INSTITUTO PROFESIONAL DE SANTIAGO
ESCUELA DE INFORMATICA
INGENIERIA DE EJECUCION EN COMPUTACION
E INFORMATICA

INFORME FINAL DE PRACTICA PROFESIONAL PARA OPTAR
AL TITULO DE INGENIERO DE EJECUCION
EN COMPUTACION E INFORMATICA

AXEL HEKENKOTTER MARTIN

TITULO DEL INFORME :

CONVERSION DE SISTEMAS DE DATA GENERAL A WANG - VS

PRACTICA REALIZADA EN LA EMPRESA :
SISTEMAS DE COMPUTACION S. A. "SISTECO"

Compilador Cobol VS
Enformat (Generador de Pantallas)
Report (Generador de Listados)
Tapecopy (Cobgen Generador De Programas Cobol)
Disprint (Permite Ver Listados en Pantalla)
Control (Generan Una Descripción de Archivo)

Profesor Guía : Sr. Jorge Braghiroli R.
Equipos Utilizados : Wang VS-65 y Data General 130
Principal Software Utilizado :

05 ENERO DE 1986

INSTITUTO PROFESIONAL DE SANTIAGO.

ESCUELA DE INFORMATICA.

Carrera de Ingeniería de Ejecución en Computación e Informática

Informe final de práctica profesional para optar al título de Ingeniero de Ejecución en computación e Informática.

AXEL HEILENKOTTER MARTIN.

N#. de matrícula : 247355-24.

Título del informe :

CONVERSION DE SISTEMAS DE DATA-GENERAL A WANG-VS.

Práctica realizada en la empresa :

SISTEMAS DE COMPUTACION S.A. "SISTECO".

Profesor Guía: Sr. Jorge Braghiroli R.

Equipos utilizados : Wang VS-65 y Data General 120.

Principal Software utilizado :

- Compilador COBOL-VS
  EZFORMAT (generador de pantallas)
  REPORT (generador de listados)
  TAPECOPY (copiador de cintas)
  COBGEN (generador de programas cobol)
  DISPRINT (permite ver listados en pantalla).
  CONTROL (genera una descripción de archivo).

Fecha presentación del informe terminado: 05 de Enero de 1986.

INDICE DE MATERIAS

INTRODUCCION.

Mi práctica profesional fué realizada en la empresa Sistemas de Computación S.A. "SISTECO", ubicada en Av. Vicuña Mackenna 152, Santiago. La principal actividad de esta empresa es representar en Chile los computadores WANG de procedencia norteamericana, además de esto es reprensentante de aproximadamente 100 productos más, dentro de los cuales se encuentran las impresoras C.ITOH, los medios magnéticos MEMOREX, los cajeros automáticos DIEBOL, los terminales bancarios BUNKERS RAMO, los modems PARADINE y los computadores NORAND.
Dentro de la cartera de clientes de "SISTECO" algunos de los mas importantes son; La Bolsa de Comercio de Santiago, ESSO Chile, EMPORCHI (Empresa Portuaria de Chile) y La Chilena Consolidada, en donde se instaló el primer equipo WANG VS-300 el equipo mas grande de la línea VS.

La práctica me tocó desarrollarla específicamente en el departamento de SOPORTE A CLIENTES dependiente de la SUB-GERENCIA DE SOFTWARE. Esta división de la empresa es la que se encarga de solucionarle al cliente todos los problemas que se relacionen con el software, ya sea de sistema operativo o de programas y sistemas de aplicación.

El proyecto en el cual participé fué una conversión de sistemas desarollados en Cobol, desde un equipo DATA GENERAL a un equipo WANG VS-65 que fué adquirido por la CLINICA LAS CONDES S.A..
La cantidad de programas a convertir fué del orden de 450, que forman parte de 14 sistemas entre los cuales estan contabilidad, existencias, admisión pacientes, cuenta corriente pacientes y estadísticas de producción.
El grupo de trabajo que fué asignado para el proyecto estuvo formado por 4 personas trabajando a tiempo completo y un jefe de proyecto encargado de supervisar y administrar los recursos asignados.

El antiguo equipo de la CLINICA LAS CONDES S.A. era un DATA GENERAL 120 con 256 K.byte de memoria real, 2 unidades de disco de 40 Mega bytes c/u y una unidad de cinta, y el nuevo equipo adquirido un WANG VS-65 con 1 Mega bytes de memoria real, 4 unidades de disco, dos fijas de 70 Mega bytes c/u y otras dos removibles de 170 Mega bytes c/u.
Una de las mayores diferencias entre ambos equipos, la cual es una ventaja de la VS-65 es que los computadores WANG son totalmente orientados al desarollo y explotación de sistemas por lo cual todo su software esta orientado a tales áreas y además generalmente orientado a lenguaje Cobol.

Los programas a convertir fueron recibidos en cintas grabadas en el equipo DATA GENERAL y leidas en SISTECO en un equipo WANG

VS-100 para copiarlas a un disco donde se almacenaban todas las cintas con programas enviadas por la clínica, luego de esto los programas eran distribuidos entre los miembros del grupo de conversión para su modificación a Cobol-VS.

Los mayores problemas encontrados durante la conversión de los programas fué que no se contaba con una documentación adecuada y actualizada de los sistemas, por lo cual en muchos casos había que consultar directamente a los programadores de la clínica sobre detalles de los programas. Otro problema se sucitó ya que en muchos casos las cintas enviadas por la clínica con programas venían defectuosas al igual que ciertos archivos de datos.
Para solucionar estos últimos problemas enunciados se debió desarrollar algunos programas para eliminar ciertos carácteres de control de los programas y para arreglar los archivos de datos, ya que estos también debieron ser convertidos. Todos estos problemas influyeron en el avance normal del projecto.

Los puntos más rescatables en cuanto a la experiencia adquirída durante esta práctica son; la necesidad de contar con una buena documentación de programas y sistemas, la importancia de la programación estructurada y ordenada y la realización de una adecuada programación de actividades tomando en cuenta los numerosos imprevistos que se pueden producir durante el desarrollo de un proyecto.

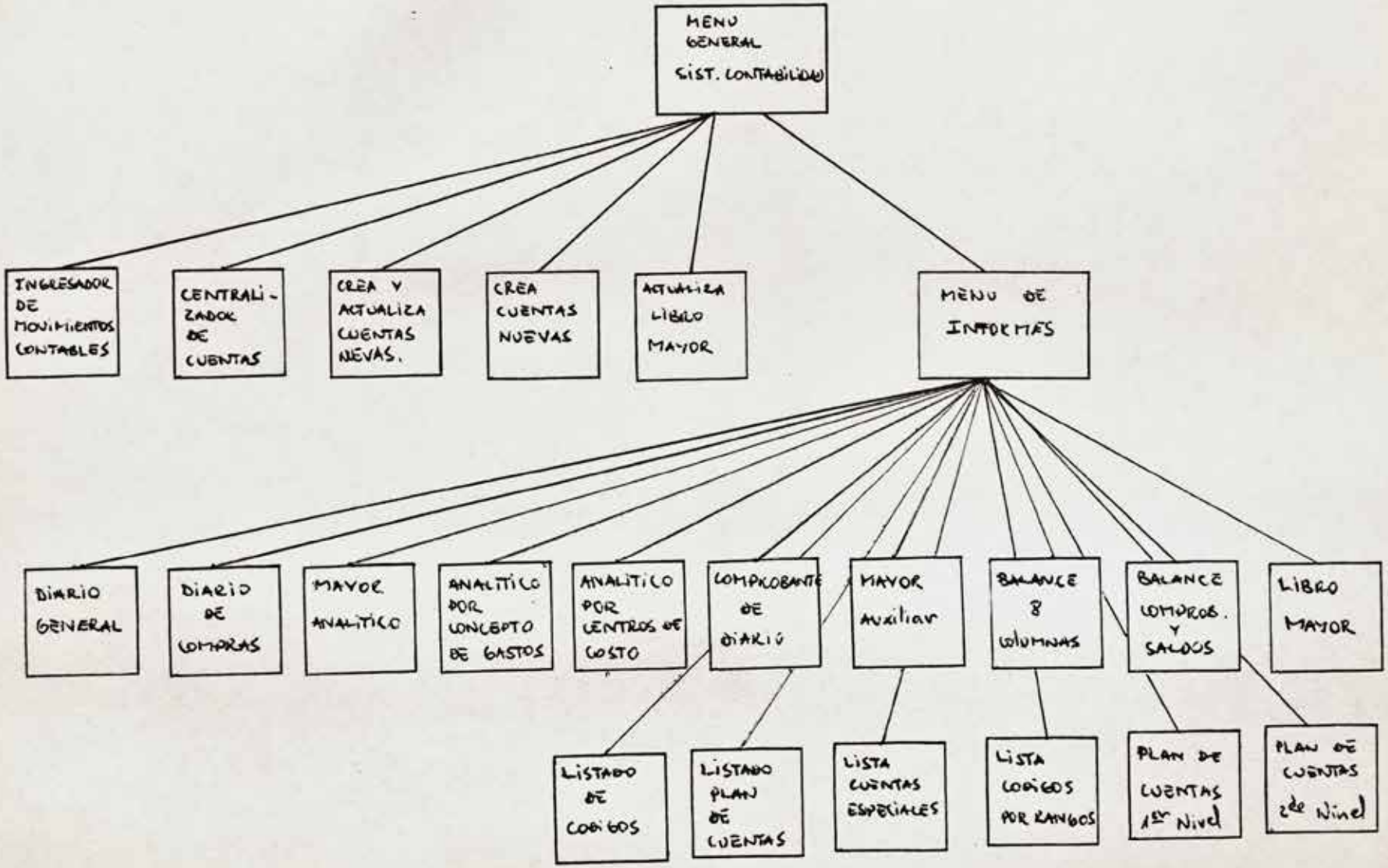
DESCRIPCION DETALLADA DE LOS SISTEMAS CONVERTIDOS.

2.1- Sistema de contabilidad.

Este fué el primer sistema con el cual se dió comienzo a la conversión de sistemas para la clínica por lo cual fué en donde se presentaron la mayor cantidad de problemas por la poca y mas bien inexistente experiencia en transformar programas desde un equipo totalmente desconocido para nuestro grupo de trabajo como lo era el DATA GENERAL, hacia otro equipo en este caso desconocido para mi como lo era en esos instantes el WANG VS-65. Otro punto relevante en la complejidad de la conversión de estos programas fué que una gran cantidad de estos venían ya desde la clínica con problemas, es decir muchos de ellos era totalmente imposible que funcionaran. Como por ejemplo un caso típico de error que se encontró fué que en muchos casos se movian espacios a variables declaradas como numéricas y vice versa, otra deficiencia era que en distintos programas, archivos iguales diferían en la cantidad de caracteres de la definición del archivo, habían programas que utilizaban un archivo X con 232 caracteres y otros programas que utilizaban el mismo archivo X con 250 caracteres.

El sistema de contabilidad consta de 95 programas entre los cuales encontramos ingresadores de datos , listadores, y programas que realizan otros procesos.
Este sistema de contabilidad sale un poco de lo tradicional por algunos detalles especiales que este posee. Estos son; el gran volúmen de datos manejados, existe un programa cuya función es centralizar ciertas cuentas para con estos poder reducir el volúmen de información. Esta centralización consiste en que para ciertos códigos de cuentas específicos se crea una nueva cuenta en donde se registran todos estos. Otra característica un poco especial es que este sistema es capaz de procesar todas las sociedades por las cuales se compone la clínica, (6 sociedades) y de este motivo todos los archivos de datos del sistema de contabilidad estan en 6 versiones una para cada sociedad. Esto último no significa que también se sextuplique el tamaño de los archivos, ya que existen varias sociedades que solo tienen dos o tres movimientos contables por período.
Referente a los listados que emite el sistema, algunos de estos son; listado mayor analítico, analítico por concepto de gastos, analítico por centro de costo, balance a ocho columnas y libro mayor. En el diagrama de bloques (figura 1) es posible visualizar la totalidad de los informes que emite el sistema.

FIGURA 1


MENU GENERAL SIST. CONTABILIDAD
INGRESADOR DE MOVIMIENTOS CONTABLES
CENTRALIZADOR DE CUENTAS
CREA Y ACTUALIZA CUENTAS NUEVAS.
CREA CUENTAS NUEVAS
ACTUALIZA LIBRO MAYOR
MENU DE INFORMES
DIARIO GENERAL
DIARIO DE COMPRAS
MAYOR ANALITICO
ANALITICO POR CONCEPTO DE GASTOS
ANALITICO POR CENTROS DE COSTO
COMPROBANTE DE DIARIO
MAYOR Auxiliar
BALANCE 8 COLUMNAS
BALANCE COMPROB. Y SALDOS
LIBRO MAYOR
LISTADO DE CODIGOS
LISTADO PLAN DE CUENTAS
LISTA CUENTAS ESPECIALES
LISTA CODIGOS POR RANGOS
PLAN DE CUENTAS 1er Nivel
PLAN DE CUENTAS 2de Nivel

2.2- Sistema de existencias.

La conversión de este sistema resultó ser un poco mas sencilla que la del sistema anterior (contabilidad), ya que se contaba con un poco más de experiencia y en mi caso particular ya podía manejar completamente y de manera eficiente las capacidades que provee la VS-65 para el desarrollo de aplicaciones en lenguaje Cobol.

Este sistema consta de 65 programas en donde los mas complejos son los ingresadores de datos, esto por la gran cantidad de tipo de movimientos diferentes que se manejan, como ingreso a bodega, devolución a bodega, ajuste por salida, ajuste por ingreso, devolución a proveedor, devolución de centro, etc..., y para cada uno de estos tipos de movimientos existe un ingresador diferente, que en algunos casos son totalmente distintos unos de otros.

Una característica resaltante de este sistema es la inmensa cantidad de items de inventario que se manejan, esta es de aproximadamente 10.000.

Este sistema se encuentra en constante funcionamiento en la clínica, ya que por el gran movimiento de productos sería imposible digitar y procesar los movimientos con una frecuencia menor, cada dos o tres días por ejemplo.

Otro punto importante dentro de este sistema, es que este maneja 6 bodegas diferentes,esto debido a la gran cantidad y mas que nada a la gran diferencia de tipos de productos que se manejan en la clínica, como por ejemplo materiales de oficina y plasma. Estos productos se encuentran separados por tipos en las diferentes bodegas. Se maneja una bodega de materiales quirúrgicos, otra de productos que deben estar refrigerados, etc...

2.3- Sistema de estadísticas de producción.

Este fué el tercer sistema que se convirtió, de acuerdo a una lista de prioridades que fué entregada por la clínica en las reuniones preliminares antes de comenzar el proyecto en sí.
El sistema de estadísticas de producción es uno de los sistemas no tradicionales que se utilizan en la clínica actualmente.
Este esta compuesto por 33 programas dentro de los cuales se cuentan 4 ingresadores de datos, donde los dos mas importantes son: el ingresador de las planillas de consultas de los médicos y el otro el ingresador de las planillas de cargos varios.
El propósito prinicipal de este sistema mas que llevar una estadística de la producción de los médicos, es que este sirve prioritariamente, para mediante la determinación de la producción de cada médico poder cancelar a cada uno de ellos sus honorarios profesionales por todos los tipos de atenciones que éste realizó (intervenciones quirúrgicas, consultas, tratamientos, etc...). Otro tipo de información que entrega este sistema son, listados resumen de produccción por centro de costo, producción en pabellones por tipo de intervención, un cuadro estadístico, listados de prestaciones por especialidad, listado de examenes, etc...

La conversión de este sistema resultó bastante menos costosa en cuanto que la clínica de comun acuerdo con SISTECO resolvió enviar a trabajar en la conversión de cada sistema a una persona de la clínica, con el objeto de agilizar los posibles problemas y consultas que se sucitaran y tambien para que los futuros usuarios de estos sistemas se fueran acostumbrando al nuevo equipo e interiorizando de los cambios realizados en los programas, esto último debido a que en algunos casos por la complejidad de los programas resultó menos costoso desarrollarlos nuevamente y no tratar de convertirlos. Generalmente los programas ingresadores de datos resultó mas fácil y rápido realizarlos nuevamente solo tomando como referencia del antiguo algunas partes, como las validaciones y subrutinas de grabación.

2.4- Sistema de admisión de pacientes.

Como su nombre lo indica este sistema se utiliza para registrar los pacientes que ingresan a la clínica. Este es un sistema que está en línea por el sencillo hecho que los pacientes pueden llegar en cualquier momento y para tal efecto es necesario saber que habitaciones estan disponibles.
Este sistema consta de 36 programas en donde uno solo de ellos es un ingresador por el cual se registran los datos de los pacientes ingresados, datos tales como, nombre del paciente, fecha de ingreso, edad, sexo, habitación en la que se encuentra, hora de ingreso, hora y fecha de egreso, estado civil, nombre y dirección de la persona responsable por dicho paciente, etc...
Se manejan con este sistema dos archivos maestros, uno el maestro de pacientes y otro el maestro de pacientes ambulatorio. En este último se registran todos los pacientes a los cuales se les realizan intervenciones menores, exámenes y consultas.
La principal información que entrega el sistema son listados de, maestro de habitaciones, censo de pacientes, informe de traslados, informe del porcentaje de ocupación de cada habitación, informe de disponibilidad de camas, ingresos del día, altas del día y censo por unidad de tratamiento.

El tiempo empleado en la conversión de estos 36 programas fué relativamente corto, por el hecho que solo uno de ellos era un ingresador de datos, que es el tipo de programa que mas cambios requiere, solo se emplearon 3 días para realizar la conversión de estos programas.
Obviamente la conversión de los programas no es realmente la parte que mas trabajo demanda ni la mas costosa del proyecto, posteriormente a la conversión de todos los sistemas viene una etapa de depuración final de estos y de puesta en marcha, en donde se deberán probar con datos reales y en forma integrada todos los sistemas para poder detectar los posibles problemas y defectos que aun tengan los programas.

2.5- Sistema de cuenta corriente pacientes.

El sistema de cuenta corriente de pacientes resultó ser uno de los mas complejos para la conversión, hasta el momento, ya que muchos de sus programas son ingresadores de datos muy sofisticados. Los dos ingresadores de datos mas importantes de este sistema son el ingreasador de TCP (tarjetas de cargo a paciente) y el ingresador de reversas de TCP, que son programas de aproximadamente 2500 líneas, que realizan la función específica de registrar todos los medicamentos, apositos, comida, y otros que son suministrados a cada paciente durante su estadía en la clínica.
Es decir si a un paciente lo van a intervenir, el médico solicita 15 jeringas pero solo se ocupan 10, las 15 son cargadas a la cuente corriente del paciente mediante el ingresador de TCP y las 5 que no se utilizaron son descontadas utilizando el ingresador de reversas de TCP. Esta última operación de los registros de las TCP se efectúa diariamente ya que este sistema alimenta por medio del archivo de cargos a un sistema de pre-facturación mediante el cual se emite día a día una pre-factura que es enviada al paciente o a su responsable todos los días para que este tenga conocimiento del monto de dinero que lleva acumulado hasta el momento, esto para que no se produzca que una persona no pueda cancelar la cuenta final al dársele de alta.

PROBLEMAS QUE SE PRESENTARON DURANTE LA CONVERSION.

3.1- Coordinación del grupo de trabajo.

Al comenzar la conversión se sucitaron diversos problemas ya que era la primera vez que los que formabamos el grupo de conversión, trabajabamos en un proyecto de tal especie y en grupo donde se debía interactuar lo más posible, en cuanto a ayudarse a solucionar problemas que se fueran sucitando a medida que se transformaran los programas y los archivos de datos.
Como la labor del jefe de proyecto era la de coordinar la relacion entre SISTECO y la Clínica Las Condes en cuanto a obtener los archivos de datos, las cintas con programas y ver el plan de actividades para realizar la conversión, toda la labor de coordinación interna del grupo de trabajo, la distribución de los programas, la medición de ciertos indicadores para poder determinar con exactitud las fechas de término de la conversión de cada sistema con el objeto de solicitar con cierta anticipación los programas del sistema que se debía convertir a continuación, tuvo que ser asumida por nosotros mismos.

El primer sistema que convertimos fué el sistema de Contabilidad. Durante la conversión de este se produjo un atraso en las fechas estimadas de término para la conversión de este sistema, ya que en ese momento no contabamos con la experiencia previa de este tipo de trabajo, ni con ninguna herramienta que nos permitiera organizarnos para aumentar la productividad de conversión. Se tenía estimado que para cumplir con la fecha de entrega de la conversión terminada había que transformar dos programas diarios, meta que estuvo muy lejos durante la conversión del primer sistema.

Ya a partir del segundo sistema a convertir (existencias), comenzaron a surguir ciertas ideas para ordenarse un poco, porque nadie sabía donde estaban los programas a convertir, en que librería, como se llamaban y en donde estaban los programas ya terminados y listos para pasar a la etapa de prueba y depuración. A todo este desorden interno se sumaba el desorden que había por parte del cliente, en cuanto a la documentación que era prácticamente inexistente, aparte de unos diagramas de bloques y algunas especificaciones de archivos, y la continua demanda de pequeños cambios en los programas, los cuales derivaban en algunos casos en tener que hacer un programa completamente denuevo.

La primera acción que se llevó a cabo para ordenarse en cuanto a los programas a convertir, fué hacer un programa en Cobol (ver anexo 5) para separar los programas y obtener un listado con sus nombres, tamaños y en que librería se encontraban. Los programas había que separarlos ya que estos venían en cintas desde la clínica que contenían un solo archivo donde se encontraban todos los programas a convertir uno tras el otro. Originalmente,

durante la conversión del primer sistema esta separación la hacía cada persona del grupo copiando en el editor la cantidad de líneas que correspondían al programa a convertir desde el archivo que se obtenía de la cinta en donde estaban todos los programas.
Luego tratamos de organizar las librerías en donde ibamos a trabajar. Se crearon varias librerías, la CLCFTE , CLC para identificar el nombre Clínica Las Condes y FTE para identificar que se trataban de programas fuente. En esta librería se debían poner todos los programas fuentes en Cobol que se iban a convertir. La CLCOBJ en donde estarían todos los programas objetos (ejecutables) de todos los sistemas. La CLCORIG en la que se encontrarían los archivos originales correspondientes a las cintas que enviaba la clínica. La CLCLISTx, en estas librerías se deberían dejar los programas totalmente terminados después de su etapa de conversión, la letra minuscula x debía ser la primera letra del nombre del sistema al cual pertenecían los programas. Por ejemplo la librería CLCLISTE contenía los programas listos de existencias. Las últimas dos librerías que se crearon fueron la CLCDATA y la CLCINFO. En la CLCDATA estarían todos los archivos de datos que se enviaban de la clínica y los archivos que generaban los programas al ser ejecutados. La otra librería la CLCINFO sería una librería de impresión, aqui estarían todos los informes impresos que emitirían los programas. De esta manera se pudo organizar y racionalizar el uso del disco y ordenar los nombres de las bibliotecas para que cada integrante del grupo supiera que contenía cada biblioteca. Ya que al comienzo cada persona sabía que había en sus propias bibliotecas y para buscar un programa había que pasar por la mayoría de ellas hasta dar con el.

Otra de las cosas que se hicieron para organizar el grupo de trabajo fué que al comienzo de la conversión de un sistema, se decidió asignar un responsable de dicho sistema. Esto implicaba que esta persona debía distribuir y asignar los programas a convertir a cada persona, darle nombre a la librería en donde quedarían los programas terminados, listos para ser probados por el usuario y de solicitar al jefe de proyecto los archivos de datos necesarios para probar superficialmente los programas, debido a que la prueba final debía ser en presencia del usuario.
En relación con el punto anteriormente enunciado sobre las pruebas frente al cliente y la contínua demanda de modificaciones a los programas, se diseño un pequeño formaulario de aceptación de programas en donde al momento de probarse el programa en frente al usuario final este debía poner en dicho formulario los cambios que se le debían de hacer o bien poner la firma para dar por aprobado definitivamente el programa. Este documento tuvo dos utilidades. La primera fué que sirvió para cuando el usuario hacía ver que la conversión iba muy lenta poder demostrarle que se habían producido retrasos debido a que se había solicitado la modificación de ciertos programas específicos. Y la otra que se les dió, fué poder contar con un

documento firmado por el usuario en donde se comprometía a aceptadar el programa en tales condiciones, esto sirvió de respaldo para poder frenar un poco la demanda de continuos cambios en los programas por parte del mismo usuario.

Otro punto deficiente que se detectó fué que no se contaba con algun tipo de anotación en donde poder encontrar el nombre de la persona que había convertido un programa dado. Esto se hizo notar cuando comenzó la primera prueba de sistemas con participación del usuario final del sistema. En estas pruebas surgieron muchos programas con errores los cuales había que corregir, y quién mas indicado para hacerlo que la propia persona que había convertido tal programa. De allí la necesidad de contar el mencionado formulario donde se recapituló información como :

- nombre del programa a convertir.
- fecha de entraga del programa al programador.
- fecha de término de la conversión del programa.
- nombre del programador que convirtió el programa.
- nombre del sistema a que pertenece el programa.

Este formulario era manejado por la persona que se asignaba como responsable del sistema.
A partir de la información de este formulario de control, se pudo extraer dos cosas más. Primero, como se anotaban en este las fechas de comienzo de la conversión del programa y la fecha de término, se pudo calcular un dato estadístico de cuanto tiempo se necesitaba para convertir cierto tipo de programa (listador, ingresador o de proceso solamente). Y por lo tanto con este tipo de información fué posible estimar a priori el tiempo, con un cierto margen de error, empleado en convertir un sistema de acuerdo a la cantidad y tipo de programas. Mas o menos hacia la mitad de la conversión los valores estimados para convertir un programa listador eran de entre 30 a 40 min., en cuanto que para un programa ingresador de datos eran de entre uno a uno y medio días. Esto último dependiendo de la cantidad de pantallas que el programa tuviera.
Como segundo punto de acuerdo a la información de este formulario de control, se pudo determinar que logrando una especialización del tipo de programa a convertir por cada integrante del grupo se podría aumentar la productividad. Por lo tanto unos integrantes se especializaron en ingresadores de datos otros en programas de proceso solamente y otros en programas listadores. Con esta medida se logro disminuir el tiempo de conversión de los listadores a un tiempo de entre 20 a 30 min. y para los ingresadores de entre 7 a 11 horas.

3.2- Problemas relativos a los programas.

Los programas enviados desde la clínica eran recibidos en cinta, ya que este era el único medio posible para tranpasar archivos desde el equipo DATA GENERAL al WANG-VS, pero de todas maneras este medio de transpaso no fué totalmente óptimo.
En la clínica uno de los editores que se utilizaban para desarrollar los programas, era un editor de texto que trabajaba con el texto como si este fuera un archivo indexado utilizando como llave el número de línea del editor. Además las líneas eran tomadas como registros de largo variable. Esto último fué el punto que generó serios problemas para el transpaso de las cintas entre los dos equipos. Como los registros eran de largo variable, el editor colocaba al final de cada línea un carácter de control que indicaba el fin del registro y además rellenaba este con algún carácter especial hasta completar los 80 caracteres de largo. Digo carácter especial, porque fué imposible detectar que carácter era el que en esas posiciones venía, ya que este en el equipo DATA GENERAL no tenía ninguna incidencia, ni generabas errores al momento de la compilación y en el WANG este no se veía en la pantalla, ni tenía representación alguna de acuerdo a la tabla de caracteres. Para poder solucionar este problema la gente de la clínica decidío desarrollar un programa para insertar espacios a todos los programas después del carácter de control de termino de cada línea. Esto funcionó bien para las primeras cintas que se empezaron a recibir después de detectado el problema, pero después comenzaron a llegar todas las cintas con programas cortados, con líneas sin puntos, otras sin paréntesis e instrucciones truncadas en la mitad. Como era obvio que este problema había que solucionarlo como fuera, ya que era un trabajo demasiado costoso el tener que pasar línea por línea de programa poniendo puntos o digitando lo que faltaba basandose en los listados de programas que se enviaban junto con los sistemas o bien teniendo que,tambien pasar línea por línea borrando todo lo que se suponía que había después del término de cada línea, desarrollamos un programa que tomando la cinta original enviada sin aplicarle el proceso que usaban en la clínica para supuestamente arreglar los programas, intercalara espacios a cada línea después del último carácter válido encontrado (ver programa anexo 6). Y con este programa logramos solucionar tal problema y por consiguiente ahorrar un tiempo y un trabajo que habría atrasado considerablemente el término del proyecto de conversión.

Otro problema que se presentó con los programas, fué que estos, en un gran porcentaje venían con fallas que en ningún tipo de computador podían ser compilados sin obtener errores. Como por ejemplo mover espacios a un campo declarado como numérico. El problema de esto no era en realidad el que se le movieran espacios a un campo numérico como en el ejemplo, sino que esto hacía suponer que los programas enviados no eran los que

realamente estaban siendo explotados en esos momentos en la clínica, y por lo tanto se estaba convirtiendo algo que no se iba a utilizar y que probablemente después, al momento de las pruebas no realizaría lo esperado. Luego de visto esto, se pidió que el cliente asegurara si los programas enviados eran realmente los programas que se estaban utilizando en la clínica en esos momentos para la explotación. Y la respuesta de éste fué que los programas enviados sí eran los que se estaban utilizado en esos momentos para la explotación de los sistemas. Por lo tanto durante toda la conversión, entre los integrantes del grupo, salía a relucir la pregunta un tanto irónica "funcionarían estos programas en el equipo de la clínica ? ".

A todos estos problemas anteriormente explicados se sumaba el problema que para cada sistema que se convirtió, o bien faltaban programas o sobraban programas. Esto es que con cada sistema se enviaba una copia del diagrama de bloques a nivel de programas del sistema y los programas recibidos en las cintas no concordaban con lo que en tales papeles aparecía. Algunas veces habían programas que no existían en el diagrama de bloques pero que sí venían el la cinta enviada, y por lo tanto no se sabía ni que es lo que hacían y otras veces habían programas que aparecían en el diagrama de bloques y que no venían en la cinta. Cada vez que se sucitaba este error se redactaba un informe, que era enviado a traves del jefe de proyecto a la clínica, de lo recibido en la cinta y de los programas que faltaban o sobraban según el diagrama de bloques, para que se nos informara qué era lo que realmente había que convertir.

Dentro de la conversión misma de los programas el mayor problema se presentó al convertir los programas ingresadores de datos. Esto debido a que la gran diferencia entre el lenguaje Cobol de los dos equipos es el manejo de pantallas. En el equipo DATA GENERAL las pantallas son desplegadas línea a línea independientemente una de otra como se aprecia en la definición de la SCREEN SECTION de un programa original en la figura 3.2.1. Esto tiene un gran inconveniente que también incide en el grado de dificultad y de entendimiento de un programa para su conversión, y es que como además de las pantalla se manejan teclas programadas, como las teclas PF en WANG, para poder saltar de una línea a otra en la pantalla, los programas desarrollados para la clínica eran totalmente faltos de estructuración utilizando una cantidad desmesurada de sentencias GO TO. La razón dada fué que se habían hecho de esa manera para no tener que hacer un algoritmo demasiado complicado para poder devolverse a la línea anterior de la pantalla. Se puede apreciar una parte de un programa sin convertir en la figura 3.2.2.
A diferencia de esto el equipo WANG-VS provee la facilidad de poder diseñar pantallas utilizando el utilitario EZFORMAT y generar con este mismo el trozo de código en lenguaje Cobol para ser insertado en la WORKING STORAGE de un programa, ver figura 3.2.3. Estas pantallas son desplegadas completas mediante una

FIGURA 3.2.1

```
NG VS INTEGRATED EDITOR - VERSION  6.13.01       11:39 12/29/86   PAGE      2
PUT FILE IS 00000013 IN LIBRARY CLCEX    ON VOLUME SISTCO

5800  01 MENU-2.
5900     02 BLANK SCREEN.
6000     02 LINE 1 COL 25 "INGRESO DE CODS. DE PROVEEDORES".
6100     02 LINE 2 COL 25 "**********************************".
6200  01 MENU-3.
6300     02 BLANK SCREEN.
6400     02 LINE 1 COL 25 "CORRECCION DE CODS. DE PROVEEDORES".
6500     02 LINE 2 COL 25 "**********************************".
6600  01 MENU-4.
6700     02 BLANK SCREEN.
6800     02 LINE 1 COL 25 "ELIMINA CODS. DE PROVEEDORES".
6900     02 LINE 2 COL 25 "**************************".
7000     02 LINE 6 COL 16 "INGRESE CODIGO DE PROV. A ELIMINAR: ".
7100     02 LINE 10 COL 16 "FIN = BREAK/ESC".
7200  01 ACEPTA-4.
7300     02 LINE 14 COL 38 TO CODIGO PIC X(4).
7400  01 MENU-GEN.
7500     02 LINE 5 COL 5 "1.- CODIGO   ".
7600     02 LINE 6 COL 5 "2.- DESCRIPCION  ".
7700  01 DAT-CORR.
7800     02 LINE 20 COL 5 "DATOS CORRECTOS? (SI=1/NO=2)".
7900  01 ACEPTA-D-CORR.
8000     02 LINE 20 COL 35 TO D-CORR PIC X.
8100  01 FIN-DAT.
8200     02 LINE 14 COL 5 "FIN = BREAK/ESC".
8300  01 MODIF.
8400     02 LINE 23 COL 5 "INGRESE NUMERO DE CAMPO A MODIFICAR : ".
8500  01 ACEPTA-MODIF.
8600     02 LINE 23 COL 43 TO NUM-LIN PIC 9.
8700  01 DAT-1.
8800     02 LINE 5 COL 30 TO CODIGO PIC X(4).
8900  01 DAT-2.
9000     02 LINE 6 COL 30 TO DESCRIPCION PIC X(30).
9100  01 MENS-E.
9200     02 LINE 24 COL 3 "NO EXISTE REGISTRO A ELIMINAR".
9300  01 DAT-01.
9400     02 LINE 5 COL 30 USING CODIGO PIC X(4).
9500  01 DAT-02.
9600     02 LINE 6 COL 30 USING DESCRIPCION PIC X(30).
9700  01 LIMPIA.
9800     02 LINE 24 COL 3 BLANK LINE.
9900  01 LIMPIA-PAG.
0000     02 BLANK SCREEN.
0100  PROCEDURE DIVISION.
0200  PRINCIPAL.
0300       DISPLAY "COMIENZA PROGRAMA EXPINO3X".
0400       OPEN I-O ARCH-PROV.
0500       PERFORM PROCESO UNTIL RESP-1 = 4.
0600       CLOSE ARCH-PROV.
0700       DISPLAY LIMPIA-PAG.
0800       DISPLAY "FIN PROGRAMA EXINPO3X".
0900       CALL PROGRAM "CLCOD/I".
1000
1100  PROCESO.
1200 *********
1300       MOVE 0 TO ESCAPE-CODE.
1400       DISPLAY MENU1.
```

FIGURA 3.2.2

```
G VS INTEGRATED EDITOR - VERSION  6.13.01        14:29 10/16/86   PAGE     25
UT FILE IS PRUEBA22 IN LIBRARY CLCFTE    ON VOLUME SOPVTA

200             IF ESCK = FK1
300                MOVE TBODREC (1) TO BOD-2
400                 DISPLAY MENU-BODREC2D
500             ELSE
600                IF ESCK = FK2
700                   MOVE TBODREC (1) TO BOD-2
800                   DISPLAY MENU-BODREC2D
900                   GO TO LEE-LINEA-MOD
000                ELSE
100                   DISPLAY MENU-BODREC2D
200                   PERFORM VALIDA-BODEGA-RECEPTORA.
300 *
400 *
500  SEL-OPCION-MODI.
600 *---------------
700 *
800 *
900          ACCEPT MENU-OPDOCM.
000          ACCEPT ESCK FROM ESCAPE KEY.
100          IF ESCK = FK5 GO TO MENU-DOCUMENTOS.
200          IF ESCK = FK1 OR ESCK = FK2 OR ESCK = FK3
300                    PERFORM MUESTRA-DATO-AUX
400                    GO TO SEL-OPCION-MODI.
500          IF ESCK = FK4
600             PERFORM GRABA-TOT-DOC
700             PERFORM VER-ENCABEZADO
800             PERFORM GRABAR-DOCUMENTOS THRU GRAB-EXIT
900               VARYING I FROM 1 BY 1     UNTIL I > REGLEI
000             DISPLAY NUEVO-MOD
100             ACCEPT NUEVO-MOD
200             IF CORR-2 = 0 GO TO MENU-DOCUMENTOS
300             ELSE
400               MOVE  0 TO CONTADOR
500               GO TO MODIFICAR-DOCUMENTO.
600       IF OPCION-DOC > 4 OR ESCK NOT = 0
700          MOVE "ERROR EN OPCION" TO MENSAJE
800          DISPLAY MENU-ERROR
900          GO TO SEL-OPCION-MODI.
000          IF OPCION-DOC = 0
100             PERFORM GRABA-TOT-DOC
200             PERFORM VER-ENCABEZADO
300             PERFORM GRABAR-DOCUMENTOS THRU GRAB-EXIT
400               VARYING I FROM 1 BY 1     UNTIL I > REGLEI
500             GO TO MENU-DOCUMENTOS.
600          IF OPCION-DOC = 3
700             GO TO LEE-LINEA-MOD.
800          IF OPCION-DOC = 1
900             DISPLAY NUMERO-LINEA
000             ACCEPT NUMERO-LINEA
100            PERFORM V-LINEA
200            DISPLAY NUMERO-LINEAD
300             MOVE 0 TO LINEA
400             PERFORM VALIDA-LINEA VARYING I FROM 1 BY 1
500                    UNTIL I > REGLEI OR LINEA = 1
600             IF LINEA = 1
700                MOVE "LINEA YA EXISTE" TO MENSAJE
800                DISPLAY MENU-ERROR
```

FIGURA 3.2.3

```
NG VS INTEGRATED EDITOR - VERSION  6.13.01        9:14 12/29/86   PAGE
PUT FILE IS EXPIN03X IN LIBRARY CLCLISTE ON VOLUME SISTCO

 5800  77  MENSAJE              PIC X(51)  VALUE SPACES.
 5900  77  MENSAJE2             PIC X(50)  VALUE SPACES.
 6000  77  SW-1                 PIC 9      VALUE ZERO.
 6100
 6200  01  MENU1  USAGE IS DISPLAY-WS.
 6300      05  FILLER           PICTURE IS  X(13)       ROW 01 COLUMN 03
 6400          VALUE IS "PROG: EXPIN03".
 6500      05  FILLER           PICTURE IS  X(35)       ROW 02 COLUMN 24
 6600          VALUE IS "INGRESO Y MANTENCION DE PROVEEDORES".
 6700      05  FILLER           PICTURE IS  X(35)       ROW 03 COLUMN 24
 6800          VALUE IS "**********************************".
 6900      05  FILLER           PICTURE IS  X(31)       ROW 07 COLUMN 26
 7000          VALUE IS "(PF1).....INGRESO DE CODIGOS".
 7100      05  FILLER           PICTURE IS  X(34)       ROW 09 COLUMN 26
 7200          VALUE IS "(PF2).....MANTENCION DE CODIGOS".
 7300      05  FILLER           PICTURE IS  X(28)       ROW 11 COLUMN 26
 7400          VALUE IS "(PF3).....ELIMINA CODIGOS".
 7500      05  FILLER           PICTURE IS  X(34)       ROW 13 COLUMN 26
 7600          VALUE IS "(PF4).....CONSULTAR PROVEEDORES".
 7700      05  FILLER           PICTURE IS  X(18)       ROW 15 COLUMN 26
 7800          VALUE IS "(PF5).....F I N".
 7900
 8000  01  MENU2  USAGE IS DISPLAY-WS.
 8100      05  FILLER           PICTURE IS  X(13)       ROW 01 COLUMN 03
 8200          VALUE IS "PROG: EXPIN03".
 8300      05  F1-TITULO-1      PICTURE IS  X(36)       ROW 02 COLUMN 24
 8400          SOURCE IS TITULO-1      .
 8500      05  F1-TITULO-2      PICTURE IS  X(36)       ROW 03 COLUMN 24
 8600          SOURCE IS TITULO-2      .
 8700      05  FILLER           PICTURE IS  X(21)       ROW 07 COLUMN 16
 8800          VALUE IS "CODIGO DE PROVEEDOR :".
 8900      05  F1-CODIGO        PICTURE IS  X(04)       ROW 07 COLUMN 39
 9000          SOURCE IS CODIGO-P       OBJECT IS CODIGO-P      .
 9100      05  FILLER           PICTURE IS  X(21)       ROW 09 COLUMN 16
 9200          VALUE IS "DESCRIPCION          :".
 9300      05  F1-DESCRIPCION PICTURE IS  X(30)         ROW 09 COLUMN 39
 9400          SOURCE IS DESCRIPCION-P  OBJECT IS DESCRIPCION-P .
 9500      05  F1-MENSAJE       PICTURE IS  X(51)       ROW 22 COLUMN 16
 9600          SOURCE IS MENSAJE.
 9700      05  F1-MENSAJE2      PICTURE IS  X(50)       ROW 15 COLUMN 16
 9800          SOURCE IS MENSAJE2.
 9810
 9900  PROCEDURE DIVISION.
 9910 ********************
 0000  PRINCIPAL.
 0010 *----------
 0100       OPEN I-O ARCH-PROV  WS.
 0200       MOVE 0 TO PF-KEY.
 0300       PERFORM PROCESO UNTIL PF-KEY = 5.
 0400       CLOSE ARCH-PROV WS.
 0500       STOP RUN.
 0600 *     CALL PROGRAM "CLC00/I".
 0700
 0800  PROCESO.
 0900 *--------
 1000       MOVE SPACES TO CODIGO-P DESCRIPCION-P.
 1100       MOVE 0 TO PF-KEY SW-VALIDA SW-ENCONTRADO.
```

sola instrucción, la instrucción DISPLAY AND READ (ver figura 3.2.4). En esta misma instrucción es posible definir las teclas PF que se pueden utilizar en tal pantalla para realizar algún proceso determinado. La mayor facilidad que reviste el uso de las pantallas en los equipos WANG-VS es que para moverse de un campo a otro de la pantalla se pueden utilizar las teclas marcadas con las flechas o bien la tecla TAB y no es necesario diseñar un algoritmo especial para tal efecto, como en el caso de DATA GENERAL.

Las otras instrucciones que hubo que cambiarle a los programas no revistieron dificultad alguna ya que solo se trataba de diferencias en la sintaxis de estas y por tal razón no se requirió un cambio en la lógica del programa como en el caso del despliege e ingresos de datos desde el terminal, explicado en el párrafo anterior. Estas diferencias de sintaxis se encuentran por ejemplo en la sentencia SELECT. Esta sentencia en lenguaje Cobol DATA GENERAL se escribe de la siguiente manera :

```
SELECT  ARCHIVO
        ASSIGN TO DISK, "ARCH-1"
        ORGANIZATION IS INDEXED
        ACCESS MODE  IS RANDOM
        RECORD KEY   IS COD-1
        ALTERNATE RECORD KEY IS NOMBRE WITH DUPLICATES
        FILE STATUS  IS FIL-1.
```

Y esta misma en Cobol del equipo WANG-VS sería:

```
SELECT ARCHIVO
       ASSIGN TO "ARCH-1", "DISK", NODISPLAY
       ORGANIZATION IS INDEXED
       ACCESS MODE  IS RANDOM
       RECORD KEY   IS COD-1
       ALTERNATE RECORD KEY 01 IS NOMBRE WITH
DUPLICATES
       FILE STATUS  IS FIL-1.
```

En el anexo número 1 y número 2 es posible apreciar con mayor exactitud un programa sin convertir y uno ya convertido.

FIGURA 3.2.4

```
ENG VS INTEGRATED EDITOR - VERSION  6.13.01        9:14 12/29/86   PAGE      3
INPUT FILE IS EXPIN03X IN LIBRARY CLCLISTE ON VOLUME SISTCO

 1200       DISPLAY AND READ  MENU1 ON WS
 1300       ONLY PFKEYS 01, 02, 03, 04, 05.
 1400       IF PF-KEY = 1
 1500         MOVE 0 TO PF-KEY
 1600         PERFORM INGRESO UNTIL PF-KEY = 16.
 1700       IF PF-KEY = 2
 1800         MOVE 0 TO PF-KEY
 1900         PERFORM MODIFICACION UNTIL PF-KEY = 16.
 2000       IF PF-KEY = 3
 2100         MOVE 0 TO PF-KEY
 2200         PERFORM ELIMINA UNTIL PF-KEY = 16.
 2300      IF PF-KEY = 4
 2400         MOVE 0 TO PF-KEY
 2500         PERFORM MUESTRA UNTIL PF-KEY = 16.
 2600  INGRESO.
 2700 *-------
 2800       MOVE SPACES TO CODIGO-P  DESCRIPCION-P.
 2900       MOVE "   INGRESO DE CODIGOS DE PROVEEDORES" TO TITULO-1.
 3000       MOVE "   **********************************" TO TITULO-2.
 3100       MOVE "(ENTER) = GRABAR  (PF16) = MENU PRINCIPAL" TO MENSAJE.
 3200       MOVE 0 TO PF-KEY SW-VALIDA.
 3300       DISPLAY AND READ MENU2 ON WS
 3400       PFKEYS  16.
 3500       IF PF-KEY  = 0  THEN
 3600             PERFORM VALIDA-CODIGO.
 3700       IF SW-VALIDA = 1 AND PF-KEY = 0  THEN
 3800          MOVE CODIGO-P TO CODIGO
 3900          MOVE DESCRIPCION-P TO DESCRIPCION
 4000          WRITE REG-PROV
 4100          MOVE SPACES TO  CODIGO-P DESCRIPCION-P.
 4200  VALIDA-CODIGO.
 4300 *--------------
 4400          MOVE 0 TO  SW-CODIGO.
 4500          MOVE 0 TO SW-LECTURA.
 4600          MOVE CODIGO-P TO CODIGO.
 4700          MOVE DESCRIPCION-P TO DESCRIPCION.
 4800          READ ARCH-PROV INVALID KEY MOVE 1 TO SW-LECTURA.
 4900          IF SW-LECTURA = 0 THEN DISPLAY "PROVEEDOR YA EXISTE!!!!".
 5000          IF CODIGO-P < "1000" OR CODIGO-P > "3000" THEN
 5100               MOVE 1 TO SW-CODIGO
 5200               DISPLAY "RANGO DE CODIGO INVALIDO!!!!!!!".
 5300          IF SW-LECTURA = 1 AND SW-CODIGO = 0 THEN
 5400               MOVE 1 TO SW-VALIDA.
 5500  ELIMINA.
 5600 *--------
 5700      MOVE SPACES TO CODIGO-P DESCRIPCION-P.
 5800      MOVE "     ELIMINA CODIGOS DE PROVEEDORES" TO TITULO-1.
 5900      MOVE "     ******************************" TO TITULO-2.
 6000      MOVE "(ENTER) =CONTINUAR   (PF16) = MENJ PRINCIPAL" TO
 6100      MENSAJE.
 6200      MOVE 0 TO PF-KEY SW-ENCONTRADO.
 6300      MOVE PROT TO FAC OF F1-DESCRIPCION.
 6400      DISPLAY AND READ MENU2 ON WS
 6500      PFKEYS  16.
 6600      MOVE DESPROT TO FAC OF F1-DESCRIPCION.
 6700      IF PF-KEY = 0  THEN
 6800            PERFORM BUSCA-PROVEEDOR.
```

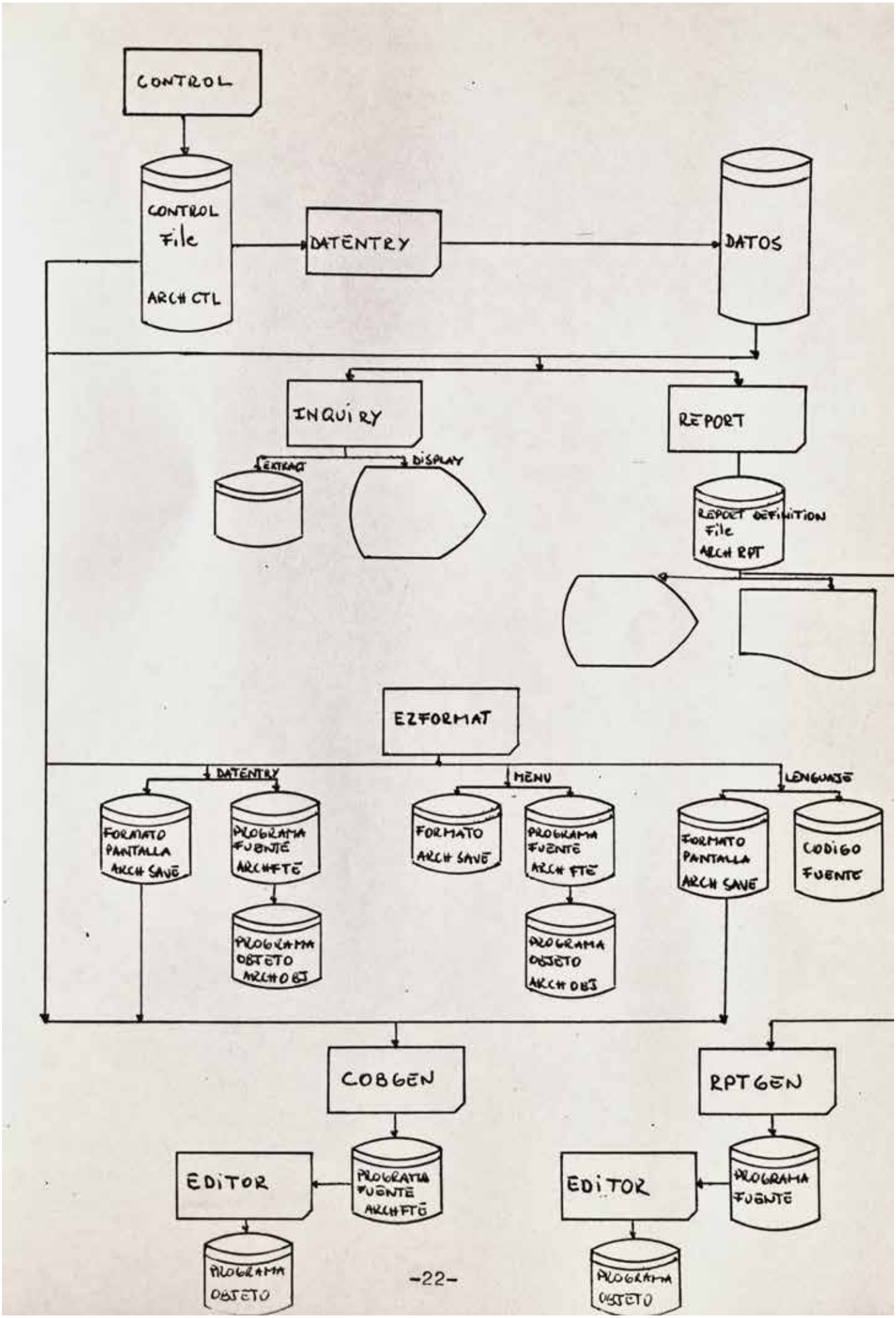
3.3- Problemas relativos a los archivos de datos.

Otro punto de la conversión aparte de la de los programas, fué la conversión de los archivos de datos. Esta presentó algunos problemas, que más que ser complicados de solucionar resultaron ser muy costosos en cuanto a tiempo.
El problema prinicipal se produjo en los archivos de datos que contenían ciertos campos cuyo formato incluía un signo, esto porque el equipo DATA GENERAL para almacenar un valor positivo o negativo, calcula al parecer un cierto complemento del número y ese número es el que graba en el archivo. No así el WANG-VS que para indicar si el campo es negativo o positivo incluye un carácter extra al final del campo, que es un + en el caso de un número positivo y un - en el caso contrario. En realidad este problema fué detectado en el momento de ejecutar algunos programas, ya que aparecía en la pantalla un error cuyo mensaje decía que el programa necesitaba un archivo con, por ejemplo 250 caracteres de largo y el archivo especificado solo tenía 241 caracteres. Obviamente los caracteres que faltaban eran los signos de los valores numéricos que el equipo DATA GENERAL no utilizaba.
El primer paso que dimos para solucionar esto fué mostrar el archivo en pantalla mediante el uso del utilitario DISPRINT. Con esto nos pudimos percatar realmente que le faltaban los signos a ciertos valores numéricos, pero que además al principio de cada uno de estos números aparecía un carácter extraño que no estaba definido en la tabla de caracteres del WANG-VS. Después de esto hice un programa que leía secencialamente el archivo con un formato sin signos y lo grababa en un archivo de salida con signos. El resultado de este proceso no fué realmente positivo. Como el archivo de entrada contenía caracteres no reconocibles para el equipo WANG el archivo generado por este proceso resultó tener solamente basura. Entonces modifiqué el programa poniendo un FILLER en la posición en donde se encontraban los caracteres irreconosibles, en la descripción del formato de lectura en la FD del archivo, para así poder ignorarlos y solo leer el numero para luego grabarlo en el archivo de salida con un formato con signo (ver programa PONESIG en el anexo 6). Este último proceso si dió resultado y con esto se pudo probar los programas que utilizaban tal archivo. El único problema de esto último fué que todos los valores quedaron positivos por lo tanto esta solución solo fué parcial y sirvió solamente para las pruebas iniciales de los programas.
Como había que solucionar definitívamente este problema para poder convertir los archivos de datos y hacer las pruebas finales de los sistemas, la gente del depto. de computación de la clínica tuvo que desarrollar un programa para convertir cada archivo de datos, que fuera leyendo secuencialmente los registros y para los campos con signo, preguntando si este era positivo o negativo para grabarle al final un carácter + o un - adicional dependiendo de si el valor era positivo o negativo.

El otro problema que tuvimos con los archivos de datos fué que, en un gran porcentaje de ellos, campos que eran numéricos y deberían estar en cero contenían espacios. La solución que se le dió a este problema fué similar a la del problema anterior. Hubo que leer cada uno de estos archivos en forma secuencial para ir generando un archivo de salida donde se reemplazaran los campos con espacios por los mismos pero con ceros y de esta manera poder reconstruir el archivo a como debería de haber estado.

DESCIPCION DE LOS UTILITARIOS UTILIZADOS.

4.1- Esquema de los utilitarios de WANG-VS.


CONTROL
CONTROL File
ARCH CTL
DATENTRY
DATOS
INQUIRY
EXTRACT
DISPLAY
REPORT
REPORT DEFINITION File
ARCH RPT
EZFORMAT
DATENTRY
MENU
LENGUAJE
FORMATO PANTALLA ARCH SAVE
PROGRAMA FUENTE ARCH FTE
PROGRAMA OBJETO ARCH OBJ
FORMATO ARCH SAVE
PROGRAMA FUENTE ARCH FTE
PROGRAMA OBJETO ARCH OBJ
FORMATO PANTALLA ARCH SAVE
CODIGO FUENTE
COBGEN
RPTGEN
EDITOR
PROGRAMA FUENTE ARCH FTE
PROGRAMA OBJETO
EDITOR
PROGRAMA FUENTE
PROGRAMA OBJETO

4.2- Utilitario EZFORMAT.

Este utilitario provee tres funciones que facilitan el desarrollo interactivo de programas.
Estas funciones son :

- Crear un programa de entrada de datos en lenguaje Cobol usando un formato de pantalla diseñado por el usuario y un archivo de control correspondiente al archivo que se accesará.

- Permitir al usuario diseñar un formato de pantalla y generar automáticamente el código fuente ya sea un lenguaje ASSEMBLER, BASIC, COBOL o RPG II.

- Crear un programa menú completo, que asocia funciones especificadas por el usuario con teclas de funcion PF. En este caso el usuario diseña el formato de la pantalla que describe este menú.

En la figura 4.1.1 se ejemplifican las diferentes opciones que posee este utilitario.

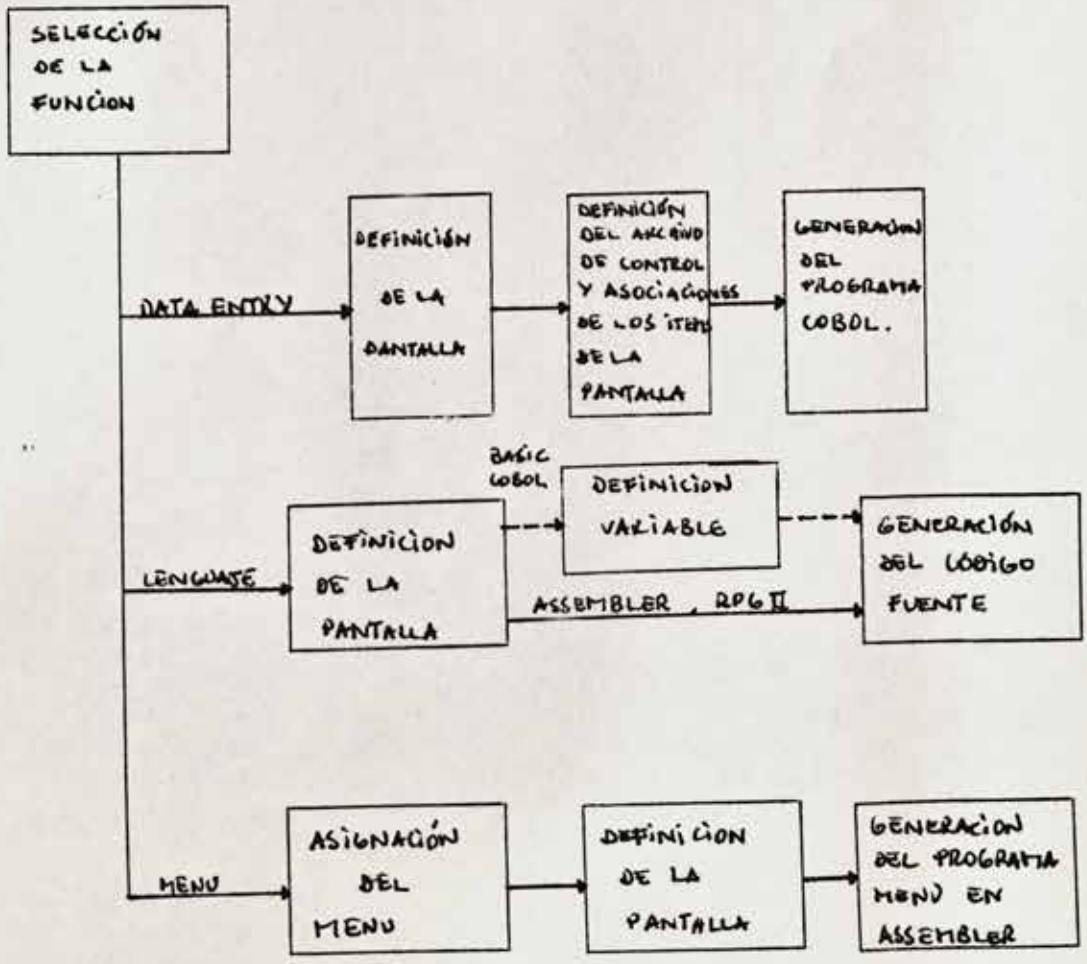


FIGURA 4.2.1

```
5 VS INTEGRATED EDITOR - VERSION  6.13.01       11:38 10/21/86   PAGE    13
UT FILE IS EXINTM03 IN LIBRARY CLCFTE   ON VOLUME SOPVTA

500          SOURCE IS DIGDET .
500      05  F10-DESCRIPCION PICTURE IS  X(25)       ROW 15 COLUMN 18
700          SOURCE IS DESCRIPCION  .
300      05  F10-MEDIDA-I    PICTURE IS  X(05)       ROW 15 COLUMN 45
900          SOURCE IS MEDIDA-I .
000      05  F10-CANDET      PICTURE IS  Z(06).ZZ    ROW 15 COLUMN 53
 00          SOURCE IS CANDET    .
:00      05  F10-MENSAJE     PICTURE IS  X(40)       ROW 22 COLUMN 10
:00          SOURCE IS MENSAJE.
 00      05  FILLER          PICTURE IS  X(77)       ROW 24 COLUMN 03
 00          VALUE IS "(ENTER)= CONTINUAR                (PF1)= ELIMINA
 00 -                  "                (PF16)= M E N U".
 00 *-------------------------------------------------------------
 00  01  MODITOT      USAGE IS DISPLAY-WS.
 00      05  FILLER          PICTURE IS  X(14)       ROW 01 COLUMN 03
 00          VALUE IS "PROG: EXINTM03".
 00      05  FILLER          PICTURE IS  X(23)       ROW 02 COLUMN 30
 00          VALUE IS "MODIFICACION DE TOTALES".
 00      05  FILLER          PICTURE IS  X(23)       ROW 03 COLUMN 30
 00          VALUE IS "_______________________".
 00      05  FILLER          PICTURE IS  X(07)       ROW 07 COLUMN 38
 00          VALUE IS "TOTALES".
 00      05  FILLER          PICTURE IS  X(07)       ROW 08 COLUMN 38
 00          VALUE IS "*******".
 00      05  FILLER          PICTURE IS  X(15)       ROW 11 COLUMN 28
 00          VALUE IS "TOTAL DETALLE :".
 00      05  F7-TOTDET       PICTURE IS  Z(08),ZZ    ROW 11 COLUMN 45
 00          SOURCE IS TOTDET          OBJECT IS TOTDET        .
 00      05  FILLER          PICTURE IS  X(65)       ROW 24 COLUMN 09
 00          VALUE IS "(ENTER) = GRABA MODIF.     (PF1) = CONTINUAR
 00 -                "  (PF16) = M E N U".
 00 *-------------------------------------------------------------
 00  01  ELITOT       USAGE IS DISPLAY-WS.
 00      05  FILLER          PICTURE IS  X(14)       ROW 01 COLUMN 03
 00          VALUE IS "PROG: EXINTM03".
 00      05  FILLER          PICTURE IS  X(22)       ROW 02 COLUMN 30
 00          VALUE IS "ELIMINACION DE TOTALES".
 00      05  FILLER          PICTURE IS  X(23)       ROW 03 COLUMN 30
 00          VALUE IS "_______________________".
 00      05  FILLER          PICTURE IS  X(07)       ROW 07 COLUMN 38
 00          VALUE IS "TOTALES".
 00      05  FILLER          PICTURE IS  X(07)       ROW 08 COLUMN 38
 00          VALUE IS "*******".
 00      05  FILLER          PICTURE IS  X(15)       ROW 11 COLUMN 28
 00          VALUE IS "TOTAL DETALLE :".
 00      05  F11-TOTDET      PICTURE IS  Z(10)       ROW 11 COLUMN 45
 00          SOURCE IS TOTDET     .
 00      05  FILLER          PICTURE IS  X(65)       ROW 24 COLUMN 09
 00          VALUE IS "(ENTER) = CONTINUAR        (PF1) = ELIMINAR
 00 -                "  (PF16) = M E N U".
 00 *-------------------------------------------------------------
 00  01  CONSU2       USAGE IS DISPLAY-WS.
 00      05  FILLER          PICTURE IS  X(08)       ROW 01 COLUMN 03
 00          VALUE IS "REMESA :".
 00      05  F7-NUMERO-REME  PICTURE IS  Z(06)       ROW 01 COLUMN 13
 00          SOURCE IS NUMERO-REMESA .
 00      05  FILLER          PICTURE IS  X(03)       ROW 01 COLUMN 38
```

Generación de código fuente en lenguaje Cobol.

EZFORMAT utilizando la descripción de títulos y campos definidos por el usuario en la pantalla, construye el código fuente en lenguaje Cobol que contiene la DATA DIVISION del registro del terminal, apropiado para hacer referencia a este mediante una instrucción DISPLAY AND READ ubicada en cualquier parte a lo largo de la PROCEDURE DIVISION.
Para los campos de la pantalla es posible definir el tipo de dato solamente como numérico o alfabético, si se requiere una pantalla con datos numéricos editados por ejemplo, es necesario modificar el formato del campo en el registro del terminal, cuando el código ya fué copiado al programa fuente en Cobol, utilizando el editor. Otra ventaja que provee el utilitario es la de poder realizar una validación de rango de los valores ingresados en las variables del registro de pantalla.
A continuación en la figura 4.2.1 se muestra un trozo de código en Cobol generado mediante EZFORMAT.

Generación de programas menú mediante EZFORMAT.

Esta opción del utilitario permite crear un programa fuente en lenguaje RPG II o bien en ASSEMBLER, en el cual se despliega la pantalla diseñada y se ejecuta la opción de proceso de acuerdo a la tecla PF que presione el usuario. Es posible modificar y recompilar este programa utilizando el editor.
Un programa menú permite realizar las siguientes funciones:

- Ejecutar un programa específico (de aplicación o utilitario).
- Salir del menú.
- Dejar al usuario fuera del sistema.

4.3- Utilitario REPORT.

El utilitario REPORT provee un método para crear un programa fuente que emita un informe impreso a partir de un archivo de datos.
REPORT permite definir cosas como títulos, encabezamientos de página, campos de corte de control, supresión de ceros en ciertos campos, e inclusión de signos especiales y caracteres monetarios, dentro de sus opciones.
Existe una sola restricción para generar programas mediante este utilitario, y es que solo se puede utilizar un archivo de datos a la vez, para generar en "report".

4.4- Utilitario COBGEN.

Sin duda el utilitario COBGEN es uno de los que mayor ayuda provee para el desarrollo de programas. Su finalidad es facilitar el trabajo mecánico de codificar programas fuente en lenguaje Cobol. COBGEN genera automáticamente el código fuente correspondiente a las cuatro divisiones de un programa Cobol, el cual se puede complementar y compilar por medio del utilitario EDITOR.

4.5- Utilitario CONTROL.

El utilitario CONTROL permite definir los atributos de un archivo de datos y de sus registros. Estos atributos se graban en un archivo de control que sirve de directorio para describir los atributos de un archivo de datos.
Un archivo de control consiste en :

- Un registro descriptor de archivo, (HEADER)
- Registros descriptores de campos, uno por cada campo del archivo de datos.
- Registros de llaves alternas.
- Opcionalmente, uno a tres registros de comentarios.

Todo archivo de datos que se crea o modifica mediante otro utilitario llamado DATENTRY y se mantiene a traves de un programa de entrada de datos generado por EZFORMAT o accesado por REPORT, debe tener por lo menos un archivo de control asociado.

4.6- Utilitario VS-INTEGRATED EDITOR.

El proceso de desarrollar programas en WANG VS se compone de cuatro pasos :

- Ingresar y/o editar el texto fuente.
- Compilar el texto fuente en un programa objeto, usando un traductor de lenguaje apropiado, ASSEMBLER, BASIC, COBOL, FORTRAN o RPG II.
- Encadenar programas objeto, si esto es necesario utilizando el LINKER.
- Ejecutar el programa objeto.

El trabajo de desarrollar un programa, por lo tanto involucra el uso de diferentes programas del sistema incluyendo el editor y un traductor de lenguaje.

Además, si el programa objeto se compone de dos o más subprogramas independientes, estos deben encadenarse por medio del programa LINKER antes de ejecutarse.

Algunos programas del sistema, tales como SORT, COPY y EZFORMAT se pueden ejecutar para propositos específicos relacionados con la creación , compilación, prueba y ejecución de los programas del usuario. Los programas del sistema se pueden ejecutar independientemente desde el menú del procesador de comandos o desde el editor. Aún más, estando dentro del programa editor es posible volver a invocar a este para trabajar con otro programa fuente, este anidamiento de llamadas del editor a través de el mismo es posible realizarlo hasta 14 veces.

Frecuentemente el programador debe editar, compilar y ejecutar un programa muchas veces antes que este, esté totalmente depurado y sin errores. Este proceso consume mucho tiempo cuando los parámetros necesarios se deben ingresar cada vez que el programa se modifica y se ejecuta. Para simplificar el proceso de desarrollar programas, la VS proporciona un programa de sistema, el editor, cuyo propósito es crear y sostener un medio de programación integrado.

El utilitario editor integra todas las funciones necesarias para crear, editar, compilar, y ejecutar programas.
La edición de un programa fuente, que comprende la creación de un texto y las modificaciones de un texto existente, se ubica en un archivo de trabajo temporal creado por el editor.

Cuando se crea un programa fuente, el programa completo se graba en el archivo de trabajo. Cuando se edita un programa existente, el programa se copia primero en el archivo de trabajo. Cuando el usuario lo solicita, el texto se tranfiere desde el archivo de trabajo a un archivo fuente permanente como un nuevo archivo, o bien reemplaza al archivo existente. El archivo fuente original

no se altera hasta que el programador lo reemplaza explicitamente por la versión editada del archivo de trabajo. En el anexo 3 se encuentran la principales pantallas de este utilitario.

4.7- Utilitario PROCGEN.

El utilitario PROCGEN se usa para generar procedimientos. El archivo de salida creado por PROCGEN se puede ejecutar a través del procesador de comandos o bien, se puede modificar y ejecutar por medio del utilitario EDITOR.
PROCGEN puede generar el código en lenguaje de procedimientos para las instrucciones RUN, SET y SCRATCH. Tambien puede generar la instrucciones ENTER de cualquier pantalla (GETPARAM) para los sigientes utilitarios del sistema:

- BELL
- SORT
- TRANSL
- CONVERTC
- CREATE
- COPY
- DISPLAY
- TAPECOPY
- COPY2200
- DISKINIT
- EDITOR
- EZFORMAT
- DATENTRY
- REPORT
- ASSEMBLER
- BASIC
- COBOL
- RPG II
- LINKER
- MOUNT
- BACKUP
- FLOPPY-DUP.

El propósito de PROCGEN es servir de ayuda al usuario que desea escribir procedimientos, no pretende eliminar completamente la necesidad de comprender los procedimientos.

Un caso específico en cual se utilizó PROCGEN en la conversión, fué cuando se detectó la necesidad de contar con un programa que fuera fácil de operar cuando se necesitara tranferir la información desde las cintas que eran enviadas por la clínica al disco en el cual estabamos trabajando. Al comienzo cada véz que se nesecitaba hacer esto era necesario ejecutar el utilitario TAPECOPY y proporcionar toda la información que este requería en sus pantallas, nombre del archivo de entrada, numero de la unidad de cinta en que se encontraba, nombre del archivo de salida, nombre de la librería, nombre del volúmen, etc... Por lo tanto para hacer mas rápido este proceso se generó mediante PROCGEN un procedimiento que solo había que ingresarle el nombre de la cinta, el largo del registro y el nombre del archivo de salida. Un listado de este procedimiento se encuentra en el anexo 4.

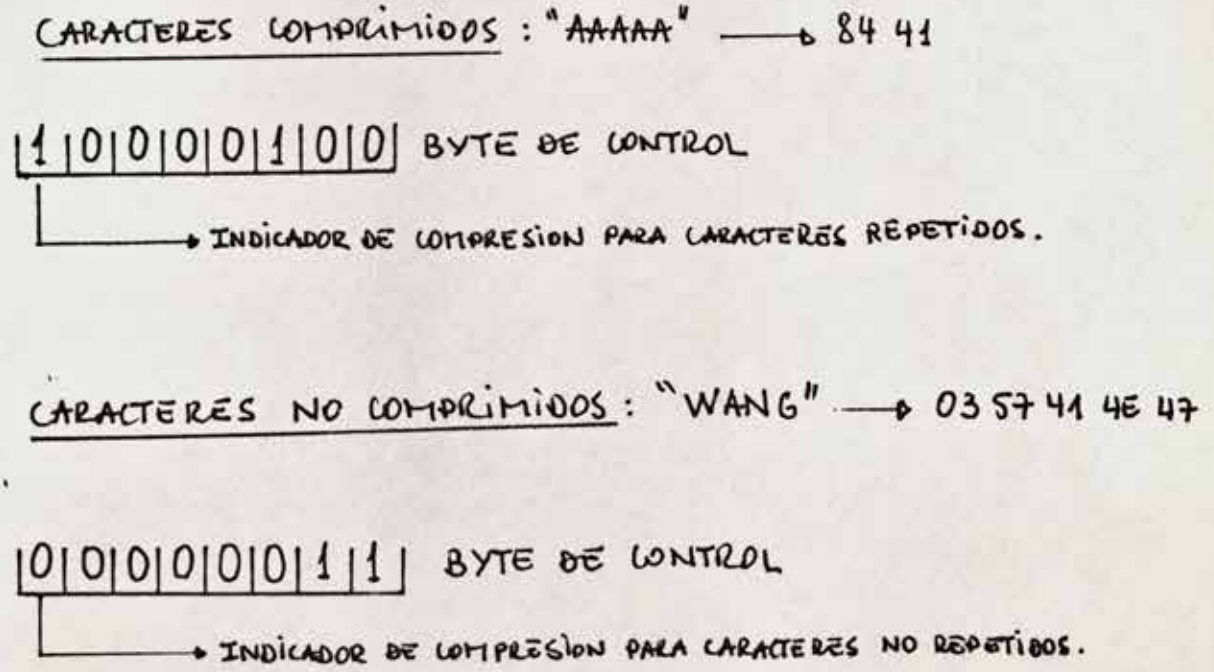
METODOS UTILIZADOS PARA MEJORAR EL RENDIMIENTO DE LOS PROGRAMAS

5.1- Generalidades del DMS y como aumentar el rendimiento del sistema

En WANG-VS para utilizar un archivo, ya sea este un programa un procedimiento o un archivo de datos, es necesario identificar tres parámetros que son; el nombre del archivo, la librería en que se encuentra y el volúmen. Existen algunas restricciones en cuanto que los nombres, los archivos y las librerías solo pueden ser de ocho caracteres de largo y los volúmenes de seis. Por ejemplo un nombre completo para identificar un archivo sería:

FILE : mae$val LIBRARY : clcdata VOLUME : fijo00

Existe en DMS un concepto denominado de compresión de archivos, éste se refiere a que un archivo en el cual existen caracteres que se repiten y si este archivo obviamente fué creado con opción de compresión estas repeticiones de dos o más veces se almacenan en dos bytes. En el primer byte se almacena el factor de repetición y en el segundo el carácter que se repite.
Esta opción permite ahorrar un espacio considerable en disco. Al momento de transferir un registro en formato comprimido desde disco a memoria, este es descomprimido automáticmente por DMS. En el siguiente gráfico se puede apreciar mas claramente la diferencia entre un formato comprimido y uno no comprimido.



Durante la conversión de los programas fué necesario fijarse mucho en ciertos factores que degradaban el rendimiento de las aplicaciones. Estos factores son :

- La organización de los archivos.
- Los modos de apertura.
- La compresión.
- las estrategias de uso de buffers.
- Las opciones de compilación.

De acerdo a mejorar el rendimiento de los programas en base a estos factores se efectuaron cambios en los modos de apertura de los archivos. Muchos programas utilizaban el primer registro del archivo como un registro índice para verificar si una remesa(1) estaba en uso por otro usuario, esto se hacía de la siguiente manera. El programa leía el primer registro correspondiente a la remesa y si un determinado campo de este estaba en el valor 1, esto quería decir que la remesa se estaba ocupando, y si el valor era 0 esta se podía utilizar pero previo a esto el programa regrababa tal campo con el valor 1 para dejar en estado de ocupada tal remesa.
Para realizar esto, en el programa se abría solo una vez el archivo en modo de acceso I/O, ya que el programa leía y grababa. Pero solamente el archivo era grabado en una sola parte del programa, en donde se verificaba si la remesa estaba ocupada. Por lo tanto para mejorar el rendimiento se decidió abrir el archivo en modo I/O para utilizarlo en la rutina inicial en donde se verificaba la utilización de la remesa y luego cerrarlo y volverlo a abrir en modo INPUT solamente.

A propósito del factor de compresión, explicado en un punto anterior, se decidió no comprimir los archivos de datos de organización indexada, ya que al momento de tranferir estos de disco a memoria y ser por la tanto descomprimidos, se producía una baja de rendimiento en el proceso de lectura, esto debido sobre todo al tamaño del registro por el uso de llaves alternas.
Con las técnicas de manejo de buffers es posible disminuir el tiempo de acceso a los archivos minizando las operaciones de input/output a disco. Existen dos técnicas utilizadas, estas son:

- Large buffer.
- Buffer pooling.

(1) REMESA : término utilizado para describir un paquete con una cierta cantidad determinada de documentos.

La técnica Large buffer se utiliza para archivos consecutivos con modo de acceso secuencial, este permite agrandar el tamaño del buffer en múltiplos de 2 Kbytes, hasta 18 Kbytes. El tamaño del buffer se determina con la cláusula BUFFER SIZE IS integer BLOCKS, dentro de la cláusula SELECT del archivo.
La otra técnica, Buffer pooling, sirve para archivos de organización indexada abiertos en modo I/O, esta determina un área de buffers que es compartida por varios archivos abiertos en un mismo programa. Es posible asignar un Buffer pooling de 60 buffers de 2 Kbyte cada uno.
Esta técnica es utilizada automáticamente para archivos indexados abiertos en modo SHARED. La cantidad de buffers es asignada mediante el uso de la clausulas RESERVE integer AREA(S) y SAME AREA FOR file-name-1 file-name-2 ...
Es necesario mantener un cierto balance de estas técnicas, ya que si se utiliza un tamaño de Large buffer muy grande o una cantidad de Buffer pooling muy alta, esto puede conllevar a un aumento en la tasa de paginamiento, degradando en vez de mejorar el rendimiento del sistema.

Durante la conversión fué necesario obviamente probar los programas y como muchos de ellos no funcionaban correctamente se debió utilizar ciertas opciones para la compilación que permitieran detectar los errores que estos tenían, que en la grán mayoría de los casos eran de lógica. Esto último debido a que los programas enviados por la clínica contenian fallas garrafales con las cuales era imposible que realizaran el proceso deseado y más aún que no arrojaran errores de sintaxis al momento de compilarlos. Las opciones de compilación que se utilizaron para tales efectos fueron SYMB y SUBCHK.
La opción SYMB se utiliza para poder ejecutar un programa mediante el uso del utilitario DEBUG y la opción SUBCHK es utilizada para chequear los rangos de subíndices de los arreglos. Estas dos opciones son muy útiles durante la etapa de depuración de los programas pero al momento de generar los programas objeto definitivos fué necesario recompilar estos sin utilizar la opcion SYMB ya que ésta incide en el rendimiento de la aplicación durante la ejecución de esta. La opción SYMB genera un segmento extra en el programa objeto llamado "symbolic block" que contiene la información que requiere el utilitario DEBUG para ejecutar un programa, uno de estos datos es el nombre de la sección (nombre de la PROGRAM ID.en el caso de un programa Cobol). Y por otra parte la opción SUBCHK hace mas lenta la etapa de desarrollo de aplicaciones, en este caso específico, de conversión de programas, ya que chequea todos los subíndices posibles de todos los arreglos durante la compilación del programa.

5.2- Utilización de PAGE POOL.

El PAGE POOL es un área fija de un disco que contiene archivos de paginación de los segmentos 2 de los usuarios.
El tamaño del PAGE POOL es manejado por el administrador del sistema, y dependiendo de su tamaño e utilización dentro de la configuración hace que esta tenga un mejor o peor rendimiento. Este rendimiento generalmente se refiere al tiempo de respuesta del equipo, el cual es uno de los factores mas importantes desde el punto de vista de los usuarios y explotadores de sistemas.
El tamaño del PAGE POOL se asigna mediante el utilitario INIT al momento de inicializar el disco, ya que un disco puede o no contener áreas de paginamiento. Generalente las áreas de paginación son definidas en un sistema en los discos que mayor permanencia tienen dentro del sistema, esto se refiere que no es recomendable definir áreas de paginación o PAGE POOL para un disco que se supone va a ser desmontado frecuentemente en el sistema. Se recomienda esto ya que si un disco se desea desmontar, no puede haber ningún usuario que tenga un área de paginación definida en tal disco, en dicho caso para retirar el disco el usuario deberá desconectarse del sistema y volverse a conectar una vez que el disco este desmontado. Esta situación se produce porque el sistema operativo, cuando se conecta un usuario le asigna un área de paginación en algún disco del sistema que contenga PAGE POOL. Es posible determinar el tamaño óptimo del PAGE POOL utilzando la fórmula :

PAGE POOL (Mb) = (5 + (0.5 * S * U)) / N

donde :

S = promedio de los segmentos dos de los usuarios en Mb.

U = número de tareas concurrentes en máxima carga.

N = número de volumenes en que se permitirá paginamiento.

Es posible monitorear el uso de PAGE POOL mediante el uso del utilitario POOLSTAT, que entrega datos como el número de tareas asignadas al PAGE POOL y cuanto espacio tiene. Algunas de las ventajas de utilizar un tamaño apropiado para el PAGE POOL, son :

- Se logra una contiguidad de los bloques de paginación.
- Se puede controlar su ubicación.
- Reducción del tiempo de busqueda.

Para mayor información sobre PAGE POOL referirse al manual SYSTEM UTILITIES REFERECE Capítulo 21.

5.3- Reorganización de archivos.

La reorganización de archivos, más que una técnica para mejorar el rendimiento de las aplicaciones durante la etapa de conversión, fué una recomendación que se le dió al cliente, para que utilizara cada cierto tiempo a partir de la puesta en marcha de los sistemas.

La reorganización es conveniente realizarla cada cierto tiempo a archivos que tengan muchas actualizaciones, vale decir archivos a los cuales se le ingresen nuevos registros y se le eliminen otros. Ya que los archivos que han recibido muchas actualizaciones no tienen todos sus bloques asignados contiguos y mediante la reorganización esto se puede lograr, lo cual lleva a tener un tiempo de acceso bastante más bajo, sobre todo en archivos indexados en donde no se leen los registros en forma consecutiva sino que por medio de las llaves.
Para reorganizar un archivo es necesario copiar este usando el utilitario COPY con la opcion REORG = YES.

5.4- Utilización del utilitario DEBUG-HOOK.

Uno de los puntos recalcados por el jefe de proyecto al comienzo de la conversión fué que en la medida que fuera posible se tratara de mejorar los tiempos de proceso de los programas, pero sin que esto demandara un esfuerzo adicional demasiado grande.
Para poder cumplir con este requerimiento se utilizó una herramienta del equipo WANG-VS, el utilitario DEBUG-HOOK.
Este utilitario permite compilar y ejecutar un programa obteniendo diversos informes con datos estadísticos de la ejecución de éste. Utilizando el informe que se puede encontrar en el anexo 7 y específicamente tomando como referencia los datos del tiempo de CPU y el porcentaje del total del tiempo de ejecución para cada una de las instrucciones, se pudo efectuar en algunos programas ciertos cambios que disminueyeron el tiempo total de ejecución de cada uno de estos.
Los cambios mas frecuentes que se hicieron fué cambiar las instrucciones "COMPUTE" por simples sumas, restas, multiplicaciones y divisiones y poner la aperturas de archivos después de desplegar las pantallas, ya que no era necesario abrir estos si el usuario aún no confirmaba si deseaba ejecutar el programa.

CONCLUSIONES.

Ventajas y desventajas entre el WANG-VS y otros equipos.

Los equipos WANG-VS poseen una grán ventaja dentro del campo de las aplicaciones administrativas, ya que es un equipo especialmente diseñado para operar en este campo.
Si entraramos en una comparación entre un equipo DIGITAL, por ejemplo un VAX-750, y un equipo WANG-VS 65, probablemente el equipo DIGITAL posee una gran cantidad de software pero por otra parte el WANG posee herramientas para el desarrollo de programas que no se encuentran en DIGITAL. Un ejemplo de esto pueden ser los utilitarios COBGEN, REPORT y EZPRINT.
Este punto anteriormente explicado es muy importante en un ambiente de desarrollo, ya que contar con toda la gama de utilitarios que posee WANG-VS hace aumentar notoriamente la productividad de los programadores.
Otra ventaja que caracteriza a los equipos WANG es la facilidad de manejo del equipo y de sus utilitarios que provee el usuario final. Esto se debe a que todo su sistema operativo esta basado en pantallas, en donde se le explica al usuario con mensajes la información que debe suministrar y las diversas acciones que puede ejecutar presionando una de las 32 teclas "PF" programadas.
Las deventajas que a juicio particular poseen los computadores WANG, siempre comparando estos con DIGITAL, son su sistema de protecciones, el manejo de archivos y ciertos detalles en la administración de los recursos de la configuración.
Contrariamente a todas las facilidades de operación que posee el equipo WANG la administración de los archivos no es tan transparente para el usuario, ya que para poder utilizar un archivo debe especificar el nombre del volúmen (disco por ejemplo) y el nombre de la librería de dicho volúmen donde se encuentra el archivo deseado.
La otra deficiencia en cuanto a la administración de los recursos, es que cada usuario no tiene una restricción en cuanto al espacio en memoria secundaria que puede utilizar. Si un solo usuario desea, este puede llenar un disco con un archivo y no dejarle espacio a ningun otro usuario. Por lo cual cada usuario debe tener la suficiente responsabilidad de ir eliminando los archivos que no utiliza para liberar espacio en el disco. Otro punto relacionado con la administración de los recurso, es que en la cola de impresión cualquier usuario puede borrar o correr hacia atras un listado que tenga la misma clase que la que este esta ocupando.
Principalmente por estas dos últimas razones es que personalmente no encuentro una opción muy acertada el poner un equipo WANG en un ambiente universitario, donde se le prestan servicios prioritariamente a los estudiantes. Pero por otro lado creo que es una opción muy acertada dentro del medio ambiente empresarial.

Necesidad de documentación actualizada, completa y adecuada.

Un punto muy importante pero poco tomado en cuenta generalmente cuando se llega al final de un proyecto es la necesidad de confeccionar una documentación completa de los programas y sistemas.
Este fué uno de los puntos que bastantes problemas causó durante la conversión, el no contar con una buena documentación de los programas y sistemas que se estaban modifcando. Esto se notó principalmente ya que cada vez que surgía una duda sobre qué era lo que realizaba un cierto programa, en véz de obtener dicha información de la documentación era necesario hacer la consulta de modo verbal al usuario, sobre cuál era el propósito de tal programa.

Importancia de la participación del cliente.

Durante la conversión misma se produjo un cambio bastante grande en el momento en que la clínica envió una persona para que formara parte del grupo de trabajo. Esto fué realmente beneficioso ya que se contaba con la ayuda directa de una persona que estaba ya interiorizada de los procesos y la forma como se habían desarrollado los programas. Quizás éste punto no habría sido tan importante si se hubiera contado con una buena documentación de los sistemas como se explicó en un punto anterior, pero desgraciadamente la experiencia indica que en la mayoría de los casos esto no es así, de aquí entonces la conveniencia de solicitar en este tipo de proyectos la participación directa en el trabajo de conversión de una persona que conozca a cabalidad los sistemas que se van a convertir.

Planificación adecuada de los plazos de entrega.

Otra conclusión muy importante se refiere a los plazos de entrega prometidos al cliente. Al comienzo del proyecto se efectuó una planificación de los tiempos que tomaría la conversión de cada uno de los sistemas y cual sería la fecha final de término del proyecto. Pero más adelante, en conformidad con el cliente, hubo que reevaluar estos plazos ya que durante el cominezo de la conversión surgieron problemas que segúramente no se habían presupuestado en primera instancia. Esto deja la enseñanza que en proyectos de este tipo es mejor presupuestar un poco mas de tiempo que el que realmente se supone que durará el

proyecto, esto ya que la cantidad de imprevistos que se deben tomar en cuenta para la planificación son bastantes, por ejemplo cosas como, disponibilidad de equipo, problemas de compatibilidad entre los medios de traspaso de archivos, disminución en cantidad de personas del equipo de trabajo por enfermedades,.etc..., y obviamente otros imprevistos que van surgiendo a medida que avanza el proyecto.

Importancia del uso de las técnicas de programación estructurada.

Realmente cuando a uno le asignan la tarea de modificar un programa que no fué desarrollado por uno mismo y por lo tanto tiene que tratar primero que nada de entender que y como esta hecho el programa, es posible darse cuenta lo difícil que es tratar, digo tratar porque en algunos casos es mejor desarrollar un nuevo programa, modificar un programa que no sigue una secuencia lógica y ordenada en sus instrucciones y no atiende relación alguna con alguna técnica conocida de programación. De esto se puede sacar por lo tanto como conclusión la importancia de utilizar las técnicas de modularización y estructuración en el desarrollo de programas.

Utilización de ciertos estandares de programación.

Otra cosa que se utilizó durante la conversión fué definir algunos estandares de programación. Esto se refiere a estandarizar por ejemplo cosas como, los nombres de variables,los títulos y encabazados de programas, las formas de indicar el comienzo de un párrafo y el formato de la pantalla, para la totalidad de los programas convertidos. Ya que los programas originales no contaban con esto y cada vez que se tomaba un nuevo programa el formato de este era totalmente diferente al anterior. Por lo tanto con esta estandarización se logró algo muy importante, un mayor orden.

ANEXOS

ANEXO 1

```
 NG VS INTEGRATED EDITOR - VERSION  6.13.01        11:39 12/29/85   PAGE
 PUT FILE IS 00000013 IN LIBRARY CLCEX    ON VOLUME SISTCO

 0100   IDENTIFICATION DIVISION.
 0200   PROGRAM-ID. EXPIN03X.
 0300   AUTHOR. ALVARO VASQUEZ LANGER.
 0400 *
 0500 *
 0600 *
 0700 *      INGRESO Y MANTENCION DE CODIGOS DE PROVEEDORES
 0800 *
 0900 *
 1000 *
 1100 *
 1200   ENVIRONMENT DIVISION.
 1300   CONFIGURATION SECTION.
 1400   SPECIAL-NAMES.
 1500       DECIMAL-POINT IS COMMA.
 1600   INPUT-OUTPUT SECTION.
 1700   FILE-CONTROL.
 1800         SELECT ARCH-PROV  ASSIGN TO DISK   "EXCLPROV"
 1900                ORGANIZATION IS INDEXED
 2000                ACCESS MODE IS RANDOM
 2100                RECORD KEY IS CODIGO
 2200                ALTERNATE RECORD KEY  IS DESCRIPCION
 2300                STATUS IS FILE-STATUS.
 2400   DATA DIVISION.
 2500   FILE SECTION.
 2600   FD ARCH-PROV
 2700         RECORD CONTAINS  34 CHARACTERS
 2800         LABEL RECORDS ARE OMITTED.
 2900   01 REG-PROV.
 3000      02 CODIGO      PIC X(4).
 3100      02 DESCRIPCION PIC X(30).
 3200 *
 3300   WORKING-STORAGE SECTION.
 3400   77 FLG-MOD1 PIC 9 VALUE 0.
 3500   77 RESP-1 PIC 9 VALUE 0.
 3600   77 RESP-2 PIC 9 VALUE 0.
 3700   77 FLG PIC 9 VALUE 0.
 3800   77 ESCAPE-CODE PIC 9 VALUE 0.
 3900   77 FILE-STATUS PIC X(2).
 4000   77 D-CORR   PIC X.
 4100   77 NUM-LIN PIC 9.
 4200   01 REG-AUX.
 4300      02 FILLER PIC X(4).
 4400      02 BUFF  PIC X(30).
 4500 *
 4600   SCREEN SECTION.
 4700   01 MENU1.
 4800      02 BLANK SCREEN.
 4900      02 LINE 01 COLUMN 02 "PROG. EXPIN03".
 5000      02 LINE 03 COLUMN 25 "INGRESO Y MANTENCION DE PROVEEDORES".
 5100      02 LINE 04 COLUMN 25 "**********************************".
 5200      02 LINE 12 COLUMN 08 "1.- INGRESO DE CODIGOS.".
 5300      02 LINE 13 COLUMN 08 "2.- MANTENCION DE CODIGOS.".
 5400      02 LINE 14 COLUMN 08 "3.- ELIMINA CODIGOS.".
 5500      02 LINE 15 COLUMN 08 "4.- FIN.".
 5600   01 ACEPTA1.
 5700      02 LINE 16 COLUMN 33 TO RESP-1 PIC 9.
```

```
017200        ACCEPT ACEPTA-MODIF.
017300   VER-CAMBIO-LLAVE.
017400  ****************
017500        IF RESP-1 = 2 AND NUM-LIN = 1
017600          MOVE 2 TO FLG-MODI
017700          MOVE REG-PROV TO REG-AUX
017800          DELETE ARCH-PROV RECORD INVALID KEY DISPLAY "QUEEE!!!!"
017900          CALL PROGRAM "#W".
018000   ACEPTA-CAMPOS.
018100  **************
018200        GO TO ACEPTA-DAT1, ACEPTA-DAT2 DEPENDING ON NUM-LIN.
018300   ACEPTA-DAT1.
018400        MOVE 0 TO FLG.
018500        PERFORM ACEPTA-VALIDA-DAT1 UNTIL FLG = 1 OR ESCAPE-CODE = 1.
018600        GO TO FIN-ACEPTA.
018700   ACEPTA-DAT2.
018800        PERFORM ACEPTA2.
018900        GO TO FIN-ACEPTA.
019000   FIN-ACEPTA.
019100  ************
019200        EXIT.
019300   ACEPTA2.
019400        ACCEPT DAT-2.
019500   MANTENCION.
019600  ***********
019700        DISPLAY MENU-3.
019800        DISPLAY MENU-GEN.
019900        DISPLAY FIN-DAT.
020000        MOVE 0 TO FLG-MODI.
020100        MOVE 0 TO FLG.
020200        PERFORM ACEPTA-DAT1-MOD UNTIL FLG = 2 OR ESCAPE-CODE = 1.
020300        IF FLG = 2
020400            DISPLAY DAT-01
020500            DISPLAY DAT-02
020600            MOVE 0 TO D-CORR
020700            PERFORM RESTO UNTIL D-CORR = 1.
020800   ACEPTA-DAT1-MOD.
020900  ****************
021000        MOVE 0 TO FLG.
021100        DISPLAY DAT-1.
021200      * ACCEPT DAT-1.
021300        ACCEPT ESCAPE-CODE FROM ESCAPE KEY.
021400        IF ESCAPE-CODE NOT = 1
021500           PERFORM VAL-PARA-MODI.
021600          IF ESCAPE-CODE = 1
021700              GO TO PROCESO.
021800   VAL-PARA-MODI.
021900  ***************
022000        READ ARCH-PROV INVALID KEY MOVE 1 TO FLG.
022100        IF FLG NOT = 1
022200           MOVE 2 TO FLG.
022300   GRABA-REG.
022400  **********
022500        IF RESP-1 = 1
022600          WRITE REG-PROV INVALID KEY DISPLAY "IMPOSIBLE ES INGRESO"
022700          CALL PROGRAM "#W"
022800        ELSE
```

ANEXO 2

```
000100  IDENTIFICATION DIVISION.
000200 *-----------------------
000300  PROGRAM-ID. EXPIN03X.
000400  AUTHOR. ALVARO VASQUEZ LANGER.
000500 ************************************************************
000600 *     INGRESO Y MANTENCION DE CODIGOS DE PROVEEDORES       *
000700 ************************************************************
000800  ENVIRONMENT DIVISION.
000900 *--------------------
001000  CONFIGURATION SECTION.
001100 *---------------------
001200  FIGURATIVE-CONSTANTS.
001300 *---------------------
001400         PROT IS "85"
001500         DESPROT IS "81".
001600  INPUT-OUTPUT SECTION.
001700 *---------------------
001800  FILE-CONTROL.
001900 *-------------
002000         SELECT ARCH-PROV  ASSIGN TO "EXCPROV", "DISK", NODISPLAY
002100                ORGANIZATION IS INDEXED
002200                ACCESS MODE IS RANDOM
002300                RECORD KEY IS CODIGO
002400                ALTERNATE RECORD KEY 01 IS DESCRIPCION.
002500         SELECT WS ASSIGN TO "WS", "DISPLAY"
002600                ORGANIZATION IS SEQUENTIAL
002700                ACCESS MODE IS RANDOM
002800                PFKEY IS PF-KEY.
002900  DATA DIVISION.
003000 *--------------
003100  FILE SECTION.
003200 *--------------
003300   FD WS   LABEL RECORD IS OMITTED.
003400   01  REG-PAN       PIC X(1924).
003500
003600   FD ARCH-PROV
003700         LABEL RECORDS ARE STANDARD
003800         VALUE OF FILENAME IS "EXCPROV"
003900                  LIBRARY  IS "CLCDATA"
004000                  VOLUME   IS "FIJOJO".
004100   01 REG-PROV.
004200     02 CODIGO      PIC X(4).
004300     02 DESCRIPCION PIC X(30).
004400
004500  WORKING-STORAGE SECTION.
004600 *------------------------
004700  77  TITULO-1            PIC X(36)  VALUE SPACES.
004800  77  COD-P               PIC X(04)  VALUE SPACES.
004900  77  DESC-P              PIC X(30)  VALUE SPACES.
005000  77  PF-KEY              PIC 99     VALUE ZERO.
005100  77  TITULO-2            PIC X(36)  VALUE SPACES.
005200  77  CODIGO-P            PIC X(04)  VALUE SPACES.
005300  77  DESCRIPCION-P       PIC X(30)  VALUE SPACES.
005400  77  SW-ENCONTRADO       PIC 9      VALUE ZERO.
005500  77  SW-VALIDA           PIC 9      VALUE ZERO.
005600  77  SW-LECTURA          PIC 9      VALUE ZERO.
005700  77  SW-CODIGO           PIC 9      VALUE ZERO.
```

```
WANG VS INTEGRATED EDITOR - VERSION  6.13.01        9:14 12/29/86   PAGE
INPUT FILE IS EXPIN03X IN LIBRARY CLCLISTE ON VOLUME SISTCO

016900  BUSCA-PROVEEDOR.
017000 *----------------
017100      MOVE CODIGO-P TO CODIGO.
017200      READ ARCH-PROV WITH HOLD INVALID KEY MOVE 1 TO Sw-ENCONTRADO.
017300      IF SW-ENCONTRADO = 1 THEN DISPLAY "PROVEEDOR NO EXISTE!!!!!".
017400      IF Sw-ENCONTRADO = 0 THEN
017500       MOVE CODIGO TO CODIGO-P
017600       MOVE DESCRIPCION TO DESCRIPCION-P
017700       MOVE 0 TO PF-KEY
017800       MOVE PROT TO FAC OF F1-CODIGO FAC OF F1-DESCRIPCION
017900      MOVE "(ENTER)= ELIMINAR  PF2=CONTINUAR  PF16=MENU" TO MENSAJE
018000       DISPLAY AND READ MENU2 ON WS PFKEYS  02, 16.
018100      IF SW-ENCONTRADO = 0 AND PF-KEY = 0 THEN
018200         DELETE ARCH-PROV.
018300      MOVE DESPROT TO FAC OF F1-CODIGO FAC OF F1-DESCRIPCION.
018400  MODIFICACION.
018500 *------------
018600      MOVE SPACES TO CODIGO-P DESCRIPCION-P.
018700      MOVE "MODIFICACION DE CODIGOS DE PROVEEDORES" TO TITULO-1.
018800      MOVE "**************************************" TO TITULO-2.
018900      MOVE "(ENTER) =CONTINUAR   (PF16) = MENU PRINCIPAL" TO
019000      MENSAJE.
019100      MOVE 0 TO PF-KEY SW-ENCONTRADO.
019200      MOVE PROT TO FAC OF F1-DESCRIPCION.
019300      DISPLAY AND READ MENU2 ON WS
019400      PFKEYS  16.
019500      IF PF-KEY = 0  THEN
019600            PERFORM BUSCA-MODIFICA.
019700      MOVE DESPROT TO FAC OF F1-DESCRIPCION.
019800  BUSCA-MODIFICA.
019900 *---------------
020000      MOVE 0 TO SW-1.
020100      MOVE DESPROT TO FAC OF F1-DESCRIPCION.
020200      MOVE CODIGO-P TO CODIGO.
020300      MOVE DESCRIPCION-P TO DESCRIPCION.
020400      READ ARCH-PROV WITH HOLD INVALID KEY MOVE 1 TO SW-ENCONTRADO.
020500      IF SW-ENCONTRADO = 1 THEN DISPLAY "PROVEEDOR NO EXISTE!!!!!".
020600      IF SW-ENCONTRADO = 0 THEN
020700       MOVE CODIGO      TO COD-P
020800       MOVE DESCRIPCION TO DESC-P
020900       MOVE DESCRIPCION TO DESCRIPCION-P
021000       DELETE ARCH-PROV
021100       MOVE 0 TO PF-KEY
021200       MOVE "(ENTER)= GRABA MODI. PF2=CONTINUAR  PF16=MENU PRI." TO
021300       MENSAJE
021400       DISPLAY AND READ MENU2 ON WS PFKEYS  02, 16.
021500      IF SW-ENCONTRADO = 0 AND PF-KEY = 0 THEN
021600         MOVE CODIGO-P TO CODIGO
021700         READ ARCH-PROV WITH HOLD INVALID KEY MOVE 1 TO SW-1.
021800      IF SW-ENCONTRADO = 0 AND PF-KEY = 0 AND SW-1 = 1 THEN
021900         MOVE CODIGO-P TO CODIGO
022000         MOVE DESCRIPCION-P TO DESCRIPCION
022100         WRITE  REG-PROV.
022200      IF SW-ENCONTRADO = 0 AND PF-KEY = 0 AND SW-1 = 0 THEN
022300         DISPLAY "CODIGO YA EXISTE NO SE PUEDE MODIFICAR!!!!!!!!
022400         MOVE COD-P TO CODIGO
022500         MOVE DESC-P TO DESCRIPCION
```

ANEXO 3

```
WORKSTATION 127  -  USER 352  -  DAVID PARADA 531

 5:51:23 PM    TUESDAY       DECEMBER 23, 1986


********************************************************************************
****         1         2         3         4         5         6         7         8 ****
**** 12345678901234567890123456789012345678901234567890123456789012345678901234567890 ****
********************************************************************************
*  *                                                                           *  *
* 1*  ***  MESSAGE SENT BY EDITOR                                             * 1*
* 2*                                                                           * 2*
* 3*                    INFORMATION REQUIRED BY  PROGRAM  EZSESION             * 3*
* 4*                          TO DEFINE INPUT                                  * 4*
* 5*                       ACTIVE SUBPROGRAM IS WE1EDIT                        * 5*
* 6*                                                                           * 6*
* 7*    *** WANG VS INTEGRATED PROGRAM DEVELOPMENT EDITOR - VERSION  6.13.01 *** * 7*
* 8*                                                                           * 8*
* 9*  LANGUAGE = COBOL**** (ASSEMBLER, BASIC, CC, COBOL, FORTRAN,              * 9*
*10*                       GLOSSARY, PLI, PROCEDURE, RPGII)                    *10*
* 1*                                                                           * 1*
* 2*  FILE     = ********   PLEASE SPECIFY THE NAME OF THE FILE TO BE EDITED.  * 2*
* 3*                       WHEN CREATING A NEW FILE, LEAVE THE FILE NAME BLANK.* 3*
* 4*                                                                           * 4*
* 5*  PLEASE SPECIFY THE LIBRARY AND VOLUME FOR:                               * 5*
* 6*     SOURCE FILES . . . . . LIBRARY  = CLOFTE** ON VOLUME   = SISTCO       * 6*
* 7*     OBJECT FILES . . . . . PLIBRARY = CLCOBJ** ON PVOLUME  = SISTCO       * 7*
* 8*     LISTING FILES  . . . . LLIBRARY = #S52PRT* ON LVOLUME  = SYSTEM       * 8*
* 9*     DEBUG FILES (ASSEMBLER ONLY) . . DLIBRARY = S52DEBUG ON DVOLUME  = SISTCO * 9*
*20*                                                                           *20*
* 1*  WHEN COMPILING PROGRAMS, SHOULD EXISTING OBJECT, DEBUG,                  * 1*
* 2*  AND LISTING FILES BE SCRATCHED WITHOUT WARNING?          SCRATCH  = YES   * 2*
* 3*                                                                           * 3*
* 4*                      (PRESS PF15 TO END EDITOR PROCESSING)                * 4*
*  *                                                                           *  *
********************************************************************************
****         1         2         3         4         5         6         7         8 ****
**** 12345678901234567890123456789012345678901234567890123456789012345678901234567890 ****
********************************************************************************
```

```
WORKSTATION 127  -  USER 552  -  DAVID PARADA 531

 3:52:07 PM    TUESDAY       DECEMBER 23, 1986


*****************************************************************************
****          1         2         3         4         5         6         7    8 ****
**** 12345678901234567890123456789012345678901234567890123456789012345678901234567890 ****
*****************************************************************************
*  *                                                                        *  *
* 1*     *** WANG VS INTEGRATED PROGRAM DEVELOPMENT EDITOR - VERSION  6.13.01 ***      * 1*
* 2*  INPUT FILE IS CINTA     IN LIBRARY CLCPRO   ON VOLUME SISTCO                    * 2*
* 3*  THERE ARE        47 LINES IN THE EDITED TEXT.                                  * 3*
* 4*                                                                                 * 4*
* 5*  PLEASE SELECT THE DESIRED FUNCTION AND PRESS THE APPROPRIATE PF KEY:           * 5*
* 6*                                                                                 * 6*
* 7*    (1) DISP   - RETURN TO DISPLAY MODE AND RESUME TEXT EDITING                  * 7*
* 8*    (2) FIRST  - DISPLAY THE FIRST 20 LINES IN THE FILE                          * 8*
* 9*    (3) LAST   - DISPLAY THE LAST 5 LINES IN THE FILE                            * 9*
*10*    (4) PREV   - MOVE THE DISPLAY BACK 15 LINES                                  *10*
* 1*    (5) NEXT   - ADVANCE THE DISPLAY 15 LINES                                    * 1*
* 2*    (6) DOWN   - MOVE THE DISPLAY BACK 1 LINE                                    * 2*
* 3*    (7) UP     - ADVANCE THE DISPLAY 1 LINE                                      * 3*
* 4*    (8) FIND   - FIND THE NEXT OCCURRENCE OF A SPECIFIED TEXT                    * 4*
* 5*    (9) MOD    - ALLOW MODIFICATION OF DISPLAYED LINES, OR ADD LINES IF AT END   * 5*
* 6*   (10) CHNG   - CHANGE OCCURRENCES OF A TEXT STRING WITHIN A SPECIFIED RANGE    * 6*
* 7*   (11) INS    - INSERT A NEW LINE AFTER THE CURSOR LINE                         * 7*
* 8*   (12) DEL    - DELETE A SPECIFIED RANGE OF LINES                               * 8*
* 9*   (13) MOVE   - MOVE A SPECIFIED RANGE OF LINES TO AFTER THE CURSOR LINE        * 9*
*20*   (14) COPY   - COPY A SPECIFIED RANGE OF LINES TO AFTER THE CURSOR LINE        *20*
* 1*   (15) COL    - SHOW CURSOR COLUMN POSITION                                     * 1*
* 2*   (16) MENU   - ACTIVATE SPECIAL MENU (SET, RESTART, CREATE, REPLACE, RENUMBER,  * 2*
* 3*                 EXTERNAL COPY, UTILITIES, PRINT, EXIT)                         * 3*
* 4*  PRESS ENTER FOR A DESCRIPTION OF RANGE, TARGET, AND FIND SPECIFICATIONS.      * 4*
*  *                                                                        *  *
*****************************************************************************
****          1         2         3         4         5         6         7    8 ****
**** 12345678901234567890123456789012345678901234567890123456789012345678901234567890 ****
*****************************************************************************
```

ANEXO 4

```
VS VS INTEGRATED EDITOR - VERSION  6.13.01        15:53 12/23/86   PAGE     1
PUT FILE IS CINTA     IN LIBRARY CLCPRO   ON VOLUME SISTCO

                                                                               000
                                                                               000
 PROCEDURE    CINTA                                                            000
 AUTHOR       ROLANDO SALAS 365                                                000
 DATE         06/22/86                                                         000
                                                                               000
   DECLARE &CINTA AS STRING (5)                                                000
   DECLARE &NOMBRE AS STRING (6)                                               000
   DECLARE &SIZE INTEGER                                                       000
   PROMPT   CENTER LINE   "DIGITE EL NOMBRE DE LA CINTA A LEER:" ;             000
            CENTER UPPER &CINTA ;                                              000
   PROMPT  CENTER LINE   "DIGITE EL NOMBRE DEL ARCHIVO DE OUTPUT:" ;           000
            CENTER UPPER &NOMBRE ;                                             000
                                                                               000
   PROMPT  CENTER LINE   "DIGITE EL LARGO DEL REGISTRO DE INPUT :" ;           000
            CENTER NUMERIC  &SIZE   ;                                          000
                                                                               001
*******************************************************************************  *** 001
RUN TAPECOPY   IN USERAID    ON SYSTEM                                         001
*******************************************************************************  *** 001
                                                                               001
                                                                               001
STEP101:  ENTER    INPUT                                                       001
       FILE    =' '     , LIBRARY =' '     , VOLUME  =&CINTA  ,               001
       DEVICE  =TAPE    , CONVERT =N       , MULTYPE =N       ,               001
       FSEQ    ='1'     , LABEL   =NL      , HEADER2 =YES                     001
                                                                               002
STEP102:  ENTER    MOUNT                                                       002
       DEVICE  =72      , USAGE   =SH                                         002
                                                                               002
STEP103:  ENTER    TAPEFILE                                                    002
       RECFORM =F       , RECSIZE =&SIZE   , BLOCKED =N       ,               002
       BLKSIZE =80      , COMPRESS=N                                          002
                                                                               002
STEP104:  ENTER    OUTPUT                                                      002
       FILE    =&NOMBRE , LIBRARY =CLCORIG , VOLUME  =SISTCO  ,               002
       DEVICE  =DISK                                                         003
                                                                               003
STEP105:  ENTER"   DISKFILE                                                    003
       NRECS   =5000    , FILEORG =C       , RECFORM =V       ,               003
       COMPRESS=Y       , FILECLAS=S       , RETAIN  =90                     003
            ,                                                                  003
                                                                               003
                                                                               003
STEP106:  ENTER    EOJ     TO                                                  003
                                                                               003
STEP107: DISMOUNT TAPE &CINTA                                                  003
RETURN                                                                         004
```

ANEXO 5

```
COBOL VS INTEGRATED EDITOR - VERSION  6.13.01       16:01 12/23/86   PAGE     1
INPUT FILE IS TRASPASO IN LIBRARY CLCFTE   ON VOLUME SISTCO

000100  IDENTIFICATION DIVISION.
000200  PROGRAM-ID. TRASPASO.
000300  ENVIRONMENT DIVISION.
000400  CONFIGURATION SECTION.
000500  INPUT-OUTPUT SECTION.
000600  FILE-CONTROL.
000700      SELECT ENTRADA        ASSIGN TO "ENTRADA" "DISK".
000800      SELECT SALIDA         ASSIGN TO "SALIDA"  "DISK".
000900      SELECT LISTADO        ASSIGN TO "LISTADO" "PRINTER".
001000  DATA DIVISION.
001100  FILE SECTION.
001200  FD  ENTRADA               LABEL RECORDS ARE STANDARD
001300                            VALUE OF FILENAME "ENTRADA".
001400  01  REG-1.
001500      03  NUMERO-1         PIC 9(06).
001600      03  LINEA-1.
001700          05  FILLER       PIC X(01).
001800          05  PROG-ID-1    PIC X(12).
001900          05  NOM-PROG-1   PIC X(12).
002000          05  FILLER       PIC X(41).
002100      03  RESTO-1          PIC X(06).
002200
002300  FD  SALIDA                LABEL RECORDS ARE STANDARD
002400                            VALUE OF FILENAME NOM-FILE
002500                                    LIBRARY   "CLCFAC "
002600                                    VOLUME    "SISTCO"
002700                                    SPACE     REG-SAL.
002800  01  REG-2.
002900      03  NUMERO-2         PIC 9(06).
003000      03  LINEA-2.
003100          05  FILLER       PIC X(01).
003200          05  PROG-ID-2    PIC X(12).
003300          05  NOM-PROG-2   PIC X(12).
003400          05  FILLER       PIC X(41).
003500      03  RESTO-2          PIC X(06).
003600
003700  FD  LISTADO               LABEL RECORDS ARE STANDARD
003800      "                     VALUE OF FILENAME "LISTADO".
003900  01  REG-LISTADO           PIC X(132).
004000
004100  WORKING-STORAGE SECTION.
004110  01  LINEA01-PRT.
004120      03  FILLER           PIC X(05)       VALUE SPACES.
004130      03  FILLER           PIC X(13)       VALUE
004131      "NOMBRE PROGRAMA".
004140      03  FILLER           PIC X(04)       VALUE SPACES.
004150      03  FILLER           PIC X(14)       VALUE
004151      "NOMBRE EXTERNO".
004160      03  FILLER           PIC X(06)       VALUE SPACES.
004170      03  FILLER           PIC X(12)       VALUE
004175      "LINEAS PGM  ".
004180
004200  01  LINEA02-PRT.
004300      03  FILLER           PIC X(10)       VALUE SPACES.
004400      03  NOMBRE-PGM       PIC X(12).
004500      03  FILLER           PIC X(10)       VALUE SPACES.
004600      03  NUMERO-PGM       PIC X(06).
```

```
                05  FILLER              PIC X(10)        VALUE SPACES.
                05  LINEAS-PGM          PIC Z(05)9       VALUE ZEROS.

            77  CONT-LINEAS-PGM         PIC 9(06)        VALUE ZEROS.
            77  REG-SAL                 PIC 9(06)        VALUE 200.
            77  NOM-FILE                PIC X(08)        VALUE SPACES.
            77  NUM-PROG                PIC 9(06)        VALUE 0.
            77  SW                      PIC 9(01)        VALUE ZEROS.

        PROCEDURE DIVISION.

        INICIO.
            OPEN INPUT  ENTRADA.
            OPEN OUTPUT LISTADO.
            ACCEPT NUM-PROG.
            PERFORM ABRE-SALIDA.
            WRITE REG-LISTADO FROM LINEA01-PRT AFTER PAGE.

        LEE-ENTRADA.
            READ ENTRADA AT END
                 GO TO FIN-LECTURA.
            IF PROG-ID-1 = "IDENTIFICATI" AND SW = 1
               CLOSE SALIDA
               PERFORM IMPRIME-LISTADO
               PERFORM ABRE-SALIDA.
            IF PROG-ID-1 = "PROGRAM-ID."
               THEN MOVE NOM-PROG-1 TO NOMBRE-PGM
                    MOVE 1 TO SW.
            ADD 100        TO NUMERO-2.
            MOVE LINEA-1 TO LINEA-2.
            MOVE SPACES  TO RESTO-2.
            WRITE REG-2.
            ADD 1 TO CONT-LINEAS-PGM.
            GO TO LEE-ENTRADA.

        ABRE-SALIDA.
            ADD 1 TO NUM-PROG.
            MOVE NUM-PROG TO NOM-FILE NUMERO-PGM.
            OPEN OUTPUT SALIDA.
            MOVE 0 TO NUMERO-2 CONT-LINEAS-PGM.

        IMPRIME-LISTADO.
            MOVE CONT-LINEAS-PGM TO LINEAS-PGM.
            WRITE REG-LISTADO FROM LINEA02-PRT.

        FIN-LECTURA.
            CLOSE ENTRADA.
            PERFORM IMPRIME-LISTADO.
            CLOSE  LISTADO.
            STOP RUN.
```

ANEXO 6

```
000100  IDENTIFICATION DIVISION.
000200  PROGRAM-ID. SACAGORO.
000300  ENVIRONMENT DIVISION.
000400  CONFIGURATION SECTION.
000410  FIGURATIVE-CONSTANTS.
000420       GORRO IS "0A".
000500  INPUT-OUTPUT SECTION.
000600  FILE-CONTROL.
000700      SELECT ENTRADA       ASSIGN TO "ENTRADA" "DISK".
000800      SELECT SALIDA        ASSIGN TO "SALIDA"  "DISK"
000810             ORGANIZATION IS SEQUENTIAL
000820             ACCESS MODE IS DYNAMIC.
000900  DATA DIVISION.
001000  FILE SECTION.
001100  FD  ENTRADA                  LABEL RECORDS ARE STANDARD
001200                               VALUE OF FILENAME "ENTRADA".
001300  01  REG-1 PIC X(80).
001800
001900  FD  SALIDA                   LABEL RECORDS ARE STANDARD.
002000  01  REGISTRO    PIC X(80).
002100
002200  WORKING-STORAGE SECTION.
002201  01  REG-W.
002210      03  NUMEROS-1           PIC 9(6).
002220      03  FILLER              PIC X(01).
002230      03  TABLA-REG-1 OCCURS 73 TIMES.
002240          04  REG-1           PIC X(01).
002300  77  STAT                  PIC 9  VALUE ZEROS.
002400  77  SW-TERMINO              PIC 9      VALUE 0.
002500  77  I                       PIC 99     VALUE 0.
002600  77  J                       PIC 99     VALUE 0.
002700  PROCEDURE DIVISION.
002800  INICIO.
002810      MOVE SPACES TO REG-W
002900      OPEN INPUT  ENTRADA.
003000      OPEN OUTPUT SALIDA.
003100      PERFORM LEE UNTIL STAT = 1.
003200      CLOSE ENTRADA SALIDA.
003300      STOP RUN.
003400  LEE.
003500      MOVE 0 TO SW-TERMINO.
003600      READ ENTRADA RECORD INTO REG-W AT END MOVE 1 TO STAT.
003700      IF STAT NOT = 1 THEN
003710         MOVE 0 TO I
003800         PERFORM LIMPIA-LINEA UNTIL I > 73
003900                                 OR SW-TERMINO = 1
004100         WRITE REGISTRO FROM REG-W.
004200  LIMPIA-LINEA.
004210         ADD 1 TO I
004300      IF TABLA-REG-1(I) = GORRO THEN
004350         MOVE 1 TO SW-TERMINO
004355         MOVE I TO J
004400         PERFORM OTRA  UNTIL J > 73.
004500  OTRA.
004700      MOVE SPACES TO TABLA-REG-1(J).
004710      ADD 1 TO J.
004800
```

```
VS INTEGRATED EDITOR - VERSION  6.13.01        16:02 12/23/86   PAGE     1
T FILE IS PONESIG  IN LIBRARY CLCFTE   ON VOLUME SISTCO

                                                                        83021
100  IDENTIFICATION DIVISION.                                           84072
200  PROGRAM-ID. EXPIN4X.                                               84091
300  AUTHOR. SISTEMAS.                                                  83021
400  ENVIRONMENT DIVISION.                                              83021
500  CONFIGURATION SECTION.                                             83021
600  INPUT-OUTPUT SECTION.                                              83021
700  FILE-CONTROL.                                                      84072
800          SELECT ARCH-MAE ASSIGN TO "CARDEX", "DISK"
900                 ORGANIZATION IS INDEXED
1000                ACCESS MODE IS RANDOM
1100                RECORD KEY IS COD-CLOS                             84072
1200                STATUS IS MAE-STATUS.
1300         SELECT ARCH-ENT ASSIGN TO "EXCL0396", "DISK"              84072
1400                ORGANIZATION IS SEQUENTIAL
1500                ACCESS MODE IS SEQUENTIAL.
1600 DATA DIVISION.                                                     83021
1700 FILE SECTION.                                                      83021
1800 FD  ARCH-MAE
1900          LABEL RECORDS ARE STANDARD
2000          VALUE OF FILENAME IS "EXC03016"
2100                   LIBRARY  IS "CLCDATA"
2200                   VOLUME   IS "PIJ000"
2300                   SPACE    IS REC-NUM2.
2400 01  REG-MAE.                                                       84072
2500     02 COD-CLOS.                                                   84072
2600       03 COD-NUM-CLOS        PIC X(5).                            84072
2700       03 COD-DV-CLOS         PIC X.
2800     02 BUFF2.                                                      84072
2900       03 BO-ANT-1            PIC S9(3)V99.                        84072
3000       03 BO-ANT-2            PIC S9(3)V99.                        84072
3100       03 BO-ANT-3            PIC S9(3)V99.                        84072
3200       03 BO-ANT-4            PIC S9(3)V99.                        84072
3300       03 BO-ANT-5            PIC S9(3)V99.                        84072
3400       03 BO-ANT-6            PIC S9(3)V99.                        84072
3500       03 BO-ANT-7            PIC S9(3)V99.                        84072
3600       03 BO-ANT-8            PIC S9(3)V99.                        84072
3700       03 BO-ANT-9            PIC S9(3)V99.                        84072
3800       03 BO-ANT-10           PIC S9(3)V99.                        84072
3900       03 BO-ACT-1            PIC S9(3)V99.                        84072
4000       03 BO-ACT-2            PIC S9(3)V99.                        84072
4100       03 BO-ACT-3            PIC S9(3)V99.                        84072
4200       03 BO-ACT-4            PIC S9(3)V99.                        84072
4300       03 BO-ACT-5            PIC S9(3)V99.                        84072
4400       03 BO-ACT-6            PIC S9(3)V99.                        84072
4500       03 BO-ACT-7            PIC S9(3)V99.                        84072
4600       03 BO-ACT-8            PIC S9(3)V99.                        84072
4700       03 BO-ACT-9            PIC S9(3)V99.                        84072
4800       03 BO-ACT-10           PIC S9(3)V99.                        84072
4900       03 COSTO-CORR          PIC S9(10)V99.                       84072
5000       03 COSTO-PROM-ANT      PIC S9(3)V9(4).                      84072
5100       03 COSTO-PROM-ACT      PIC S9(3)V9(4).                      84072
5200       03 COSTO-ULT-COMPR     PIC S9(10)V99.                       84072
5300       03 TASA-CONS-ANUAL     PIC S9(3)V99.                        84072
5400       03 STOCK-MIN-CANT      PIC S9(3)V99.                        84072
5500       03 STOCK-MAX-CANT      PIC S9(3)V99.                        84072
5600       03 STOCK-MIN-DIAS      PIC S9(3)V99.                        84072
5700       03 STOCK-MAX-DIAS      PIC S9(3)V99.
```

```
                05 TIP-MAT                PIC X.                                 840
                05 DIAS-CONSUMO           PIC 9(2).                              840
                05 IND-NAC-IMPORT         PIC 9.                                 840
                05 FECH-ULT-COMPR         PIC 9(6).                              840
                05 FECH-ULT-MOVIM         PIC 9(6).                              840
                05 BLANCOS                PIC X(10).                             840
            FD  ARCH-ENT
                    LABEL RECORDS ARE STANDARD
                    VALUE OF FILENAME IS "EXCLOS96"
                             LIBRARY  IS "CLCDRIG"
                             VOLUME   IS "FIJOCO"
                             SPACE    IS REC-NUME.
            01  REG-MAE-E.                                                        840
                02 COD-CLOS-E.                                                    840
                   05 COD-NUM-CLOS-E      PIC X(5).                              840
                   05 COD-DV-CLOS-E       PIC X.                                 840
                02 BUFF2-E.
                   05 BO-ANT-1-E          PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
                   05 BO-ANT-2-E          PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
                   05 BO-ANT-3-E          PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
                   05 BO-ANT-4-E          PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
                   05 BO-ANT-5-E          PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
                   05 BO-ANT-6-E          PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
                   05 BO-ANT-7-E          PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
                   05 BO-ANT-8-E          PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
                   05 BO-ANT-9-E          PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
                   05 BO-ANT-10-E         PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
                   05 BO-ACT-1-E          PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
                   05 BO-ACT-2-E          PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
                   05 BO-ACT-3-E          PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
                   05 BO-ACT-4-E          PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
                   05 BO-ACT-5-E          PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
                   05 BO-ACT-6-E          PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
                   05 BO-ACT-7-E          PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
                   05 BO-ACT-8-E          PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
                   05 BO-ACT-9-E          PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
                   05 BO-ACT-10-E         PIC 9(7)V99.                           840
                   05 FILLER              PIC X.
```

```
                03 COSTO-CORR-E          PIC  9(09)V99.                    84072
                03 FILLER                PIC  X.
                03 COSTO-PROM-ANT-E      PIC  9(7)V9(4).                   84072
                03 FILLER                PIC  X.
                03 COSTO-PROM-ACT-E      PIC  9(7)V9(4).                   84072
                03 FILLER                PIC  X.
                03 COSTO-ULT-COMPR-E     PIC  9(09)V99.                    84072
                03 FILLER                PIC  X.
                03 TASA-CONS-ANUAL-E     PIC  9(7)V99.                     84072
                03 FILLER                PIC  X.
                03 STOCK-MIN-CANT-E      PIC  9(7)V99.                     84072
                03 FILLER                PIC  X.
                03 STOCK-MAX-CANT-E      PIC  9(7)V99.                     84072
                03 FILLER                PIC  X.
                03 STOCK-MIN-DIAS-E      PIC  9(7)V99.                     84072
                03 FILLER                PIC  X.
                03 STOCK-MAX-DIAS-E      PIC  9(7)V99.                     84072
                03 FILLER                PIC  X.
                03 TIP-MAT-E             PIC  X.                           84072
                03 DIAS-CONSUMO-E        PIC  9(2).                        84072
                03 IND-NAC-IMPORT-E      PIC  9.                           84072
                03 FECH-ULT-COMPR-E      PIC  9(6).                        84072
                03 FECH-ULT-MOVIM-E      PIC  9(6).                        84072
                03 BLANCOS-E             PIC  X(10).                       83021
            WORKING-STORAGE SECTION.
            77  REG-NUM1          PIC 9(06) VALUE 35000.
            77  REG-NUM2          PIC 9(06) VALUE 35000.                   83030
            77  MAE-STATUS        PIC X(2)  VALUE SPACES.
            77  SW-FIN            PIC 9     VALUE 0.
            77  SW                PIC 9     VALUE 0.                       83021
            PROCEDURE DIVISION.                                           84072
           *---------------------                                         83021
            PRINCIPAL.
                MOVE 0 TO SW-FIN.
                OPEN INPUT ARCH-ENT OUTPUT  ARCH-MAE.                     84072
                PERFORM LEE-GRABA  UNTIL SW-FIN = 1.
                CLOSE ARCH-MAE ARCH-ENT.
                STOP RUN.
            LEE-GRABA.
                READ ARCH-ENT NEXT RECORD AT END MOVE 1 TO SW-FIN.
                IF SW-FIN NOT = 1 THEN
                  PERFORM PONE-CEROS
                  MOVE COD-CLOS-E          TO COD-CLOS                    84072
                  MOVE BO-ANT-1-E          TO BO-ANT-1                    84072
                  MOVE BO-ANT-2-E          TO BO-ANT-2                    84072
                  MOVE BO-ANT-3-E          TO BO-ANT-3                    84072
                  MOVE BO-ANT-4-E          TO BO-ANT-4                    84072
                  MOVE BO-ANT-5-E          TO BO-ANT-5                    84072
                  MOVE BO-ANT-6-E          TO BO-ANT-6                    84072
                  MOVE BO-ANT-7-E          TO BO-ANT-7                    84072
                  MOVE BO-ANT-8-E          TO BO-ANT-8                    84072
                  MOVE BO-ANT-9-E          TO BO-ANT-9                    84072
                  MOVE BO-ANT-10-E         TO BO-ANT-10                   84072
                  MOVE BO-ACT-1-E          TO BO-ACT-1                    8407
                  MOVE BO-ACT-2-E          TO BO-ACT-2                    8407
                  MOVE BO-ACT-3-E          TO BO-ACT-3                    8407
                  MOVE BO-ACT-4-E          TO BO-ACT-4
```

```
                                                                                84072
           MOVE BO-ACT-5-E          TO BO-ACT-5                                 84072
           MOVE BO-ACT-6-E          TO BO-ACT-6                                 84072
           MOVE BO-ACT-7-E          TO BO-ACT-7                                 84072
           MOVE BO-ACT-8-E          TO BO-ACT-8                                 84072
           MOVE BO-ACT-9-E          TO BO-ACT-9                                 84072
           MOVE BO-ACT-10-E         TO BO-ACT-10                                84072
           MOVE COSTO-CORR-E        TO COSTO-CORR                               84072
           MOVE COSTO-PROM-ANT-E    TO COSTO-PROM-ANT                           84072
           MOVE COSTO-PROM-ACT-E    TO COSTO-PROM-ACT                           84072
           MOVE COSTO-ULT-COMPR-E   TO COSTO-ULT-COMPR                          84072
           MOVE TASA-CONS-ANUAL-E   TO TASA-CONS-ANUAL                          84072
           MOVE STOCK-MIN-CANT-E    TO STOCK-MIN-CANT                           84072
           MOVE STOCK-MAX-CANT-E TO STOCK-MAX-CANT                              84072
           MOVE STOCK-MIN-DIAS-E    TO STOCK-MIN-DIAS                           84072
           MOVE STOCK-MAX-DIAS-E    TO STOCK-MAX-DIAS                           84072
           MOVE TIP-MAT-E           TO TIP-MAT                                  84072
           MOVE DIAS-CONSUMO-E      TO DIAS-CONSUMO                             84072
           MOVE IND-NAC-IMPORT-E    TO IND-NAC-IMPORT                           84082
           MOVE FECH-ULT-COMPR-E    TO FECH-ULT-COMPR                           84072
           MOVE FECH-ULT-MOVIN-E    TO FECH-ULT-MOVIN                           84072
           MOVE BLANCOS-E           TO BLANCOS.                                 84072
       IF SW = 0 THEN
            MOVE ZEROES TO REG-MAE
            WRITE REG-MAE INVALID KEY STOP RUN.
       IF SW = 1 THEN
            WRITE REG-MAE INVALID KEY STOP RUN.
       MOVE 1 TO SW.
   PONE-CEROS.
       IF BO-ANT-1-E = SPACES THEN MOVE ZEROES TO BO-ANT-1.
       IF BO-ANT-2-E = SPACES THEN MOVE ZEROES TO BO-ANT-2.
       IF BO-ANT-3-E = SPACES THEN MOVE ZEROES TO BO-ANT-3.
       IF BO-ANT-4-E = SPACES THEN MOVE ZEROES TO BO-ANT-4.
       IF BO-ANT-5-E = SPACES THEN MOVE ZEROES TO BO-ANT-5.
       IF BO-ANT-6-E = SPACES THEN MOVE ZEROES TO BO-ANT-6.
       IF BO-ANT-7-E = SPACES THEN MOVE ZEROES TO BO-ANT-7.
       IF BO-ANT-8-E = SPACES THEN MOVE ZEROES TO BO-ANT-8.
       IF BO-ANT-9-E = SPACES THEN MOVE ZEROES TO BO-ANT-9.
       IF BO-ANT-10-E = SPACES THEN MOVE ZEROES TO BO-ANT-10.

       IF BO-ACT-1-E = SPACES THEN MOVE ZEROES TO BO-ACT-1.
       IF BO-ACT-2-E = SPACES THEN MOVE ZEROES TO BO-ACT-2.
       IF BO-ACT-3-E = SPACES THEN MOVE ZEROES TO BO-ACT-3.
       IF BO-ACT-4-E = SPACES THEN MOVE ZEROES TO BO-ACT-4.
       IF BO-ACT-5-E = SPACES THEN MOVE ZEROES TO BO-ACT-5.
       IF BO-ACT-6-E = SPACES THEN MOVE ZEROES TO BO-ACT-6.
       IF BO-ACT-7-E = SPACES THEN MOVE ZEROES TO BO-ACT-7.
       IF BO-ACT-8-E = SPACES THEN MOVE ZEROES TO BO-ACT-8.
       IF BO-ACT-9-E = SPACES THEN MOVE ZEROES TO BO-ACT-9.
       IF BO-ACT-10-E = SPACES THEN MOVE ZEROES TO BO-ACT-10.
```

ANEXO 7

ANEXO 8

```
      PROGRAM PERFORMANCE EVALUATION REPORT FOR REPIPRE1
PARAGRAPH NAME

 NI  CPU_TIME  SOURCE_STATEMENT_FROM_WHICH_TRANSFER_OF_CONTROL_OCCURRED_

  1      0.00  018700      ON PFKEY 16   PERFORM DESPROTEGE   GO TO MENU-1.
  1       .00  019100      PERFORM DESPROTEGE
  1       .00  023700      ON PFKEY 16   PERFORM DESPROTEGE   GO TO MENU-1.
      .02

  2       .02  015900      IF PF-KEY = 3 PERFORM ELIMINA UNTIL PF-KEY = 16.
      .02

  1      0.00  024200      PERFORM ELIMINAR.
     0.00

  1       .00  017000      PERFORM GRABAR.
      .00

  2       .03  015700      IF PF-KEY = 1 PERFORM INGRESA  UNTIL PF-KEY = 16.
      .03

  1       .06  014900 SEC2 SECTION.
      .06

  2       .00  018800      PERFORM LECTURA.
  1       .02  019400      PERFORM LECTURA.
  1       .00  023800      PERFORM LECTURA.
  1       .02  024400      PERFORM LECTURA.
      .06

  4       .05  015300      PERFORM LIMPIAR-CAMPOS
  1       .00  017500      PERFORM LIMPIAR-CAMPOS.
      .06

  1       .00  015100      OPEN I-O ARCH-PRE PANTALLA.
  1      0.00  016800      ON PFKEY 16   GO TO MENU-1.
  1       .00  018700      ON PFKEY 16   PERFORM DESPROTEGE   GO TO MENU-1.
  1      0.00  023700      ON PFKEY 16   PERFORM DESPROTEGE   GO TO MENU-1.
      .02

  2       .00  015800      IF PF-KEY = 2 PERFORM MODIFICA UNTIL PF-KEY = 16.
  1       .00  019010                          GO TO MODIFICA.
      .02

  1      0.00  019200      PERFORM MODIFICAR.
     0.00

  3       .04  018000      PERFORM PROTEGE.
  2       .02  023000      PERFORM PROTEGE.
      .06

  1      0.00  015100      OPEN I-O ARCH-PRE PANTALLA.
```

CONTROL DE PROGRAMAS

HOJA NRO:
DE :

NOMBRE PROGRAMA	NOMBRE EJECUTOR	FECHA ENTREGA	FECHA TERMINO