran a una organización como se muestra en la figura 1.1

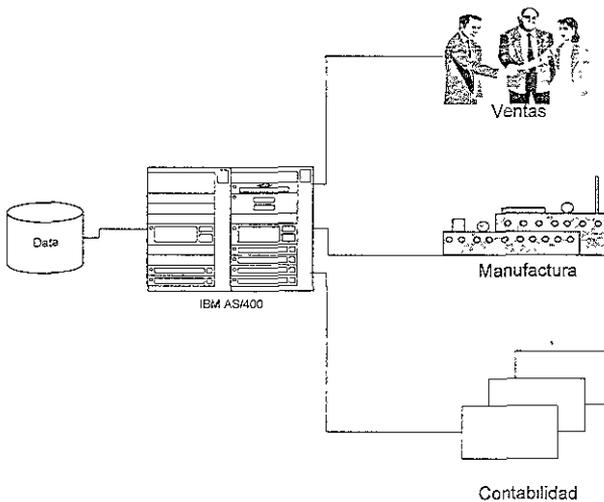


FIG. I.1 Manejo de información de las áreas que integran a una organización

1.2 OPCIONES DE CONECTIVIDAD

El sistema AS/400 tiene muchas facilidades de comunicación, opción de conectarse con servidores de red, conexión con impresoras de funciones avanzadas, sistemas expertos, multimedia, telefonía, etc. (ver figura 1.2). Estas opciones de conectividad representan grandes ventajas, por ejemplo, los usuarios al conectarse a través de la red, pueden acceder al sistema por medio de su propia computadora personal y no es necesario que tengan una terminal propia de AS/400, por medio de la red se pueden conectar a las mismas impresoras y compartirlas con las PC's, evitando así la adquisición de impresoras especiales.

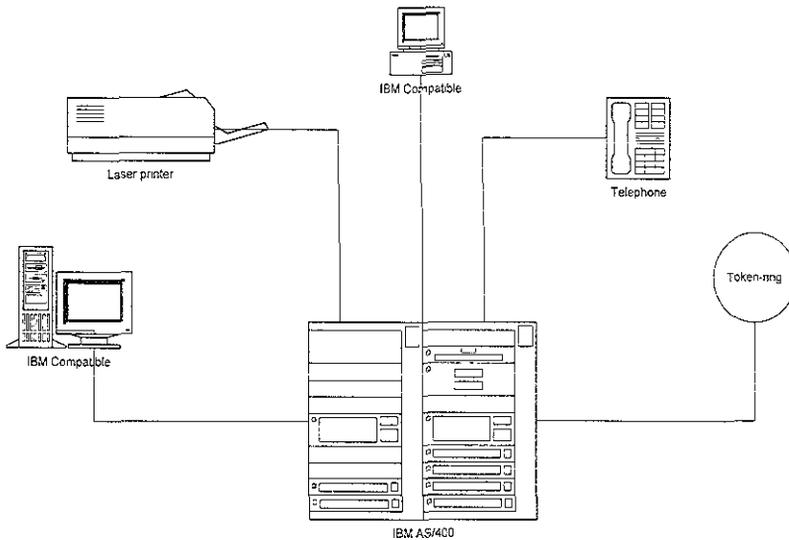


FIG. 1.2 Conectividad del sistema AS/400

I.3 ARQUITECTURA DEL SISTEMA AS/400

El AS/400 está definido por 5 principios arquitectónicos fundamentales:

- Independencia tecnológica.
- Un diseño con base de objetos.
- Integración del hardware.
- Integración del software.
- Almacenamiento de un solo nivel.

Estos 5 principios representan lo que los clientes valoran del AS/400 como computadora de negocios.

Independencia tecnológica.

El AS/400 no está definido por el hardware, esto es, los programas no hablan directamente al hardware sino a una interfase tecnológicamente independiente. Entre esta interfase y el software real existen varios millones de líneas de software del sistema operativo llamado System Licensed Internal Code (Código Interno Autorizado del Sistema - SLIC). Esto aísla los programas de aplicación de las características relevantes del hardware. De esta manera cuando el procesador cambia es suficiente con sobrescribir los componentes SLIC para preservar la integridad de la interfase.

Debido a este diseño de tecnología los programas son inconscientes de los cambios tecnológicos. Estos programas pueden ser migrados a otra máquina con diferente arquitectura sin tener que ser modificados o compilados.

Diseño con base de objetos.

En el AS/400 todo dentro del sistema (programas, archivos, perfiles de usuario), son un objeto. Los objetos constan de dos partes una descriptiva y una parte de información. La parte descriptiva define las formas válidas de uso del objeto. Si un objeto es definido como programa, la parte descriptiva establecerá que la parte de información será tratada como ejecutable, y las únicas operaciones permitidas serán aquellas que tienen sentido para un programa. Si un objeto ingresa al AS/400 retendrá por siempre sus características; es decir, no puede ingresar como información y después convertirse en código ejecutable. Esta es una de las razones por las cuales el AS/400 goza de una reputación de seguridad e integridad debido a que, en otras máquinas, un mecanismo por el cual los virus entran a los sistemas es entrando como información y después el virus trata de convertirse en ejecutable.

Integración de Hardware.

En el ambiente Ingeniero - Científico de la computación se realizan operaciones complejas con una relativamente pequeña cantidad de información, pero en el ambiente general de negocios se realizan operaciones simples con una gran cantidad de información. Esto es tomado en cuenta para el diseño del AS/400 optimizándolo para el ambiente general de negocios con características de hardware que le permiten un gran desempeño con grandes cantidades de información.

Esto se logra teniendo procesadores de entrada/salida (IOP) dedicados al dispositivo de disco. Cuando el procesador principal está ejecutando un programa y llega a una petición de información, por ejemplo una lectura del disco, delega esta tarea al procesador de entrada/salida, entonces el procesador puede desviar su atención a otra aplicación regresando a la original solo cuando la petición de información anterior esté disponible en el almacén principal.

Integración del Software.

En todos los negocios se necesitan algunas características de software comunes. Adicionalmente a los drivers del sistema operativo que manejan los dispositivos de entrada y salida, siempre se requiere en los negocios software para funciones estándar tales como las comunicaciones, seguridad, respaldo y recuperación. En el sistema AS/400 todo este software de negocios está integrado en el sistema operativo, y cuando hay algún cambio en alguno de estos componentes se les entrega a todos los clientes un sistema operativo nuevo, completo y totalmente probado, evitando así, posibles conflictos de liberaciones entre los componentes individuales del AS/400.

En los demás sistemas los clientes deben comprar componentes adicionales para satisfacer sus necesidades, y deben asegurarse que estos componentes sean compatibles entre sí, pudiendo tener problemas además con las versiones de los mismos.

De esto sobresalen inmediatamente dos beneficios para el cliente: un costo bajo de adquisición y un rápido despliegue de nuevas soluciones de negocios.

Almacenamiento de un solo Nivel.

El espacio de almacenamiento masivo del AS/400 es un almacén real. Pero el AS/400 al administrar sus tecnologías de almacenamiento automáticamente hace innecesario que los clientes tengan que conocerlas o estar conscientes de ellas. Todos los programas y la información residen en este espacio masivo. Los usuarios no deben preocuparse por dónde reside un programa, ellos solo tienen que referirse a él por su nombre. Tampoco es necesario preocuparse por hacer extensiones a archivos llenos, el AS/400 maneja esto de manera automática también. Al agregarse otro dispositivo de almacenamiento al AS/400 el sistema reconoce el nuevo almacenamiento disponible y lo utiliza, lo que en otros casos es necesario hacer por medio de una redistribución de información.

Los procesos de aplicaciones de negocios en un ambiente multi-aplicaciones y multi-servidores con frecuencia implican la conexión entre tareas diferentes. En el AS/400 conectar una nueva tarea es tan fácil como ejecutar una instrucción en la localidad donde ésta reside a diferencia de otros sistemas donde es necesario recrear un espacio de direccionamiento separado antes de que la ejecución de una nueva tarea pueda comenzar. Este almacenamiento de un solo nivel no solo simplifica el almacenamiento de información y la conexión frecuente de tareas que caracteriza al ambiente de negocios, sino que también brinda un excelente desempeño.

I.4 SISTEMA OPERATIVO

El sistema operativo proporcionado por IBM para el AS/400 es el OS/400. En el OS/400 se controla como fue visto en la arquitectura del AS/400 la independencia tecnológica mediante el System Licensed Internal Code, el diseño con base a objetos, es decir, que trato se le debe dar a cada objeto y sus formas validas de uso. También se integran todas las funciones estándar como comunicaciones, seguridad, respaldo y recuperación, y el almacenamiento de un solo nivel.

Este sistema operativo contiene muchos objetos y comandos para soportar las comunicaciones entre dos máquinas, tales como transferencia de archivos, accesos remotos a archivos, llamados a comandos remotos, etc. De esta manera puede ser posible cuando así se requiera por cuestiones de volumen de información, manejar los sistemas de una compañía en dos o más máquinas pudiendo compartir datos entre ellas para la integración de los sistemas.

Los objetos de comunicación del AS/400 consisten en servidores de red, interfases de red, líneas de comunicación, controladores, y dispositivos. El hardware remoto necesario para establecer una comunicación consiste en cualquier controlador remoto y un dispositivo conectado al sistema mediante una línea de comunicación. El servidor de red es usado cuando se desea conectar al AS/400 con una red a través de la cual se comparten datos, usuarios, etc. Del mismo modo, la interfase de red es solo necesaria para la comunicación entre el AS/400 y un servidor de red. Una línea de comunicación es el enlace físico entre un sistema y otro controlador o red. Un controlador de comunicaciones

es un dispositivo que está conectado al sistema a través de una línea de comunicación. El controlador permite a otro dispositivo remoto comunicarse con el sistema local. Un dispositivo puede ser, por ejemplo, asíncrono que está conectado a un controlador de comunicaciones que existe en el otro lado de la línea, éste puede ser otro AS/400.

El AS/400 tiene un sistema propio de autoridades, las cuales se pueden dar a usuarios en particular, a usuarios que forman un grupo, al público integrado por todos los usuarios. Estas autoridades tienen diferentes niveles, por ejemplo para un objeto se puede tener autoridad para verlo, usarlo o modificarlo. Este sistema de autoridades restringe el acceso a las bases de datos a los usuarios dejando el uso o modificación de éstas solo a los usuarios deseados.

El sistema operativo OS/400 para el manejo de la información y procesos utiliza librerías para la agrupación de diferentes objetos como:

1.4.1 Archivo Físico

Es un archivo de base de datos que contiene registros de datos. Los registros de datos son agrupados en miembros del archivo físico y cada uno de los miembros tiene su propia ruta de acceso a la información. Normalmente los archivos físicos tienen solo un miembro, el cual, es adicionado al archivo por default cuando éste es creado. Para la creación de un archivo físico se usa un archivo fuente, el archivo fuente contiene la especificación del formato de registro del archivo físico, sus campos y la ruta de acceso para el archivo y sus miembros. El formato de registro es el nombre que se le da a cada uno de los arreglos de campos que tiene el archivo físico.

I.4.2 Archivo Lógico

Es una “vista” del archivo de base de datos el cual describe como será ordenada la información contenida en uno o más archivos físicos para su lectura en un programa. El archivo lógico no contiene registros de datos. Los registros de datos están contenidos en los archivos físicos asociados con el archivo lógico. Los registros de datos contenidos en un archivo físico están agrupados en miembros de archivo físico. El archivo lógico accesa los registros a través de uno o más miembros de archivo lógico. Cada miembro de archivo lógico describe los datos contenidos en uno o más miembros de archivo físico y cada miembro de archivo lógico tiene su propia ruta de acceso a los datos. Para la creación del archivo lógico es necesario un archivo fuente en el cual se especifica el formato de registro, la ruta de acceso para el archivo y sus miembros; y el nombre del archivo físico en el cual se basará.

I.4.3 Pantallas (Display File)

Es un archivo donde se encuentra la forma de la presentación de una pantalla. Esta pantalla puede contener campos de entrada o salida para la presentación o introducción de datos al sistema, información adicional como fecha hora etc. Para la creación de estas pantallas también es necesario un archivo fuente con el cual crear la pantalla. Es posible también utilizar en los display files los SUBFILES, los cuales son una especie de archivo residente solo en la memoria de la máquina, los cuales se pueden grabar, leer, actualizar y presentar en una pantalla con el formato que se le haya dado.

I.4.4 Programas RPG

Los programas RPG son usados para la interacción con la base de datos. Los programas en este lenguaje son los más usados en el AS/400. Es en estos programas donde la mayoría de las veces se lee, consulta, graba o actualiza información en la base de datos debido a que es donde se tiene más opciones de manipulación y procesamiento de los mismos. Son también estos programas los que nos permiten el manejo de las pantallas o Display Files así como de los Printer Files o reportes de una manera más óptima y completa

I.4.5 Programas CL

Los programas CL son usados para el control de los procesos indicando los pasos que deben seguir los mismos, están codificados en lenguaje de control propio del sistema el cual es muy flexible, con una gran variedad de comandos que pueden ser utilizados de manera que proporcionen ciertas facilidades, tales como: manejar pantallas, hacer llamados a programas RPG, decisiones lógicas en el programa, recuperación de atributos de trabajo y usuario, envío de mensajes, comandos de comunicación, monitoreo de mensajes de error para tomar una acción, creación de objetos, cambio de los atributos de los objetos, sometimiento de trabajos, etc.

I.4.6 Perfiles de usuario.

Para acceder al sistema es necesario contar con un User Profile, en este objeto se agrupan las diferentes características con las que va a trabajar el usuario. El user profile identifica a un usuario para el sistema y permite personalizar la manera como aparece el sistema.

Este objeto consta de las siguientes características y atributos principales:

- Una clave de usuario: es el nombre con el que el sistema identifica al usuario.
- Un password o código de seguridad: que le permitirá al usuario abrir una sesión en el sistema, este sólo debe ser conocido por el usuario.
- Autoridad especial: especifica la autoridad especial otorgada al usuario. Para ciertas funciones en el sistema es necesario tener autoridades especiales.
- La más alta prioridad que puede tener el usuario en el sistema: el sistema maneja prioridades las cuales le indican a quién debe dedicar más tiempo de procesamiento.
- Grupo: el usuario puede ser parte de un grupo de usuarios, mediante los cuales se pueden dar autoridades a objetos a todos los miembros del grupo sin necesidad de indicar a cada uno.

I.4.7 Áreas de Datos.

Un área de datos es una cadena de caracteres o números de longitud variable que está almacenada en la el sistema en una librería específica. Cada vez que se cambia esta área de datos es actualizada en la memoria auxiliar y con esto se asegura que los cambios hechos a ésta no se perderán en caso de una falla en el programa o en el sistema.

Si el área de datos es usada solo por un usuario, ésta puede estar explícitamente asignada al trabajo apropiado. Si el área de datos es usada por dos o más usuarios a la vez, ésta es protegida contra cambios simultáneos que ocurren de diferente trabajo. Un área de datos

se modifica mediante el comando CHGDTAARA. El sistema no permite que dos comandos cambien la misma área de datos al mismo tiempo.

1.4.8 UTILERÍAS

Para el desarrollo de archivos físicos, lógicos, pantallas, reportes y programas en el sistema AS/400 se cuenta con diferentes herramientas que facilitan en especial el diseño y creación de pantallas y reportes. A continuación se presenta una breve explicación de la forma de entrar a estas herramientas del sistema. En la figura 1.3 se presenta el menú principal del AS/400.

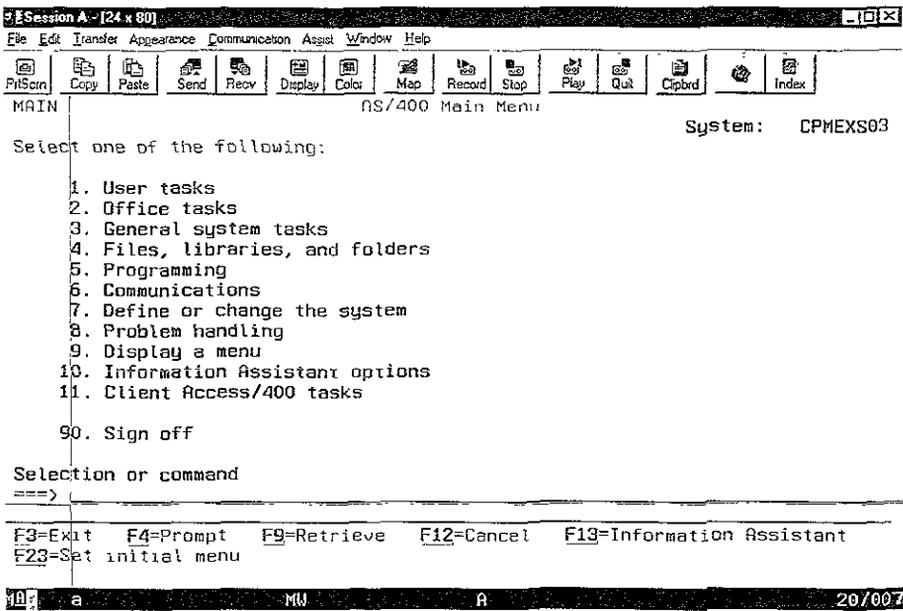


FIGURA 1.3 Menú principal del Sistema AS/400

En la opción 5 del menú anterior se encuentra el menú de programación, este es el menú donde se encuentran las opciones utilizadas para el desarrollo tanto de archivos, reportes pantallas y programas en el sistema AS/400.

En la figura 1.4. se muestra el menú de programación.

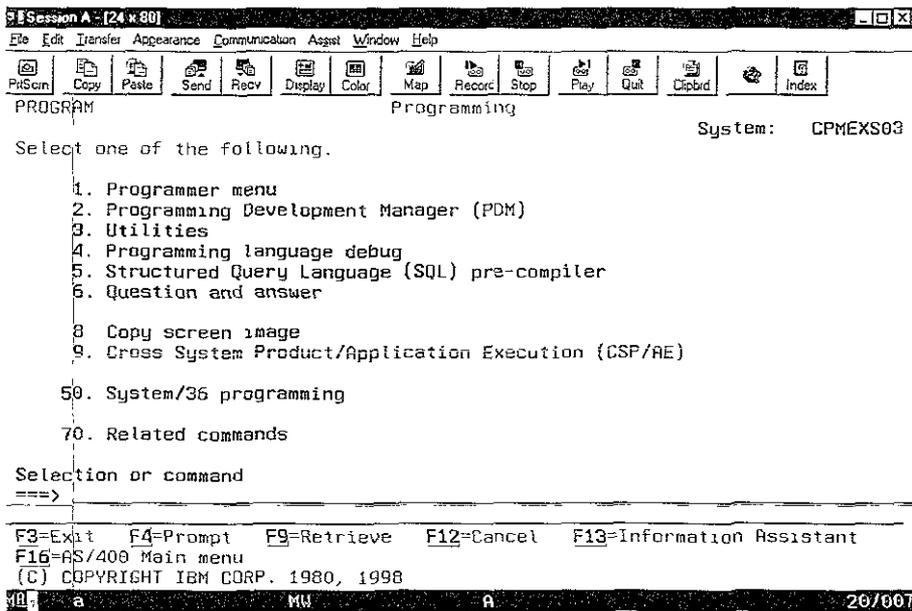


FIGURA 1.4 Menú de Programación

En este menú se encuentra la opción 2 que es el administrador de desarrollo de programas (PDM). Este administrador es el que nos organiza, presenta y facilita el manejo, desarrollo y creación de archivos y programas.

Para la creación de archivos, pantallas o programas es necesario primero codificarlo en un fuente, para después con este crear el objeto deseado mediante la compilación de su fuente. Estos archivos fuente son agrupados en miembros de archivos los cuales a su vez son agrupados en librerías. El PDM nos ayuda a presentar editar y compilar estos miembros de archivos mediante editores y herramientas. La figura 1.5 muestra el menú del PDM.

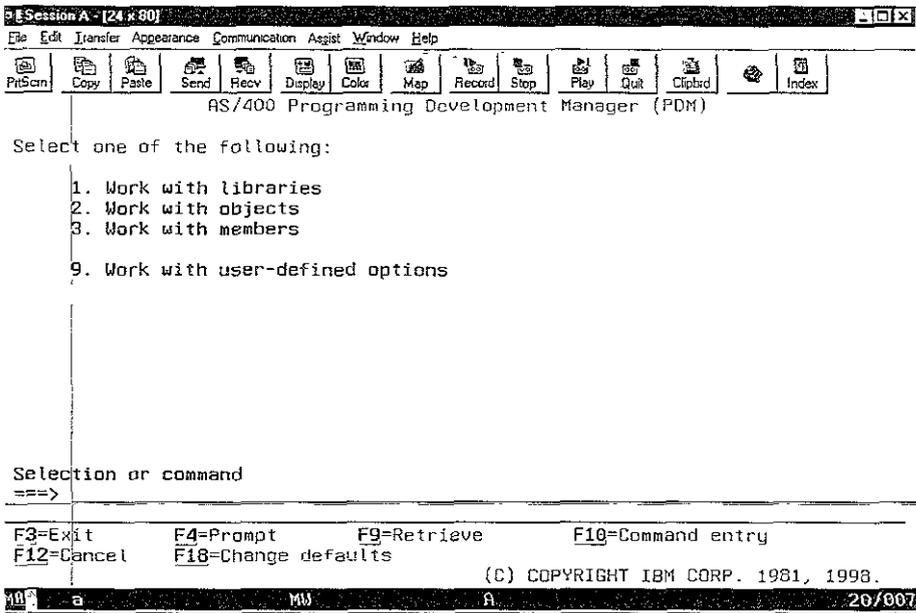


FIGURA 1.5 Menú del Administrador de Desarrollo de Programas

Este menú cuenta con tres opciones, las cuales, permiten trabajar con librerías, objetos y miembros.

La opción uno es para trabajar con librerías, se puede seleccionar una o varias en forma genérica, las cuales serán desplegadas en una pantalla dando opción de borrarlas, crearlas, modificar su texto descriptivo o trabajar con todos los objetos que contiene una librería.

Con la segunda opción, se permite trabajar con objetos de una librería especificada, los cuales se despliegan en una pantalla que proporciona las opciones para borrar, visualizar datos del objeto, cambiar su texto descriptivo, renombrar o mover los objetos a otra librería.

La opción tres permite seleccionar un archivo y una librería que contenga este archivo para que sean presentados todos sus miembros o fuentes; estos archivos pueden tener cualquier nombre y pueden contener fuentes de cualquier tipo, pero lo más común es que se dividan por tipos. El archivo que contiene fuentes de programas en RPG suele denominarse como QRPGSRC, el que contiene programas de Lenguaje de Control (CL) QCLSRC etc. La figura 1.6 muestra la presentación que hace el PDM de un archivo QRPGSRC en la cual se tienen diferentes opciones para editar, copiar, renombrar, borrar, compilar, visualizar, etc. Además cuenta con diferentes funciones para la creación de nuevos fuentes, salir, tener diferentes vistas, etc.

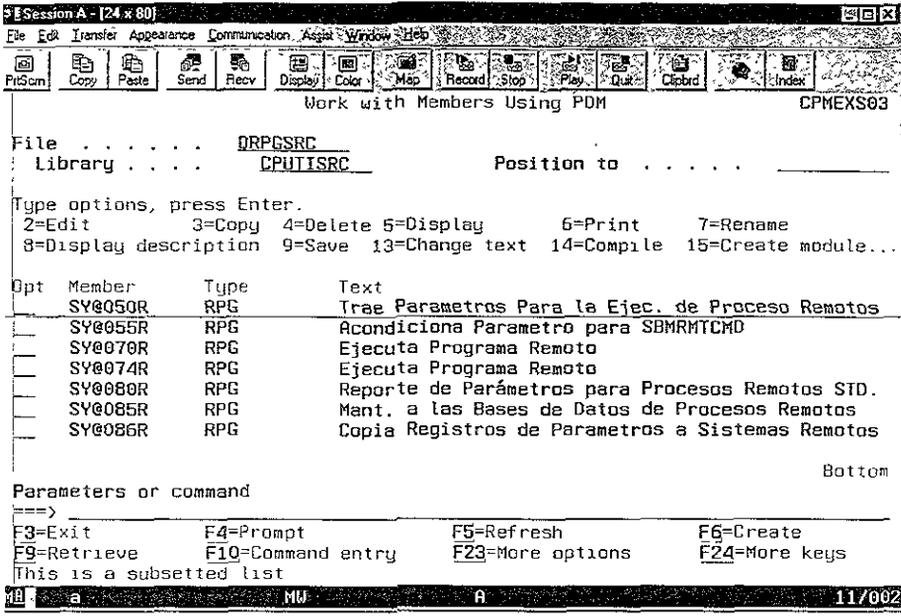


FIGURA 1.6 Presentación de Fuentes en PDM

En el caso de programas en RPG el lenguaje es posicional es decir las instrucciones, variables, archivos, indicadores, etc., tienen una posición dentro del archivo fuente del programa a diferencia de los programas CL en los cuales no es necesario respetar ninguna posición.

La utilidad para el desarrollo de los programas es un editor que permite borrar, copiar insertar y modificar cadenas de caracteres o líneas completas. La figura 1.7 muestra este editor.

```

Session A - [24 x 80]
File Edit Transfer Appearance Communication Assistance Window Help
Print Screen Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipboard Index

Columns . . . . : 1 71 Browse CPUTISRC/QRPGSRC
SEU==> SY@085R
FMT * ..... * 1 ..... 2 ..... 3 ..... 4 ..... 5 ..... 6 ..... 7
0102.00 * Despliega Pantalla No. 2 de Datos Generales. *
0103.00 *-----*
0104.00 C SR0012 BEGSR
0105.00 C *IN03 DOWEQ '0'
0106.00 C ASTER ANDEQ '*'
0107.00 C WRITESY@085M2
0108.00 C EXFMTSY@085P2
0109.00 C CLEARPMSG
0110.00 C MOVEA '00000' *IN,51
0111.00 *
0112.00 C *IN03 IFEQ '1'
0113.00 C *IN12 DREQ '1'
0114.00 C LEAVE
0115.00 C ENDIF
0116.00 *
0117.00 C SHTIPD IFNE '*LCL'
0118.00 C SHTIPD ANDNE '*RMT'

F3=Exit F5=Refresh F9=Retrieve F10=Cursor F11=Toggle F12=Cancel
F16=Repeat find F24=More keys

```

FIGURA 1.7 Utilería para el desarrollo de programas

En el caso de las pantallas y archivos de impresión el PDM cuenta con herramientas especiales para su diseño donde se puede ver la posición de un campo dentro de la pantalla o el reporte; y moverlo o copiarlo a otro lado en la pantalla, modificarlo, borrarlo, adicionar etc. y su código es generado automáticamente.

La herramienta para el diseño de pantallas se llama SDA a la cual se puede entrar una vez estando en la presentación del fuente de la pantalla deseada con la opción 17 (cambiar usando SDA de la figura 1.8) la cual presenta otra pantalla donde se pueden ver los registros contenidos en este fuente.

Cada registro contenido en el fuente es en sí una pantalla; esto es para facilitar la programación cuando es necesario en un mantenimiento, el uso de más de una pantalla al no tener que utilizar varios archivos en el mismo programa.

Al entrar a uno de estos registros con la opción 12 de diseñar imagen, se entra a otra pantalla donde se presenta el código fuente del display file con los campos en las posiciones en las que se presentarán en la pantalla cuando el objeto sea creado y utilizado. Al adicionar un campo a esta pantalla la utilería lo adiciona automáticamente al archivo fuente de esta, del mismo modo al borrar o modificar un campo se modifica o borra este campo del archivo fuente de la pantalla. Esta utilería contiene ayudas en las que se puede consultar la manera de copiar, borrar, mover, etc.,

También es posible definir atributos a los campos, tales como códigos de edición, color, alta intensidad, si son campos de entrada, salida; rangos de valores válidos, mensajes de error, posicionamiento del cursor, así como funciones válidas en la pantalla.

Es también en estas dos pantallas donde se definen los subfiles, los cuales, presentan un arreglo de “n” campos horizontales repetidos “n” veces verticalmente. El subfile siempre debe ir asociado a un archivo de control en el cual se define su tamaño, la forma como se puede borrar y grabar en él la información.

Session A - [24 x 80] CPMEXS03

File Edit Transfer Appearance Communication Assist Window Help

PrintScreen Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipboard Index

Work with Members Using PDM

File QDPSRC
 Library CPUTSRC Position to

Type options, press Enter.
 16=Run procedure 17=Change using SDR 19=Change using RLU
 25=Find string 54=Compare 55=Merge . . .

| Opt | Member | Type | Text |
|-----|---------|------|--|
| 17 | SY@080P | DSPF | Reporte de Parametros para Procesos Remotos STD. |
| | SY@085P | DSPF | Mant. a las Bases de Datos de Procesos Remotos |

Parameters or command
 ==>

| | | | |
|-------------|-------------------|------------------|---------------|
| F3=Exit | F4=Prompt | F5=Refresh | F6=Create |
| F9=Retrieve | F10=Command entry | F23=More options | F24=More keys |

Bottom

a MU A 12/002

Session A - [24 x 80]

File Edit Transfer Appearance Communication Assist Window Help

PrintScreen Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipboard Index

Work with Display Records

File : QDPSRC Member : SY@085P
 Library : QTEMP Source type : DSPF

Type options, press Enter.
 1=Add 2=Edit comments 3=Copy 4=Delete
 7=Rename 8=Select keywords 12=Design image

| Opt | Order | Record | Type | Related Subfile | Date | DDS Error |
|-----|-------|----------|--------|-----------------|----------|-----------|
| | 10 | SY@085HD | RECORD | | 08/05/97 | |
| | 20 | SY@085S1 | SFL | | 08/05/97 | |
| | 30 | SY@085C1 | SFLCTL | SY@085S1 | 23/01/98 | |
| | 40 | SY@085P1 | RECORD | | 08/05/97 | |
| | 50 | SY@085P2 | RECORD | | 08/05/97 | |
| | 60 | SY@085S2 | SFL | | 08/05/97 | |
| | 70 | SY@085C2 | SFLCTL | SY@085S2 | 08/05/97 | |
| | 80 | SY@085S3 | SFL | | 08/05/97 | |
| | 90 | SY@085C3 | SFLCTL | SY@085S3 | 08/05/97 | |

F3=Exit F12=Cancel F14=File-level keywords
 F15=File-level comments F17=Subset F24=More keys

(C) COPYRIGHT IBM CORP. 1981, 1998.

a MU A 11/002

FIGURA 1.8 Utilería para el desarrollo de pantallas

1.5 NECESIDAD DE COMPARTIR INFORMACIÓN CON UN SISTEMA REMOTO

Por el mismo volumen de información que puede ser manejado en las compañías se requiere a veces la utilización de más de un equipo AS/400, no pudiendo aislar cada equipo del otro, por razones obvias de compartir información, se vuelve una necesidad la comunicación entre programas de los sistemas locales y programas de los sistemas remotos. Estas necesidades se pueden dar, por ejemplo, al tener separados en una compañía los sistemas de manufactura y ventas al cliente en diferente equipo, si se quiere calcular la producción de acuerdo a la estadística de ventas, es necesario entonces acceder del equipo del sistema de manufactura al equipo del sistema de ventas para poder extraer las ventas y así calcular la producción. En muchas aplicaciones es necesario compartir volúmenes grandes de información, lo cual complica aún más el problema ya que se requiere mayor tiempo para procesar todos los datos y esto se ve reflejado en el desempeño del equipo en general. Para lograr la comunicación entre equipos es necesario hacer uso tanto del directorio de distribución como de las colas de distribución (distribution queues).

1.5.1 Directorio de Distribución.

Para que un usuario pueda hacer uso de la distribución y de ciertos comandos remotos es necesario que esté dado de alta en el directorio de distribución. El directorio de distribución del sistema contiene información del usuario como su clave de usuario, su dirección, el nombre del sistema, teléfono y otros datos sobre los usuarios que tienen autorización para mandar y recibir distribución en la red. Es necesario que cada usuario

tenga una clave de red y dirección, el cual, debe ser único en el diccionario de distribución. Esto es para que pueda ser identificado de una forma única en la red. Esta debe ser de 8 caracteres considerando que los caracteres usados puedan ser tecleados por todos los usuarios (por usuarios de sistemas con lenguajes diferentes), si se define con minúsculas el sistema lo convierte a mayúsculas y el primer caracter no puede ser blanco. También es necesario saber el user profile al cual pertenece esta clave de red, y por lo cual, también éste se encuentra en el directorio. El user profile no se especifica cuando se trata de un usuario remoto, ya que en este caso no existe en el sistema local, pero si es local el profile debe existir en el sistema local y tiene que ser especificado. Si el profile se especifica en un usuario remoto éste debe ser válido. Se debe especificar además el nombre del sistema el cual puede constar de dos partes, se compone tanto del nombre del sistema como del nombre del grupo del sistema. Un usuario remoto puede darse de alta en el directorio de distribución antes de que el nombre y grupo del sistema sean definidos en las tablas de red, pero no se puede tener acceso a la distribución con ese usuario hasta que se definan el nombre y grupo del sistema.

1.5.2 Colas de Distribución

Las Distribution Queues o colas de distribución son usadas para enviar la distribución a otros sistemas, es en ellas donde se almacena la información antes de ser enviada. Asocia un sistema remoto con el sistema local y contiene información concerniente a cada uno, el sistema remoto es a donde se enviará la distribución fuera del sistema local. El nombre de un sistema remoto para una nueva cola de distribución debe ser único porque solo una cola de distribución puede ser definida para cada sistema remoto. El nombre del sistema local es el que utilizarán los sistemas remotos para identificarlo en la red.

I.6 OPCIONES PARA COMPARTIR INFORMACIÓN EN EL AS/400.

Existen varias opciones para compartir información en el sistema AS/400, tales como:

- DDM File.
- FTP (File Transfer Protocol).
- Data Mirror.
- Comandos.

I.6.1 Administrador de Distribución de Datos (DDM)

La necesidad de compartir información puede tener una solución al hacer uso del DDM (Distributed Data Management). El DDM es parte del sistema operativo OS/400. El DDM que soporta el sistema AS/400 permite a usuarios o programas de aplicación acceder archivos de datos que residen en un sistema remoto y también permiten a sistemas remotos acceder archivos de datos en el sistema AS/400 local. Estos sistemas remotos pueden ser otros diferentes al AS/400 que sean compatibles con la arquitectura del DDM, como se muestra en la figura 1.11

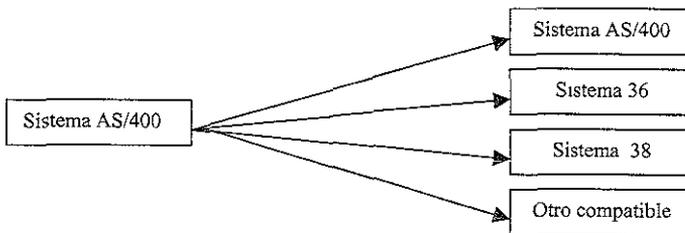


FIG. 1.11 Sistemas Remotos Compatibles

El DDM controla el procesamiento de archivos remotos, permitiendo a los programas de aplicación que están corriendo en un AS/400 acceder a archivos de datos ubicados en otro

sistema que soporta el DDM. Similarmente otro sistema que tiene DDM puede acceder archivos de datos del sistema local AS/400. El DDM hace más fácil la distribución del procesamiento de archivos entre dos o más sistemas. (Ver fig. 1.12)

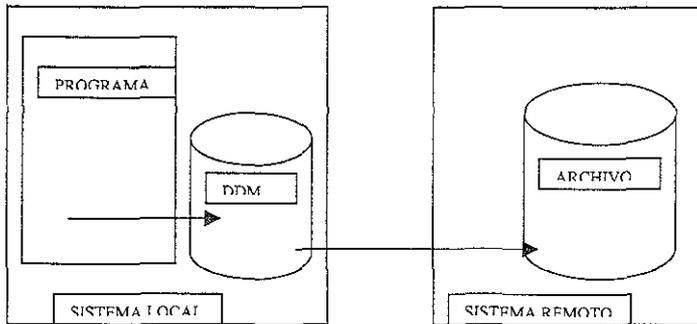


FIG. 1.12 Uso de un DDM para administrar el intercambio de información

Cuando se utiliza un DDM para el intercambio de información se procede de la siguiente manera:

El sistema local abre el archivo DDM y manda un requerimiento al sistema remoto para abrir el archivo remoto, el archivo es abierto en el sistema remoto y se manda una notificación de terminación al sistema local, éste pide acceso a un registro para cambiarlo posteriormente, el sistema remoto lo accesa y lo bloquea para que no pueda ser modificado por algún otro programa, y lo envía al sistema local, éste lo modifica y manda el requerimiento para la modificación en el sistema remoto, en el sistema remoto se cambia el registro y se desbloquea, una vez terminada la operación el sistema local

manda el requerimiento para cerrar el archivo, el sistema remoto lo cierra y manda la notificación de terminación. (Ver fig. 1.13)

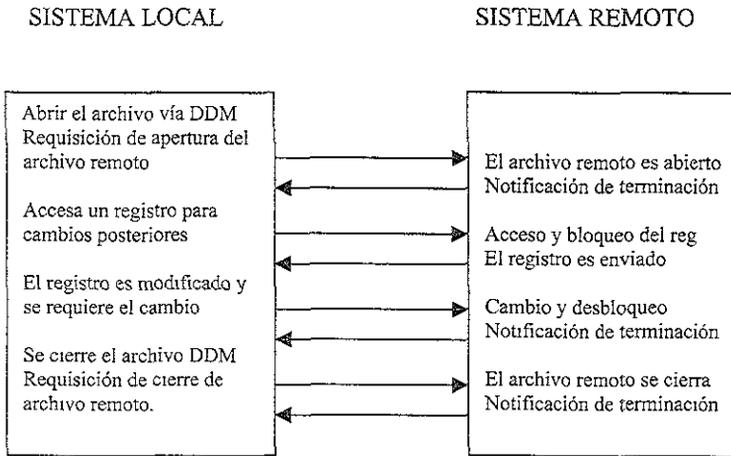


FIG. 1.13 Intercambio de información entre un sistema local y un sistema remoto

Cuando se usa un DDM, ni la aplicación ni el programa necesitan saber si el archivo es local o es remoto. El DDM maneja el acceso al archivo remoto casi de la misma forma como un acceso a un archivo local es manejado en el sistema local, y el programa de aplicación normalmente no recibe ningún indicador de donde se localiza el archivo.

Los programas de RPG/400 y los programas de Lenguaje de Control (CL) pueden usar los archivos DDM. Generalmente los nombres de estos archivos pueden ser especificados en cualquier lugar donde un nombre de base de datos pueda estar. Cuando un programa de RPG/400 abre un archivo DDM en el sistema local, los siguientes tipos de operaciones de entrada y salida pueden ser ejecutadas en el archivo del sistema remoto: CHAIN,

CLOSE, DELET, EXCPT, OPEN, READ, READE, READP, SETGT, SETLL, UPDAT y WRITE. Estas operaciones son usadas para realizar lecturas, actualizaciones, adición o borrado de registros.

Los programas de lenguaje de control pueden usar archivos DDM en muchos de los comandos relacionados con archivos de base de datos como el CPYF, DSPFD, OVRDBF, etc.

Antes de que los archivos puedan ser accedidos por un sistema AS/400 los archivos DDM deben ser creados en el sistema local especificándole un nombre, la librería donde será creado, el nombre y librería del archivo remoto accedido, del cual está haciendo uso el DDM cada vez que se utiliza; y el nombre del sistema remoto donde reside.

Para un usuario final el acceder un archivo local es muy parecido a acceder un archivo remoto. Sin embargo es requerido un tiempo adicional cuando se accesan archivos remotos a través de una línea de comunicaciones. La diferencia en el proceso entre los archivos locales y los remotos es proporcional al número de accesos al archivo remoto, la longitud del registro de datos y la velocidad de transmisión.

Una diferencia adicional entre el procesamiento de archivos locales y remotos es que las operaciones de entrada o salida a un archivo local pueden no resultar inmediatamente en una operación física en el disco porque el sistema transfiere bloques de datos desde el disco y escribe bloques de datos al disco. Por ello es esencial el conocimiento del diseño

de una aplicación y la cantidad y tipo de accesos a archivos para considerarlos cuando se determine cuales van a ser accedidos remotamente a través de un DDM.

El tiempo adicional para cada acceso remoto está compuesto de:

- Procesamiento adicional de sistema para convertir las interfaces de archivo del sistema local a interfaces DDM.
- Cantidad de datos transmitidos a través de la línea de comunicación.
- Cantidad de procesamiento en el sistema remoto de las operaciones en el archivo.
- La velocidad en la línea de comunicación.

El tiempo de la línea de comunicación representa la mayor cantidad del tiempo adicional así pues, el tiempo real depende de la velocidad y cantidad de actividad en la línea durante la función del DDM.

El estar accedando un archivo remoto por un DDM no es siempre lo más conveniente ya que al estar utilizando vías de comunicación y, dependiendo de la distancia de separación entre los dos sistemas, el tiempo de respuesta se puede ver afectado. Esto puede representar una gran desventaja cuando hablamos de un nivel alto de datos que se tienen que utilizar.

Otra desventaja al utilizar un archivo DDM es cuando una misma información es requerida por más de un proceso, y al acceder por DDM es necesario hacer las mismas lecturas varias veces y al tomar en cuenta que estamos utilizando un archivo remoto es

claro que esto se verá reflejado en el tiempo-máquina requerido para realizar todos los procesos, comparándolo con el tiempo requerido si el archivo residiera en el mismo equipo.

El archivo DDM es también usado cuando un comando de lenguaje de control (CL) es sometido al sistema remoto. El comando SBMRMTCMD (Submit Remote Command) somete un comando CL a través de un DDM para que corra en el sistema remoto. La información de la localización remota en el archivo DDM es usada para determinar la línea de comunicación a usar; y así indirectamente, identifica el sistema remoto que recibirá el comando. El archivo remoto, normalmente asociado con el archivo DDM, se ignora cuando éste es usado para someter comandos en el sistema remoto.

El principal propósito de este comando es permitir a un usuario o programa del sistema local ejecutar operaciones de acceso a los archivos localizados en el sistema remoto. El usuario debe tener la autoridad necesaria para ejecutar el comando que va a ser sometido y para los objetos del sistema remoto sobre los cuales éste va a operar.

1.6.2 Protocolo de Transferencia de Archivos (FTP).

Existe otra opción de transferencia de información entre máquinas AS/400 denominada Protocolo de Transferencia de Archivos (File Transfer Protocol), en la cual se envían archivos completos al sistema remoto.

Con este protocolo el envío de archivos se hace en unas cuantas operaciones simples. En el FTP se requiere conectarse a la máquina remota para lo cual, se necesita un usuario y

password válidos. Una vez establecida la conexión, el envío se realiza con una sola instrucción; la cual debe seguir la sintaxis mostrada en la figura 1.14

PUT LIBORG/ARORG LIBDES / ARDES

| | | |
|--------------------|---------------------------|---------------------------|
| Instrucción | Librería / Archivo | Librería / Archivo |
| | Origen Origen | Destino Destino |

FIG. 1.14 Sintaxis de la instrucción de envío por FTP

Para enviar un archivo a la máquina remota se utiliza la instrucción PUT y para traer un archivo desde la máquina remota se utiliza la instrucción GET. La LIBRERÍA y el ARCHIVO ORIGEN nos indican el archivo que deseamos transmitir; asimismo la LIBRERÍA y el ARCHIVO DESTINO indican dónde se recibirá.

El envío de archivos por FTP tiene la ventaja de permitir el envío a otro tipo de máquinas como por ejemplo UNIX, pudiendo controlar este envío desde un programa CL en el AS/400, esto es, se pueden tener procesos que envíen archivos a otro tipo de máquinas en forma automática.

Una desventaja de la transferencia por FTP se presenta cuando se envían archivos que contienen campos numéricos y empacados. En estos casos la transmisión no es completa debido a que los números empacados se reemplazan en el FTP por caracteres diferentes con lo cual se pierde información. Los campos empacados son comunes en el AS/400 y son usados para un mejor aprovechamiento de las áreas de almacenamiento de información.

Debido a que muchos archivos tienen campos numéricos y empacados, el FTP no puede brindar una solución genérica para la transferencia de información.

1.6.3 Espejo de Datos.

El principio del funcionamiento del Espejo de Datos (Data Mirror) es el de mantener la misma información disponible en ambas máquinas, usando para esto un juego de archivos, uno en la máquina local y otro en la remota y mantener los dos archivos con la misma información. Esto se logra programando un trabajo en las máquinas para que cada cierto intervalo de tiempo se copie la información del archivo local al remoto.

Con esta opción se tiene la ventaja de tener la información disponible en ambas máquinas en todo momento, sin tener que hacer procesos especiales en el momento que se requiera esta. Sin embargo, tiene algunos inconvenientes como por ejemplo, el desperdicio de espacio en disco al tener la información duplicada lo cual con niveles grandes de información es más notorio. Además se tiene otra desventaja, al tener un trabajo replicando la información cada cierto intervalo de tiempo, se eleva el consumo de los recursos de CPU. Si el intervalo de tiempo con el cual se replica la información entre las máquinas se hiciera mayor para evitar un elevado consumo de recursos de CPU, se tiene el riesgo de no tener los mismos datos en ambas máquinas al momento de necesitarlos.

I.6.4 Transferencia con comandos del AS/400.

Existe una gran cantidad de comandos que ayudan a la comunicación en este sistema.

Algunos ejemplos de ellos son:

SENDNETF

El comando de enviar un archivo de red (Send Net File), envía un archivo de salvado o un miembro de un archivo físico de base de datos a un usuario en el sistema local o a un sistema remoto. Este comando puede ser usado para:

- Enviar archivos de datos a un usuario.
- Enviar archivos fuentes a un usuario
- Enviar otros tipos de objetos guardados en un archivo de salvado.

Un archivo de salvado es un objeto en el cual se pueden agrupar varios objetos de diferentes tipos, este nos da la ventaja de poder enviar todo lo que necesitemos en un solo paso. Estos objetos necesitan ser salvados previamente antes de enviar el archivo de salvado y al llegar a su destino es necesario recibirlos y después restaurarlos.

Cuando un archivo llega a su destino un mensaje de notificación es enviado a ambos, al que envió y al que recibió el archivo. Para utilizar este comando el usuario debe estar enrolado en el directorio de distribución. El tamaño máximo de un archivo para el comando SNDNETF es de aproximadamente dos billones de bytes.

Al usar este comando se debe especificar el nombre del archivo a enviar y la librería donde se encuentra éste. El archivo que será enviado puede ser un archivo físico o un archivo de salvado, no se pueden enviar archivos lógicos o archivos de pantallas a no ser dentro de un archivo de salvado. También se debe especificar el usuario al cual se le enviará este archivo, así como el nombre del sistema al que será enviado. Se puede especificar el nombre del miembro a enviar para un archivo físico o de fuentes si no se especifica, se enviará el primer miembro del archivo. El nombre de miembro no es permitido para archivos de salvado.

WRKNETF

El comando WRKNETF (trabajar con archivos de red), muestra en pantalla o imprime una lista de los archivos que han llegado para un usuario, o crea un archivo de salida que contiene la lista de los archivos. Cuando la lista es mostrada se puede:

- Recibir el archivo en un archivo del usuario.
- Borrar el archivo.
- Visualizar el miembro del archivo (esto no es válido para archivos de salvado).

Un usuario con autoridad de oficial de seguridad puede visualizar los archivos de red de cualquier usuario, los demás usuarios solo pueden visualizar aquellos enviados a ellos o a su grupo de profile.

RCVNETF

El comando RCVNETF (recibir archivo de red), permite recibir un archivo de red y copiar sus registros en un archivo físico de base de datos. Si el archivo original es un archivo de salvado el archivo debe ser recibido en un archivo de salvado. Antes de que un archivo sea recibido, el archivo especificado en el comando en donde se va a recibir el archivo de red debe existir.

Cuando el archivo es un archivo físico, la longitud del registro del archivo donde se va a recibir debe ser cuando menos del mismo tamaño que el archivo original. Si es más grande la longitud de registro del archivo donde se van a recibir los registros se rellenan con el valor por default para el archivo de recepción.

Un usuario con autoridad de oficial de seguridad puede recibir archivos de red de cualquier usuario. Los otros usuarios solo pueden recibir los archivos de red que fueron enviados a ellos o a su grupo de profile. El usuario debe tener autoridad a la librería que contiene el archivo de recepción y autoridad para usar y adicionar información al archivo.

SBMRMTCMD

EL comando SBMRMTCMD (someter un comando remoto), manda un comando de lenguaje de control a través de un DDM (Distribution Data Management) para que sea procesado en el sistema especificado por el archivo DDM. El DDM determina la línea de comunicación a usarse e indirectamente identifica el sistema remoto que recibe el comando.

Este comando solo envía comandos de lenguaje de control a sistemas remotos que soportan el lenguaje de control. El principal propósito de este comando es el permitir a un usuario de un sistema fuente el realizar operaciones de administración y actividades de autorización en archivos localizados en un sistema remoto.

Las siguientes operaciones pueden ser realizadas usando este comando:

- Crear o borrar archivos físicos, lógicos, pantallas o fuentes.
- Dar o quitar autoridades sobre archivos remotos.
- Checar, renombrar, o mover archivos u otros objetos.
- Salvar o restaurar archivos u otros objetos.

Usando este comando se pueden realizar operaciones no relacionadas con archivos, como someter comandos de escritura del usuario o hacer un llamado remoto a un programa en el sistema remoto.

Los procesos en el sistema remoto operan independientemente de los procesos en el sistema fuente. Las salidas generadas por el comando sometido (como archivos de impresión), no son enviadas de regreso al sistema fuente. Es por esto que los comandos de visualización no deben de ser enviados porque las salidas permanecen en el sistema remoto, así también los comandos que puedan causar mensajes para ser mandados al operador del sistema no deben ser enviados.

El comando a ejecutarse puede ser de hasta 2000 caracteres y representa el comando que será corrido en el sistema remoto. El comando debe de permitirse en ambos ambientes, interactivo y batch en el sistema remoto.

Usando este comando se pueden mandar órdenes desde un sistema fuente a un sistema remoto para iniciar un proceso.

En las aplicaciones el mejor desempeño se logra cuando el archivo se encuentra en el mismo sistema que el programa que esta corriendo. En casos cuando la localización del archivo requiere para un mejor resultado en el tiempo de proceso mover la aplicación al sistema remoto, el uso del comando SBMRMTCMD (someter comando remoto) debe ser considerado. Este trabaja mejor cuando no lo hace de forma interactiva donde cada programa espera a que el programa anterior termine. Por ejemplo, si un programa A utiliza archivos locales, y un programa B utiliza archivos reinotos, el programa A debe correr en el sistema local y se puede utilizar el comando SBMRMTCMD para correr el programa B en el sistema remoto.

En ciertos casos un programa remoto y uno local pueden proporcionar un mejor desempeño que un DDM. Por ejemplo, si 10 registros del sistema remoto van a ser utilizados, cuando es usado el DDM 10 transmisiones son enviadas y 10 transmisiones son recibidas para un total de 20 transmisiones. Un programa de escritura puede ayudar a que el requerimiento de los datos y la transmisión de los mismos pueda ser en 2 en lugar

de 20 transmisiones, una requisición de todos los registros que necesita el sistema local y una respuesta conteniendo estos 10 registros.

Es conveniente cuando se tiene que manejar un volumen grande de información, mandarla toda junta en un archivo al sistema donde se va a procesar. Para lograr esto es necesario seleccionar solo la información que nos interesa para no enviar todo el archivo lo cual, resultaría también contraproducente. La forma de lograrlo es seleccionando la información por medio de un programa, y después enviar estos datos para que puedan ser usados en el sistema fuente. Para poder coordinar un proceso como este es necesario el lograr establecer una buena comunicación entre programas de diferentes equipos para poder hacer la requisición de información, seleccionar ésta en el sistema remoto, mandarla al sistema fuente, recibirla y después procesarla.

Codificar este proceso de selección y envío de información en cada programa que lo requiera necesita de mucho tiempo de programación. Lo ideal es tener un módulo que tome el control de los envíos, recepciones y los llamados a programas remotos. Esto no esta contenido en los procesos de funciones estándar que vienen integrados en el OS/400.

La creación de una aplicación de control de envío, recepción y llamado de programas se explica por las ventajas que puede proporcionar el manejo de programas remotos y envíos de archivos en cuanto a tiempo de procesamiento en una aplicación. Estas aplicaciones no incluyen aquellas en las que se requiere una actualización de la información en línea.

I.7 Hardware Básico.

El AS/400 contiene, tanto en hardware como en su sistema operativo, los elementos necesarios para establecer comunicaciones entre dos máquinas, las cuales pueden conectarse por medio de una línea de transmisión, un modem o una red.

En la parte del hardware se deben tener en cuenta algunas consideraciones para la elección de los componentes necesarios para la conexión, a continuación se presentan algunas de ellas:

- Tipo de línea.
- Interfase de línea requerida
- Velocidad de la línea requerida.
- Tipo de conector de fin de línea requerido.
- Opciones telefónicas.
- Características requeridas de un modem (en caso de usarlo).
- Tipo de conexión.

Para una conexión de 16 o 4 Mbps con Token Ring por ejemplo, es necesario una tarjeta adaptadora #2626 (16/4 Mbps Token-Ring Network Adapter/A), y cable de par torcido con un conector RJ45 de ocho posiciones. Este cable es generalmente referido como STP, Tipo TTP RJ45 categoría 3 (4Mbps) o categoría 4 (16Mbps).

Para una conexión con Ethernet se requiere un tarjeta #2617 (Ethernet/IEEE 802.3 Network Adapter/HP), y cable de par torcido con un conector RJ45 de 15 pines.

Para lograr éstas conexiones se deben crear y configurar tres objetos principales en el AS/400:

- Descripción de línea.
- Descripción del Controlador.
- Descripción de dispositivo.

Descripción de Línea.

La descripción de línea describe el sistema local y el tipo de comunicación que éste utiliza. Define la interfase física entre el sistema local y un sistema remoto, controlador o red y el protocolo usado para las comunicaciones. También puede incluir información acerca de la velocidad de la línea y la dirección de la red o el teléfono del sistema local.

Las descripciones de línea se crean por medio de comandos de lenguaje de control (CL); el comando utilizado depende del protocolo de comunicaciones. Para comunicar dos máquinas AS/400 se puede utilizar una línea de comunicaciones asíncronas, la cual, se crea con el comando CRTLINASC.

Descripción del Controlador.

La descripción del controlador describe las características del sistema remoto, controlador o red que va a comunicarse con el sistema local. Contiene información de la dirección del sistema remoto, información para describir la comunicación entre el sistema local y el remoto. Se pueden asociar muchas descripciones de controlador con una sola descripción de línea.

El comando específico para crear la descripción del controlador depende del tipo de comunicaciones empleado. Para la comunicación entre dos máquinas AS/400 se crea un controlador asíncrono de comunicaciones, utilizando el comando CRTCTLASC.

Descripción de Dispositivo.

La descripción de dispositivo contiene la descripción del dispositivo lógico o físico que se va a comunicar con el sistema local; en este caso se describe un dispositivo lógico que representa una sesión de comunicaciones. Se pueden asociar muchas descripciones de dispositivo con una sola descripción de controlador. Esta descripción incluye alguna forma de información de dirección, incluyendo el nombre del sistema remoto y el nombre de la descripción del controlador asociado con el dispositivo.

También en este caso el comando utilizado para la creación de la descripción depende del tipo de comunicaciones. Para la comunicación entre dos máquinas AS/400 se crea una descripción de dispositivo asíncrono utilizando el comando CRTDEVASC.

En el caso de querer conectar dos máquinas que se encuentran separadas físicamente por una distancia considerable, se puede utilizar un par de modems. Para este tipo de conexión es necesario configurar estos modems de la siguiente manera.

Configuración del Modem.

Para realizar esta configuración se debe entrar al menú de configuración de TCP/IP punto a punto con el comando `CFGTCPPTP`, y después se selecciona la opción 11 para ver las entradas por default para los modems en el AS/400 y checar si existe una configuración por default para el modelo del modem. Si el modem existe se va al siguiente paso de la configuración, si no existe se puede hacer de dos formas diferentes:

1. Tratar con la conexión por default para el modem genérico.
2. Adicionar una nueva conexión para el modem.

Para determinar cuál es la mejor de estas opciones se debe observar el manual del modem tomando en cuenta los principales parámetros de la configuración, si estos concuerdan con los del modem genérico, entonces es mejor utilizar esta configuración. Si el modem tiene diferentes parámetros de configuración, se debe crear una nueva entrada para éste.

Para adicionar una nueva configuración del modem se realiza con el comando `CFGTCPPTP` y después la opción 1 en el AS/400 presenta una pantalla para adicionar un nuevo modem. La guía de usuario del modem contiene los parámetros necesarios para adicionar una nueva configuración en el AS/400.

CAPÍTULO II

LENGUAJES DE PROGRAMACIÓN

CAPÍTULO II LENGUAJES DE PROGRAMACIÓN

II.1 LENGUAJE DE CONTROL

El lenguaje de control es el uso de los comandos de lenguaje de control, los cuales pueden realizar funciones del sistema operativo, así como de varios lenguajes y utilerías. Cuando se va a realizar una de estas funciones es necesario especificar en el comando CL el nombre del objeto sobre el que se va a realizar dicha función.

Los objetos son unidades que existen; es decir ocupan espacio en memoria, que tienen un nombre y sobre los cuales se realizan las funciones por el sistema operativo OS/400. La mayoría se guarda en librerías. Estos objetos del sistema operativo OS/400 son los que proveen la forma a través de la cual toda la información de procesamiento de datos es guardada y procesada por el sistema AS/400. Cada objeto tiene unos atributos descriptivos que son definidos cuando el objeto es creado. El nombre del objeto que va a ser usado por el sistema para realizar una función específica debe ser referido en el comando CL que realizará la función. Muchos objetos son creados y usados en el lenguaje de control.

Generalmente cada objeto es independiente de todos los demás, sin embargo, algunos deben ser creados antes de que otros objetos puedan ser creados también. Por ejemplo, un

archivo lógico no puede ser creado si el archivo físico en el cual está basado no existe. Cada objeto debe ser creado antes de cualquier operación que use el objeto antes de que sea realizada.

II.2 COMANDOS QUE OPERAN EN LOS OBJETOS OS/400

Todos los tipos de objetos OS/400 tienen comandos que se pueden usar en ellos. La mayoría de los objetos tienen comandos que realizan las siguientes funciones:

- Crear (CRT): Crea el objeto y especifica sus atributos.
- Borrar (DLT): Borra el objeto del sistema.
- Cambiar (CHG): Modifica los atributos y/o el contenido del objeto.
- Mostrar (DSP): Muestra el contenido de los objetos.
- Trabajar con (WRK): Trabaja con los atributos y/o contenido de los objetos. A diferencia de los comandos DSP en estos comandos se permite al usuario operar en los objetos y modificar aplicaciones.

Los nombres de los comandos consisten en una combinación de un verbo y del objeto sobre el cual se va a actuar. Por ejemplo, se puede crear, borrar o mostrar un programa; así las abreviaciones de verbos CRT, DLT y DSP se juntan con la abreviación de programa PGM. El resultado son tres comandos que pueden usarse en un programa: CRTPGM, DLTPGM y DSPPGM.

Adicionalmente a los comandos que se pueden usar en objetos de un solo tipo, existen comandos que se pueden usar con varios tipos de objetos; por ejemplo:

- Mover un objeto: Mueve un objeto de una librería a otra.
- Renombrar un objeto: Especifica el nuevo nombre de un objeto.
- Salvar un objeto: Salva un objeto y su contenido en disco, cinta o en un archivo de salvado.
- Restablecer objeto: Restablece la versión salvada del objeto de un disco, cinta o archivo de salvado.

Algunos de estos comandos sólo pueden actuar sobre un objeto a la vez, pero pueden ser utilizados en varios tipos de objetos.

Otros comandos, como el comando para mostrar la descripción de un objeto, pueden ser utilizados en objetos de diferentes tipos a la vez. Especificando varios tipos en el mismo comando se puede mostrar la descripción de un grupo de objetos

Existen también comandos que contienen más de una palabra en sus nombres descriptivos. Entre estos comandos están los llamados a programas (CALL), el condicional (IF o ELSE), el de una orden (DO), etc.

Con el lenguaje de control se pueden crear programas para el manejo de un proceso. Las facilidades que nos dan estos programas son, entre otras cosas, poder definir los pasos

que tiene un proceso, el orden en que se van a realizar y poder detectar posibles errores que se corrijan mediante una acción predeterminada. En estos programas se deben seguir ciertas reglas para el manejo de los comandos.

II.3 SINTAXIS DEL LENGUAJE DE CONTROL

En los programas de lenguaje de control se tiene una sintaxis especial, esta debe ser usada para codificar los comandos CL. Cuando el programa corre, cada comando es procesado por el sistema operativo OS/400 para realizar la función sobre el objeto especificado.

Un comando CL tiene las siguientes partes: etiqueta del comando que es opcional, nombre del comando, y uno o más parámetros. Un parámetro incluye una palabra clave y un valor. (Ver fig. 2.1)

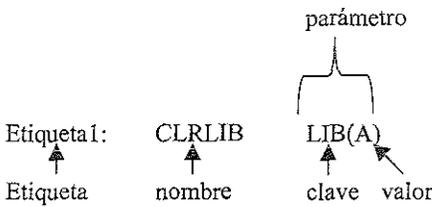


FIG. 2.1 Elementos de un comando CL

II.3.1 ETIQUETA DEL COMANDO

Las etiquetas de los comandos principalmente son usadas para hacer referencia a este comando (o línea en el programa) desde otro y mandar la secuencia de ejecución a éste.

La etiqueta del comando se escribe exactamente antes del nombre del comando. Esta etiqueta puede ser de hasta diez caracteres, y debe ser seguida inmediatamente por dos puntos; y pueden existir espacios en blanco entre los dos puntos y el nombre del comando.

Las etiquetas no son obligatorias pero se puede poner una etiqueta en cualquier comando, si una etiqueta se pone en un comando que no puede correr (por ejemplo, el usado para declarar una variable CL), y el programa manda a esta etiqueta como el siguiente comando a procesarse, el programa se mueve al siguiente comando ejecutable. Similarmente solo una etiqueta se puede poner en cada línea, si una etiqueta se coloca en una línea donde no existe comando, el programa brinca hasta el siguiente comando que puede ser corrido.

Para especificar etiquetas múltiples, cada etiqueta adicional debe estar en una línea separada precediendo al comando como se muestra a continuación:

```
LABEL1:
```

```
LABEL2:  CMDX
```

No se permiten los caracteres de continuación de línea (+ o -) en las líneas precedentes.

II.3.2 NOMBRE DEL COMANDO

El nombre del comando identifica la función que será procesada por el programa que corrió el comando. Como otros objetos, un nombre de comando puede ser opcionalmente calificado por un nombre de librería. El nombre del comando es una abreviación de la descripción del comando; por ejemplo, el nombre MOV OBJ identifica al comando Mover Objeto, el cual mueve un objeto de una librería a otra.

II.3.3 PARÁMETROS DEL COMANDO

La mayoría de los comandos CL tienen uno o más parámetros que especifican los objetos y valores usados para correr el comando. Cuando un comando es ejecutado, el usuario especifica los nombres de los objetos, el nombre del parámetro y sus valores usados para este comando. El número de parámetros especificados depende del comando, es decir, algunos no tienen parámetros mientras que otros cuentan con uno o más. Un parámetro identifica un valor individual o un grupo de valores usados por el comando.

Un comando puede tener parámetros que, forzosamente, deben ser especificados y otros que son opcionales. A los parámetros opcionales usualmente se les asigna un valor por default si no se especificaron al ser ejecutados.

A) PARÁMETROS EN FORMA DE PALABRA CLAVE

Consiste en una palabra clave seguida inmediatamente por un valor (o una lista de valores separados por espacios en blanco) entre paréntesis. No se debe poner espacios en blanco entre la palabra clave y el paréntesis izquierdo que precede al valor del parámetro. Es

posible colocar espacios en blanco entre el paréntesis y el valor cuando todos los parámetros de un comando son especificados en la forma de palabra clave y pueden colocarse sin importar el orden. Por ejemplo, los siguientes dos comandos:

```
CTRLIB LIB(MYLIB) TYPE(*TEST)
```

```
CTRLIB TYPE(*TEST) LIB(MYLIB)
```

B) PARÁMETROS EN FORMA POSICIONAL

Un parámetro en esta forma no cuenta con la palabra clave, solo contiene el valor (o valores, si es una lista), cuya función es determinada por su posición en el comando. Los valores de los parámetros están separados entre sí mismos y el nombre del comando por uno o más espacios en blanco. Debido a que los parámetros en forma posicional solo pueden tener una secuencia, la forma posicional del comando anterior es:

```
CTRLLIB MYLIB *TEST
```

Si no se quiere especificar el valor de uno de los parámetros, el valor predeterminado *N (Nulo) puede ser especificado en la posición de este parámetro. El sistema reconoce *N como un parámetro omitido, asignándole un valor por default o dejándolo nulo.

C) PARÁMETROS EN AMBAS FORMAS

Un comando puede también tener sus parámetros en una combinación de las formas posicional y de palabra clave. Los siguientes ejemplos muestran tres formas de definir el comando Declarar una Variable CL (DCL):

En forma de Palabra Clave:

```
DCL VAR (&QTY) TYPE(*DEC) LEN(5) VALUE(0)
```

En forma Posicional:

```
DCL &QTY *DEC 5 0
```

Ambas formas juntas:

```
DCL &QTY *DEC VALUE(0)
```

En el último ejemplo, como el parámetro LEN no fue especificado, el parámetro VALUE debe estar en la forma de palabra clave. Una vez descontinuada la forma posicional, no puede volver a comenzar.

II.3.4 SINTAXIS DE LOS COMANDOS

Los comandos tienen la siguiente sintaxis general. Los brackets indican que el valor que está entre ellos es opcional, sin embargo, el valor de los parámetros puede o no puede ser opcional dependiendo de los requerimientos del comando.

[?] [nombre de la etiqueta] [nombre de la librería/] nombre del comando [parámetros]

II.3.5 DELIMITADORES DE COMANDO

Son caracteres especiales o espacios que identifican el inicio y el final de un grupo de caracteres en un comando. Los delimitadores son usados para separar un arreglo de caracteres en las partes individuales que forman un comando: etiqueta, nombre, palabras clave de parámetros y valores de los parámetros. Los valores de los parámetros pueden ser constantes, variables, listas o expresiones como se muestra en la figura 2.2

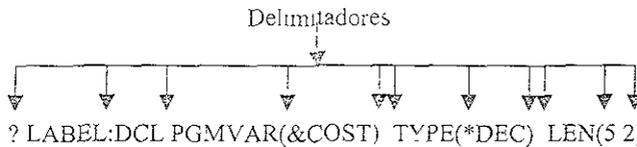


FIG. 2.2 Delimitadores de Comando

Los siguientes delimitadores se usan en el lenguaje de control del OS/400 :

- Los dos puntos marcan el fin de la etiqueta del comando y pueden ser usados para separarla del nombre del comando y para separar una hora en sus partes.
- El espacio en blanco separa el nombre del comando de sus parámetros y a sus parámetros entre sí, también separan valores en una lista. Varios espacios en blanco seguidos son tratados como uno solo a excepción de cuando están en un arreglo de caracteres entre comillas o un comentario en apóstrofes. Un espacio en blanco puede separar una palabra clave de su valor

- Los paréntesis son usados para separar los valores de los parámetros de sus palabras claves o para grupos de listas de valores.
- La diagonal conecta las partes de un nombre calificado (librería y nombre del objeto).
- Un punto o una coma pueden ser usados como punto decimal.
- Los apóstrofes son usados para delimitar un arreglo de caracteres.
- Los caracteres / . , - son usados para separar una fecha en sus partes (día, mes y año)
- Los caracteres /* y */ son usados para comentarios
- Los símbolos + y - son usados para continuar un comando en la línea siguiente

II.3.6 CARACTERES ESPECIALES

- Los símbolos *AND, *OR, *EQ, NE, *GT, *LT y *NOT son operadores lógicos.
- Los símbolos *CAT *BCAT y *TCAT son usados para juntar arreglos de caracteres.
- El & identifica el nombre de una variable CL cuando está en la primera posición.
- El % identifica una función cuando está en la primera posición.
- Los símbolos +, -, *, y / son usados como operadores aritméticos.

Cuando se tienen múltiples operadores en una expresión, se evalúan en un cierto orden, dependiendo de los operadores usados en ésta. Los paréntesis pueden ser usados para cambiar el orden de evaluación de la expresión. La figura 2.3 muestra la prioridad de todos los operadores:

| PRIORITY | OPERADOR |
|----------|---|
| 1 | Los signos + y -, valores decimales y la negación *NOT. |
| 2 | Multiplicación y división. |
| 3 | + y - cuando son usados como suma y resta. |
| 4 | *CAT, *BCAT y *TCAT |
| 5 | *GT, *LT, *EQ, *GE, *LE, *NE, *NG Y *NL. |
| 6 | *AND. |
| 7 | *OR. |

FIG. 2.3 Prioridad de Operadores

La prioridad más alta es la uno (o sea, los signos son evaluados primero), y la prioridad 7 es la prioridad más baja (es decir, la función *OR es evaluada hasta el final). Cuando se encuentran operadores de diferente prioridad en una expresión, se evalúan de acuerdo a ésta. Cuando existen varios operadores con la misma prioridad se ejecutan de izquierda a derecha.

II.4 LENGUAJE RPG

Para leer o modificar un archivo de base de datos de AS/400 se utiliza por lo regular un programa en RPG por las facilidades que da para leer, grabar, modificar o borrar registros del archivo. Entre estas facilidades se encuentran las de poder leer o borrar con una llave de acceso, actualizar solo los campos que se desean en un registro sin afectar los otros, encadenarse a otros archivos, facilita la generación de reportes etc.

El lenguaje RPG es escrito en una variedad de “hojas” de especificaciones donde cada una tiene sus funciones (ver fig. 2.4). Estas especificaciones deben tener un orden dentro del programa, cualquiera de ellas puede faltar pero al menos una de ellas debe estar presente en el programa.

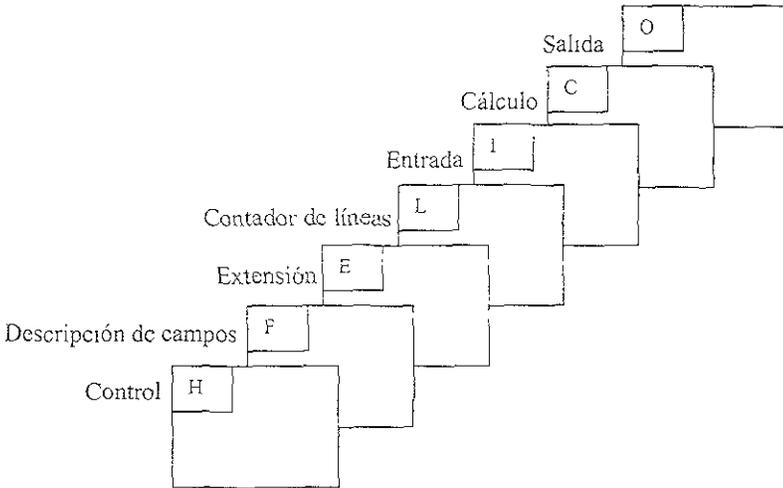


FIG. 2.4 Orden de los tipos de especificación en un Programa RPG

II.4.1 CONTROL

La especificación de control identificada por una H en la columna 6, nos da información acerca de la generación y corrida del programa compilado, aquí se puede indicar si se permite las operaciones de DEBUG y DUMP en el programa. El DEBUG es usado para correr un programa línea por línea y saber el valor de las variables en cualquier momento, esto es muy útil para encontrar errores en el programa. El DUMP es usado cuando un programa termina anormalmente y nos genera un reporte con las condiciones en el programa (valores en las variables, número de línea, etc)

También se puede definir el formato de fecha que se va a usar (AA/MM/DD, DD/MM/AA, MM/DD/AA), la notación decimal (Punto o coma), etc. Solo una especificación de control es permitida por programa.

II.4.2 DESCRIPCIÓN DE ARCHIVOS

Esta especificación define todos los archivos usados por el programa. Todos los archivos usados por el programa deben ser definidos, un máximo de 50 archivos totales pueden ser usados por programa, un solo archivo primario y 8 archivos de impresión.

Se pueden especificar los siguientes tipos de archivos:

- Entrada.
- Salida.
- Actualización.
- Combinado (entrada/Salida)

También se define si el archivo es descrito dentro o fuera del programa (esto es, si la definición de la longitud del registro, de los campos, el nombre de los campos, etc, es dada en el programa o fuera de él), la forma de procesar el archivo (leer el archivo en forma consecutiva o por medio de una llave), el indicador de fin de página en caso de un archivo de impresión, la designación del archivo (primario, secundario o full procedural), si se pueden adicionar registros al archivo, el control para abrir o cerrar el archivo, si es un archivo de impresión, base de datos o pantalla.

II.4.3 EXTENSIÓN

Describe todos los arreglos y tablas, el número de elementos la longitud y los decimales de los mismos, el número de elementos definidos por línea en caso de ser arreglo de tiempo de compilación, el tipo de arreglo, archivo de donde serán cargados en caso de ser arreglos que se cargan antes de la corrida del programa, el nombre de su tabla asociado en caso de ser tabla.

II.4.4 CONTADOR DE LÍNEAS

Indica la cantidad de líneas por hoja y la longitud de la forma para cada archivo de impresión del programa.

II.4.5 ENTRADA

Para un archivo de entrada esta especificación describe los campos en el registro, indicadores basados en el contenido de los campos, etc. Para los archivos descritos externamente esta especificación es opcional y puede usarse para adicionar funciones de RPG/400 a la descripción externa. Las especificaciones de entrada se usan también para describir estructuras de datos y constantes.

II.4.6 CÁLCULO

Describe los cálculos que se realizarán en el programa, así como el orden de éstos. Las especificaciones de cálculo pueden controlar ciertas operaciones de entrada y salida de ó a los archivos. Cada operación se especifica en una línea a excepción de cuando hay líneas con operaciones lógicas de AND /OR. Las operaciones de cálculo deben estar en el

orden en que van a ser realizadas, cada expresión de cálculo se puede dividir en tres partes:

- Las condiciones cuando se realizará el cálculo: en cada línea de cálculo se pueden especificar indicadores que condicionan cuándo y bajo qué condiciones se realizará la operación.
- Clase de operación: el tipo de operación, con qué campos se realizará y en qué campo se guardará el resultado.
- Calificado del resultado de la operación. existen indicadores para determinar el resultado de la operación, estos indicadores varían de acuerdo a la operación y pueden usarse para condicionar los siguientes cálculos o salidas.

Es posible definir variables de trabajo en las líneas de cálculo, estas variables solo existen mientras el programa está corriendo.

II.4.7 SALIDA

Describe los registros y el formato de los campos en un archivo de salida descrito en el programa e indica qué campos van a ser escritos o actualizados. Las especificaciones de salida son opcionales para un archivo descrito externamente.

II.5 CÓDIGOS DE OPERACIÓN

El lenguaje RPG/400 es un lenguaje posicional (cada instrucción debe comenzar en una columna específica), que nos permite en las especificaciones de cálculo hacer muchas

operaciones diferentes con la información. Cualquier operación tiene cuatro partes esenciales:

| Factor1 | Código de operación | Factor2 | Resultado |
|---------|---------------------|---------|-----------|
|---------|---------------------|---------|-----------|

El factor 1 y factor 2 son los valores con los cuales se va a efectuar la operación. Los códigos de operación indican el tipo de operación a realizar, usualmente estos códigos son abreviaciones de los nombres de las operaciones. El resultado es donde se va a dejar el valor obtenido en la operación.

Los códigos de operaciones pueden ser agrupados en categorías:

II.5.1 OPERACIONES ARITMÉTICAS

Las operaciones aritméticas son:

- ADD (adicionar).
- DIV (dividir).
- MULT (multiplicar).
- MVR (mover el residuo).
- SQRT (raíz cuadrada).
- SUB (substracción).
- XFOOT (suma los elementos de un arreglo).
- Z-ADD (borra y adiciona).
- Z-SUB (borra y subtrae).

Se debe recordar lo siguiente cuando se realizan operaciones aritméticas:

- Las operaciones aritméticas pueden ser hechas solo sobre campos numéricos.
- La alineación decimal es hecha para todas las operaciones, aún cuando se truncan decimales, la posición del punto decimal no se ve afectada.
- Solo se afecta el factor del resultado.
- Cualquier valor puesto en el campo de resultado reemplaza al valor anterior.
- El campo de resultado debe ser lo suficientemente grande para contener el valor obtenido en la operación, ya que el programa no genera ningún error; si el campo no es lo suficientemente grande los dígitos se truncan para cualquiera o ambos lados, dependiendo de la posición del punto decimal.
- Se puede usar el redondeo para los resultados de las operaciones.

La longitud de cualquier campo usado en operaciones aritméticas no puede exceder de 30 dígitos. Todas las operaciones aritméticas se hacen en forma algebraica, o sea, siguiendo las siguientes reglas:

Adición: Si los dos factores de la operación tienen el mismo signo, el factor resultante tendrá este signo también. Si los factores tienen signos diferentes, el signo del resultado será el del factor con el valor absoluto mayor.

Substracción: Se cambia el valor del factor 2 y se aplican las mismas reglas de la suma.

Multiplicación: Si los factores tienen signos iguales el resultado será positivo, si tienen signos diferentes el resultado será negativo.

División: Si los signos de los factores son iguales el resultado será positivo, si sus signos son diferentes el resultado es negativo. El signo del residuo es el mismo del primer factor.

Para la suma, resta, división y multiplicación el primer factor no es obligatorio; si el factor 1 no se especifica, la operación se hace como si éste y el campo de resultado fueran el mismo

II.5.2 OPERACIONES DE MOVER

Estas operaciones son:

- MOVE (mover).
- MOVEA (mover arreglo)
- MOVEL (mover a la izquierda)

Las operaciones de mover transfieren todo o parte del factor 2 al campo de resultado. El factor 2 no cambia, y el factor 1 no se debe especificar. Generalmente el factor 2 y el resultado son del mismo tipo, sin embargo se pueden mover datos de un campo carácter a uno numérico y viceversa.

La operación MOVE transfiere los datos comenzando de derecha a izquierda del factor 2, mientras que MOVEL los transfiere comenzando de izquierda a derecha.

II.5.3 OPERACIONES DE COMPARACIÓN

Estas operaciones son:

- COMP (comparar)
- CABxx (comparar e ir)
- CASxx (invocación de una subrutina condicionada)
- DOUxx (hacer hasta que)
- DOWxx (hacer mientras)
- IFxx (condicional)
- WHxx (cuando – entonces)

En las operaciones CABxx, CASxx, DOUxx, DOWxx, IFxx, y WHxx, xx puede ser:

| xx | Significado |
|----|--------------------------------------|
| GT | factor1 es mayor que factor2. |
| LT | factor1 es menor que factor2. |
| EQ | factor1 es igual al factor2. |
| NE | factor1 no es igual al factor2. |
| GE | factor1 es mayor o igual al factor2 |
| LE | factor1 es menor o igual al factor2. |

Estas operaciones son ejecutadas si las condiciones se cumplen, ningún campo es cambiado por estas operaciones. Se debe recordar lo siguiente cuando se usan estas operaciones:

- Si se comparan campos numéricos o de caracteres de diferente longitud, el campo más pequeño es llenado con ceros o con blancos a la derecha o izquierda del punto decimal para que sean de la misma longitud y compararlos.
- Todas las comparaciones numéricas son algebraicas, esto es un número con signo positivo es siempre más grande que uno negativo.
- No se puede comparar campos numéricos con campos de caracteres.
- El nombre de un arreglo no se puede especificar en estas operaciones, solo se puede especificar uno de sus elementos.
- Las operaciones ANDxx, y ORxx pueden ser usadas dentro de un DOUxx, DOWxx, IFxx, Y WHxx.

II.5.4 OPERACIONES DECLARATIVAS

Estas operaciones no causan ninguna acción y pueden ser especificadas en cualquier lugar de las especificaciones de cálculo. Las siguientes operaciones son declarativas:

- DEFN (definición de campos)
- KFLD (definición de una parte de una llave de acceso)
- KLIST (define una llave compuesta)
- PARM (identifica los parámetros)
- PLIST (identifica una lista de parámetros)
- TAG (destino)

La operación DEFN define un campo basado en los atributos de otro campo o define un campo como una data área. Las operaciones KLIST y KFLD se utilizan para indicar el nombre con el cual será referida una llave compuesta y los campos que la componen. Una llave compuesta es la que contiene una lista de campos con los cuales se va a acceder al archivo. Las operaciones PLIST y PARM son usadas con el comando CALL para permitir a un programa mandar valores que serán usados por el programa llamado. La operación TAG nombra la línea de destino para una las operaciones GOTO y CABxx.

II.5.5 OPERACIONES DE INICIALIZACIÓN

Permiten limpiar o inicializar todos los elementos de una estructura (formato de registro, estructura de datos, arreglo o tabla) o una variable. Estas operaciones son:

- CLEAR (limpiar).
- RESET (reiniciar).

La operación CLEAR pone todos los elementos de una estructura o variable en cero, blancos o '0', dependiendo del tipo de campo (numérico, caracter, o indicador), mientras que RESET los pone en su valor inicial.

II.5.6 OPERACIONES CON SERIES DE CARACTERES

Estas operaciones incluyen concatenación, búsqueda y selección de series de caracteres, solo se pueden usar en campos de tipo carácter. Las operaciones con series son:

- CAT (concatena o junta dos series de caracteres)
- SCAN (busca una serie de caracteres en un campo)

- SUBST (selecciona una serie de caracteres de un campo)

La operación CAT junta dos series de caracteres para formar una sola. La operación SCAN busca si se encuentra una serie de caracteres dentro de una segunda serie de caracteres. La operación SUBST extrae una serie específica de caracteres de una serie de base de caracteres.

II.5.7 OPERACIONES DE SUBRUTINAS

Una subrutina en RPG/400 es un grupo de instrucciones de cálculo en un programa que pueden ser ejecutadas muchas veces en el mismo, una subrutina puede ser llamada desde cualquier parte del programa. Estas operaciones son:

- BEGSR (comienza subrutina)
- ENDSR (fin de subrutina)
- EXSR (ejecuta una subrutina)
- CASxx (ejecuta una subrutina en forma condicional)

II.5.8 OPERACIONES DE PROGRAMACIÓN ESTRUCTURADA

Estas operaciones son:

- ANDxx (condicional “Y”)
- DO (hacer)
- ELSE (en el otro caso)
- END o ENDyy (fin del grupo de operaciones)
- ITER (iteración)

- LEAVE (salir del grupo de operaciones)
- ORxx (condicional “O”)
- OTHER (en cualquier otro caso)
- SELEC (comienza un grupo de selección)

La operación DO permite procesar un grupo de cálculos una o más veces, comenzando con el valor del factor 1 incrementando este valor con el valor de la operación ENDDO asociada hasta que el límite especificado en el factor 2 sea alcanzado. Las operaciones ANDxx y ORxx son usadas con las operaciones condicionales para formar condiciones más complejas que la comparación de un factor 1 con un factor 2. La operación ELSE es usada con la operación IFxx para formar otro grupo de instrucciones que serán ejecutadas en caso de que las condiciones xx no se cumplan. La operación ENDyy nos indica el fin de un grupo de instrucciones. Cuando se especifica una operación ITER ésta nos manda a realizar nuevamente el grupo de instrucciones en el que está la operación; y la operación LEAVE nos manda fuera de él. La operación SELEC comienza un grupo de selección en el que solo se realiza una de varias secuencias alternativas de operaciones. Esta secuencia se selecciona mediante la operación WHxx y en caso de que ninguna de las condiciones indicadas se cumpla se realiza la secuencia indicada en la operación OTHER.

II.5.9 OPERACIONES CON ARCHIVOS

- CHAIN (acceso a un registro basado en una llave de acceso)
- CLOSE (cierra un archivo)

- DELET (borra un registro del archivo)
- EXCPT (salida a un archivo en tiempo de calculo)
- EXFMT (presenta un formato de pantalla)
- OPEN (abre un archivo)
- READ (lee un registro)
- READC (lee el siguiente registro que fue modificado en una pantalla)
- READE (lee un registro igual a la llave de acceso)
- SETGT (busca el primer registro mayor a la llave de acceso)
- SETLL (busca el último registro menor a la llave de acceso)
- UPDAT (actualiza un registro en el archivo)
- WRITE (escribe un registro en el archivo)

II.5.10 OPERACIONES DE LLAMADO DE PROGRAMAS

- CALL (llama a un programa)
- FREE (libera a un programa que fue llamado con anterioridad)
- PARM (identifica los campos utilizados como parámetros)
- PLIST (identifica una lista de parámetros)

II.5.11 OPERACIONES CON ARREGLOS

- LOKUP (busca un valor en los elementos de un arreglo)
- MOVEA (mueve un arreglo)
- SORTA (ordena los elementos de un arreglo por el valor contenido en ellos)
- XFOOT (suma los valores contenidos en los elementos del arreglo)

II.5.12 OPERACIONES CON ÁREAS DE DATOS

- IN (recupera el valor de un área de datos)
- OUT (actualiza el valor del área de datos)
- UNLCK (libera el área de datos)

CAPÍTULO III

DESARROLLO DE UN SISTEMA

CAPÍTULO III DESARROLLO DE UN SISTEMA

III.1 SISTEMA

Existen diferentes tipos de sistemas, de hecho, casi todo aquello con lo cual entramos en contacto durante nuestra vida cotidiana es un sistema o bien parte de un sistema (o ambas cosas), por lo cual también existen varias definiciones de un sistema:

1. Grupo de elementos interdependientes o que interactúan regularmente formando un todo, por ejemplo, un sistema numérico.
2. Un grupo de cuerpos que interactúan bajo la influencia de fuerzas relacionadas, por ejemplo, un sistema gravitacional.
3. Un grupo de órganos del cuerpo que juntos llevan a cabo una o más funciones vitales, por ejemplo, el sistema digestivo.
4. Un grupo de aparatos o una organización que forma una red, especialmente para distribuir algo o para servir a un propósito común, por ejemplo, un sistema de proceso de datos.

Como se puede observar, la palabra sistema tiene un significado muy extenso, podemos dividir los sistemas en sistemas naturales (como el sistema digestivo) y en sistemas hechos por el hombre (como sistemas de transporte, de comunicación, de manufactura, etc.). Dentro de los sistemas hechos por el hombre, actualmente la mayoría incluye

computadoras y muchos no podrían sobrevivir sin ellas, pero algunos continúan sin computarizar y podrían seguir así por décadas.

Consideremos la frase “Juan tiene un sistema para llevar a cabo su trabajo” o “María tiene una manera sistemática de hacer su trabajo”. Estas frases no significan necesariamente que ellos han computarizado su trabajo, pero implica la división del mismo en una serie de pasos definidos para lograr algún propósito.

El propósito de automatizar un sistema es para hacerlo más eficiente, hacer más fácil y rápidamente las operaciones y tener un mejor control del proceso en general. La forma de automatizar un sistema es usando una o más computadoras para controlarlo, implementando de esta forma un sistema de información.

III.2 METODOLOGÍA DE IMPLEMENTACIÓN

Para la implementación de un sistema de información existen diferentes metodologías cuyo objetivo es el ayudar a la organización y control de todo el proceso. Estas metodologías definen los pasos que se deben seguir para llevar a cabo el desarrollo de un sistema asegurando que cumpla con las expectativas.

- 1) Podemos dividir una metodología en las siguientes partes principales como se muestra en la figura 3.1

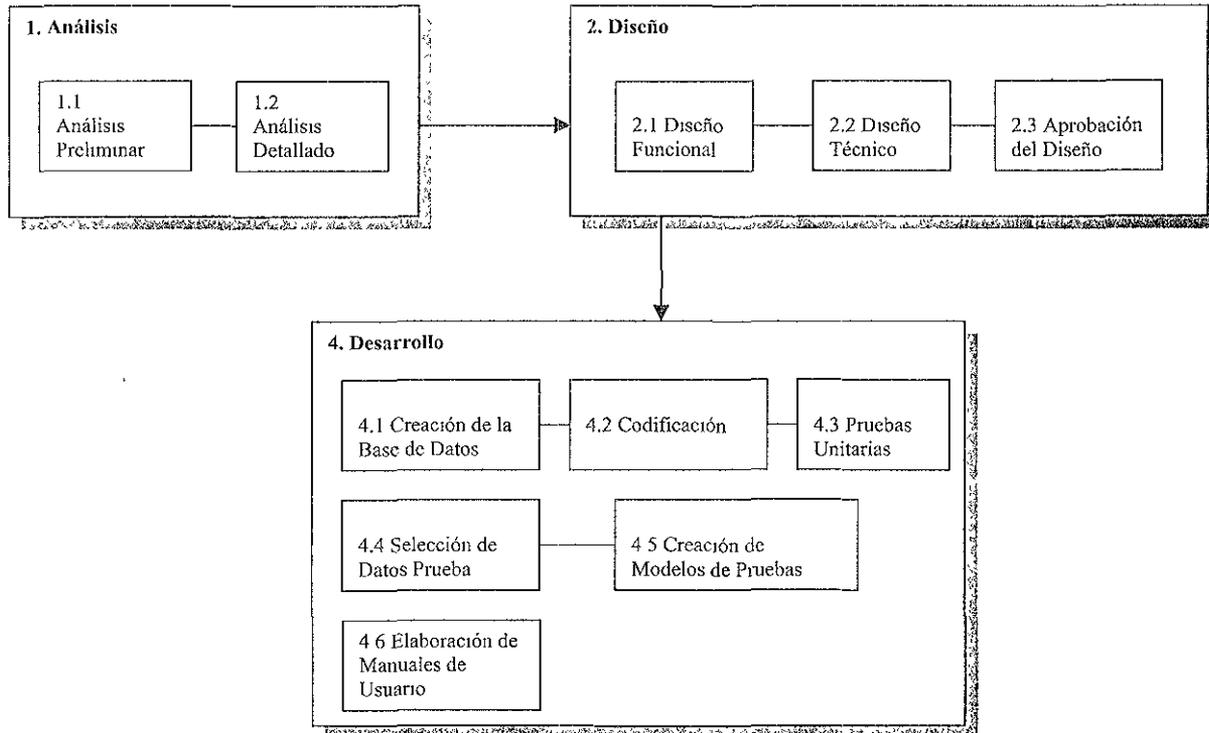


FIG. 3.1 Implementación de Sistemas

- 1) Análisis.
- 2) Diseño.
- 3) Desarrollo.

III.2.1 ANÁLISIS

El análisis es la parte donde se define el objetivo y alcances del sistema que se desea implementar, esto es imprescindible para su realización se estudian las necesidades y requerimientos del sistema, se recaba información del sistema actual (el que se está usando), si existe, para tener una idea conceptual del sistema lo cual será de ayuda para el diseño del mismo.

El análisis está muy ligado al diseño, sin un buen análisis no se tiene la información necesaria para el diseño, ya que éste está basado en el concepto, objetivos y alcances del sistema que se obtienen de un buen análisis.

El análisis se puede dividir en dos partes análisis preliminar y análisis detallado. (Ver figura 3 2).

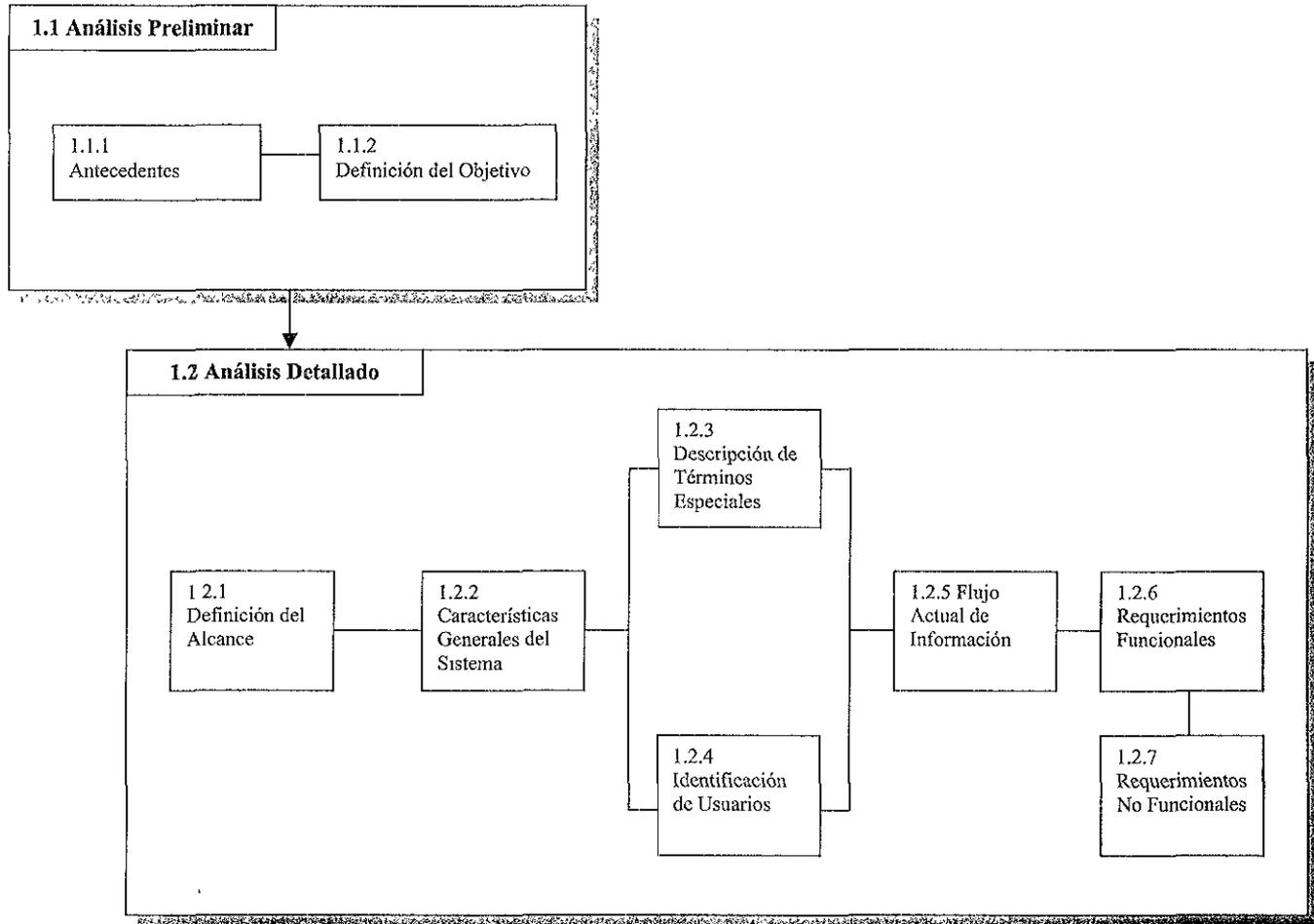


FIG. 3.2 Análisis

III.2.1.1 ANÁLISIS PRELIMINAR

Antecedentes

Para el desarrollo de un sistema es muy importante el entender cómo está funcionando ese sistema en la actualidad, para ello es necesario recabar la mayor información acerca del sistema y su funcionamiento. Esto se logra teniendo contacto con las personas que realizan el trabajo (los usuarios del sistema), tomando en cuenta también lo que esperan del sistema computarizado, sus necesidades y posibilidades futuras para el sistema.

También es importante tener una idea general de los sistemas que interactúan con el sistema a desarrollarse, esto significa que este sistema puede estar relacionado o formar parte de otros sistemas, así se puede estar trabajando en un sistema de nómina que a su vez forma parte de un sistema de recursos humanos, que a su vez forma parte de alguna organización (que en sí es un sistema), que a su vez es parte de un sistema económico.

En esta parte se recaban todos los antecedentes del sistema que se pretende implementar, es decir, se describe de manera ordenada y completa las características generales de la operación actual. Se toman en cuenta puntos como los siguientes:

- Forma de procesamiento actual (manual o automatizada).
- Elementos principales considerados en la operación.
- Principales procesos del sistema actual
- Descripción general de las fuentes de datos.
- Descripción de la problemática con el sistema actual.

- Usuarios a nivel de puestos y su participación en el sistema.
- Crear diagramas y/o gráficas que muestren la estructura de los datos que se deban procesar.

La forma de procesamiento actual, el impacto de la implantación del nuevo sistema, así como qué tanto se debe cambiar la forma como se han venido realizando los procesos para automatizarlo o si es factible el hacerlo.

Los elementos principales de la operación muestran aspectos claves que deben ser tomados en cuenta para el diseño del sistema.

Con los procesos principales de la operación se puede tener una idea de lo que el sistema hace y la importancia de cada paso de la operación actual.

La información que alimenta al sistema se obtiene de la descripción de las fuentes de datos, y así se pueden definir las formas de entrada de datos al sistema que se adapten a estas fuentes.

En la descripción de la problemática del sistema actual se tienen las mejoras a realizar o problemas a resolver en el nuevo sistema, estas son algunas de las expectativas que el usuario tiene del nuevo sistema.

La importancia del sistema se puede medir en parte por los puestos de los usuarios que lo utilizan y de esto depende muchas veces la velocidad de respuesta requerida en el sistema.

Definición de Objetivos

También en el análisis preliminar se debe definir el objetivo de manera clara y específica, sin profundizar en los elementos y componentes que debe incluir. Se debe redactar específicamente lo que como actividad principal se espera del nuevo sistema.

III.2.1.2 ANÁLISIS DETALLADO

Definición del Alcance

Se deben indicar todas las funciones que debe contener el sistema, se deben listar todas las actividades y procesos que el sistema debería realizar, con esto se logra dimensionar el proyecto. Algunos ejemplos de estas funciones son:

- Permitir mantenimientos.
- Extraer información por medio de interfases.
- Realización de cálculos.
- Impresión de reportes diarios.

Características Generales del Sistema

Se da también las principales características que debe tener el sistema para poder ser utilizado por los usuarios, ejemplo:

- Menú con el menor número de opciones.
- Diseñar pantallas con descripción de todas las funciones.
- Eliminar lo más posible la captura de datos.

Descripción de Términos Especiales

Estas características son puntos donde se debe poner especial atención en el diseño del sistema por ser puntos claves.

Identificación de Usuarios

Se realiza una identificación de usuarios clasificándolos en directos e indirectos. Los directos son aquellos que actualizan la información por medio de captura o procesos, los indirectos son aquellos que usan la información pero no la actualizan. Esta clasificación ayuda a planear las pruebas del sistema.

Flujo Actual de Información

Se describe el flujo actual de la información de forma secuencial esto es, las etapas de operación actual del sistema mostrando la actividad y su detalle.

Requerimientos Funcionales y No funcionales

Clasificación de requerimientos en funcionales y no funcionales:

A) Funcionales

Requerimientos indispensables para cubrir el objetivo principal del sistema, es decir, describir de manera clara y concreta las actividades necesarias para que el sistema funcione, ejemplos:

- Mantenimientos.
- Interfases con sistemas para tomar información.
- Procesos de generación de resultados
- Impresión de reportes de operación.

B) No funcionales

Requerimientos deseables pero no indispensables para cubrir el objetivo principal del sistema. Ejemplo:

- Identificar los elementos de una pantalla con diferentes colores.
- Generación de información para transferencia y proceso en excel.

Esta referencia de requerimientos nos sirve para asignar prioridades y/o poder quitar

III.2.2 DISEÑO

El diseño es la parte de la implementación donde se definen las características del sistema, es decir cómo manejará los datos, de cuántos archivos constará, cuántos procesos, pantallas, reportes y programas tendrá.

Se comienza con un diseño en forma conceptual donde se define en forma clara pero solo en concepto el funcionamiento del sistema, después se realiza el diseño de todos sus componentes (archivos, reportes, programas, etc.) como se muestra en la figura 3.3

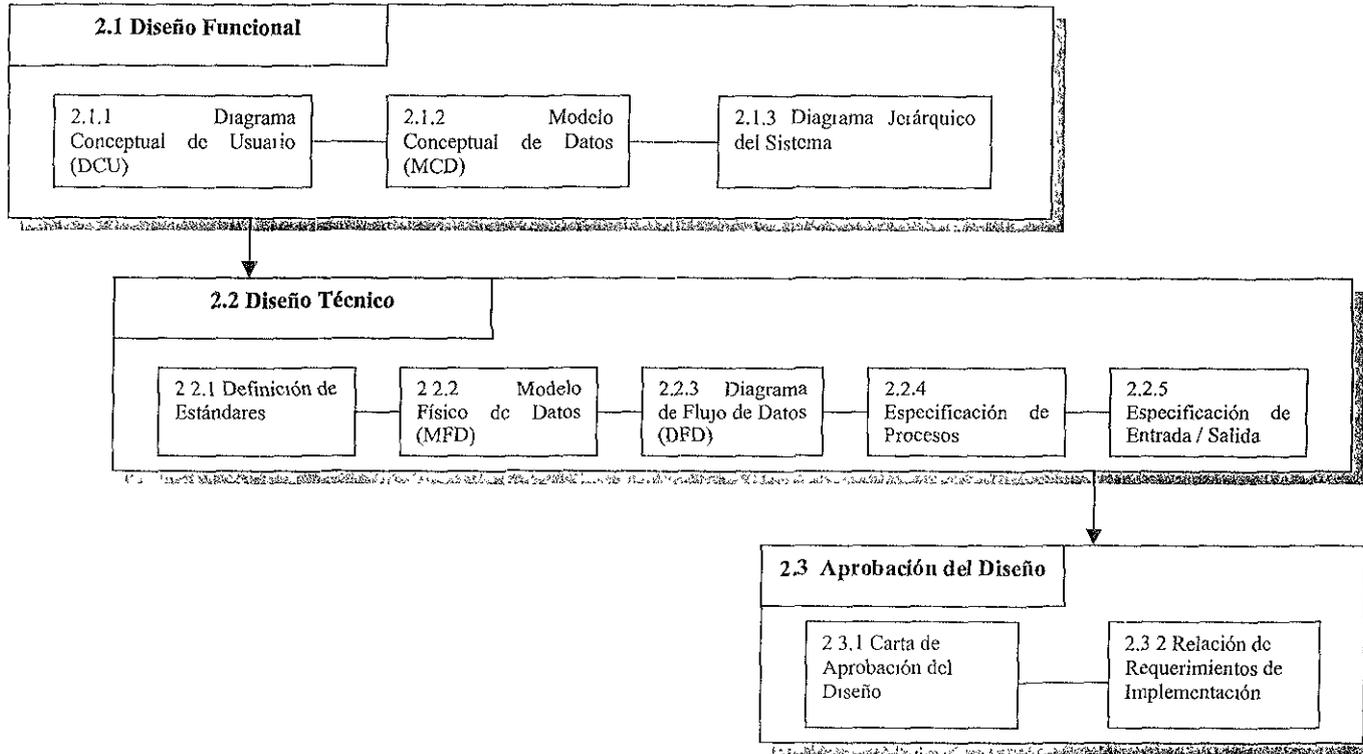


FIG. 3.3 Diseño

III.2.2.1 DISEÑO FUNCIONAL

Es la representación conceptual del sistema, ésta se puede representar por medio de un diagrama conceptual de usuario, un diagrama jerárquico y un modelo conceptual de datos.

Diagrama Conceptual de Usuario

El diagrama conceptual de usuario es un esquema gráfico que refleja los componentes principales del sistema, con indicadores para identificar los bloques. El nivel de detalle será determinado por el analista de acuerdo con los requerimientos del sistema.

Modelo Conceptual de Datos

El modelo conceptual de datos se puede especificar por medio de un diagrama entidad relación (DER), el cual debe reflejar la forma en que se relacionan los datos del sistema unos con otros.

Diagrama Jerárquico del Sistema

El diagrama jerárquico se puede obtener a partir del diagrama conceptual de usuario y en éste se muestran las diferentes opciones del sistema, con los cuales se puede obtener los menús que necesita el sistema.

III.2.2.2 DISEÑO TÉCNICO

Definición de Estándares

En el diseño técnico es muy importante la definición de los estándares porque permite unificar el criterio de desarrollo del sistema. Se deben listar todas estas especificaciones con las que el sistema debe cumplir, por ejemplo:

- Plataforma en la que deberá operar el sistema.
- Lenguaje de programación.
- Sistema manejador de base de datos.
- Estructura y datos de identificación de párrafos en programas.
- Variables de memoria e identificadores de valores
- Teclas de funciones.
- Estructura en pantallas y reportes.
- Lista de instrucciones de programación excluidas

Modelo Físico de Datos

El modelo físico de datos se representa mediante un diagrama de archivos y referencias generado en base del modelo conceptual de datos. Aquí se deben establecer completamente los detalles de almacenamiento y acceso de datos que manejará el sistema. Se debe considerar lo siguiente:

- Archivos.
- Índices (llaves primarias).
- Vistas (lógicos).

Diagrama de Flujo de Datos

Es una representación gráfica de la forma en que los datos pasan de un proceso a otro hasta convertirse en información para el usuario. Este diagrama sirve para definir el esquema general del procesamiento de los datos fuente hasta los resultados o información de usuario, su detalle depende de la complejidad del sistema.

Especificación de Procesos

Se lleva a cabo una especificación de procesos en la cual se definen las actividades que se deben realizar para cada función del sistema. La descripción de operaciones básicas se debe hacer con un texto narrativo de las operaciones o funciones que debe cubrir cada proceso.

Especificación de Entrada / Salida

Se deben especificar las entradas y salidas de datos al sistema, con esto se tiene la descripción de las características con que deben contar las pantallas y reportes para realizar cada función del sistema.

III.2.3 DESARROLLO

El desarrollo es la creación de todos los elementos del sistema tomando en cuenta todas las características y definiciones obtenidas en el diseño; es decir, los archivos físicos, lógicos, de impresión, pantallas y programas. (Ver figura 3.4)

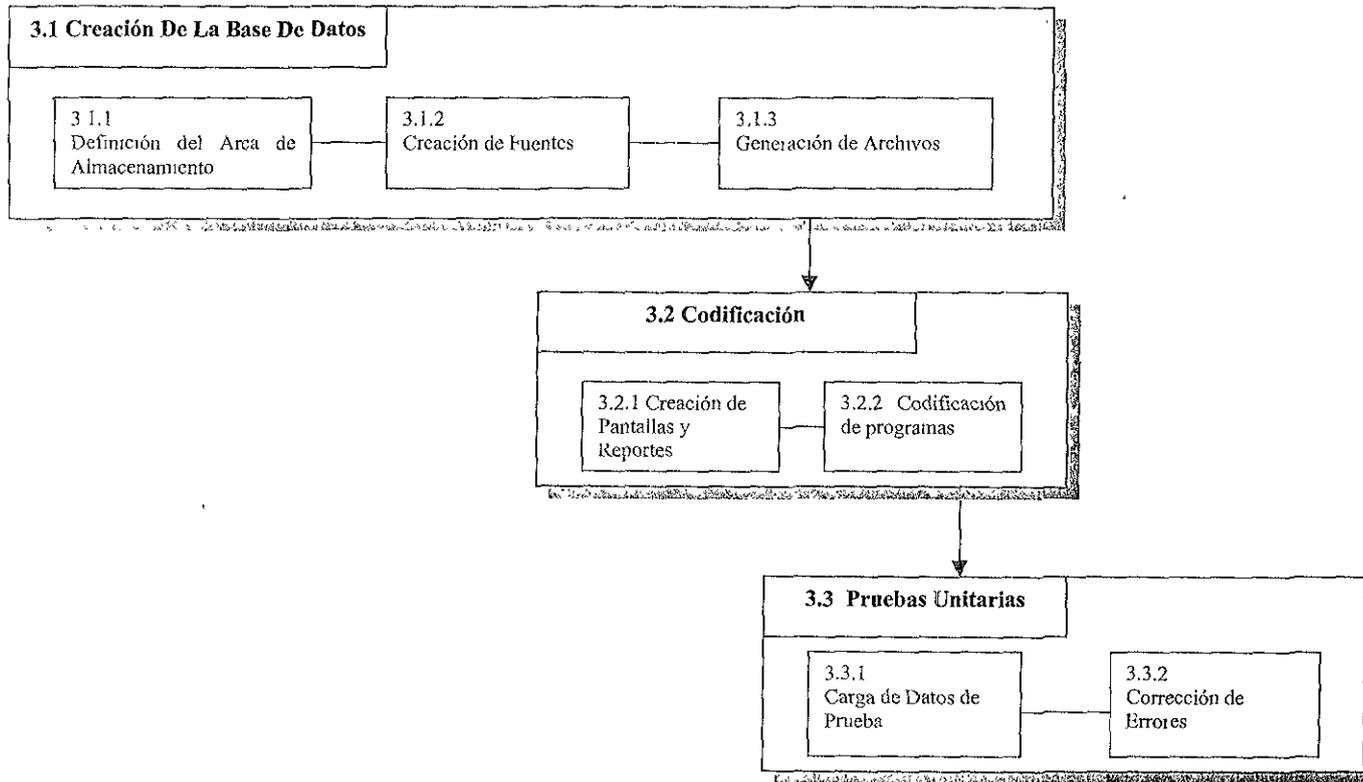


FIG. 3.4 Desarrollo

III.2.3.1 CREACIÓN DE LA BASE DE DATOS

Definición del Área de Almacenamiento

Aquí se les asigna nombre a los archivos físicos y lógicos que contendrán la información, así como a los campos y registros de los mismos. Estos nombres son con los que serán identificados por los programas del sistema cuando se quiera acceder a ellos.

Creación de Fuentes

Para crear un archivo físico o lógico es necesario un archivo fuente de los mismos. El archivo fuente es usado por el compilador para tomar las propiedades que debe tener el archivo a crear. Estos archivos fuentes se deben crear con las especificaciones obtenidas del diseño de la base de datos.

Generación de Archivos

Una vez teniendo los archivos fuente, el siguiente paso es crear los archivos de la base de datos compilando estas fuentes. La compilación es la creación de los archivos tomando las características del archivo fuente. Primero se deben crear los archivos físicos y después los lógicos; ya que estos últimos al ser solo una lista lógica de un archivo físico necesita un archivo en el que se pueda basar para su ordenamiento

III.2.3.2 CODIFICACIÓN

Creación de Pantallas y Reportes

Sus sistemas utilizan pantallas y reportes como dispositivos de entrada y salida de datos. En las pantallas se puede presentar información contenida en el sistema o capturar información que va a ingresar al sistema; por lo cual deben definirse sus campos que pueden ser de entrada, salida o entrada/salida.

Los reportes son una presentación de la información contenida en el sistema, en estos, todos los campos son de salida.

Codificación de Programas

Los programas de un sistema son los que contienen los procedimientos o formas como se debe registrar, presentar o procesar la información en el sistema. Estas acciones a tomar se especifican al computador por instrucciones de algún lenguaje. Se debe de especificar qué archivos, pantallas, reportes, etc., se usaron en el programa, así como los campos de éstos. También se debe especificar lo que se hace con la información leída, qué operaciones se realizan con ella, si se presenta en pantallas o en reportes y cómo se graba la información en archivos de salida o actualización.

Para la creación de los programas se definen todas estas acciones en un archivo fuente en algún lenguaje de programación con el cual, a través de su respectivo compilador se crea el programa.

III.2.3.3 PRUEBAS UNITARIAS

Carga de Datos de Prueba

Se deben seleccionar datos que sirvan para probar la aplicación. En estos datos hay algunos que son imprescindibles, como son catálogos, datos de autoridades a alguna aplicación, archivos maestros, etc. Estos datos deben ser cargados al nuevo sistema para poder realizar las pruebas.

Corrección de Errores

En las pruebas del sistema se detectan ciertas fallas o ajustes que deben hacerse o algunos cambios a pantallas o reportes. Estos deben corregirse en los archivos fuentes y volver a crear los objetos (pantallas, reportes, programas, etc.) y realizar nuevas pruebas hasta afinar bien todos los detalles del sistema.

CAPÍTULO IV

DESARROLLO DEL SISTEMA PARA COMUNICACIÓN ENTRE PROGRAMAS

CAPÍTULO IV DESARROLLO DEL SISTEMA PARA COMUNICACIÓN ENTRE PROGRAMAS

IV.1 ANÁLISIS

Para el desarrollo de este sistema como en cualquier otro la parte más importante es el análisis ya que de éste parte el diseño del mismo. Este análisis se presenta dividido en dos partes principales análisis preliminar y análisis detallado.

IV.1.1 ANÁLISIS PRELIMINAR

En el análisis preliminar se pretende comprender cómo funciona el sistema actualmente y establecer un objetivo claro del mismo.

IV.1.1.1 ANTECEDENTES

La forma como trabaja el sistema actual es de forma automática, por medio de DDM's o mediante diferentes comandos dentro de los programas. Cada programa que requiere acceder, mandar o recibir información a otro sistema que no sea en el que reside realiza un proceso que puede ser solo el uso de un DDM o el de varios comandos para la tarea requerida.

Se tienen dos o más sistemas que residen en diferentes máquinas los cuales necesitan compartir información de los archivos de sus bases de datos, algunas veces es necesario correr procedimientos que seleccionan la información deseada, esto casi siempre se realiza por medio de los DDM's.

Los principales procesos pueden ser el acceso a las estadísticas de ventas de un sistema a otro. En manufactura se tiene un sistema que calcula la producción basado en el promedio de ventas de las últimas semanas, para esto debe seleccionar un rango de datos de los archivos del sistema de estadísticas de ventas.

Otro proceso importante es el requerido por los sistemas de apoyo a la toma de decisiones, estos sistemas deben tener datos de todos los sistemas para poder tener una visión más amplia de la empresa. Para estos sistemas es importante la transferencia de información desde todos los demás sistemas para poder organizar toda esta información y presentarla de una manera apropiada.

Las fuentes de datos son muy variadas, por tratarse de acceso o transferencia de información entre dos o más sistemas, es mucha la variedad de archivos que pueden usarse.

Estos procesos en que se comparte información son en sí interfases entre los sistemas de la empresa, al permitir conjuntar los diferentes sistemas.

El principal problema que presenta el sistema actual es la velocidad de acceso a los sistemas remotos, debido a que estos accesos se realizan la mayoría de las veces por DDM's y a la manera de acceso a éstos comparada con el acceso a un archivo local (son

más las transmisiones en un DDM como se vio en el capítulo I), los procesos son más lentos dependiendo del volumen de datos que se maneje.

Esto se puede ver claramente cuando se necesita la información de las ventas del último mes por ejemplo, si el acceso es por medio de un DDM se lee registro por registro haciendo todas las respectivas transmisiones.

Otro problema se presenta cuando no se utilizan DDM's y se utilizan comandos para el envío de los datos. Al utilizar estos comandos dentro de los programas se deben codificar secuencias de instrucciones repetitivas en todos los programas con transferencia de información. Además al quedar estas instrucciones dentro de los programas en donde se especifica la dirección del sistema remoto, llamados a programas, archivos a enviar y recibir así como sus librerías, se dificulta darle flexibilidad a estos programas. En la mayoría de las veces es necesario un cambio al código de los programas con instrucciones para transferencia de datos cuando hay un cambio de las condiciones en las que se creó al inicio, esto es entre qué computadoras se va establecer la comunicación, qué programas se van a llamar en cada una, cuáles archivos se van a enviar o recibir, etc., y dependiendo del volumen de programas de este tipo puede llegar a ser un problema demasiado grande por el número de programas que se deben modificar y volver a crear

En ocasiones por motivo de una contingencia es necesario juntar dos sistemas que antes estaban separados en la misma máquina, cuando se tienen programas que se requieran cambiar eliminando los accesos remotos (ya que ahora los archivos del otro sistema

residen en la misma máquina), como DDM's, comandos remotos, envío y recepción de archivos, el tiempo requerido para implementar la contingencia es muy elevado y esta suele ser muy problemática.

Estos procesos de acceso o transferencia de información solo son usados por gente de desarrollo de sistemas al estar diseñando o modificando sus aplicaciones que requieren compartir información con otros sistemas.

IV.1.1.2 DEFINICIÓN DE OBJETIVOS

Ayudar a mejorar el tiempo requerido para comunicaciones, procesamiento, desarrollo de programas y dar mayor flexibilidad en los procesos de acceso y transferencia de información con sistemas remotos.

IV.1.2 ANÁLISIS DETALLADO

Para mejorar el tiempo requerido para las comunicaciones entre procesos, tomando en cuenta las características del AS/400, se puede enviar toda la información requerida en una sola transacción de información entre máquinas. Para realizar esto en el AS/400 se debe controlar el envío de los archivos entre máquinas y la recepción de estos en las librerías requeridas.

Asimismo, para evitar enviar información que no es necesaria, se debe tener programas que puedan seleccionar la información requerida para cada proceso, y llenar con esta los archivos que se enviarán de una máquina a otra.

Para poder automatizar los procesos (esto es, evitar la intervención de un operador para el control de los mismos), es necesario que los procesos tengan la posibilidad de realizar estas requisiciones de información.

IV.1.2.1 DEFINICIÓN DE LOS ALCANCES

El sistema debe dar a un proceso en la máquina local las siguientes opciones:

- Enviar información a la máquina remota en archivos (cuando sea necesario).
- Recibir estos archivos en la máquina remota en las librerías especificadas.
- Llamar a un programa en la máquina remota para realizar una selección de información.
- Enviar los archivos con la información seleccionada a la máquina local.
- Recibir los archivos en la máquina local en las librerías especificadas
- Llamar a una programa en la máquina local desde la máquina remota para continuar con el proceso utilizando la información recibida.
- Coordinación entre envíos de archivos y llamado de programas, esto es no continuar un proceso hasta no haber recibido los archivos con la información requerida para el mismo.
- Manejo de lista de librerías con la cual trabajarán los programas llamados en el sistema local y remoto.
- Permitir el uso de parámetros en le llamado de los programas en la máquina remota y local

La selección de información, por ser muy específica para cada proceso, no se puede adicionar a este sistema, por lo cual debe ser realizada por los desarrolladores que harán uso del mismo.

Para tener un sistema flexible este debe tener las siguientes características:

- Debe permitir el manejo de diferentes plantas.
- Considerar un posible cambio en la localización de los sistemas.
- Manejar toda la información necesaria para cada una de las comunicación entre procesos requeridas mediante una base de datos identificándolas mediante una clave.

IV.1.2.2 CARACTERÍSTICAS GENERALES DEL SISTEMA

- A) Se debe evitar lo más posible la captura de código en los programas que utilicen este sistema
- B) Contar con un proceso eficiente para disminuir las cargas de procesamiento
- C) Eliminar lo más posible la captura de datos.

IV.1.2.3 DESCRIPCIÓN DE TÉRMINOS ESPECIALES

Clave de Proceso: Es un identificador con el cual se accesa a los archivos de base de datos del sistema para obtener cuáles deben ser los archivos a enviar, qué programa se debe correr en el sistema remoto y en el local, con qué lista de librerías, etc

Clave de Planta: Esta clave se utiliza debido a que pueden existir más de una planta que se manejan separadamente (por ejemplo en diferentes librerías).

Clave de Transferencia de Información: Es un identificador que se utiliza para diferenciar los archivos que se mandan a la máquina remota de los que se reciben desde la máquina remota. Estas claves son las siguientes:

INIC : Archivos que se envían de la máquina local a la remota.

RETO: Archivos que se envían de la máquina remota a la local.

Clave de Identificación de Librerías: Es un identificador que sirve para diferenciar las librerías de la lista de librerías de la máquina local y la remota.

INIC : Librería de la máquina remota.

RETO Librería de la máquina local.

Clave de Orden: Es un número que indica el orden que deben tener las librerías en cada una de las listas de librerías o el orden con el que serán enviados los archivos.

Clave de Tipo de Proceso: Es un identificador para saber si el proceso se debe correr en la máquina remota o en la máquina local; por ejemplo, en casos de contingencia cuando dos sistemas que originalmente radican en diferentes máquinas, por fallas en alguna de ellas, deben coexistir en la misma máquina. En este caso con solo cambiar esta clave, el

sistema enviará, recibirá y correrá en la máquina local los archivos y programas que anteriormente se ejecutaban en la máquina remota.

Adicionar o reemplazar información: Identifica la información del archivo receptor será reemplazada o solo se adicionará la nueva información.

Parámetros de Programa: Son valores que se envían al programa antes de su ejecución, los cuales serán tomados como condiciones iniciales de éste.

IV.1.2.4 IDENTIFICACIÓN DE USUARIOS

Directos: Son las personas de desarrollo de sistemas, quienes se encargan de la actualización de la base de datos.

Para los usuarios en general, es sólo una interfase, la cual por sí misma, no presenta ni genera información, sólo la transmite para ser utilizada por otros sistemas.

IV.1.2.5 FLUJO ACTUAL DE INFORMACIÓN

| ACTIVIDAD | DESCRIPCIÓN |
|----------------------------|---|
| 1) Preparación del Usuario | a) Alta de la clave de usuario en las máquinas a utilizar. b) Alta del usuario en el directorio de distribución de ambas máquinas. |

c) Proporcionar autoridad al usuario sobre librerías, archivos y comandos necesarios.

2) Alimentar datos del proceso remoto

a) Captura de la clave del proceso y la clave de planta.

b) Captura de la dirección de las máquinas que intervienen en el proceso, así como el DDM usado para la conexión.

c) Especificar si se trata de un proceso remoto o local.

d) Especificar si tiene programa a ejecutarse en el sistema remoto y/o local. Si es así, se debe especificar cuáles son y sus respectivas librerías a utilizar con su respectivo orden.

c) Especificar los archivos a enviar, sus librerías; y la librería y archivos donde se recibirán.

3) Codificación de Programas

a) Codificación del llamado al programa principal del sistema en los programas que harán uso de éste, así como un campo de 256 caracteres con los

parámetros que se desean pasar a los programas y su recuperación del área de datos SY@010D en los mismos.

IV.1.2.6 REQUERIMIENTOS FUNCIONALES

Mantenimiento de Parámetros de los procesos de transferencia.

Proceso de envío de archivos.

Proceso de recepción de archivos.

Proceso para el llamado remoto de programas.

IV.1.2.7 REQUERIMIENTOS NO FUNCIONALES

Proceso de copia de datos de los archivos del sistema de transferencia de información de una máquina a otra.

Reporte de parámetros por clave de proceso

IV.2 DISEÑO

Toda la información recabada y el análisis realizado con anterioridad son usados en el diseño del sistema de transmisión de información lo cual nos ayuda a un mejor diseño del sistema para hacerlo funcional y que cubra todos los requerimientos.

IV.2.1 DISEÑO FUNCIONAL

A continuación se presenta una idea conceptual del sistema y sus elementos necesarios basados en el análisis de los requerimientos y objetivos del mismo.

IV.2.1.1 DIGRAMA CONCEPTUAL DE USUARIO

De las figuras 4.1 a 4.3 se muestran los niveles del diagrama de transferencia de información.



FIG. 4.1 Nivel I sistema de transferencia de información.

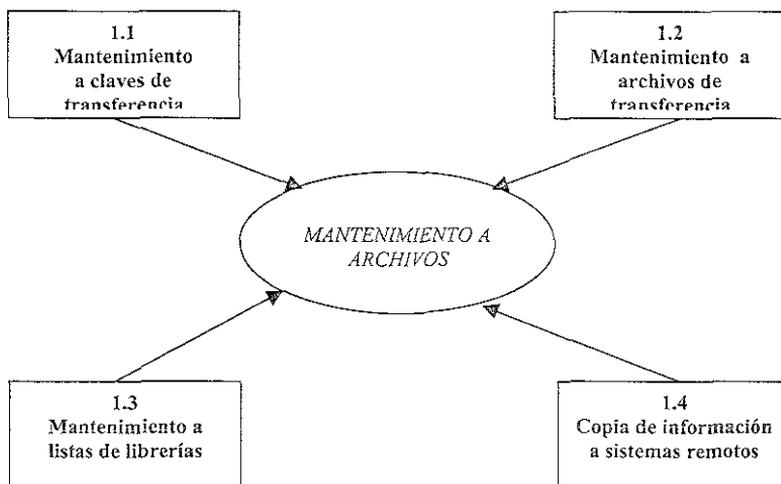


FIG. 4.2 Nivel 2 mantenimiento a archivos.

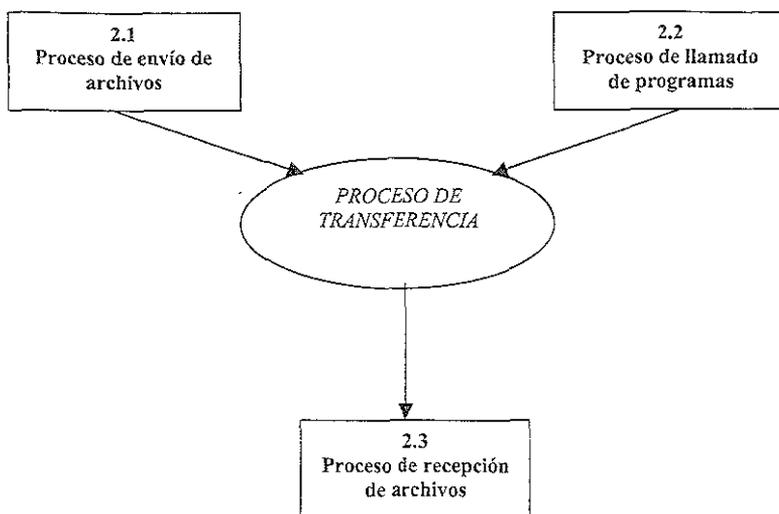


FIG. 4.3 Nivel 2 proceso de transferencia.

IV.2.1.2 DIAGRAMA JERÁRQUICO DEL SISTEMA

En la figura 4.4 se muestra el diagrama jerárquico del sistema.

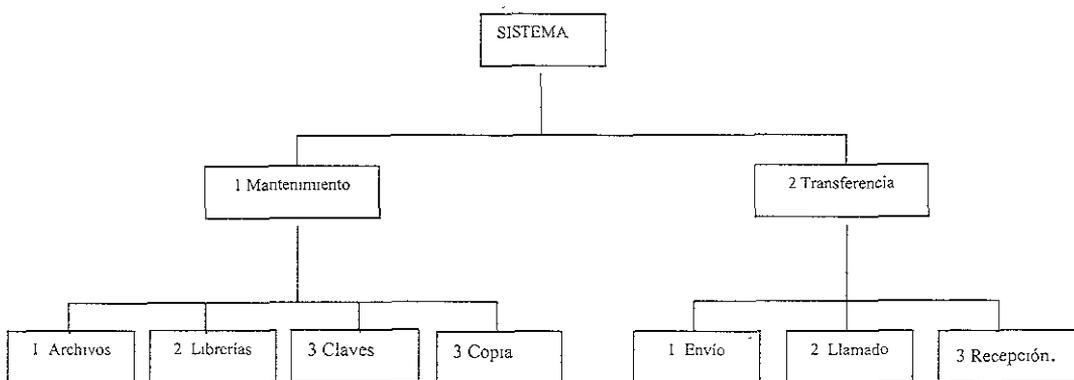


FIG 4.4 Diagrama jerárquico del sistema.

IV.2.1.3 MODELO CONCEPTUAL DE DATOS

El modelo conceptual de datos se obtiene de un análisis del sistema para la identificación de las entidades principales del mismo, así como la relación que guardan entre ellas. Las relaciones entre las entidades pueden ser de uno a uno, de una a muchos o de muchos a muchos. En la figura 4.5 se muestra este modelo conceptual de datos del sistema.

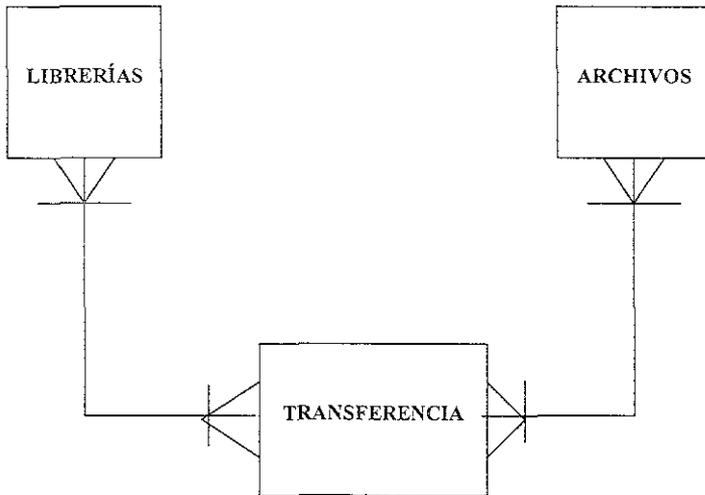


FIG. 4.5 Modelo conceptual de datos.

IV.2.2 DISEÑO TÉCNICO

Una vez teniendo el diseño conceptual se puede pasar al diseño de base de datos y demás objetos basado en lo anterior.

IV.2.2.1 DEFINICIÓN DE ESTÁNDARES

- La plataforma en que opera el sistema es AS / 400.
- Los lenguajes de programación en que se desarrollará el sistema es el RPG y el CL.
- Su sistema manejador de base de datos es el OS / 400
- Los encabezados de los programas deben contar con los siguientes datos:
 1. Sistema.
 2. Nombre del programa.
 3. Objetivo del programa.
 4. Autor.
 5. Fecha de creación
 6. En caso de tener modificaciones se le debe adicionar una historia de modificaciones con los siguientes datos:
 - Autor.
 - Causa de la modificación.
 - Fecha de la modificación.
 - Las subrutinas deben tener un encabezado con el nombre de la misma y su objetivo
- Las variables utilizadas en los programas deberán tener un nombre que ayude a identificar el uso que se les da en los mismos.

➤ Funciones:

- La función F3 es utilizada para salir de las opciones.
- La función F12 se utiliza para regresar a la pantalla anterior.
- La función F7 se utiliza para enviar la información de los archivos del sistema al sistema remoto.

➤ Las pantallas deben contar con los siguientes datos:

1. En el lado superior izquierdo:

- Nombre de la pantalla.
- Clave del usuario.

2. En el centro:

- Nombre del sistema.

3. En el lado superior derecho:

- Fecha.
- Hora.

4. En la parte inferior:

- Funciones válidas con su acción.

➤ Se excluyen las instrucciones de programación: ITER (Iteración) y GO TO (Ir a).

IV.2.2.2 MODELO FÍSICO DE DATOS

El modelo físico de datos se obtiene a partir del modelo conceptual de datos en donde se representan las principales entidades del sistema. Se realiza un análisis de cada una de esas entidades para definir sus propiedades y se elige un nombre, tipo de dato y extensión para cada una de ellas. La figura 4.6 muestra el modelo físico de datos.

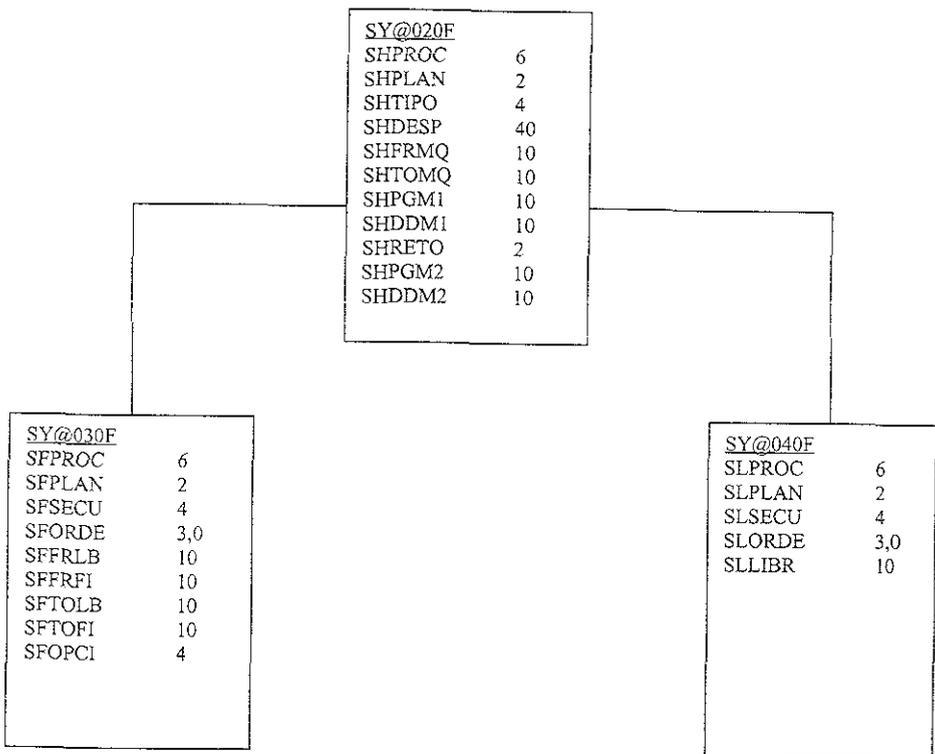


FIGURA 4.6 Modelo físico de datos.

IV.2.2.3 DIAGRAMA DE FLUJO DE DATOS.

En el caso del proceso de mantenimiento el flujo de los datos es de los tres archivos y en ambos sentidos, esto es para la presentación de la información el proceso lee los archivos y en el caso de una alta o modificación el proceso actualiza la información en ellos (fig. 4.7)

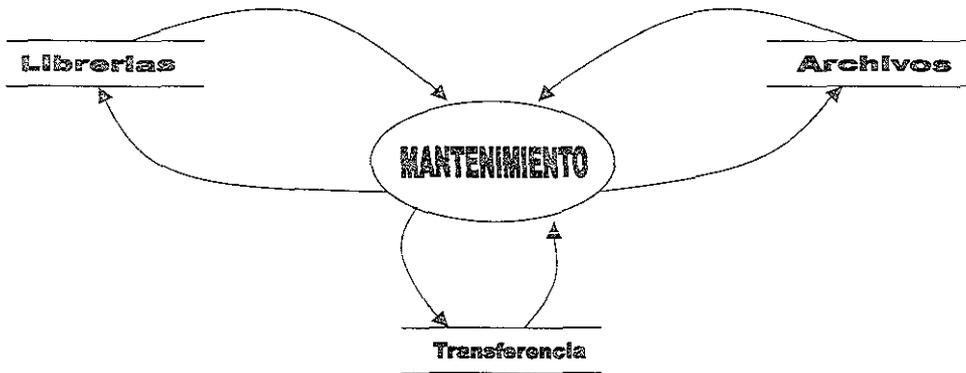


FIGURA 4.7 Flujo de datos en el proceso de mantenimiento.

En el proceso de transferencia el flujo de los datos es en una sola dirección, debido a que solo se toma información de los archivos para la transferencia y llamados de programas pero no se tiene la necesidad de cambiarlos o modificarlos (fig. 4 8).

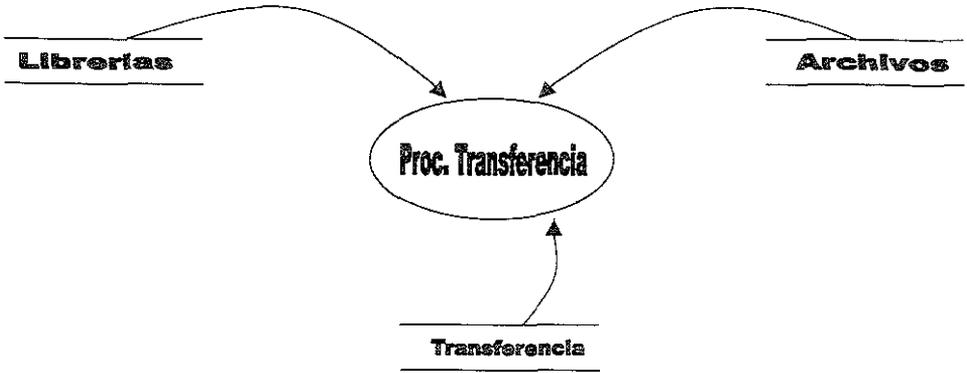


FIG. 4.8 Flujo de datos en el proceso de transferencia.

El flujo en el proceso de copiado de información es en ambos sentidos, esto es porque para evitar capturar los datos necesarios para el funcionamiento de la transferencia en ambas máquinas, la de origen y destino de la transferencia, este proceso lee la información de los archivos de una de ellas y los actualiza en la otra máquina (fig. 4.9).

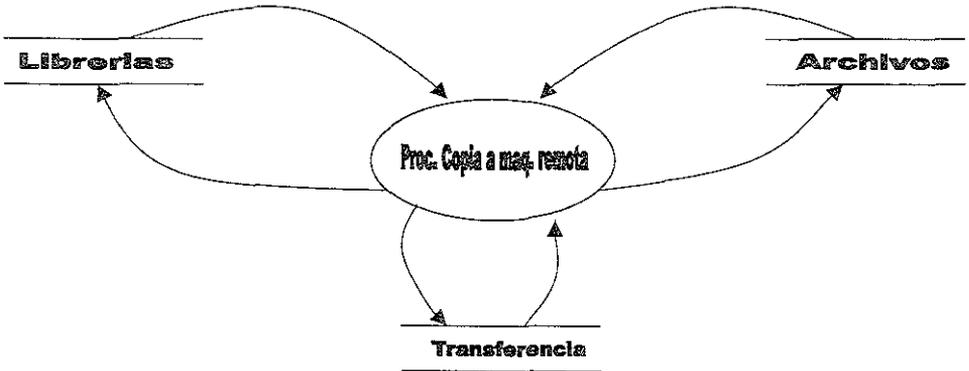


FIG. 4.9 Flujo de datos en el proceso de copia de información.

IV.2.2.4 ESPECIFICACIÓN DE PROCESOS

ESPECIFICACIÓN DE PROCESO

| | |
|--|--|
| SISTEMA Sistema de transferencia de información | FECHA |
| PROCESO A.1.2 Mantenimiento a parámetros del sistema pgm SY@085R | |
| OBJETIVO Realizar Altas, Bajas, Cambios y Copiado a las claves de transferencia | |
| FRECUENCIA Indeterminada, como se requiera crear, modificar o eliminar claves | |
| OPERACIONES BÁSICAS 1. Mostrar pantalla con Claves de transferencia existentes con su descripción y tipo de transf. <ul style="list-style-type: none"> • Si se pulsa 2 e intro, Mostrar información de la clave donde pueden ser modificados. <ul style="list-style-type: none"> • Descripción de la clave. • Dirección de la máquina remota y la local • Existencia de programa remoto para ser llamado en la otra máquina • Nombre del programa a llamar en la máquina remota si existe. • Nombre del DDM que se utiliza para la transmisión. • Nombre de archivo a enviar de una máquina o otra y sus respectivas librerías. • Sentido del envío de los archivos • Nombre de las librerías en la máquina local y remota. • Identificador de librerías locales y remotas • Secuencia de las librerías en la lista correspondiente. • Si se pulsa 3 e intro, se copia la información a una nueva clave. • Si se pulsa 4 e intro, se borra la información correspondiente a esa clave • Si se pulsa 7 e intro se copia la información de esa clave al sistema remoto. 2. Crear una clave de transferencia. Con la función F6 se tiene acceso a la alta de claves | |
| RELACIÓN DE ARCHIVOS - ARCHIVOS (SY@020F) - LIBRERIAS (SY@030F) - TRANSFERENCIAS (SY@040F) | |
| ARCHIVOS DE ENTRADA - SY@020F - SY@030F - SY@040F | ARCHIVOS DE SALIDA - SY@020F - SY@030F - SY@040F |

ESPECIFICACIÓN DE PROCESO

| | |
|--|--|
| SISTEMA Sistema de transferencia de información | FECHA |
| PROCESO A.1.2 Envío de archivos pgm. SY@050CL | |
| OBJETIVO Enviar los archivos especificados de una máquina a otra | |
| FRECUENCIA Indeterminada, como se requiera por un proceso. | |
| OPERACIONES BÁSICAS 1. Envío de archivos especificados a la máquina remota <ul style="list-style-type: none"> • Se obtiene cada uno de los archivos a enviar y recibir y sus respectivas librerías • Se eliminan los archivos de red para cada archivo a enviar. • Se envía cada uno de los archivos al usuario a la máquina remota • Ejecución del proceso de llamado de programas. 2. Monitoreo de mensajes de error. | |
| RELACIÓN DE ARCHIVOS - ARCHIVOS (SY@020F) - LIBRERIAS (SY@030F) - TRANSFERENCIAS (SY@040F) | |
| ARCHIVOS DE ENTRADA - SY@020F - SY@030F - SY@040F | ARCHIVOS DE SALIDA - SY@020F - SY@030F - SY@040F |

ESPECIFICACIÓN DE PROCESO

| | |
|---|---|
| SISTEMA Sistema de transferencia de información | FECHA |
| PROCESO A.1.2 Llamado Remoto de Programas Pgm. SY@054CL | |
| OBJETIVO Someter un job en una máquina AS/400 remota. | |
| FRECUENCIA Indeterminada, como se requiera por una proceso | |
| OPERACIONES BÁSICAS 1 Prepara parámetros para someter programa remoto 2 Selecciona si el proceso es local o remoto y llama al programa enviándole los parámetros. "Selecciona si corre en batch o interactivo. "Adiciona las librerías especificadas para el proceso (INIC o RETO). "Prepara parámetros recibidos para su uso. "Crea área de datos con los parámetros recibidos "Ejecuta proceso de recepción de archivos. "Llama a los programas especificados en la base de datos del sistema (INIC O RETO). "Si el proceso es de inicio y el job debe regresar a la máquina origen ejecuta procesos de envío de archivos y llamado remoto de programas 3 Monitorea mensajes de error. | |
| RELACIÓN DE ARCHIVOS - ARCHIVOS (SY@020F) - LIBRERIAS (SY@030F) - TRANSFERENCIAS (SY@040F) | |
| ARCHIVOS DE ENTRADA - SY@020F - SY@030F - SY@040F - Los especificados en las claves de proceso | ARCHIVOS DE SALIDA - Los especificados en los parámetros de cada clave de proceso |

ESPECIFICACIÓN DE PROCESO

| | |
|--|--|
| SISTEMA | FECHA |
| Sistema de transferencia de información | |
| PROCESO | |
| A.1.2 Proceso de Recepción de Archivos SY@072CL | |
| OBJETIVO | |
| Recibir aArchivos de Red Especificados en la Base de Datos del Sistema | |
| FRECUENCIA | |
| Indeterminada, como se requiera por una proceso | |
| OPERACIONES BÁSICAS | |
| <ol style="list-style-type: none"> 1. Selecciona si adiciona o reemplaza la información en el archivo 2. Si es proceso es local copia el archivo origen al archivo destino 3. Si el proceso es remoto recibe el archivo de red en el archivo de la librería indicado. 4. Reintenta la recepción del archivo de red en 5 ocasiones, si no lo puede recibir envía un mensaje de error al usuario | |
| RELACIÓN DE ARCHIVOS | |
| <ul style="list-style-type: none"> - ARCHIVOS (SY@020F) - LIBRERIAS (SY@030F) - TRANSFERENCIAS (SY@040F) | |
| ARCHIVOS DE ENTRADA | ARCHIVOS DE SALIDA |
| <ul style="list-style-type: none"> - SY@020F - SY@030F - SY@040F - Los especificados en las claves de proceso | <ul style="list-style-type: none"> - Los especificados en los parámetros de cada clave de proceso |

IV.3 DESARROLLO

A continuación se presenta el desarrollo del sistema el cual está definido por el diseño presentado anteriormente. Este desarrollo se presenta en tres partes, la creación de la base de datos, la codificación (de pantallas reportes y programas), y pruebas unitarias.

IV.3.1 CREACIÓN DE LA BASE DE DATOS.

Para la creación de la base de datos se requiere en primer lugar la definición del área de almacenamiento, que es el lugar donde estarán contenidos los archivos. A continuación se generan los fuentes de los archivos y a partir de éstos se generan los mismos.

IV.3.1.1 DEFINICIÓN DEL AREA DE ALMACENAMIENTO.

Debido al almacenamiento de un solo nivel del AS/400, el área de almacenamiento del sistema puede ser una sola librería. El área de almacenamiento de dicho sistema es la librería QGPL en ambas máquinas. En el AS/400 esta librería es utilizada para propósitos generales, y por lo tanto existe en cualquier máquina AS/400. Esta librería se puede adicionar a la lista de librerías de cualquier usuario al iniciar una nueva transmisión con lo cual se asegura que cualquier aplicación que utilice este sistema encontrará todos los objetos necesarios para la transferencia de información.

IV.3.1.2 CREACIÓN DE FUENTES.

Los fuentes de los archivos son listados con instrucciones que reconoce el compilador del AS/400 para crear los archivos tanto físicos como lógicos, estos fuentes contienen el

nombre del archivo que va a ser creado, cada uno de sus campos (sus nombres, longitudes y tipo de datos), sus llaves de acceso, etc.

Estos fuentes son creados a partir del modelo físico de datos obtenido en la etapa del diseño técnico, donde se definieron los archivos con sus campos (nombres y tipo de datos de estos)

```

*-----*
*  NOMBRE      : SY@020F                                *
*  DESCRIPCION: PARAMETROS DE PROCESOS REMOTOS HEADER  *
*  *
*  (PROCESOS QUE CUENTAN CON VARIAS PARTES LAS        *
*  CUALES SE PUEDEN EJECUTAR EN MAS DE UNA MÁQUINA)*
*  *
*  EJEMPLO.                                           *
*  *-----P R O C E S O-----*
*  INICIO  ----> PARTE2  ----> FINAL
*  MAQUINA1      MAQUINA2      MAQUINA1
*  *
*  ESTE ARCHIVO CONTIENE TODOS LOS PARAMETROS QUE    *
*  SE REQUIEREN PARA EJECUTAR EL PROCESO REMOTO    *
*  NOMBRES DE LAS MÁQUINAS, ARCHIVOS DDM's, ETC.   *
*  *
*  TAMBIEN SE PUEDE ESPECIFICAR QUE LAS PARTES DEL *
*  PROCESO SE EJECUTAN EN LA MISMA MÁQUINA.       *
*  (CLAVES DE ENVÍO)
*  REALIZO    . EDUARDO MARTÍNEZ SALINAS
*  INSTALACION.
*  FECHA      :
*-----*
A      R SY@020R
A      SHPROC      6A      TEXT('PROCESO')
A      COLHDG('PROCESO')
A      SHPLAN      2A      TEXT('PLANTA')
A      COLHDG('PLANTA')
A      SHTIPO      4A      TEXT('TIPO DE PROCESO')
A      COLHDG('TPO.' 'PROC.')
A      SHDESP      40A     TEXT('DESCRIPCION DEL PROCESO')
A      COLHDG('DESC. DEL PROCESO')
A      SHFRMQ      10A     TEXT('FROM-MÁQUINA')
A      COLHDG('FROM' 'MÁQUINA')
A      SHTOMQ      10A     TEXT('TO-MÁQUINA')
A      COLHDG('TO' 'MÁQUINA')
A      SHPGM1      10A     TEXT('PROGRAMA REMOTO')
A      COLHDG('PROGRAMA' 'REMOTO')
A      SHDDM1      10A     TEXT('DDM ENLACE REMOTO')
A      COLHDG('DDM' 'ENLACE' 'REMOTO')
A      SHRETO      2A      TEXT('EXISTE PROCESO DE RETORNO')
A      COLHDG('PROC.' 'RETORNO')
A      SHPGM2      10A     TEXT('PROGRAMA REMOTO RETORNO')
A      COLHDG('PGM.' 'REMOTO' 'RETORNO')
A      SHDDM2      10A     TEXT('DDM DE ENLACE RETORNO')
A      K SHPROC
A      K SHPLAN

```

```

*-----*
* NOMBRE      : SY@030F *
* DESCRIPCION: PARAMETROS DE PROCESOS REMOTOS (SEND FILE) *
* *
* (PROCESOS QUE CUENTAN CON VARIAS PARTES LAS *
* CUALES SE PUEDEN EJECUTAR EN MAS DE UNA MÁQUINA)*
* *
* EJEMPLO. *
* **-----P R O C E S O-----** *
* INICIO  ----> PARTE2  ----> FINAL *
* MAQUINA1      MAQUINA2      MAQUINA1 *
* *
* ESTE ARCHIVO CONTIENE TODOS LOS PARAMETROS QUE *
* SE REQUIEREN PARA EJECUTAR EL PROCESO REMOTO *
* ARCHIVOS A ENVIAR A LOS PROCESOS REMOTOS *
* (ARCHIVOS A ENVIAR) *
* *
* REALIZO  - EDUARDO MARTÍNEZ SALINAS *
* INSTALACION *
* FECHA *
*-----*

```

```

A      R SY@030R
A      SFPROC      6A      TEXT('PROCESO')
A      COLHDG('PROCESO')
A      SFPLAN      2A      TEXT('PLANTA')
A      COLHDG('PLANTA')
A      SFSECU      4A      TEXT('SECUENCIA DE PROCESO')
A      COLHDG('SEC.' 'PROC')
A      SFORDE      3 0      TEXT('ORDEN DE ENVIO/RECEPCION')
A      COLHDG('ORDEN' 'ENV.REC')
A      SFPRLB      10A     TEXT('FROM LIBRERIA')
A      COLHDG('FROM LIBRERIA')
A      SFFRFI      10A     TEXT('FROM ARCHIVO')
A      COLHDG('FROM' 'ARCHIVO')
A      SFTOLB      10A     TEXT('TO LIBRERIA')
A      COLHDG('TO LIBRERIA')
A      SFTOFI      10A     TEXT('TO ARCHIVO')
A      COLHDG('TO' 'ARCHIVO')
A      SFOPCI      4A      TEXT('ADICIONAR O REMPLAZAR REGIS')
A      COLHDG('*ADD' '*REP' 'REGTOS')
A*
A      K SFPROC
A      K SFPLAN
A      K SFSECU
A      K SFORDE

```

```

*-----*
* NOMBRE      : SY@040F *
* DESCRIPCION: PARAMETROS DE PROCESOS REMOTOS LISTA-LIBRERIAS *
* *
* (PROCESOS QUE CUENTAN CON VARIAS PARTES LAS *
* CUALES SE PUEDEN EJECUTAR EN MAS DE UNA MÁQUINA)*
* *
* EJEMPLO. *
* **-----P R O C E S O-----** *
* INICIO ----> PARTE2 ----> FINAL *
* MAQUINAL1 MAQUINA2 MAQUINAL *
* *
* ESTE ARCHIVO CONTIENE TODOS LOS PARAMETROS QUE *
* SE REQUIEREN PARA EJECUTAR EL PROCESO REMOTO *
* LISTA DE LIBRERIAS *
* *
* TAMBIEN SE PUEDE ESPECIFICAR QUE LAS PARTES DEL *
* PROCESO SE EJECUTAN EN LA MISMA MÁQUINA. *
* (LISTA DE LIBRERÍAS) *
* REALIZO      : EDUARDO MARTÍNEZ SALINAS *
* INSTALACION: *
* FECHA *
*-----*
A      R SY@040R
A      SLPROC      6A      TEXT('PROCESO')
A      COLHDG('PROCESO')
A      SLPLAN      2A      TEXT('PLANTA')
A      COLHDG('PLANTA')
A      SLSECU      4A      TEXT('SECUENCIA DE PROCESO')
A      COLHDG('SEC.' 'PROC')
A      SLORDE      3 0      TEXT('ORDEN DE REPLAZO')
A      COLHDG('ORDEN' 'REPLAZO')
A      SLLIBR      10A     TEXT('LIBRERIA')
A      COLHDG('LIBRERIA')
A*
A      K SLPROC
A      K SLPLAN
A      K SLSECU
A      K SLORDE

```

IV.3.1.3 GENERACIÓN DE ARCHIVOS.

Para la generación de los archivos solo se necesita compilar estos fuentes e indicar en que librería serán creados, así como darle la autoridad necesaria sobre ellos a los usuarios. Esta autoridad debe permitir la actualización a los desarrolladores y la lectura a los usuarios en general.

IV.3.2 CODIFICACIÓN.

Para la creación de pantallas, formatos de reportes y programas se necesita codificar las instrucciones necesarias en un archivo fuente que reconoce el compilador del AS/400 igual que para los archivos. Las pantallas y los formatos de reportes son los primeros que se deben de crear, debido a que los programas los utilizan y por lo tanto son necesarios para la creación de éstos. A continuación se presentan tanto la codificación de las pantallas y formatos de reportes como de los programas.

IV.3.2.1 CODIFICACIÓN DE PANTALLAS Y REPORTES.

Para la codificación de las pantallas se utiliza la utilería SDA, la cual presenta una pantalla a la que se pueden agregar los campos necesarios, definir sus tipos de datos, moverlos, agregarles formatos de edición, color; definir si son de entrada, salida o ambas; protegerlos, adicionar valores permitidos, etc., generándose el código automáticamente.

De la misma manera para la codificación de los reportes se utiliza la utilería RLU, la cual también permite manipular los campos, subrayarlos, remarcarlos, etc., dentro de una pantalla que muestra las posiciones que cada campo ocupará en el reporte, y genera el código automáticamente.

Estas dos utilerías facilitan la codificación de las pantallas y reportes debido a la gran cantidad de palabras clave que se utilizan para los atributos de los campos, las cuales, gracias a las utilerías, no se deben aprender de memoria. A continuación se presenta la codificación de los fuentes tanto de las pantallas como de los reportes.

PANTALLA SY@085P , MANTENIMIENTO A PARAMETROS DEL SISTEMA

```

A*%EC
A
A          DSPSIZ(24 80 *DS3)
A          PRINT
A          CF03(03 'F3-Salir')
A          CF12(12 'F12-Cancelar')
A
R SY@085HD
A
A          1 2'SY@085P'
A          1 23'GRUPO COLGATE PALMOLIVE S.A. DE -
A          C.V.'
A          DSPATR(HI)
A          1 72TIME
A          2 2USER
A          2 18'Mant. a las Bases de Datos de Pro-
A          cesos Remotos'
A          2 72DATE
A          EDTCDE(Y)
A          4 2'
A
A          '
A          DSPATR(UL)
A          SFL
R SY@085S1
A          SFOPC          1A I 9 3
A          SHPROC         6A O 9 7
A          SHPLAN         2A O 9 16
A          SHTIPO         4A O 9 21
A          SHDESP         40A O 9 27
R SY@085C1
A          SFLCTL(SY@085S1)
A          SFLSIZ(0012)
A          SFLPAG(0011)
A          CF06(06 'F6-Alta')
A          OVERLAY
A          SFLCSRRRN(&RRCS1)
A          SFLDSP
A          SFLDSPCTL
A          SFLCLR
A          SFLEND(*MORE)
A          26
A N25
A 25
A 27
A          NRR1          5S 0H
A          NRCD1         4S 0H
A          RRCS1         5S 0H
A          SFLRCDNER(CURSOR)
A          5 2'Opciones:'
A          DSPATR(HI)
A          5 14'2=Cambio'
A          5 25'3=Copiar'
A          5 36'4=Eliminar'
A          5 49'7=Copia Sist. Remotos'
A          7 2'Opc'
A          DSPATR(UL)
A          DSPATR(HI)
A          7 7'Proceso'
A          DSPATR(UL)
A          DSPATR(HI)
A          7 16'Pta'
A          DSPATR(UL)
A          DSPATR(HI)
A          7 21'Tipo'
A          DSPATR(UL)
A          DSPATR(HI)
A          7 27'Descripción
A          '
A          DSPATR(UL)

```

```

A                                     DSPATR(HI)
A      CTPROC                        6A B 8 7CHANGE(35)
A      CTPLAN                        2A B 8 16CHANGE(35)
R SY@085P1
A                                     OVERLAY
A                                     5 68'Modo:'
A                                     DSPATR(HI)
A      PADMOV                        6A O 5 74
A                                     7 2'Proceso : '
A                                     DSPATR(HI)
A      SHPROC                        6A B 7 12
A                                     DSPATR(RI)
A 51                                     DSPATR(PC)
A 51                                     8 2'Planta. : '
A                                     DSPATR(HI)
A      SHPLAN                        2A B 8 12
A 52                                     DSPATR(RI)
A 52                                     DSPATR(PC)
R SY@085P2
A                                     OVERLAY
A                                     5 68'Modo:'
A                                     DSPATR(HI)
A      PADMOV                        6A O 5 74
A                                     7 2'Proceso : '
A                                     DSPATR(HI)
A      SHPROC                        6A O 7 12
A                                     8 2'Planta. : '
A                                     DSPATR(HI)
A      SHPLAN                        2A O 8 12
A                                     10 2'Tipo de Proceso . . . : '
A      SHTIPO                       4A B 10 30
A 51                                     DSPATR(RI)
A 51                                     DSPATR(PC)
A 34                                     DSPATR(PR)
A                                     11 2'Descripción del Proceso. : '
A      SHDESP                       40A B 11 30
A                                     DSPATR(RI)
A 52                                     DSPATR(PC)
A 52                                     DSPATR(PR)
A 34                                     12 2'From-máquina . . . . . : '
A      SHFRMQ                       10A B 12 30
A                                     DSPATR(RI)
A 53                                     DSPATR(PC)
A 53                                     DSPATR(PR)
A 34                                     13 2'To-máquina . . . . . : '
A      SHTOMQ                       10A B 13 30
A                                     DSPATR(PR)
A 34                                     DSPATR(RI)
A 54                                     DSPATR(PC)
A 54                                     14 2'Programa Remoto. . . . . : '
A      SHPGM1                       10A B 14 30
A                                     DSPATR(PR)
A                                     15 2'DDM Enlace Remoto. . . . . : '
A      SHDDM1                       10A B 15 30
A                                     DSPATR(PR)
A                                     16 2'Existe Proceso de Retorno: '
A      SHRETO                       2A B 16 30
A                                     DSPATR(RI)
A 55                                     DSPATR(PC)
A 55                                     DSPATR(PR)
A 34                                     17 2'Programa Remoto Retorno. : '
A      SHPGM2                       10A B 17 30
A 34                                     DSPATR(PR)

```

```

A          18 2'DDM de Enlace de Retorno : '
A          A          SHDDM2          10A B 18 30
A 34          DSPATR(PR)
A          R SY@085S2          SFL
A          S2SECU          4A B 8 2
A          DSPATR(RI)
A 51          DSPATR(PC)
A 51          DSPATR(PR)
A 34          S2ORDE          3Y 0B 8 9
A          DSPATR(RI)
A 52          DSPATR(PC)
A 52          EDTCDE(Z)
A          DSPATR(PR)
A 34          S2FRLB          10A B 8 18
A          DSPATR(RI)
A 53          DSPATR(PC)
A 53          DSPATR(PR)
A 34          S2FRFI          10A B 8 29
A          DSPATR(RI)
A 54          DSPATR(PC)
A 54          DSPATR(PR)
A          S2TOLB          10A B 8 44
A          DSPATR(RI)
A 55          DSPATR(PC)
A 55          DSPATR(PR)
A 34          S2TOFI          10A B 8 55
A          DSPATR(RI)
A 56          DSPATR(PC)
A 56          DSPATR(PR)
A 34          S2OPCI          4A B 8 71
A          DSPATR(RI)
A 57          DSPATR(PC)
A 57          DSPATR(PR)
A 34          R SY@085C2          SFLCTL(SY@085S2)
A          SFLSIZ(0100)
A          SFLPAG(0012)
A          OVERLAY
A          SFLDSP
A 26          SFLDSPCTL
A N25          SFLINZ
A 25          SFLEND(*MORE)
A 27          NRR2          5S 0H
A          5 2'Proceso : '
A          DSPATR(HI)
A          SHPROC          6A O 5 12
A          5 23'Planta: '
A          DSPATR(HI)
A          SHPLAN          2A O 5 31
A          5 68'Modo: '
A          DSPATR(HI)
A          PADMOV          6A O 5 74
A          7 2'Sec. '
A          DSPATR(UL)
A          DSPATR(HI)
A          7 8'Orden'
A          DSPATR(UL)
A          DSPATR(HI)
A          7 18'From Lib. /File '
A          DSPATR(UL)
A          DSPATR(HI)
A          7 44'To Library/File '
A          DSPATR(UL)
A          DSPATR(HI)

```

| | | | | | | |
|-------|------------|-----|----|----|----|------------------|
| A | | | | 7 | 70 | 'Add/Rep' |
| A | | | | | | DSPATR(UL) |
| A | | | | | | DSPATR(HI) |
| A | R SY@085S3 | | | | | SFL |
| A | S3SECU | 4A | B | 8 | 2 | |
| A 51 | | | | | | DSPATR(RI) |
| A 51 | | | | | | DSPATR(PC) |
| A 34 | | | | | | DSPATR(PR) |
| A | S3ORDE | 3Y | 0B | 8 | 9 | |
| A 52 | | | | | | DSPATR(RI) |
| A 52 | | | | | | DSPATR(PC) |
| A | | | | | | EDTCDE(Z) |
| A 34 | | | | | | DSPATR(PR) |
| A | S3LIBR | 10A | B | 8 | 15 | |
| A 53 | | | | | | DSPATR(RI) |
| A 53 | | | | | | DSPATR(PC) |
| A 34 | | | | | | DSPATR(PR) |
| A | R SY@085C3 | | | | | SFLCTL(SY@085S3) |
| A | | | | | | OVERLAY |
| A 26 | | | | | | SFLDSP |
| A N25 | | | | | | SFLDSPCTL |
| A 25 | | | | | | SFLINZ |
| A 27 | | | | | | SFLEND(*MORE) |
| A | | | | | | SFLSZ(0100) |
| A | | | | | | SFLPAG(0012) |
| A | NRR3 | 5S | 0H | | | |
| A | | | | 5 | 2 | 'Proceso :' |
| A | | | | | | DSPATR(HI) |
| A | SHPROC | 6A | O | 5 | 12 | |
| A | | | | 5 | 23 | 'Planta: ' |
| A | | | | | | DSPATR(HI) |
| A | SHPLAN | 2A | O | 5 | 31 | |
| A | | | | 5 | 68 | 'Modo: ' |
| A | | | | | | DSPATR(HI) |
| A | PADMOV | 6A | O | 5 | 74 | |
| A | | | | 7 | 2 | 'Sec. ' |
| A | | | | | | DSPATR(HI) |
| A | | | | | | DSPATR(UL) |
| A | | | | 7 | 8 | 'Orden ' |
| A | | | | | | DSPATR(HI) |
| A | | | | | | DSPATR(UL) |
| A | | | | 7 | 15 | 'Libreria ' |
| A | | | | | | DSPATR(UL) |
| A | | | | | | DSPATR(HI) |
| A | R SY@085M1 | | | | | |
| A | | | | | | OVERLAY |
| A | | | | 21 | 2 | ' |
| A | | | | | | |
| A | | | | | | DSPATR(UL) |
| A | | | | 23 | 2 | 'F3-Salir' |
| A | | | | 23 | 12 | 'F6-Alta' |
| A | | | | 23 | 21 | 'F12-Cancelar' |
| A | PAMSG | 78A | O | 24 | 2 | DSPATR(HI) |
| A | R SY@085M2 | | | | | |
| A | | | | | | OVERLAY |
| A | | | | 21 | 2 | ' |
| A | | | | | | |
| A | | | | | | DSPATR(UL) |
| A | | | | 23 | 2 | 'F3-Salir' |
| A | | | | 23 | 12 | 'F12-Cancelar' |
| A | PAMSG | 78A | O | 24 | 2 | DSPATR(HI) |

PANTALLA SY@080P , REPORTE DE PARAMETROS DEL SISTEMA

```

A**%EC
A
A
A      R SY@080P1
A
A      DSPSIZ(24 80 *DS3)
A
A      CF03(03 'Salida del modulo')
A      CF05(05 'Limpiar pantalla (refresh)-
A      ')
A      CF12(12 'Salir de la pantalla')
A
A      1 2'SY@020P1'
A      DES$$$      50A O 1 16DSPATR(HI)
A      1 72DATE
A      EDTCDE(Y)
A      2 2USER
A      2 17'REPORTE DE PARAMETROS PARA PROCESO-
A      S RENOTOS STD.'
A      2 72TIME
A      6 2'Carga de Parámetros Para Obtener R-
A      eporte.'
A
A      10 13'Proceso . . . . . '
A      PAMSC1      77A O 23 2DSPATR(HI)
A      24 2'F3-Salir'
A      24 12'F5-Limpiar'
A      24 24'F12-Cancelar'
A      12 13'Planta . . . . . '
A
A      PAPROC      6A E 10 30
A      PAPTA      2A E 12 30
A      20 2'***'
A      DSPATR(HI)
A      20 5'Para seleccionar todos los registr-
A      os deje los campos en BLANCO'

```


A ' P R O C E S O D E I N I C I O '
A H I G H L I G H T
A U N D E R L I N E
A S P A C E B (0 0 2)

A*%*****
A*%SS
A*%CL 002

A*%*****
A R S Y @ 0 8 0 H 3

A*%*****
A*%RI 0000

A*%*****
A 4
A 'PROGRAMA:'
A S P A C E B (0 0 2)
A W K P G M 1 0 A O - 1
A - 7
A 'DDM ENLACE:'
A W K O D M 1 0 A O - 1
A 61
A 'ARCHIVOS A ENVIAR '
A 86
A 'FROM ARCHIVO'
A 103
A 'FROM BIBLIOTECA'
A 124
A 'TO ARCHIVO'
A 139
A 'TO BIBLIOTECA'

A*%*****
A*%SS
A*%CL 002

A*%*****
A R S Y @ 0 8 0 D 3

A*%*****
A*%RI 0000

A*%*****
A S P A C E B (0 0 1)
A S F F R F I 1 0 A O 8 6
A S F F R L B 1 0 A O 1 0 3
A S F T O F I 1 0 A O 1 2 4
A S F T O L E 1 0 A O 1 3 9

A*%*****
A*%SS

A*%*****
A R S Y @ 0 8 0 H 4

A*%*****
A*%RI 0000

A*%*****
A S P A C E B (0 0 1)
A 5 2

A
A
A
A

'BIBLIOTECAS PARA REEMPLAZO:'
SPACEB(001)
86
'BIBLIOTECA'

A*%*****
A*%SS
A*%CL 001

A*%*****
A R SY@080D4

A*%*****
A*%RI 00000

A*%*****
A SPACEB(001)
A SLLIBR 10A O 86

A*%*****
A*%SS

A*%*****
A R SY@080H5

A*%*****
A*%RI 00000

A*%*****
A SPACEB(001)
A 43
A ' P R O C E S O D E R E T O R N

-
A O'
A HIGHLIGHT
A UNDERLINE
A SPACEB(002)

A*%*****
A*%SS
A*%CL 002

A*%*****
A R SY@080L1

A*%*****
A*%RI 00000

A*%*****
A SPACEB(001)
A SPACEA(002)
A 4

- A -----

- A -----

- A -----

- A -----

A*%*****

```

A*%SS
A*%CL 002

A*%*****
A          R SY@080E1

A*%*****
A*%RI 0000

A*%*****
A          SPACEB(C01)
A          80
A          'NO SE ENCONTRARON REGISTROS POR'
A          +1
A          'PROCESAR'

A*%*****
A*%SS
A*%CP+999CRTPTF
A*%CP-   FILE (GPLCP/SY@080T)
A*%CP+   DEVTYPE (*SCS)
A*%CP-   PAGESIZE (75 198)
A*%CP+   LPI (6)
A*%CP+   CPI (15)
A*%CP+   OVRFLW (70)
A*%CP+   DUPLEX (*TUMBLE)
A*%CP+   FORMTYPE (@CLSO)
A*%CP+   MAXRCDS (*NOWAX)
A*%CP+   HOLD (*YES)
A*%CP    LVLCHK (*NO)

A*%*****

```

IV.3.2.2 CODIFICACIÓN DE PROGRAMAS.

Mientras que los archivos físicos se utilizan para guardar información y los reportes y pantallas para presentarla, los programas son utilizados para la manipulación de la misma. Para la codificación de los programas se utiliza un editor. A continuación se presentan la codificación de los programas tal cual las presenta dicho editor.

El programa SY@050CL realiza las siguientes acciones.

- Recupera el usuario que está utilizando la aplicación y su lista de librerías.
- Adiciona la librería del sistema (QGPL), a la lista de librerías.
- Llama al programa SY@050R que envía los archivos.
- Restaura la lista de librerías anterior

```

/*-----*/
/*
/* NOMBRE : SY@050CL
/* DESCRIPCION: EJECUTA PROCESOS REMOTOS (INICIO DEL PROCESO)
/*
/* Llamado por:
/* REALIZO: EDUARDO MARTÍNEZ SALINAS
/* FECHA.
/*
/* NOTA: PARA EJECUTAR ESTE PROGRAMA EL PARÁMETRO &PAPARM NO PUEDE
/* CONTENER CARACTERES (BLANCO, O ESPECIALES) SI LOS CONTIENE*/
/* EL PROCESO PUEDE TERNINAR ANORMALMENTE.
/*
/*-----*/
PGM PARM(&SHPROC &SHPLAN &MODO &PAPARM)
DCL VAR(&SHPROC) TYPE(*CHAR) LEN(6)
DCL VAR(&SHPLAN) TYPE(*CHAR) LEN(2)
DCL VAR(&MODO) TYPE(*CHAR) LEN(1)
DCL VAR(&PAPARM) TYPE(*CHAR) LEN(256)
DCL VAR(&USER10) TYPE(*CHAR) LEN(10)
DCL VAR(&PAERRO) TYPE(*CHAR) LEN(2) VALUE('NO')
DCL VAR(&INICIO) TYPE(*DEC) LEN(4 0) VALUE(1)
DCL VAR(&USRLIB) TYPE(*CHAR) LEN(275) VALUE(' ')
/*-----*/
/* RECUPERA QUE USUARIO ESTA EJECUTANDO EL PROCESO Y LISTA DE LIB. */
/*-----*/
RTVJOBA USER(&USER10) USRLIBL(&USRLIB)
/*-----*/
/* PUNTA A ARCHIVOS DE PARAMETROS
/*-----*/
ADDLIBLE LIB(GPLCP) POSITION(*LAST)
MONMSG MSGID(CPF0000 CPF9999)
OVRDBF FILE(SY@020F) TOFILE(GPLCP/SY@020F)
MONMSG MSGID(CPF0000)
OVRDBF FILE(SY@030F) TOFILE(GPLCP/SY@030F)
MONMSG MSGID(CPF0000)
/*-----*/
/* ENVIA EL ARCHIVO A LA MÁQUINA INDICADA EN LOS PARAMETROS
/*-----*/
CALL SY@050R PARM(&SHPROC &SHPLAN &MODO &PAPARM &PAERRO)
IF COND(&PAERRO='SI') THEN(DO)
    SNDUSRMSG MSG('No se Encontraron los Parametros para +
    ejecutar el proceso remoto ' *CAT &SHPROC +
    *CAT ' por favor capture los parametros y +
    ejecute este proceso nuevamente') +
    MSGTYPE(*INFO) TCUSR(&USER10)
ENDDO
/*-----*/
/* RESTAURA LISTA DE LIBRERIAS ANTERIOR.
/*-----*/
ADD. CHGLIBL LIBL(*NONE)
IF COND(%SST(&USRLIB &INICIO 11) =EQ ' ') +
    THEN(GOTO CMDLBL(FINAL))
ADDLIBLE LIB(%SST(&USRLIB &INICIO 11)) POSITION(*LAST)
MONMSG MSGID(CPF0000)
CHGVAR VAR(&INICIO) VALUE(&INICIO + 11)
IF COND(&INICIO = 276) THEN(GOTO CMDLBL(FINAL))
GOTO CMDLBL(ADD)
FINAL: RETURN
ENDPGM

```

El programa SY@050R realiza las siguientes acciones.

- Llama al proceso de eliminación de archivos de red (programa SY@056CL) para cada archivo que se va a enviar.
- Llama al proceso de envío de archivos (programa SY@052CL) para cada archivo.
- Llama al proceso que ejecuta el trabajo remoto (programa SY@054CL)

```

*-----*
* PROGRAMA      : SY@050R                               *
* DESCRPCIÓN    : TRAE PARAMETROS PARA LA EJECUCION DE PROCESOS REMOTOS. *
*                                                       *
*                                                       *
*                                                       *
* REALIZÓ...   : EDUARDO MARTÍNEZ SALINAS                *
* INSTALACIÓN   :                                         *
* FECHA.....  :                                         *
*                                                       *
*-----*
FSY@020F IF E      K      DISK
FSY@030F IF E      K      DISK
C*-----*
C*      RECEPCION DE PARAMETROS                          *
C*-----*
C          *ENTRY      PLIST
C                PARM          SHPROC          PROCESO
C                PARM          SHPLAN          PLANTA
C                PARM          MODO      1      B=BATCH; I=ITERACT.
C                PARM          PAPARM256      PARÁMETRO REMOTO
C                PARM          PAERRO  2      BANDERA DE ERROR
C*-----*
C*      PAR052.- PARAMETROS PARA ENVIO DE ARCHIVOS      *
C*-----*
C          PAR052      PLIST
C                PARM          SHTOMQ          TO MÁQUINA
C                PARM          SFFRLB          FROM LIBRERIA
C                PARM          SFFRFI          FROM ARCHIVO
C*-----*
C*      PAR054.- PARAMETROS PARA EJECUCION DEL JOB REMOTO *
C*-----*
C          PAR054      PLIST
C                PARM          SHPROC          PROCESO
C                PARM          SHPLAN          PLANTA
C                PARM          MODO          B=BATCH; I=ITERACT
C                PARM          SECPRO          SECUENCIA DE PROCESC
C                PARM          SHTIPO          TIPO DE PROCESO
C                PARM          SHTOMQ          TO MÁQUINA
C                PARM          SHDDMI          ARCHIVO DDM
C                PARM          PAPARM          PARÁMETRO DEL JOB
C*-----*
C*      PAR056 - PARAMETROS PARA ELIMINACION DE ARCHIVOS REMOTOS. *
C*-----*
C          PAR056      PLIST

```

```

C          PARM          SHTOMQ          TO MÁQUINA
C          PARM          SHDDM1          ARCHIVO DDM
C          PARM          SPFRLB          FROM LIBRERIA
C          PARM          SPFRFI          FROM ARCHIVO
C*-----*
C*  KEYPA1 - LLAVE PARA ACCESAR ARCHIVO DE PARAMETROS DEL JOB *
C*-----*
C          KEYPA1      KLIST
C          KFLD          SHPROC
C          KFLD          SHPLAN
C*-----*
C*  KEYPA2 - LLAVE PARA ACCESAR ARCHIVO DE PARAMETROS DEL JOB *
C*-----*
C          KEYPA2      KLIST
C          KFLD          SHPROC
C          KFLD          SHPLAN
C          KFLD          SECPCRO
C*
C          MOVE 'NO'      PAERRO
C          MOVE 'INIC'    SECPCRO  4
C*
C          KEYPA1      CHAINSY@020F          36
C          *IN36      IFEQ '1'
C          MOVE 'SI'      PAERRO
C          ELSE
C          EXSR SR0001
C          ENDIF
C          MOVE '1'      *INLR
C*-----
C*  SR0001.- PROCESA EJECUCION DE JOB REMOTO
C*-----
C          SR0001      BEGSR
C*-----
C*  ENVIA LOS ARCHIVOS AL JOB (SOLO SI ES JOB REMOTO)
C*-----
C          SHTIPO      IFEQ '*RMT'
C*
C*  ENVIA ARCHIVOS DE DATOS DEL JOB
C          KEYPA2      SETLLSY@030F
C          KEYPA2      READESY@030F          36
C          *IN36      DOWEQ '0'
C          CALL 'SY@056CL' PAR056
C          CALL 'SY@052CL' PAR052
C          KEYPA2      READESY@030F          36
C          ENDDO
C*
C          ENDIF
C*-----
C*  EJECUTA JOB REMOTO
C*-----
C          CALL 'SY@054CL' PAR054
C          ENDSR

```

El programa SY@052CL realiza las siguientes acciones:

- Envía el archivo a la máquina indicada y monitorea errores en el envío del mismo.

```

/*-----*/
/*
/* NOMBRE : SY@052CL
/* DESCRIPCION: EJECUTA PROCESOS REMOTOS
/* ENVIO DE ARCHIVOS A JOB REMOTO
/*
/* LLAMADO POR:
/* REALIZO: EDUARDO MARTÍNEZ SALINAS
/* FECHA:
/*
/*-----*/
PGM PARM(&SHTOMQ &SFFRLB &SFFRFI)
DCL VAR(&SHTOMQ) TYPE(*CHAR) LEN(10)
DCL VAR(&SFFRLB) TYPE(*CHAR) LEN(10)
DCL VAR(&SFFRFI) TYPE(*CHAR) LEN(10)

DCL VAR(&CONTAR) TYPE(*DEC) LEN(3 0) VALUE(0)
DCL VAR(&USER10) TYPE(*CHAR) LEN(10)
DCL VAR(&USER8) TYPE(*CHAR) LEN(8)
DCL VAR(&REPLY) TYPE(*CHAR) LEN(1) VALUE(' ')
/*-----*/
/* RECUPERA QUE USUARIO ESTA EJECUTANDO EL PROCESO
/*-----*/
RTVJOBA USER(&USER10)
CHGVAR VAR(&USER8) VALUE(&USER10)
/*-----*/
/* ENVIA EL ARCHIVO A LA MÁQUINA INDICADA EN LOS PARAMETROS
/*-----*/
ENVIA: CHGVAR VAR(&CONTAR) VALUE(0)
SNDNETF FILE(&SFFRLB/&SFFRFI) TOUSRID((&USER8 &SHTOMQ)) +
PTY(*HIGH)
MONMSG MSGID(CPF0000 CPF9999) EXEC(DO)

CHGVAR VAR(&CONTAR) VALUE(&CONTAR - 1)
IF COND(&CONTAR>5) THEN(DO)
SNDUSRMSG MSG('El programa SY@052CL ha intentado por +
mas de 5 ocaciones enviar el archivo ' +
*CAT &SFFRLB *CAT ' / ' *CAT &SFFRFI *CAT -
' a ' *CAT &SHTOMQ *CAT ' , pero NO ha +
podido debido a algun error, (FAVOR DE +
Notificar al coordinador de MAPICS') +
TOUSR(&USER10) MSGRPY(&REPLY)

IF COND(&REPLY='C') THEN(GOTO CMDLBL(FINAL))
ENDDO

GOTO ENVIA
ENDDO

FINAL: RETURN
ENDPGM

```

El programa SY@054CL realiza las siguientes acciones:

- Acondiciona parámetro para el llamado del programa de ejecución de procesos remotos (programa SY@055R).

- Selecciona si el proceso correrá en la máquina local o remota.
- Si la clave de tipo de proceso es local (*LCL) llama al programa SY@060CL.
- Si la clave de tipo de proceso es remota (*RMT) somete al programa SY@060CL en la máquina remota.

```

/*-----*/
/*                                          */
/*  NOMBRE : SY@054CL                      */
/*  DESCRIPCION: EJECUTA PROCESOS REMOTOS  */
/*                                          */
/*  LLAMADO POR:                            */
/*  REALIZO: EDURADO MARTÍNEZ SALINAS     */
/*  FECHA:                                   */
/*-----*/
PGM PARM(&SHPROC &SHPLAN &MODO &SECPRO &SHTIPO &SHTOMQ &SHDDMI &PAPARM)
DCL  VAR(&SHPROC)  TYPE(*CHAR)  LEN(6)
DCL  VAR(&SHPLAN) TYPE(*CHAR)  LEN(2)
DCL  VAR(&MODO)   TYPE(*CHAR)  LEN(1)
DCL  VAR(&SECPRO) TYPE(*CHAR)  LEN(4)
DCL  VAR(&SHTIPO) TYPE(*CHAR)  LEN(4)
DCL  VAR(&SHTOMQ) TYPE(*CHAR)  LEN(10)
DCL  VAR(&SHDDMI) TYPE(*CHAR)  LEN(10)
DCL  VAR(&PAPARM) TYPE(*CHAR)  LEN(256)
DCL  VAR(&PARAME) TYPE(*CHAR)  LEN(269)

DCL  VAR(&USER10) TYPE(*CHAR)  LEN(10)
DCL  VAR(&REPLY)  TYPE(*CHAR)  LEN(1)  VALUE(' ')
DCL  VAR(&CALL)   TYPE(*CHAR)  LEN(307) VALUE(' ')
/*-----*/
/*  RECUPERA QUE USUARIO ESTA EJECUTANDO EL PROCESO  */
/*-----*/
RTVJOBA    USER(&USER10)

/*-----*/
/*  ACONDICIONA PARÁMETRO PARA SBMRMTCMD             */
/*-----*/
CALL SY@055R PARM(&PAPARM '1')

/*-----*/
/*  ARMA PARÁMETRO A ENVIAR                          */
/*-----*/
CHGVAR VAR(&PARAME) +
      VALUE(&SHPROC *CAT &SHPLAN *CAT &MODO *CAT &SECPRO *CAT &PAPARM)

/*-----*/
/*  SI EL PROCESO ES LOCAL                            */
/*-----*/
IF COND(&SHTIPO='*LCL') THEN(DO)

LOCAL:
  IF COND(&MODO = 'B', THEN(DO)
    SBMJOB      CMD(CALL PGM(SY@060CL) PARM(&PARAME))
  ENDDO

```

```

IF COND(&MODO = 'I') THEN(DO)
  CALL      PGM(SY@060CL) PARM(&PARAME)
ENDDO

GOTO      CMDLBL(FINOK)
ENDDO
/*-----*/
/* SI EL PROCESO ES REMOTO */
/*-----*/
IF COND(&SHTIPO='*RMT') THEN(DO)
/*-----*/
/* ARMA COMANDO CALL A EJECUTAR */
/*-----*/
CHGVAR VAR(&CALL) +
  VALUE('CALL PGM(GPLCP/SY@060CL) PARM('' *CAT &PARAME *CAT ''')')
/*-----*/
/* EJECUTA PROCESO REMOTO SY@065CL EN LA MÁQUINA/LIBRERIA INDICADA */
/*-----*/

REMOTO:      SBMRMTCMD  CMD(&CALL) DDMFILE(&SHDDM1)

ENDDO
/*-----*/
/* MENSAJE DE TERMINACION NORMAL */
/*-----*/
FINOK      SNDUSRMSG  MSG('Proceso Remoto:' *CAT &SHPROC *CAT '-' +
  *CAT &SHPLAN *CAT ', Se esta Ejecutando +
  Satisfactoriamente en la máquina:' *CAT +
  &SHTOMQ *CAT ' Solicitada') +
  MSGTYPE(*INFO) TOUSR(&USER10)

FINAL      RETURN
ENDPGM

```

El programa SY@055R realiza las siguientes acciones:

- Para mandar un número de parámetros variable es necesario enviarlos en un campo carácter concatenados; debido a que en el AS/400 para enviar un parámetro a un trabajo remoto sometido no se aceptan blancos, se deben rellenar estos espacios con algún carácter. Este programa cambia los blancos por una “X” al momento de llamar al programa remoto y, cambia las “X” por blancos para poder ser utilizados en la máquina remota.

```

*-----*
* SISTEMA:      EJECUCION DE PROCESOS REMOTOS      *
* PROGRAMA:     SY@Q55R                            *
* FUNCION:      ACONDICIONA PARÁMETRO PARA SBMRMTCMD *
*
* PROGRAMAS QUE LO LLAMAN:  VARIOS                *
*
*-----*
* SI EL PARÁMETRO PAFUNC ES IGUAL A '1'          *
*
* ELIMINA DEL PARÁMETRO LOS BLANCOS DEBIDO A QUE PARA ENVIAR PA- *
* RAMETROS A UN JOB REMOTO.                        *
*      SBMRMTCMD CMD(CALL PGMX ("98059805      3702"))      *
* NO PERMITE BLANCOS INTERMEDIOS POR LO QUE AL SOMETER ESTE TRA- *
* BAJA A LOS BLANCOS SE LE PONEN 'X'              *
*-----*
* SI EL PARÁMETRO PAFUNC ES IGUAL A '2'          *
*
* ELIMINA DEL PARÁMETRO LAS 'X' CUANDO ESTE PARÁMETRO LLEGA AL *
* PROGRAMA REMOTO.                                  *
*-----*
* AUTOR:        EDUARDO MARTÍNEZ SALINAS          *
* FECHA:
*-----*

```

```

E          PAR          256  1          PARÁMETRO
C*
C* RECEPCION DE PARÁMETRO DE ENTRADA
C*
C      *ENTRY      PLIST
C          PARM          PAPARM256
C          PARM          PAFUNC  1
C*
C          MOVEAPAPARM      PAR,1
C*-----
C*      ELIMINA O PONE ' ' DEL PARÁMETRO
C*-----
C          MOVE '0'          *IN36
C          MOVE *ZEROS      X          40
C          *IN36          DOWEQ'0'
C          ADD 1          X
C          X          IFGT 256
C          MOVE '1'          *IN36
C          ELSE
C          PAFUNC          IFEQ '1'
C          PAR,X          IFEQ ' '
C          MOVE 'X'          PAR,X
C          ENDIF
C          ELSE
C          PAFUNC          IFEQ '2'
C          PAR,X          IFEQ 'X'
C          MOVE ' '          PAR,X
C          ENDIF
C          ENDIF
C          ENDIF
C          ENDIF
C          ENDDO
C          MOVEAPAPAR,1      PAPARM
C          MOVE '1'          *INLR
C          RETRN

```

El programa SY@056CL realiza las siguientes acciones:

- Acondiciona parámetro para el llamado del programa de ejecución de procesos remotos (programa SY@055R).
- Ejecuta proceso remoto de eliminación de archivos de red (programa SY@057CL).

```

/*-----*/
/*
/* NOMBRE : SY@056CL
/* DESCRIPCION: ELIMINA ARCHIVOS DE RED PENDIENTES EN PROCESO *RMT */
/*
/* LLAMADO POR:
/* REALIZO: EDUARDO MARTÍNEZ SALINAS.
/* FECHA:
/*
/*-----*/
PGM PARM(&SHTOMQ &SHDDM1 &SFRLB &SFPRFI)
DCL VAR(&SHTOMQ) TYPE(*CHAR) LEN(10)
DCL VAR(&SHDDM1) TYPE(*CHAR) LEN(10)
DCL VAR(&SFRLB) TYPE(*CHAR) LEN(10)
DCL VAR(&SFPRFI) TYPE(*CHAR) LEN(10)

DCL VAR(&CALL) TYPE(*CHAR) LEN(306) VALUE(' ')
DCL VAR(&PARAM) TYPE(*CHAR) LEN(256) VALUE(' ')
DCL VAR(&CONTAR) TYPE(*DEC) LEN(3 0) VALUE(0)
DCL VAR(&USER10) TYPE(*CHAR) LEN(10)
DCL VAR(&USER8) TYPE(*CHAR) LEN(8)
DCL VAR(&REPLY) TYPE(*CHAR) LEN(1) VALUE(' ')

/*-----*/
/* RECUPERA QUE USUARIO ESTA EJECUTANDO EL PROCESO */
/*-----*/
RTVJOBA USER(&USER10)
CHGVAR VAR(&USER8) VALUE(&USER10)

/*-----*/
/* CONCATENA NOMBRE DE LIBRERIA Y ARCHIVO A ELIMINAR EN MAQ.REMOTA */
/*-----*/
CHGVAR VAR(&PARAM) VALUE(&SFRLB *CAT &SFPRFI)
CALL PGM(SY@055R) PARM(&PARAM '1')

CHGVAR VAR(&CALL) +
VALUE('CALL PGM(GPLCP/SY@057CL) PARM(" ' *CAT &PARAM *CAT '")')
/*-----*/
/* EJECUTA PROCESO REMOTO SY@057CL EN LA MÁQUINA/LIBRERIA INDICADA */
/*-----*/
CHGVAR VAR(&CONTAR) VALUE(0)

REMOTO: SBMRMTCMD CMD(&CALL) DDVFILE(&SHDDM1)
MONMSG MSGID(CPF0000 CPF9999) EXEC(DO)

CHGVAR VAR(&CONTAR) VALUE(&CONTAR - 1)
IF COND(&CONTAR>5) THEN(DO)

```

```

        SNDUSRMSG  MSG('El programa SY@057CL ha intentado por +
                    mas de 5 ocaciones ejecutar el proceso +
                    REMOTO : *CAT &SHTOMQ *CAT +
                    '/QGPL/SY@057CL, pero NO ha +
                    podido debido a algun error, (FAVOR DE +
                    Notificar al coordinador de MAPICS)') +
                    TOUSR(&USER10) MSGRPFY(&REPLY)

        IF COND(&REPLY='C') THEN(GOTO CMDLBL(FINERROR))
        ENDDO

        GOTO      REMOTO
        ENDDO

/*-----*/
/* MENSAJE DE TERMINACION NORMAL */
/*-----*/
FINOK      SNDUSRMSG  MSG('Proceso Remoto ' *CAT &SHTOMQ *CAT +
                    '/QGPL/SY@057CL, Se Ejecuto +
                    Satisfactoriamente en la máquina +
                    Solicitada') MSGTYPE(*INFO) TOUSR(&USER10)

        GOTO      CMDLBL(FINAL)
/*-----*/
/* MENSAJE DE TERMINACION ANORMAL */
/*-----*/
FINERROR   SNDUSRMSG  MSG('El programa SY@056CL ha sido cancelado +
                    por el Usuario') MSGTYPE(*INFO) +
                    TOUSR(&USER10)

FINAL: RETURN
        ENDPGM

```

El programa SY@057CL realiza las siguientes acciones:

- Acondiciona parámetro para el llamado del programa de ejecución de procesos remotos (programa SY@055R).
- Elimina archivos de red.

```

/*-----*/
/*
/* NOMBRE : SY@057CL */
/* DESCRIPCION: ELIMINA ARCHIVOS DE RED PENDIENTES */
/*
/* LLAMADO POR: */
/* REALIZO: EDUARDO MARTÍNEZ SALINAS */
/* FECHA: */
/*
/*-----*/
PGM PARM(&PARAM)
DCL VAR(&PARAM) TYPE(*CHAR) LEN(256)
DCL VAR(&SFRLB) TYPE(*CHAR) LEN(10)

```

```

DCL  VAR(&SFFRFI)  TYPE(*CHAR)  LEN(10)
DCL  VAR(&USER10)  TYPE(*CHAR)  LEN(10)
/*-----*/
/* ELIMINA LAS 'X' DEL PARÁMETRO RECIBIDO. */
/*-----*/
CALL  PGM(GPLCP/SY@055R) PARM(&PARAM '2')

CHGVAR VAR(&SFFRLB) VALUE(%SST(&PARAM 1 10))
CHGVAR VAR(&SFFRFI) VALUE(%SST(&PARAM 11 10))
/*-----*/
/* RECUPERA QUE USUARIO ESTA EJECUTANDO EL PROCESO */
/*-----*/
RTVJOBA  USER(&USER10)

/*-----*/
/* ELIMINA ARCHIVOS DE RED. */
/*-----*/
DLTNETF  FILE(&SFFRFI) NBR(*ALL) USER(&USER10)
MONMSG   MSGID(CPF8060)

FINAL: RETURN
ENDPGM

```

El programa SY@060CL realiza las siguientes acciones:

- Acondiciona librerías necesarias para que corra el proceso (programa SY@074R)
- Llama al programa SY@057CL.
- Llama al programa SY@070CL, el cual crea el área de datos SY@010D con los parámetros recibidos.

```

/*-----*/
/* */
/* NOMBRE : SY@060CL */
/* DESCRIPCION: EJECUTA PROCESOS REMOTOS */
/* */
/* NOTA IMPORTANTE: */
/* ESTE PROGRAMA DEBE SER COMPILADO EN LA QGPL DE */
/* TODAS LAS MÁQUINAS QUE TENGAN INSTALADO EL PROCESO */
/* DE EJECUCION DE PROCESOS REMOTOS */
/* */
/* LLAMADO POR: */
/* REALIZO: EDUARDO MARTÍNEZ SALINAS */
/* FECHA: */
/* */
/*-----*/
PGM PARM(&PARAME)
DCL  VAR(&PARAME)  TYPE(*CHAR)  LEN(269)
DCL  VAR(&SHPROC)  TYPE(*CHAR)  LEN(6)

```

```

DCL  VAR(&SHPLAN)  TYPE(*CHAR)  LEN(2)
DCL  VAR(&MODO)    TYPE(*CHAR)  LEN(1)
DCL  VAR(&SECPRO)  TYPE(*CHAR)  LEN(4)
DCL  VAR(&PAPARM)  TYPE(*CHAR)  LEN(256)
DCL  VAR(&JOBNAM)  TYPE(*CHAR)  LEN(9)
DCL  VAR(&SYSNAM)  TYPE(*CHAR)  LEN(8)
DCL  VAR(&APL$$$)  TYPE(*CHAR)  LEN(10)  VALUE(' ')
DCL  VAR(&PTA$$$)  TYPE(*CHAR)  LEN(2)    VALUE(' ')
DCL  VAR(&DESS$$)  TYPE(*CHAR)  LEN(50)   VALUE(' ')
DCL  VAR(&JOB$$$)  TYPE(*CHAR)  LEN(10)   VALUE(' ')

RTVNETA SYSNAME(&SYSNAM)
CHGVAR VAR(&SHPROC) VALUE(%SST(&PARAME 1 6))
CHGVAR VAR(&SHPLAN) VALUE(%SST(&PARAME 7 2))
CHGVAR VAR(&MODO)   VALUE(%SST(&PARAME 9 1))
CHGVAR VAR(&SECPRO) VALUE(%SST(&PARAME 10 4))
CHGVAR VAR(&PAPARM) VALUE(%SST(&PARAME 14 256))

/*-----*/
/* ADICIONA LA QGPL PARA QUE EL PROGRAMA PUEDA SER EJECUTADO */
/*-----*/
CHGLIBL  LIBL(*NONE)  CURLIB(*CRTDFT)

ADDLIBLE LIB(QTEMP) POSITION(*LAST)
MONMSG   MSGID(CPF0000 CPF9999)

ADDLIBLE LIB(GPLCP) POSITION(*LAST)
MONMSG   MSGID(CPF0000 CPF9999)

ADDLIBLE LIB(QGFL) POSITION(*LAST)
MONMSG   MSGID(CPF0000 CPF9999)

/*-----*/
/* PUNTA A ARCHIVOS DE PARAMETROS */
/*-----*/
OVRDBF   FILE(SY@040F) TOFILE(GPLCP/SY@040F)
MONMSG   MSGID(CPF0000)

/*-----*/
/* REEMPLAZA LIBRERIAS */
/*-----*/
CALL SY@074R PARM(&SHPROC &SHPLAN &SECPRO)

/*-----*/
/* RECUPERA APLICACION, PLANTA Y DESCRIPCION DE LA PLANTA */
/*-----*/
IF COND(&SYSNAM *NE 'CPMEXS02') THEN( +
CALL PGM(CSS510R) PARM(&APL$$$ &PTA$$$ &DESS$$ &JOB$$$))
ELSE CMD( +
CHGVAR VAR(&JOB$$$) VALUE('QBATCH'))

/*-----*/
/* EJECUTA EL PGM-SY@070CL */
/*-----*/
CHGVAR VAR(&JOBNAM) VALUE(&SHPROC *CAT '_' *CAT &SHPLAN)
IF COND(&MODO = 'B') THEN(SBMJOB CMD(CALL +
PGM(SY@070CL) PARM(&SHPROC &SHPLAN &MODO +
&SECPRO &PAPARM)) JOB(&JOBNAM) +
JOBQ(&JOB$$$))
IF COND(&MODO = 'I') THEN( +
CALL PGM(SY@070CL) PARM(&SHPROC &SHPLAN &MODO &SECPRO &PAPARM))

RETURN
ENDPGM

```

El programa SY@070CL realiza las siguientes acciones:

- Acondiciona parámetro para el llamado del programa de ejecución de procesos remotos (programa SY@055R).
- Crea el área de datos SY@010D con los valores de los parámetros recibidos.
- Llama al programa SY@070R que recibe los archivos de red y ejecuta el programa indicado en los archivos del sistema.

```

/*-----*/
/*
/* NOMBRE . SY@070CL
/* DESCRIPCION. EJECUTA PROCESOS REMOTOS
/*
/* NOTA IMPORTANTE.
/* ESTE PROGRAMA DEBE SER COMPILADO EN LA QGPL DE
/* TODAS LAS MÁQUINAS QUE TENGAN INSTALADO EL PROCESO
/* DE EJECUCION DE PROCESOS REMOTOS.
/*
/* LLAMADO POR:
/* REALIZO. EDUARDO MARTÍNEZ SALINAS
/* FECHA.
/*
/*-----*/
PGM PARM(&SHPROC &SHPLAN &MODO &SECPRO &PAPARM)
DCL VAR(&SHPROC) TYPE(*CHAR) LEN(6)
DCL VAR(&SHPLAN) TYPE(*CHAR) LEN(2)
DCL VAR(&MODO) TYPE(*CHAR) LEN(1)
DCL VAR(&SECPRO) TYPE(*CHAR) LEN(4)
DCL VAR(&PAPARM) TYPE(*CHAR) LEN(256)
DCL VAR(&PABRRO) TYPE(*CHAR) LEN(2)
DCL VAR(&USER10) TYPE(*CHAR) LEN(10)
/*-----*/
/* RECUPERA QUE USUARIO ESTA EJECUTANDO EL PROCESO
/*-----*/
RTVJOBA USER(&USER10)
/*-----*/
/* ACONDICIONA PARÁMETRO ENVIADO POR SBMRMTCMD
/*-----*/
CALL SY@055R PARM(&PAPARM '2')
/*-----*/
/* CREA DATA AREA EN QTEMP PARA GUARDAR EL PARÁMETRO RECIBIDO DEL
/* SISTEMA ORIGEN.
/*-----*/
CRTDTAARA DTAARA(QTEMP/SY@010D) TYPE(*CHAR) LEN(256) +
VALUE(' ') TEXT('Parámetro Recibido de +
sistema Origen')
MONMSG MSGID(CPF0000)
CHGDTAARA DTAARA(SY@010D) VALUE(&PAPARM)
/*-----*/
/* APUNTA A ARCHIVOS DE PARÁMETROS
/*-----*/
OVRDBF FILE(SY@020F) POFILE(GPLCP/SY@020F)

```

```

MONMSG      MSGID(CPF0000)
OVRDEF     FILE(SY@030F) TOFILE(GPLCP/SY@030F)
MONMSG      MSGID(CPF0000)
OVRDEF     FILE(SY@040F) TOFILE(GPLCP/SY@040F)
MONMSG      MSGID(CPF0000)
/*-----*/
/*  RECIBE ARCHIVOS DEL JOB Y EJECUTA PGM.INDIC.  */
/*-----*/
CALL SY@070R PARM(&SHPROC &SHPLAN &MODO &SECPRO &PAPARM &PAERRO)
IF          COND(&PAERRO='SI') THEN(DO)
            SNDUSRMSG MSG('No se Encontraron los Parametros para +
                            ejecutar el proceso remoto ' *CAT &SHPROC +
                            *CAT ' por favor capture los parametros y +
                            ejecute este proceso nuevamente') +
            MSGTYPE(*INFO) TOUSR(&USER10)
ENDDO
FINAL:     RETURN
           ENDPGM

```

El programa SY@070R realiza las siguientes acciones:

- Llama al programa Sy@072CL que recibe archivos de red.
- Llama al programa indicado en los archivos del sistema.
- Si el proceso debe regresar a la máquina origen manda archivos (programas SY@056CL y SY@054CL).

```

*-----*
*  PROGRAMA      SY@070R
*  TRAE PARAMETROS PARA LA EJECUCION DE PROCE-
*  SOS REMOTOS.
*  INTALACION...
*  ELABORO.....: EDUARDO MARTÍNEZ SALINAS
*  FECHA...     :
*-----*
FSY@020F IF E      K      DISK
FSY@030F IF E      K      DISK
C*-----*
C*  RECEPCION DE PARAMETROS
C*-----*
C          *ENTRY  PLIST
C          PARM          SHPROC          PROCESO
C          PARM          SHPLAN          PLANTA
C          PARM          MODO 1          B=BATCH,1=INTERACT
C          PARM          SECPRO 4        SECUENCIA DE PROCESO
C          PARM          PAPARM256      PARÁMETRO DEL JOB
C          PARM          PAERRO 2        BANDERA DE ERROR
C*-----*
C*  PAR072 - PARAMETROS PARA RECEPCION DE ARCHIVOS
C*-----*
C          PAR072  PLIST
C          PARM          SHTIPO          TIPO DE PROCESO

```

```

C          PARM          SFFRLB          FROM LIBRERIA
C          PARM          SFFRFI          FROM ARCHIVO
C          PARM          SFTOLB          TO LIBRERIA
C          PARM          SFTOFI          TO ARCHIVO
C          PARM          SFOPCI          *ADD O *REPLACE
C*-----*
C*      PAR052.- PARAMETROS PARA ENVIO DE ARCHIVOS          *
C*-----*
C          PAR052      PLIST
C          PARM          SHTOMQ          TO MÁQUINA
C          PARM          SFFRLB          FROM LIBRERIA
C          PARM          SFFRFI          FROM ARCHIVO
C*-----*
C*      PAR054.- PARAMETROS PARA EJECUCION DEL JOB REMOTO  *
C*-----*
C          PAR054      PLIST
C          PARM          SHPROC          PROCESO
C          PARM          SHPLAN          PLANTA
C          PARM          MODO           B=BATCH;1=INTERACT
C          PARM          SECPRO 4       SECUENCIA DE PROCESO
C          PARM          SHTIPO          TIPO DE PROCESO
C          PARM          SHTOMQ          TO MÁQUINA
C          PARM          SHDDM2         ARCHIVO DDM
C          PARM          PAPARM         PARÁMETRO DEL JOB
C*-----*
C*      PAR056.- PARAMETROS PARA ELIMINACION DE ARCHIVOS REMOTOS. *
C*-----*
C          PAR056      PLIST
C          PARM          SHTOMQ          TO MÁQUINA
C          PARM          SHDDM2         ARCHIVO DDM
C          PARM          SFFRLB          FROM LIBRERIA
C          PARM          SFFRFI          FROM ARCHIVO
C*-----*
C*      KEYP1 - LLAVE PARA ACCESAR ARCHIVO DE PARAMETROS DEL JOB *
C*-----*
C          KEYP1      KLIST
C          KFLD          SHPROC
C          KFLD          SHPLAN
C*-----*
C*      KEYP2 - LLAVE PARA ACCESAR ARCHIVO DE PARAMETROS DEL JOB *
C*-----*
C          KEYP2      KLIST
C          KFLD          SHPROC
C          KFLD          SHPLAN
C          KFLD          SECPRO
C          MOVE 'NO'    PAERRO
C          KEYP1      CHAINSY@020F          36
C          *IN36      IFEQ '1'
C          MOVE 'SI'    PAERRO
C          ELSE
C          EXSR SR0001
C          ENDIF
C          MOVE '1'    *INLR
C*-----*
C*      SR0001.- PROCESA EJECUCION DE JOB REMOTO
C*-----*
C          SR0001      BEGSR
C* RECIBE ARCHIVOS DEL JOB
C          KEYP2      SETLLSY@030F
C          KEYP2      READESY@030F          36
C          *IN36      DOWEQ'0'
C          CALL 'SY@072CL'PAR072
C          KEYP2      READESY@030F          36

```

```

C                               ENDDO
C*-----
C*      EJECUTA JOB REMOTO
C*-----
C          SECPRO      IFEQ 'INIC'
C          SHPGM1      ANDNE*BLANKS
C                               CALL SHPGM1
C                               ENDF
C          SECPRO      IFEQ 'RETO'
C          SHPGM2      ANDNE*BLANKS
C                               CALL SHPGM2
C                               ENDF
C*-----
C*      SOLO SI ES SECUENCIA DE PROCESO INIC
C*      SI EL JOB DEBE REGRESAR A LA MÁQUINA ORIGEN INICIA EL PROCES
C*-----
C          SECPRO      IFEQ 'INIC'
C          SHRETO      ANDEQ'SI'
C                               MOVEL'RETO'      SECPRO
C                               MOVESHPRMQ      SHTOMQ
C*-----
C*      ENVIA LOS ARCHIVOS AL JOB (SOLO SI ES JOB REMOTO)
C*-----
C          SHTIPO      IFEQ '*RMT'
C          KEYP2      SETLLSY@030F
C          KEYP2      READESY@030F      36
C          *IN36      DOWEQ'O'
C                               CALL 'SY@056CL'PAR056
C                               CALL 'SY@052CL'PAR052
C          KEYP2      READESY@030F      36
C                               ENDDO
C                               ENDF
C*-----
C*      EJECUTA JOB REMOTO DE REGRESO
C*-----
C                               CALL 'SY@054CL'PAR054
C                               ENDF
C                               ENDSR

```

El programa SY@072CL realiza las siguientes acciones:

- Selecciona si adiciona o reemplaza la información en el archivo.
- Si el proceso es local copia los archivos a las librerías indicadas.
- Si el proceso es remoto recibe los archivos de red en las librerías y archivos indicados.

```

/*-----*/
/*
/* NOMBRE SY@072CL
/* DESCRIPCION EJECUTA PROCESOS REMOTOS
/*
/*
/* NOTA IMPORTANTE
/* ESTE PROGRAMA DEBE SER COMPILADO EN LA QGPL DE
/* TODAS LAS MÁQUINAS QUE TENGAN INSTALADO EL PROCESO
/* DE EJECUCION DE PROCESOS REMOTOS.
/*
/* REALIZO: EDUARDO MARTÍNEZ SALINAS
/* FECHA:
/*-----*/
PGM PARM(&SHTIPO &SFFRLB &SFFRFI &SFTOLB &SFTOFI &SFOPCI)
DCL VAR(&SHTIPO) TYPE(*CHAR) LEN(4)
DCL VAR(&SFFRLB) TYPE(*CHAR) LEN(10)
DCL VAR(&SFFRFI) TYPE(*CHAR) LEN(10)
DCL VAR(&SFTOLB) TYPE(*CHAR) LEN(10)
DCL VAR(&SFTOFI) TYPE(*CHAR) LEN(10)
DCL VAR(&SFOPCI) TYPE(*CHAR) LEN(4)
DCL VAR(&MBROPT) TYPE(*CHAR) LEN(8) VALUE('*ADD ')
DCL VAR(&CONTAR) TYPE(*DEC) LEN(3 0) VALUE(0)
DCL VAR(&ESPERA) TYPE(*DEC) LEN(3 0) VALUE(5)
DCL VAR(&TIEMPO) TYPE(*DEC) LEN(3 0) VALUE(60)
DCL VAR(&USER10) TYPE(*CHAR) LEN(10)
DCL VAR(&USER8) TYPE(*CHAR) LEN(8)
DCL VAR(&REPLY) TYPE(*CHAR) LEN(1) VALUE(' ')

RTVJOBA USER(&USER10)
CHGVAR VAR(&USER8) VALUE(&USER10)
/*-----*/
/* PONE PARÁMETRO DE OPCION ADICIONAR O REPLAZAR REGISTROS
/*-----*/
IF COND(&SFOPCI='*ADD') THEN(CHGVAR VAR(&MBROPT) VALUE('*ADD '))
IF COND(&SFOPCI='*REP') THEN(CHGVAR VAR(&MBROPT) VALUE('*REPLACE'))
/*-----*/
/* SI EL JOB ES LOCAL COPIA EL ARCHIVO A LA LIBRERIA SOLICITADA
/*-----*/
IF COND(&SHTIPO='*LCL') THEN(DO)
  IF COND(&MBROPT *NE '*ADD ') THEN(+
    CLRPFM FILE(&SFTOLB/&SFTOFI))
    COPIA. CPYF FROMFILE(&SFFRLB/&SFFRFI) +
            TOFILE(&SFTOLB/&SFTOFI) MBROPT(&MBROPT) +
            PMTOPT(*NOCHK)

MONMSG MSGID(CPF2817) /* ARCHIVO VACIO */

GOTO CMDLBL(FINAL)
ENDDO
/*-----*/
/* RECIBE ARCHIVO ENVIADO DEL SISTEMA ORIGEN
/*-----*/
RECIBE: CHGVAR VAR(&CONTAR) VALUE(0)
        RCVNETF FROMFILE(&SFFRFI) TOFILE(&SFTOLB/&SFTOFI) +
          TOMBR(*FIRST) MBROPT(&MBROPT) USER(&USER10)
        MONMSG MSGID(CPF0000 CPF9999) EXEC(DO)

        CHGVAR VAR(&CONTAR) VALUE(&CONTAR + 1)
        IF COND(&CONTAR>&ESPERA) THEN(DO)
          SNDUSRMSG MSG('El programa SY@072CL ha intentado por +
            mas de 5 ocasiones recibir el archivo ' +
            *CAT &SFFRFI *CAT ' a ' *CAT &SFTOLB *CAT +
            '/' *CAT &SFTOFI *CAT ' , y NO ha podido +

```

```

debido a algun error. Favor de +
Notificar al Coordinador de MAPICS. +
Respuestas: C=Cancelar el JOB, +
R=Reintentar Recibir el archivo +
Inmediatamente, D=Esperar Un Minuto y +
Reintentar Recibir el Archivo') +
VALUES('C' 'R' 'D') TOUSR(&USER10) +
MSGRPY(&REPLY)

```

```

IF COND(&REPLY='C') THEN(GOTO CMDLBL(FINAL))
IF COND(&REPLY='R') THEN(GOTO CMDLBL(RECIBE))
IF COND(&REPLY='D') THEN(DO)
    DLYJOB    DLY(60)
    GOTO CMDLBL(RECIBE)
ENDDO
ENDDO

DLYJOB    DLY(&TIEMPO)
CHGVAR    VAR(&TIEMPO) VALUE(300)
CHGVAR    VAR(&ESPERA) VALUE(10)
GOTO      RECIBE
ENDDO

```

```

FINAL:    RETURN
          ENDPGM

```

El programa SY@074CL realiza las siguientes acciones:

- Adiciona una librería al final de la lista de librerías.

```

/*-----*/
/*
/* NOMBRE : SY@074CL
/* DESCRIPCION: EJECUTA PROCESOS REMOTOS
/*
/* NOTA IMPORTANTE:
/* ESTE PROGRAMA DEBE SER COMPILADO EN LA QGPL DE
/* TODAS LAS MÁQUINAS QUE TENGAN INSTALADO EL PROCESO
/* DE EJECUCION DE PROCESOS REMOTOS.
/*
/* REALIZO: EDUARDO MARTÍNEZ SALINAS
/* FECHA:
/*-----*/
PGM PARM(&SLLIBR)
DCL VAR(&SLLIBR) TYPE(*CHAR) LEN(10)
DCL VAR(&REPLY) TYPE(*CHAR) LEN(1) VALUE(' ')
DCL VAR(&USER10) TYPE(*CHAR) LEN(10)

RTVJOB   USER(&USER10)
/*-----*/
/* ADICIONA LA LIBRERIA AL FINAL DE LA LISTA DE LIBRERIAS
/*-----*/
ADDLIBLE LIB(&SLLIBR) POSITION(*LAST)
FINAL:   RETURN
          ENDPGM

```

El programa SY@074R realiza las siguientes acciones:

- Recupera la lista de librerías de los archivos del sistema.
- Llama al programa que adiciona una librería al final de la lista de librerías.

```

F*-----*
F* PROGRAMA      : SY@074R                               *
F* DESCRIPCIÓN  : REEMPLAZO DE LIBRERIAS                 *
F*                                                       *
F* INTALACION... :                                       *
F* ELABORO..... : EDUARDO MARTÍNEZ SALINAS              *
F* FECHA.....   :                                       *
F*-----*
FSY@040F IF E          K          DISK
C*-----*
C* RECEPCION DE PARAMETROS                               *
C*-----*
C          *ENTRY    PLIST
C          PARM          SHPROC 6          PROCESO
C          PARM          SHPLAN 2          PLANTA
C          PARM          SECPRO 4          SECUENCIA DE PROCESO
C*-----*
C* PAR074.- PARAMETROS PARA REEMPLAZO DE LIBRERIAS      *
C*-----*
C          PAR074    PLIST
C          PARM          SLLIBR          LIBRERIA
C*-----*
C* KEYP A2 - LLAVE PARA ACCESAR ARCHIVO DE PARAMETROS DEL JOB *
C*-----*
C          KEYP A2    KLIST
C          KFLD          SHPROC
C          KFLD          SHPLAN
C          KFLD          SECPRO
C* REEMPLAZA LIBRERIAS
C          KEYP A2    SETLLSY@040F
C          KEYP A2    READESY@040F          36
C          *IN36      DOWEQ '0'
C          CALL 'SY@074CL' PAR074
C          KEYP A2    READESY@040F          36
C          ENDDO
C          MOVE '1'          *INLR

```

El programa SY@080CL realiza las siguientes acciones:

- Presenta pantalla para selección de parámetros.
- Llama al programa de selección de información para el reporte de parámetros.

```

/*-----*/
/* PROGRAMA : SY@080CL */
/* REPORTE DE PARAMETROS PARA PROCESOS REMOTOS */
/* STANDARD. */
/* */
/* AUTOR : EDUARDO MARTÍNEZ SALINAS */
/* */
/* FECHA : */
/* */
/* PROGRAMAS QUE LO LLAMAN : */
/* */
/* HISTORIA DE MODIFICACIONES : */
/* */
/* AUTOR : */
/* FECHA : */
/* MOTIVO : */
/* */
/*-----*/
PGM
DCLF FILE(SY@080P)
DCL VAR(&APL$$$) TYPE(*CHAR) LEN(10)
DCL VAR(&PTA$$$) TYPE(*CHAR) LEN(2)
DCL VAR(&DES$$$) TYPE(*CHAR) LEN(50)
DCL VAR(&JOB$$$) TYPE(*CHAR) LEN(10)

/*-----*/
/* RECUPERA LA APLICACION, PLANTA Y DESCRIPCION DE LA CIA. */
/*-----*/
CALL PGM(CSS510R) PARM(&APL$$$ &PTA$$$ &DES$$$ +
&JOB$$$)

/*-----*/
/* INICIALIZA PARAMETROS DE SELECCION */
/*-----*/
LIMPIA CHGVAR VAR(&PAPROC) VALUE(' ')
CHGVAR VAR(&PAPTA) VALUE(' ')

/*-----*/
/* PRESENTA PANTALLA DE SELECCION DE PARAMETROS. */
/*-----*/
INICIO: SNDRCVF RCDFMT(SY@080P1)
CHGVAR VAR(&PAMSG1) VALUE(' ')

IF COND(&IN03 = '1') THEN(GOTO CMDLBL(FINAL))
IF COND(&IN12 = '1') THEN(GOTO CMDLBL(FINAL))
IF COND(&IN05 = '2') THEN(GOTO CMDLBL(LIMPIA))

/*-----*/
/* SOMETE TRABAJO A LA QBATCH */
/*-----*/
SBMJOB CMD(CALL PGM(SY@081CL) PARM(&PAPTA +
&PAPROC)) JOB(REP.PARMS )
CHGVAR VAR(&PAMSG1) VALUE('Proceso en ejecuci3n, +
revise sus trabajos sometidos')

GOTO CMDLBL(LIMPIA)

FINAL RETURN
ENDPGM

```

El programa SY@080R realiza las siguientes acciones:

- Genera reporte de parámetros del sistema.

```

*-----*
*  PROGRAMA : SY@080R. *
*              REPORTE DE PARAMETROS PARA PROCESOS REMOTOS *
*              STANDARD. *
* * * * *
*  SISTEMA : MAPICS/COSTOS. *
* * * * *
*  AUTOR : EDUARDO MARTÍNEZ SALINAS *
* * * * *
*  FECHA : *
* * * * *
*  PROGRAMAS QUE LO LLAMAN . *
* * * * *
*  HISTORIA DE MODIFICACIONES *
* * * * *
*  FECHA MOD. . *
*  AUTOR MOD . *
*  TEXTO MOD . *
* * * * *
*-----*
FSY@020F IF E K DISK
FSY@030F IF E K DISK
FSY@040F IF E K DISK
FCSS505F IF E K DISK
FSY@080T O E 22 PRINTER
*****
* RUTINA PRINCIPAL *
*****
C EXSR SR0000
*
C *LOVAL SETLLSY@020F
C READ SY@020F 36
*
C *IN36 DOWEQ 0'
C EXSR SR0001
C EXSR SR0002
C SHRETO IPEQ 'SI'
C EXSR SR0003
C END
C WRITESY@080L1
C READ SY@020F 36
C ENDDO
*
C SIHUBO IPEQ 'NO'
C WRITESY@080H1
C WRITESY@080E1
C END
*
C MOVE '1' *INLR
*
*****
* SR0000- SUBROUTINA DE INICIALIZACION DE VARIABLES *
*****
C SRO000 BEGSR
* VARIABLES DE TRABAJO
*
C MOVE *BLANKS AKSECJ 4

```

```

C          MOVE 'NO'          SIHUBO  1
C          MOVE '1'           *IN22
*
* KEYSY@- ACCESO AL  LOS SIGUIENTES ARCHIVOS:
* SY@030F - ARCHIVOS A ENVIAR Y/O RECIBIR
* SY@040F - LISTA DE BIBLIOTECAS A REEMPLAZAR
*
C          KEYSY@      KLIST
C          KFLD        SHPROC
C          KFLD        SHPLAN
C          KFLD        WKSECU
*
*          RECUPERA APLICACION, PLANTA Y DESCRIPCION DE LA PLANTA
*
C          READ CSS505F          36
*
C          ENDSR
* *****
* SR0001- IMPRIME DATOS GENERALES DEL PROCESO.          *
* *****
C          SR0001      BEGSR
*
C          *IN22      IFEQ '1'
C                   WRITESY@080H1
C                   MOVE '0'          *IN22
C                   END
C                   WRITESY@080D1
*
C                   MOVE 'SI'        SIHUBO
*
C                   ENDSR
* *****
* SR0002- PARAMETROS PARA PROCESO REMOTO DE INICIO.    *
* *****
C          SR0002      BEGSR
*
C                   MOVE 'INIC'      WKSECU
C                   MOVE SHPGM1      WKPGM
C                   MOVE SHDDM1      WKDDM
C          *IN22      IFEQ '1'
C                   WRITESY@080H1
C                   MOVE '0'          *IN22
C                   END
C                   WRITESY@080H2
C                   EXSR SR0012
C                   EXSR SR0013
*
C                   ENDSR
* *****
* SR0012- IMPRIME LISTA DE ARCHIVOS A ENVIAR          *
* *****
C          SR0012      BEGSR
*
C                   WRITESY@080H3
C          KEYSY@      SETLLSY@030F
C          KEYSY@      READESY@030F          36
*
C          *IN36      DOWEQ'0'
C          *IN22      IFEQ '1'
C                   WRITESY@080H1
C                   WRITESY@080H3
C                   MOVE '0'          *IN22
C                   END

```

```

C          WRITESY@080D3
C          KEYSY@  READESY@030F          36
C          ENDDO
*
C          ENDSR
*****
* SR0013- IMPRIME LISTA DE BIBLIOTECAS PARA REEMPLAZO *
*****
C          SR0013  BEGSR
*
C          WRITESY@080H4
C          KEYSY@  SETLLSY@040F
C          KEYSY@  READESY@040F          36
*
C          *IN36   DOWEQ'0'
C          *IN22   IFEQ '1'
C          WRITESY@080H1
C          WRITESY@080H4
C          MOVE '0'          *IN22
C          END
C          WRITESY@080D4
C          KEYSY@  READESY@040F          36
C          ENDDO
*
C          ENDSR
*****
* SR0003- PARAMETROS PARA PROCESO REMOTO DE RETORNO *
*****
C          SR0003  BEGSR
*
C          MOVE 'RETO'    WKSECU
C          MOVE SHPGM2    WKPGM
C          MOVE SHDDM2    WKDDM
C          *IN22         IFEQ '1'
C          WRITESY@080H1
C          MOVE '0'          *IN22
C          END
C          WRITESY@080H5
C          EXSR SR0012
C          EXSR SR0013
*
C          ENDSR

```

El programa SY@081CL realiza las siguientes acciones:

- Llama al programa de generación de reporte de parámetros del sistema.

```

/*-----*/
/* SISTEMA : MAPICS/COSTOS. */
/* */
/* PROGRAMA : SY@081CL. */
/* REPORTE DE PARAMETROS PARA PROCESOS REMOTOS */
/* STD. */
/* */
/* AUTOR : EDUARDO MARTÍNEZ SALINAS */
/* */
/* FECHA : */
/* */
/* PROGRAMAS QUE LO LLAMAN : SY@080CL */
/* */
/* HISTORIA DE MODIFICACIONES : */
/* */
/* AUTOR . . : */
/* FECHA . . : */
/* MOTIVO . . : */
/*-----*/
PGM PARM(&PAPTA &PAPROC)

DCL VAR(&PAPROC) TYPE(*CHAR) LEN(6)
DCL VAR(&PAPTA) TYPE(*CHAR) LEN(2)
DCL VAR(&SLT1) TYPE(*CHAR) LEN(400)
DCL VAR(&VEZ1RA) TYPE(*CHAR) LEN(1)

/*-----*/
/* ARMA PARAMETROS QRYSLT, PARA SELECCION DE INFORMACION. */
/*-----*/
CHGVAR VAR(&SLT1) VALUE(' ')
CHGVAR VAR(&VEZ1RA) VALUE('N')

/*----- (1) ARMA SELECCION DE PROCESO -----*/
IF COND((&PAPROC *NE ' ')) +
THEN(DO)
CHGVAR VAR(&SLT1) VALUE('SHPROC = ' *CAT +
&PAPROC *CAT ' ')
CHGVAR VAR(&VEZ1RA) VALUE('S')
ENDDO

/*----- (2) ARMA SELECCION DE TIPO DE PRODUCTO-----*/
IF COND((&PAPTA *NE ' ')) +
THEN(DO)
IF COND(&VEZ1RA *EQ 'S') THEN(DO)
CHGVAR VAR(&SLT1) VALUE(' ' *BCAT &SLT1 *BCAT '*AND')
ENDDO
ELSE CMD(DO)
CHGVAR VAR(&SLT1) VALUE(' ')
ENDDO
CHGVAR VAR(&SLT1) VALUE(' ' *BCAT &SLT1 *BCAT +
'SHPLAN = ' *CAT &PAPTA *CAT ' ')
CHGVAR VAR(&VEZ1RA) VALUE('S')
ENDDO

IF COND(&VEZ1RA *EQ 'S') THEN(DO)
CHGVAR VAR(&SLT1) VALUE(' ' *BCAT &SLT1 *BCAT ' ')
/*-----*/
/* SELECCIONA INFORMACION DEL HEADER DE PARAMETROS */
/*-----*/

```

```

OVRDEF      FILE(SY@020F) SHARE(*YES) SEQONLY(*NO)

OPNQRYF     FILE((SY@020F)) OPTION(*INP) QRYSLT(' ' +
          *BCAT &SLT1 *BCAT ' ') KEYPLD(*FILE) +
          OPTIMIZE(*MINWAIT)

ENDDO

/*-----*/
/* GENERA REPORTE DE PARAMETROS PARA PROCESOS REMOTOS. */
/*-----*/

CALL        PGM(SY@080R)

CLOF        OPNID(SY@020F)
MONMSG      MSGID(CPF0000)

DLTOVR      FILE(*ALL)
MONMSG      MSGID(CPF0000)

RETURN
ENDPGM

```

El programa SY@085R realiza las siguientes acciones:

- Mantenimiento de los archivos del sistema.

```

*-----*
* PROGRAMA . SY@085R *
* Mantenimiento a las Bases de Datos de Procesos *
* Remotos. *
* *
* SISTEMA Mapics/Costos. *
* *
* AUTOR EDUARDO MARTÍNEZ SALINAS *
* *
* FECHA : *
* *
* PROGRAMAS QUE LO LLAMAN : Persec *
* *
* HISTORIA DE MODIFICACIONES : *
* *
* FECHA MOD: *
* AUTOR MOD: *
* TEXTO MOD: *
* *
*-----*
FSY@020F UF E K DISK A
FSY@030F UF E K DISK A
FSY@040F UF E K DISK A
FSY@085P CF E WORKSTN
F NRR1 KSFIL SY@085S1
F NRR2 KSFIL SY@085S2
F NRR3 KSFIL SY@085S3
*
E MSG 1 20 78
*
C Z-ADD1 NRCD1
C *IN03 DOWEQ'0'
C EXSR WRTSF1
C EXSR SR0C10

```

```

C          ENDDO
*
C          MOVE '1'          *INLR
*-----*
*  Despliega Subfile No. 1.  *
*-----*
C          SR0010  BEGSR
C          *IN03   DOWEQ'0'
C                   WRITESY@085HD
C                   WRITESY@085M1
C                   EXFMTSY@085C1
C                   CLEARPAMSG
*
C          *IN03   IFEQ '1'
C          *IN12   OREQ '1'
C                   MOVE '1'          *IN03
C                   LEAVE
C                   ENDIF
*
C          *IN35   IFEQ '1'
C                   Z-ADD1          NRCD1
C                   ELSE
C          RRCS1   IFGT *ZEROS
C                   Z-ADDRRCS1     NRCD1
C                   ENDIF
C                   ENDIF
*
C          *IN06   IFEQ '1'
C                   MOVE '6'          SFOPC
C                   MOVE 'Alta'     PADMOV
C                   EXSR SR0011
C                   ENDIF
*
C          *IN50   EXSR LEBSF1
C                   IFEQ '0'
C                   LEAVE
C                   ENDIF
C                   ENDDO
C                   ENDSR
*-----*
*  Despliega Pantalla No 1 de Datos Generales p/Alta, Copiar. *
*-----*
C          SR0011  BEGSR
C                   CLEAR*IN06
C                   CLEARSHPROC
C                   CLEARSHPLAN
C                   MOVE '*'          ASTER  1
*
C          *IN03   DOWEQ'0'
C          ASTER   ANDEQ'*'
C                   WRITESY@085HD
C                   WRITESY@085M2
C                   EXFMTSY@085P1
C                   CLEARPAMSG
*
C          *IN03   IFEQ '1'
C          *IN12   OREQ '1'
C                   LEAVE
C                   ENDIF
*
C          *IN50   EXSR CHKERR
C                   IFEQ '0'
C                   EXSR SR0012

```

```

C          ENDIF
C          ENDDO
C          ENDSR
*-----*
*  Despliega Pantalla No. 2 de Datos Generales.  *
*-----*
C          SR0012      BEGSR
C          *IN03      DOWEQ'0'
C          ASTER      ANDEQ'*'
C                   WRITESY@085M2
C                   EXPMTSY@085P2
C                   CLEARPAMSG
C                   MOVEA'00000'      *IN,51
*
C          *IN03      IFEQ '1'
C          *IN12      OREQ '1'
C                   LEAVE
C                   ENDIF
*
C          SHTIPO     IFNE '*LCL'
C          SHTIPO     ANDNE '*RMT'
C                   MOVE '1'          *IN51
C                   MOVEAMSG,5      PAMSG
C                   ITER
C                   ENDIF
*
C          SHDESP     IFEQ *BLANKS
C                   MOVE '1'          *IN52
C                   MOVEAMSG,6      PAMSG
C                   ITER
C                   ENDIF
*
C          SHTIPO     IFEQ '*RMT'
C          SHFRMQ     IFEQ *BLANKS
C                   MOVE '1'          *IN53
C                   MOVEAMSG,7      PAMSG
C                   ITER
C                   ENDIF
*
C          SHTOMQ     IFEQ *BLANKS
C                   MOVE '1'          *IN54
C                   MOVEAMSG,8      PAMSG
C                   ITER
C                   ENDIF
*
C          SHRETO     IFNE *BLANKS
C          SHRETO     ANDNE 'SI'
C                   MOVE '1'          *IN55
C                   MOVEAMSG,9      PAMSG
C                   ITER
C                   ENDIF
*
C          SHPGM2     IFNE *BLANKS
C          SHRETO     ANDEQ*BLANKS
C          SHDDY2     ORNE *BLANKS
C          SHRETO     ANDEQ*BLANKS
C                   MOVE '1'          *IN55
C                   MOVEAMSG,10     PAMSG
C                   ITER
C                   ENDIF
*
C          EXSR SR0020

```

```

C          ENDDO
C          ENDSR
*-----*
*   Llena el Subfile No. 1.   *
*-----*
C          WRTSF1  BEGSR
C          CLEARNRR1
C          MOVEA'100'    *IN,25
C          WRITESY@085C1
C          MOVEA'000'    *IN,25
*
C          KEYCTL  SETLLSY@020R
C          READ SY@020R          27
C          *IN27  DOWEQ'0'
C          ADD 1    NRR1
C          MOVE '1'    *IN26
C          WRITESY@085S1
C          READ SY@020R          27
C          ENDDO
*
C          NRCD1  IPGT NRR1
C          Z-ADDNRR1    NRCD1
C          ENDIF
C          ENDSR
*-----*
*   Lee el Subfile No 1. de Datos Generales.   *
*-----*
C          LEESF1  BEGSR
C          *IN03  DOWEQ'0'
C          NRR1  ANDGT*ZEROS
C          READCSY@085S1          27
C          27  LEAVE
*
C          Z-ADDNRR1    NRCD1
C          SFOPC  IFEQ '2'
C          SFOPC  OREQ '3'
C          SFOPC  OREQ '4'
C          SFOPC  OREQ '7'
C          50  EXSR CHKERR
C          LEAVE
C          MOVE '*'    ASTER
C          SELEC
C          SFOPC  WHEQ '3'
C          MOVE '9'    SFOPC
C          EXSR SR0011
C          SFOPC  WHEQ '2'
C          SFOPC  OREQ '4'
C          EXSR SR0012
C          SFOPC  WHEQ '7'
C          CALL 'SY@086R'
C          PARM          SHFROC
C          PARM          SHPLAN
C          PARM ' '    WKERR 3
C          FREE 'SY@086R'
C          WKERR  IFEQ 'ERR'
C          MOVE '1'    *IN50
C          MOVEAMSG,19  PAMSG
C          LEAVE
C          ELSE
C          MOVEAMSG,20  PAMSG
C          ENDIF
C          ENDSL
C          ENDIF

```

```

C          ENDDO
C          ENDSR
*-----*
*  Verifica Errores.  *
*-----*
C          CHKERR  BEGSR
C                   MOVE '0'          *IN34
C                   MOVEA'00000'      *IN,50
C                   MOVE SHPROC      WKPROC
C                   MOVE SHPLAN      WKPLAN
*
C          SHPROC  DO
C                   IFEQ *BLANKS
C                   MOVEA'11'        *IN,50
C                   MOVEAMSG,1      PAMSG
C                   LEAVE
C                   ENDIF
*
C          SHPLAN  IFEQ *BLANKS
C                   MOVEA'101'      *IN,50
C                   MOVEAMSG,2      PAMSG
C                   LEAVE
C                   ENDIF
*
C          KEY020  CHAINSY@020R      36
C          SFOPC   IFEQ '6'          Alta
C          *IN36  ANDEQ'0'
C          SFOPC   OREQ '9'          Copiar
C          *IN36  ANDEQ'0'
C                   MOVEA'111'      *IN,50
C                   MOVEAMSG,3      PAMSG
C                   LEAVE
C                   ENDIF
*
C          SFOPC   IFNE '6'
C          SFOPC   ANDNE'9'
C          *IN36  ANDEQ'1'
C                   MOVE '1'        *IN50
C                   MOVEAMSG,4      PAMSG
C                   LEAVE
C                   ENDIF
*
C          SLEEC
C          SFOPC   WHEQ '6'
C          *NOKEY CLEARSY@020R
C          SFOPC   WHEQ '2'
C                   MOVE 'Cambio'   PADMOV
C          SFOPC   WHEQ '3'
C                   MOVE 'Copiar'   PADMOV
C                   MOVELSHPROC     WKPRO1 6
C                   MOVELSHPLAN     WKPLA1 2
C          SFOPC   WHEQ '9'
C                   MOVELSHPROC     WKPRO2 6
C                   MOVELSHPLAN     WKPLA2 2
C                   MOVELWKPRO1     WKPROC
C                   MOVELWKPLA1     WKPLAN
C          KEY020  CHAINSY@020R      36
C                   MOVELWKPRO2     SHPROC
C                   MOVELWKPLA2     SHPLAN
C          SFOPC   WHEQ '4'
C                   MOVE 'Baja'     PADMOV
C                   MOVE '1'        *IN34
C          ENDSL

```

```

*
C          ENDDO
C          ENDSR
*-----*
*  Despliega Subfile No. 2.  *
*-----*
C          SR0020    BEGSR
C                   EXSR WRTSF2
*
C          *IN03     DOWEQ'0'
C          ASTER     ANDEQ'1'
C                   WRITESY@085HD
C                   WRITESY@085M2
C                   EXFMTSY@085C2
C                   CLEARPAMSG
C                   MOVEA'00000000'*IN,50
C                   Z-ADD*ZEROS    CONT    50
*
C          *IN03     IFEQ '1'
C          *IN12     OREQ '1'
C                   LEAVE
C                   ENDIF
*
*  Verifica si se hizo algun cambio al subarchivo.
*
C          READCSY@085S2          27
C          *IN27     IFEQ '0'
C                   Z-ADD1        CONT
C                   ENDIF
*
*  Verifica Datos de los Archivos a Enviar/Recibir.
*
C          Z-ADD1        NRR2
C          *IN27     DOUEQ'1'
C          NRR2      CHAINSY@085S2          27
C          27      LEAVE
*
C          S2SECU     IFNE *BLANKS
*
C          S2SECU     IFNE 'INIC'
C          S2SECU     ANDNE'RETO'
C                   MOVEA'11'          *IN,50
C                   MOVEAMSG,11        PAMSG
C                   LEAVE
C                   ENDIF
*
C          S2ORDE     IFLE *ZEROS
C                   MOVEA'101'          *IN,50
C                   MOVEAMSG,12        PAMSG
C                   LEAVE
C                   ENDIF
*
C          S2FRLB     IFEQ *BLANKS
C                   MOVEA'1001'          *IN,50
C                   MOVEAMSG,13        PAMSG
C                   LEAVE
C                   ENDIF
*
C          S2FRFI     IFEQ *BLANKS
C                   MOVEA'10001'          *IN,50
C                   MOVEAMSG,14        PAMSG
C                   LEAVE
C                   ENDIF

```

```

*
C      S2TOLB      IFEQ *BLANKS
C                      MOVEA'100001' *IN,50
C                      MOVEAMSG,15 PAMSG
C                      LEAVE
C                      ENDIF
*
C      S2TOFI      IFEQ *BLANKS
C                      MOVEA'1000001' *IN,50
C                      MOVEAMSG,16 PAMSG
C                      LEAVE
C                      ENDIF
*
C      S2OPCI      IFNE '*ADD'
C      S2OPCI      ANDNE '*REP'
C                      MOVEA'10000001'*IN,50
C                      MOVEAMSG,17 PAMSG
C                      LEAVE
C                      ENDIF
*
C                      ELSE
C                      CLEAR$2ORDE
C                      CLEAR$2FRLB
C                      CLEAR$2FRFI
C                      CLEAR$2TOLE
C                      CLEAR$2TOFI
C                      CLEAR$2OPCI
C                      ENDIF
*
C                      UPDATESY@085S2
C                      ADD 1 NRR2
C                      ENDDO
*
C      *IN50      IFEQ '1'
C                      UPDATESY@085S2
C                      ELSE
C      CONT      IFEQ *ZEROS
C                      EXSR SR0030
C                      ENDIF
C                      ENDIF
C                      ENDDO
C                      ENDSR
*-----*
*      Llena el Subfile No. 2.      *
*-----*
C      WRTSF2      BEGSR
C                      CLEARNRR2
C                      MOVEA'100' *IN,25
C                      WRITESY@085C2
C                      MOVEA 010' *IN,25
*
C                      Z-ADD*ZEROS NRR2
C      KEY020      SETLLSY@030R
C      KEY020      READESY@030R 27
*
C      *IN27      DOWEQ'0
C                      ADD 1 NRR2
C      NRR2      CHAINSY@085S2 27
C      27      LEAVE
*
C                      MOVE SFSECU S2SECU
C                      Z-ADDSFORDE S2ORDE
C                      MOVE SPFRLB S2FRLB

```

```

C          MOVE SFFRFI      S2FRFI
C          MOVE SFTOLB     S2TOLB
C          MOVE SFTOFI     S2TOFI
C          MOVE SFOPCI     S2OPCI
C          UPDATSY@085S2
*
C          KEY020          READESY@030R          27
C          ENDDO
C          ENDSR
*-----*
*  Despliega Subfile No. 3.  *
*-----*
C          SR0030          BEGSR
C          EXSR WRTSF3
*
C          *IN03           DOWEQ'0'
C          ASTER           ANDEQ'*'
C          WRITESY@085HD
C          WRITESY@085M1
C          EXFMYSY@085C3
C          CLEARPAMSG
C          MOVEA'0000'      *IN,50
C          Z-ADD*ZEROS     CONT
*
C          *IN03           IFEQ '1'
C          *IN12           OREQ '1'
C          LEAVE
C          ENDIF
*
*  Verifica si se hizo algun cambio al subarchivo.
*
C          READCSY@085S3          27
C          *IN27             IFEQ '0'
C          Z-ADD1            CONT
C          ENDIF
*
*  Verifica los Datos de la Lista de Librerias.
*
C          Z-ADD1            NRR3
C          *IN27             DOUEQ'1'
C          NRR3              CHAINSY@085S3          27
C          LEAVE
*
C          S3SECU           IFNE *BLANKS
*
C          S3SECU           IFNE 'INIC'
C          S3SECU           ANDNE'RETO'
C          MOVEA'11'        *IN,50
C          MOVEAMSG,11      PAMSG
C          LEAVE
C          ENDIF
*
C          S3ORDE           IFLE *ZEROS
C          MOVEA'101'        *IN,50
C          MOVEAMSG,12      PAMSG
C          LEAVE
C          ENDIF
*
C          S3LIBR           IFEQ *BLANKS
C          MOVEA'1001'        *IN,50
C          MOVEAMSG,18      PAMSG
C          LEAVE
C          ENDIF

```

```

*
C          ELSE
C          MOVE *BLANKS  S3ORDE
C          MOVE *BLANKS  S3LIBR
C          ENDIF
*
C          UPDATSY@085S3
C          ADD 1          NRR3
C          ENDDO
*
C          *INS0        IFEQ '1'
C          UPDATSY@085S3
C          ELSE
C          CONT        IFEQ *ZEROS
C          EXSR UPD020
C          EXSR UPD030
C          EXSR UPD040
C          MOVE *BLANK  ASTER
C          ENDIF
C          ENDIF
C          ENDDO
C          ENDSR
*-----*
*      Llena el Subfile No. 3
*-----*
C          WRTSF3      BEGSR
C          CLEARNRR3
C          MOVEA'100'   *IN,25
C          WRITESY@085C3
C          MOVEA'010'   *IN,25
*
C          Z-ADD*ZEROS  NRR3
C          KEY020      SETLLSY@C40R
C          KEY020      READESY@040R          27
*
C          *IN27      DOWEQ'0'
C          ADD 1          NRR3
C          NRR3        CHAINSY@085S3        27
C          27          LEAVE
*
C          MOVE SLSECU  S3SECU
C          Z-ADDSLORDE S3ORDE
C          MOVE SLLIBR  S3LIBR
C          UPDATSY@085S3
*
C          KEY020      READESY@040R          27
C          ENDDO
C          ENDSR
*-----*
*      Actualiza Archivo SY@020F.
*-----*
C          UPD020      BEGSR
C          SELEC
C          SFOPC       WHEQ '6'
C          SFOPC       OREQ '9'
C          WRITESY@020R
C          SFOPC       WHEQ '4'
C          DELETSY@020R
C          SFOPC       WHEQ '2'
C          UPDATSY@020R
C          ENDSL
C          ENDSR
*-----*

```

* Actualiza Archivo SY@030F. *

```

-----*
C      UPD030  BEGSR
C      SFOPC   IFNE '6'
C      SFOPC   ANDNE '9'
C      KEY020  SETLLSY@030R
C      KEY020  READESY@030R          27
C      *IN27   DOWEQ'0'
C      DELETSY@030R
C      KEY020  READESY@030R          27
C      ENDDO
C      ENDIF
*
C      SFOPC   IFNE '4'
C      Z-ADD1   NRR2
C      NRR2     CHAINSY@085S2        27
*
C      *IN27   DOWEQ'0'
C      S2SECU  IFNE *BLANKS
C      MOVE SHPROC  SFPROC
C      MOVE SHPLAN  SFPLAN
C      MOVE S2SECU  SFSECU
C      Z-ADDS2ORDE SFORDE
C      MOVE S2FRLB SFFRLB
C      MOVE S2FRFI SFFRFI
C      MOVE S2TOLE SFTOLE
C      MOVE S2TOFI SFTOFI
C      MOVE S2OPCI SFOPCI
C      WRITESY@030R
C      ENDIF
C      ADD 1     NRR2
C      NRR2     CHAINSY@085S2        27
C      ENDDO
*
C      ENDIF
C      ENDSR
-----*

```

* Actualiza Archivo SY@040F. *

```

-----*
C      UPD040  BEGSR
C      SFOPC   IFNE '6'
C      SFOPC   ANDNE '9'
C      KEY020  SETLLSY@040R
C      KEY020  READESY@040R          27
C      *IN27   DOWEQ'0'
C      DELETSY@040R
C      KEY020  READESY@040R          27
C      ENDDO
C      ENDIF
*
C      SFOPC   IFNE '4'
C      Z-ADD1   NRR3
C      NRR3     CHAINSY@085S3        27
*
C      *IN27   DOWEQ'0'
C      S3SECU  IFNE *BLANKS
C      MOVE SHPROC  SLPROC
C      MOVE SHPLAN  SLPLAN
C      MOVE S3SECU  SLSECU
C      Z-ADDS3ORDE SLORDE
C      MOVE S3LIBR  SLLIBR
C      WRITESY@040R
C      ENDIF

```

```

C          ADD 1          NRR3
C          NRR3          CHAINSY@085S3          27
C          ENDDO
*
C          ENDIF
C          ENDSR
*-----*
* Rutina de Inicializacion.
*-----*
C          *INZSR          BEGSR
C          KEYCTL          KLIST
C          KFLD          CTPROC
C          KFLD          CTPLAN
*
C          KEY020          KLIST
C          KFLD          WKPROC  6
C          KFLD          WKPLAN  2
C          ENDSR

```

** / Tabla de Mensajes.
 Clave de Proceso no debe ir en blancos.
 Clave de Planta no debe ir en blancos.
 Clave Duplicada.
 Registro Inexistente.
 Tipo de Proceso debe ser *LCL o *RMT
 Descripción del Proceso no debe ir en Blancos
 Campo From-Máquina no debe ir en blancos.
 Campo To-Máquina no debe ir en blancos.
 Campo Existe Proceso de Retorno debe ser 'SI' o Blanco
 Campo Existe Proceso de Retorno debe ser 'SI'
 Secuencia debe ser INIC o RETO.
 No. de Orden debe ser Mayor a Ceros.
 Campo From-Librería no debe ir en Blancos.
 Campo From-Archivo no debe ir en Blancos
 Campo To-Librería no debe ir en Blancos.
 Campo To-File no debe ir en Blancos
 Campo Adicionar o Reemplazar Registro debe ser *ADD ó *REP
 Nombre de la Librería no debe ir en Blanco.
 Existen problemas de comunicación, intente mas tarde.
 El copiado de regs. a sistemas remotos se efectuo satisfactoriamente

El programa SY@086R realiza las siguientes acciones:

- Copia información de los archivos del sistema a la máquina remota por clave de proceso.

```

*-----*
* PROGRAMA : SY@086R.
*          Copia Registro de Parametros a Sistemas Remotos.
*
* SISTEMA : Mapics/Costos.
* AUTOR   : Eduardo Martínez Salinas
* FECHA   :
*
* PROGRAMAS QUE LO LLAMAN : SY@086R
* HISTORIA DE MODIFICACIONES
*-----*

```

```

FSY@020F IF E K DISK
FSY@030F IF E K DISK
FSY@040F IF E K DISK
FSY@022F UF E K DISK A UC
F SY@020R KRENAMESY@022R A UC
FSY@032F UF E K DISK A UC
F SY@030R KRENAMESY@032R A UC
FSY@042F UF E K DISK A UC
F SY@040R KRENAMESY@042R A UC

```

```

*
ISY@020R
I SHPROC XXPROC
I SHPLAN XXPLAN
I SHTIPO XXTIPO
I SHDESP XXDESP
I SHFRMQ XXFRMQ
I SHTOMQ XXTOMQ
I SHPGM1 XXPGM1
I SHDDM1 XXDDM1
I SHRETO XXRETO
I SHPGM2 XXPGM2
I SHDDM2 XXDDM2

```

```

C OPEN SY@022F 50
C N50 OPEN SY@032F 50
C N50 OPEN SY@042F 50
C SELEC
C *IN50 WHEQ '0'
C EXSR UPD020
C EXSR UPD030
C EXSR UPD040
C CLOSESY@022F 36
C CLOSESY@032F 36
C CLOSESY@042F 36
C *IN50 WHEQ '1'
C MOVE 'ERR' WKERR
C ENDSL
C MOVE '1' *INLR

```

```

*-----*
* Actualiza Archivo SY@020F de Sistemas Remotos. *
*-----*
```

```

C UPD020 BEGSR
C KEY CHAINSY@020R 36
*
C *IN36 IFEQ '0'
C KEY CHAINSY@022R 36
C MOVELXXPROC SHPROC
C MOVELXXPLAN SHPLAN
C MOVELXXTIPO SHTIPO
C MOVELXXDESP SHDESP
C MOVELXXFRMQ SHFRMQ
C MOVELXXTOMQ SHTOMQ
C MOVELXXPGM1 SHPGM1
C MOVELXXDDM1 SHDDM1
C MOVELXXRETO SHRETO
C MOVELXXPGM2 SHPGM2
C MOVELXXDDM2 SHDDM2
C 36 WRITESY@022R
C N36 UPDATSY@022R
*
C ENDIF
C ENDSR

```

```

*-----*
```

```

* Actualiza Archivo SY@030F de Sistemas Remotos.
*
-----*
C          UPD030      BEGSR
*
* Elimina Registros de Sistemas Remotos.
C          KEY        SETLLSY@032R
C          KEY        READESY@032R          36
C          *IN36      DOWEQ'0'
C          DELETSY@032R
C          KEY        READESY@032R          36
C          ENDDO
*
* Actualiza Archivos de Sistemas Remotos.
C          KEY        SETLLSY@030R
C          KEY        READESY@030R          36
C          *IN36      DOWEQ'0'
C          WRITESY@032R
C          KEY        READESY@030R          36
C          ENDDO
*
C          ENDSR
*
-----*
* Actualiza Archivo SY@040F de Sistemas Remotos.
*
-----*
C          UPD040      BEGSR
*
* Elimina Registros de Sistemas Remotos.
C          KEY        SETLLSY@042R
C          KEY        READESY@042R          36
C          *IN36      DOWEQ'0'
C          DELETSY@042R
C          KEY        READESY@042R          36
C          ENDDO
*
* Actualiza Archivos de Sistemas Remotos.
C          KEY        SETLLSY@040R
C          KEY        READESY@040R          36
C          *IN36      DOWEQ'0'
C          WRITESY@042R
C          KEY        READESY@040R          36
C          ENDDO
*
C          ENDSR
*
-----*
* Rutina de Inicializacion.
*
-----*
C          *INZSR     BEGSR
C          KEY        KLIST
C          KFLD          WKPROC
C          KFLD          WKPLAN
C          *ENTRY     PLIST
C          PARM          WKPROC   6
C          PARM          WKPLAN   2
C          PARM          WKERR    3
C          ENDSR

```

IV.4 MANTENIMIENTO A LAS BASES DE DATOS DEL SISTEMA.

Una parte importante de este sistema es el mantenimiento a la base de datos, debido a que es ésta la que indica al sistema mismo los pasos a seguir, los archivos a enviar y recibir; y los procesos a correr. A continuación se presenta la pantalla principal del mantenimiento a la base de datos en la figura 4.10

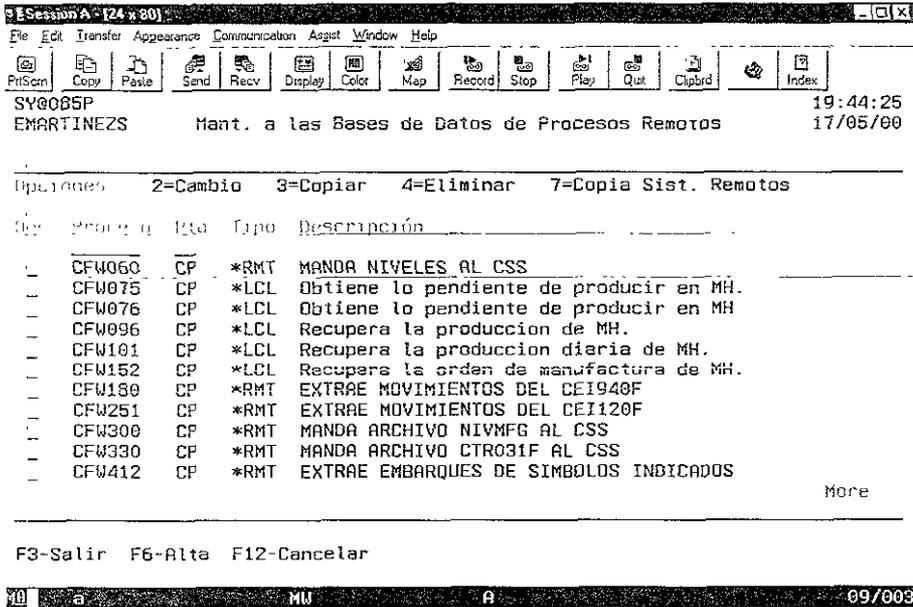


FIG. 4.10 Mantenimiento de claves de procesos remotos.

En esta pantalla se presenta la información general de cada proceso remoto, su clave, la planta, el tipo de proceso y su descripción. Se tienen diferentes opciones para el mantenimiento a esta base de datos como son cambio, copia, borrar, copiar al sistema remoto y funciones como par dar de alta nuevas claves. Por ejemplo, tomando el proceso CFW412, esta pantalla presenta la información general del mismo: la planta, si es un proceso remoto o local y su descripción.

En la pantalla de cambios de clave se puede dar mantenimiento al tipo de proceso (remoto o local), máquina local y remota, programa remoto, el cual se someterá en la máquina remota, DDM, identificador de existencia de proceso remoto de retorno, esto es si una vez terminado el proceso en la máquina remota se debe someter un programa en la máquina local, el nombre de este programa remoto de retorno si existe y su DDM de enlace. La figura 4.11 muestra esta pantalla de mantenimiento.

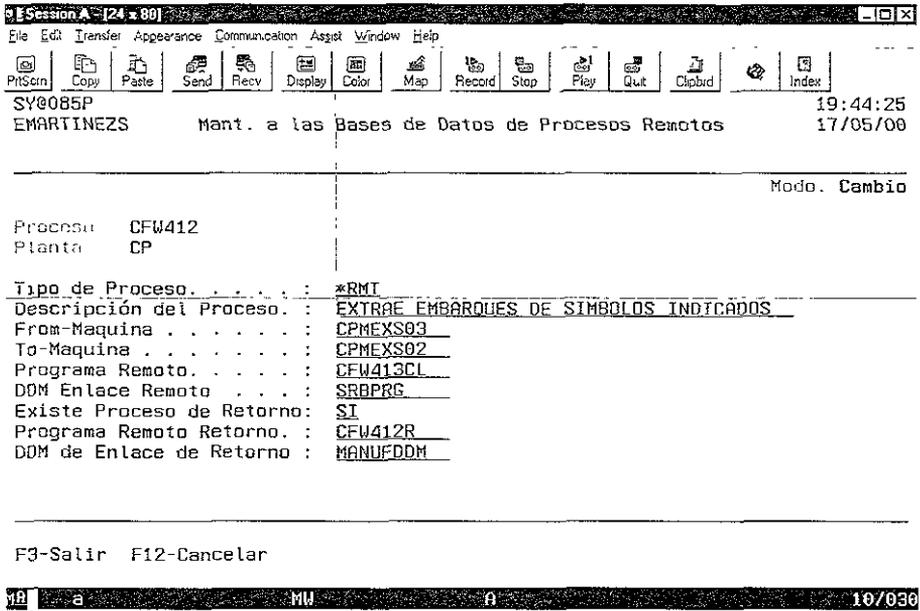


FIG. 4.11 Cambio de claves de procesos remotos.

Retomando el ejemplo del proceso CFW412, esta pantalla permite especificar si es un proceso local o remoto, una descripción del mismo, la máquina local, la máquina remota, el programa que se someterá en la máquina remota, el DDM que se utilizará para el

Para el ejemplo del proceso CFW412, la secuencia de inicio indica que se enviará el archivo CFW400F de la máquina local a la remota. Este archivo contiene los símbolos de los que se pretende extraer los embarques y las ventas para ser utilizado por el programa remoto que depositará la información en los archivos CFW414F y EMBARQUES, los cuales regresarán a la máquina local para ser utilizados por el programa CFW412R.

Otro punto importante es el mantenimiento a la lista de librerías con la que van a correr los procesos tanto en la máquina remota como en la máquina local una vez que regrese el proceso a ella. Esta lista de librerías es importante porque es en las únicas librerías donde el proceso buscare los objetos (programas, archivos, áreas de datos, etc.), y de no encontrarlos en estas terminará en error.

Se deben identificar las librerías que serán usadas en la máquina remota de las que serán usadas en la máquina local. Las de la máquina remota son marcadas con INIC en la clave de secuencia o identificador de librerías, mientras que las que serán usadas en la máquina local son marcadas con RETO.

En este caso el número de orden es muy importante ya este definirá que posición ocupará cada librería dentro de la lista de librerías. Esto es importante debido a que si un archivo existe en más de una librería, el programa lo tomará de la librería que tenga un número de orden menor. por lo que si este parámetro no es dado correctamente se puede tomar información no deseada.

La figura 4.13 nos muestra esta pantalla de mantenimiento.

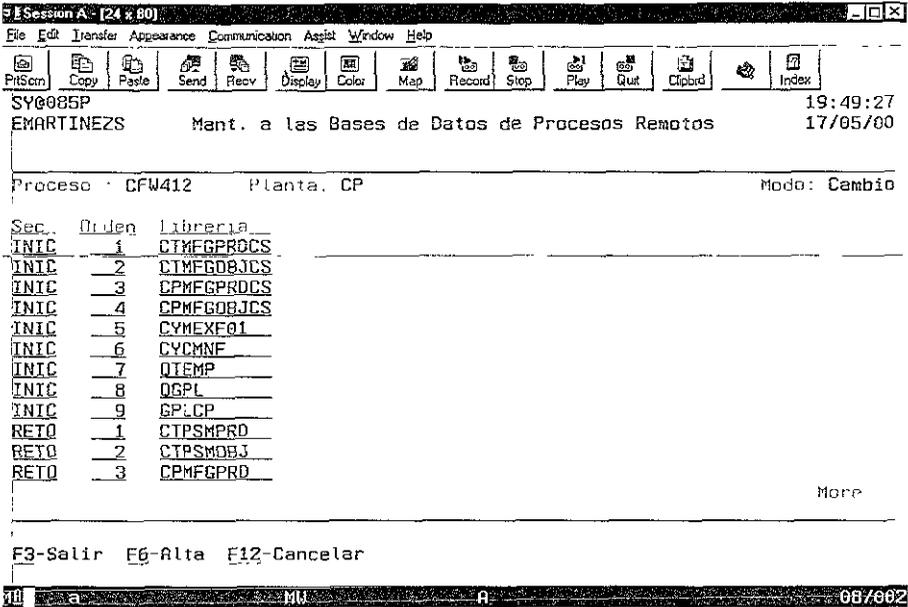


FIG. 4.13 Mantenimiento a lista de librerías

A su vez en el proceso CFW412, la secuencia INIC indica las librerías que formarán parte de la lista de librerías con que correrá el programa remoto. Por lo tanto, la secuencia RETO también señala las librerías que formarán parte de la lista de librerías con que correrá el programa local

Cabe mencionar que para el uso de parámetros en el llamado de programas, éstos se deben concatenar en una variable carácter de longitud 256 y llamar al programa inicial

del sistema con los siguientes parámetros: clave de proceso, planta, forma de proceso (Batch o Interactivo) junto con la variable anterior. La figura 4.14 muestra la forma de concatenar y llamar al programa.

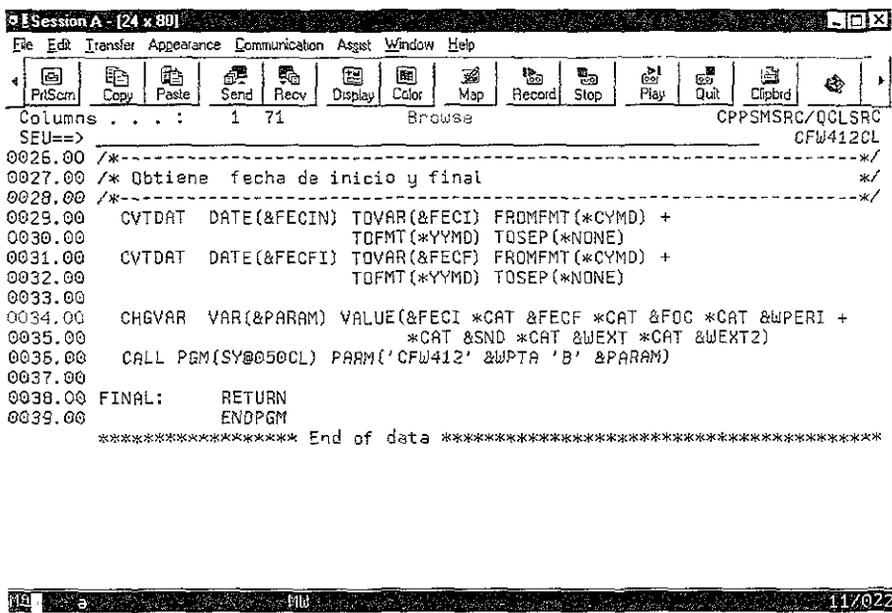


FIG. 4.14 Llamado al programa principal del sistema

En este ejemplo se envían, entre otros parámetros, una fecha inicial y una final para indicar en intervalo de ventas y embarques que se requieren.

Para utilizar los parámetros en los programas, se deben recuperar del área de datos SY@010D, separándolos en variables de la misma longitud y tipo de las originales que se concatenaron. La figura 4.15 muestra la forma de recuperar los parámetros.

```

Session B - [24 x 80]
File Edit Transfer Appearance Communication Assist Window Help
Columns . . . . : 1 71 Browse CPMFGSRCCS/QCLSRC
SEU=> CFW413CL
0019.00 /*-----*/
0020.00 /* Recupera parametros */
0021.00 /*-----*/
0022.00 RTVDTAARA DTAARA(QTEMP/SY@010D (1 8)) RTNVAR(&FECINI)
0023.00 RTVDTAARA DTAARA(QTEMP/SY@010D (9 8)) RTNVAR(&FECFIN)
0024.00 RTVDTAARA DTAARA(QTEMP/SY@010D (17 3)) RTNVAR(&FOC)
0025.00 RTVDTAARA DTAARA(QTEMP/SY@010D (32 1)) RTNVAR(&WEXT)
0026.00 RTVDTAARA DTAARA(QTEMP/SY@010D (33 1)) RTNVAR(&WEXT2)
0027.00
0028.00 CRTDUPBEJ OBJ(CFW412F) FROMLIB(CPMFGPRDCS) OBJTYPE(*FILE) +
0029.00 TOLIB(QTEMP) NEWOBJ(EMBARQUES)
0030.00 CRTDUPOBJ OBJ(CFW473F) FROMLIB(CPMFGPRDCS) OBJTYPE(*FILE) +
0031.00 TOLIB(QTEMP) NEWOBJ(EXTSROSRO)
0032.00 /*-----*/
0033.00 /* Selecciona ordenes de compra */
0034.00 /*-----*/
0035.00 IF COND(&WEXT = 'S') THEN(DD)
0036.00 QVRDBF FILE(SRGWTT) TOFILE(SRGWTT) SHARE(*YES)
0037.00 /*-----*/
0038.00 /* Selecciona ordenes de compra */
0039.00 /* WTPQTY,WTSHT */

```

FIG. 4.15 Recuperación de parámetros

En la pantalla anterior se recupera la fecha inicial de las primeras ocho posiciones del área de datos SY@010D así como la fecha final de las siguientes ocho posiciones.

CONCLUSIONES

CONCLUSIONES

Con este sistema se optimizan recursos ya que al aumentar la rapidez en la transferencia de información, se disminuye el tiempo de procesamiento de las aplicaciones, lo cual le permite a la máquina atender otras tareas aumentando de esta forma su eficiencia.

Esto es posible a través del desarrollo de dicho sistema de transferencia de información, que proporciona a los desarrolladores, que en este caso fungen como usuarios, la posibilidad de solucionar eficazmente las tareas demandadas a lo largo de los procesos diversos que se les encomiendan sin tener que utilizar códigos complejos o largos para ello en sus programas.

El sistema permite enviar archivos con campos empacados, a diferencia del FTP (File Transfer Protocol), debido a que dichos archivos son muy comunes el FTP pierde campo de aplicación.

El sistema también permite un mejor monitoreo de los mensajes de error de la transferencia; lo que permite un óptimo control del mismo.

A su vez, el sistema permite pasar el control de una máquina a otra para el procesamiento de datos en ambas; esto es de gran utilidad debido a que un proceso puede iniciar en la máquina local, después pasar el control a la máquina remota para realizar una tarea en

ella (por ejemplo una selección de la información necesaria), y finalmente continuar con el proceso en la máquina origen.

El Data Mirror nos permite tener la información en ambas máquinas teniendo de esta manera la opción de leerla en cualquiera de las dos; no siendo necesario de esta manera, transmitir información de una máquina a otra, o de tener el retraso debido a una lectura remota del mismo. Sin embargo, se necesita tiempo de proceso de la máquina para estar replicando la información de una máquina a otra, lo cual, se ejecuta independientemente de si se necesita o no la información, y al tener esta información en ambas se está utilizando mucho espacio en disco lo que también representa una desventaja. Con el sistema solo se procesa el requerimiento de información cuando ésta se requiere, y al tener la posibilidad de solo traer aquella que se necesita, se reduce de gran manera la información duplicada en ambas máquinas.

Con volúmenes grandes de información, el sistema tiene un mejor desempeño al lado del DDM (Distributed Data Management), el cual necesita una mayor transferencia de mensajes de control entre máquinas.

Otra ventaja se presenta cuando es necesario juntar los procesos e información de ambas máquinas en una sola por alguna contingencia. Para realizar esto con programas que utilizan el sistema, basta con modificar solo unos pocos datos del mismo, a diferencia de cuando se utiliza el DDM con el cual, se necesita borrar objetos y posteriormente volverlos a crear.

Sin embargo, a pesar de que este sistema tiene un excelente desempeño cuando se manejan grandes niveles de información, se pueden obtener mejores resultados con el uso del DDM cuando el volumen de información es muy pequeño.

Además, cuando se requiere hacer actualizaciones en línea, por ejemplo, en las transacciones donde se requiere bloqueo de registros, es necesario el uso del DDM; debido a que el sistema no realiza este bloqueo siendo posible que la información carezca de la integridad necesaria.

Este sistema tiene otra aplicación, cuando se tienen en la misma máquina información para diferentes plantas o compañías, la cual se almacena en diferentes juegos de librerías. En este caso se puede usar el sistema para seleccionar información de la otra planta aún cuando ambas están en la misma máquina, el proceso cambia las librerías por las de la información de la otra planta, corre el proceso requerido y vuelve a cambiar las librerías por las originales regresando al proceso original. Este es un valor agregado del sistema que facilita ciertas tareas a los programadores al desarrollar sus aplicaciones.

Finalmente, este sistema ayuda a los desarrolladores y responsables de los sistemas al permitirles el desarrollo de aplicaciones más veloces de una manera sencilla y rápida en sus programas, flexibilidad para cambios, contingencias y algunas otras aplicaciones.

También éste ayuda a los usuarios finales a tener mejor tiempo de respuesta en sus procesos y a un mejor aprovechamiento de los recursos del AS/400

BIBLIOGRAFÍA

- MARTYN, TIM y HARTLEY TIM. DB2/SQL Manual para programadores. Incluye SQL/DS, MCGRAW HILL, 1ª edición, ESPAÑA, 1991.
- INTERNATIONAL BUSSINES MACHINES (IBM), Técnicas de Procesamiento de Datos, 1ª edición, 1973.
- INTERNATIONAL BUSSINES MACHINES (IBM), Application System/400. Communications: Remote Work Station Guide versión 2, 3ª edición, ESTADOS UNIDOS, Noviembre 1993.
- INTERNATIONAL BUSSINES MACHINES (IBM), Application System/400. Communications: Management Guide versión 2, 3ª edición , ESTADOS UNIDOS, Noviembre 1993.
- INTERNATIONAL BUSSINES MACHINES (IBM), Application System/400. Central Site Distribution Guide versión 2, 3ª edición, ESTADOS UNIDOS, Noviembre 1993.
- INTERNATIONAL BUSSINES MACHINES (IBM), Application System/400. Programming Work Management Guide versión 2, 3ª edición, ESTADOS UNIDOS, Noviembre 1993.
- EMPRESAS/400, No. 17, Febrero-Marzo 2000.
- INTERNATIONAL BUSSINES MACHINES (IBM), Application System/400. Languages: Systems Application Architecture AD/Cycle RPG/400 Reference versión 2, 1ª edición, ESTADOS UNIDOS, Mayo 1991.
- INTERNATIONAL BUSSINES MACHINES (IBM), Application System/400. Communications: Distribution Services Network Administrator's Guide, 3ª edición, ESTADOS UNIDOS, Agosto 1990.

- INTERNATIONAL BUSSINES MACHINES (IBM), Application System/400, Communications: Distributed Data Management User's Guide, 2ª edición, ESTADOS UNIDOS, Septiembre 1989.
- INTERNATIONAL BUSSINES MACHINES (IBM), Application System/400, Programming Control Language Reference Volume 1, 3ª edición, ESTADOS UNIDOS, Agosto 1990.