INDICE

1. INTRODUCCION

El presente trabajo ha sido desarrollado con el propósito de participar en la preparación profesional de los estudiantes de nivel medio superior y superior, ya que se presentan bases técnicas elementales para que el alumno penetre en el mundo de la computación a través de la programación Cobol.

Asimismo, se ha buscado brindar a los alumnos de los últimos semestres de la carrera de Actuaría, un método de fácil aprendizaje de este lenguaje de programación que actualmente se imparte en los últimos cursos del plan de estudios.

Se presentan temas tales como: "Conceptos de datos e información", "Programación estructurada", "HIPO", "VTOC", con el propósito de que el estudiante en el medio de la informática cuente con el conocimiento de las técnicas empleadas actualmente.

La estructura del presente trabajo está diseñada considerando principalmente las divisiones del lenguaje Cobol como capítulos propios. Para la división de procedimientos (Procedure Division) capítulo 5, se ha decidido presentar unas de las técnicas actualmente empleadas para el análisis de problemas y desarrollo de sistemas, en el cual se hace referencia a verbos Cobol, que se presentarán posteriormente en el capítulo 6 "COMANDOS COBOL".

1.1. NOTA DE PRESENTACION.

COBOL; Common Bussines Oriented Lenguaje (Lenguaje común --- orientado a los negocios), es un lenguaje de programación diseñado bajo la estructura del idioma Inglés, encontrando su mayor campo de aplicación en el comercio y la administración, ya que maneja una gran cantidad de información a nivel de -- INPUT/OUTPUT (ENTRADA/SALIDA), permitiendo ahorro de tiempo y eliminación de errores, asimismo por sus características - de manejo, da facilidad para la creación de bancos de datos.

La información de este manual, sigue las especificaciones dadas para su uso y aplicación, presentadas por CODASYL COBOL, además las últimas modificaciones aceptadas por las instituciones que rigen el uso de este lenguaje de programación.

2. CONCEPTOS BASICOS.

2.1. OCHO ELEMENTOS BASICOS.

El lenguaje Cobol está estructurado dentro del manejo de ocho elementos, que son los siguientes:

- Caracteres (characters)
- Palabras reservadas (reserved word)
- Nombres programador (programmer-supplied names)
- Literales (literals)
- Símbolos (symbols)
- Números de nivel (level number)
- Imagen o Descripción (pictures)
- Verbos (Verbs)

Cobol tiene reglas de sintaxis propias, para el uso y manejo de estos ocho elementos.

2.1.1. Caracteres.

El elemento básico bajo el cual se estructura cual quier lenguaje, es el conjunto de símbolos permitidos por el mismo y en el caso de Cobol, se cuenta con los siguientes:

a) Caracteres alfabéticos; de la A a la Z (las letras del alfabeto inglés que son 27.).

b) Caracteres numéricos; del 0 al 9 (diez dígitos del

cero al nueve.).

c) Caracteres especiales;

espacio en blanco	∅	punto y coma	;
signo más	+	punto (punto decimal)	.
signo menos (guión)	-	comillas	"
asterisco	*	apóstrofe	'
diagonal	/	paréntesis izquierdo	(
signo igual	=	paréntesis derecho	)
coma	,	signo mayor que	>
andpersan	&	signo menor que	<
gato	#	arroba	@

2.1.2. Palabras Reservadas.

Definición: Una palabra es una cadena o sucesión de caracteres permitidos, en la cual:

1. No puede haber más de 30 caracteres.
2. No hay blancos intermedios.
3. No debe comenzar ni terminar en guión.
4. Dos palabras sucesivas deben estar separadas al menos por un blanco.

Definición: Una "palabra reservada" es una cadena de caracteres que el lenguaje permite emplear y además tiene un significado especial para el compilador Cobol.

Los tres tipos de palabras reservadas que el lenguaje permite son:

- Palabras clave o llave
- Palabras opcionales
- Conectivas.

. Palabras clave o llave.

Cualquier indicación, comando o instrucción Cobol debe contener al menos una palabra clave, sin la cual - la instrucción o indicación pierde su sentido dentro del programa, el significado de esto es que el compilador ejecutará la instrucción solo con esta palabra.

. Palabras Opcionales.

Son incluídas en cualquier versión Cobol, y son utilizadas para dar mayor habilidad al programador, dado - que pueden ser o no empleadas al escribir una instrucción o indicación.

Ejemplo:

```
WRITE LINEA                              0
WRITE LINEA AFTER 1                      0
WRITE LINEA AFTER ADVANCING 1            0
WRITE LINEA AFTER ADVANCING 1 LINES
```

. Palabras Conectivas.

Son aquellas que indican la presencia de una condi--- ción compuesta, para lo cual se tienen:

- Conectivas de identificador (OF o IN) asocia un-

nombre de dato, dentro de un área a otra área de memoria o archivo.

- Conectivas lógicas, son empleadas para formar -- condiciones y son: AND, OR, AND NOT, OR NOT.

2.1.3. "Nombres Programador"

Los "nombres programador" se refieren a nombre de datos, rutinas, subrutina, áreas de trabajo, etc. Debiéndose ajustar a las siguientes reglas:

1. No puede exceder a los 30 caracteres; puede ser una combinación de caracteres alfabéticos, numéricos, incluyendo el guión.
2. Se puede estructura con el uso del guión (-), - el cual no debe encontrarse ni al principio ni al final de la palabra.
3. No debe contener puntos intermedios.
4. No deben aparecer espacios blancos intermedios.
5. Una palabra reservada no puede aparecer como -- "nombre programador".
6. Los nombre referidos a identificar datos o pá-- rrafos (procedimientos) deben contener al menos una letra.

2.1.4. Literales.

Se entiende por literal, el valor asociado a cualquier "nombre programador" referido a un dato, ya sea cons-

tante o variable.

Existen:

1. Literal numérica. Se define como el valor numérico asignado a un "nombre programador" el cual puede ser una combinación de los dígitos 0 al 9 y no puede ser mayor a 23 dígitos.
2. Literal no numérica. Es una cadena de caracteres enmarcados por comillas (" o ') y no puede exceder a 120 caracteres.

2.1.5. Símbolos.

Los símbolos son caracteres especiales que se pueden utilizar en el desarrollo de un programa Cobol, y se clasifican en:

1. Aritméticos y Condicionales. Generalmente son aquellos que se emplean al escribir o establecer una relación, éstos son:

   Aritméticos.

+	signo más.
-	signo menos
*	signo multiplicación
/	signo división
**	signo de exponente

   Condicionales.

=	igual que

> mayor que

< menor que

En el caso de los símbolos condicionales es indistinto emplearlos de la siguiente forma:

= 0 EQUAL TO

> 0 LESS THAN

< 0 GREATER THAN

2. Puntuación.

Algunos de estos símbolos son empleados para -- hacer una codificación más clara, sin que su -- ausencia provoque errores de compilación.

. punto

, coma

; punto y coma

" o ' comillas o apóstrofe

(paréntesis izquiero

) paréntesis derecho

Cuando una instrucción antecede a un nombre de- párrafo, esta instrucción deberá terminar con - un punto.

El uso de paréntesis se ocupa para el caso de - cálculos matemáticos, por lo cual siempre debe existir dentro de la expresión el mismo número de paréntesis izquierdos que derechos.

2.2. FORMA DE CODIFICACION.

Se entiende por codificación Cobol, al hecho de transcribir una serie de especificaciones o instrucciones para el computador, en la cual se emplean los ocho elementos básicos anteriormente mencionados.

La figura número 1 muestra una hoja de codificación Cobol, que es utilizada como una herramienta de programación. La hoja de codificación es dividida en dos regiones, una para uso y control del trabajo codificado, la cual contiene datos tales como: control de páginas codificadas, nombre del responsable, etc. y la región propia de codificación, la cual se encuentra dividida en dos zonas o márgenes, que son identificados por columnas, y se emplean de la siguiente forma.

Columnas.

De la 1 a la 6 — El programador puede emplear estas columnas para incluir una secuencia por línea de su programa, se ocupan como elemento de referencia.
La información de estas columnas no ejercen efecto en la compilación del programa.
Usualmente se emplean de la columna 1 a la columna 3 para indicar la página y de la columna 4 a la 6 para la secuencia dentro de la página.

7 — Esta columna tiene distintas finalidades_

de acuerdo al caracter que contenga, los cuales pueden ser:

* indica al compilador cobol que la parte restante de una línea es considerada como un comentario.

/ indica salto de página, en la impresión del programa.

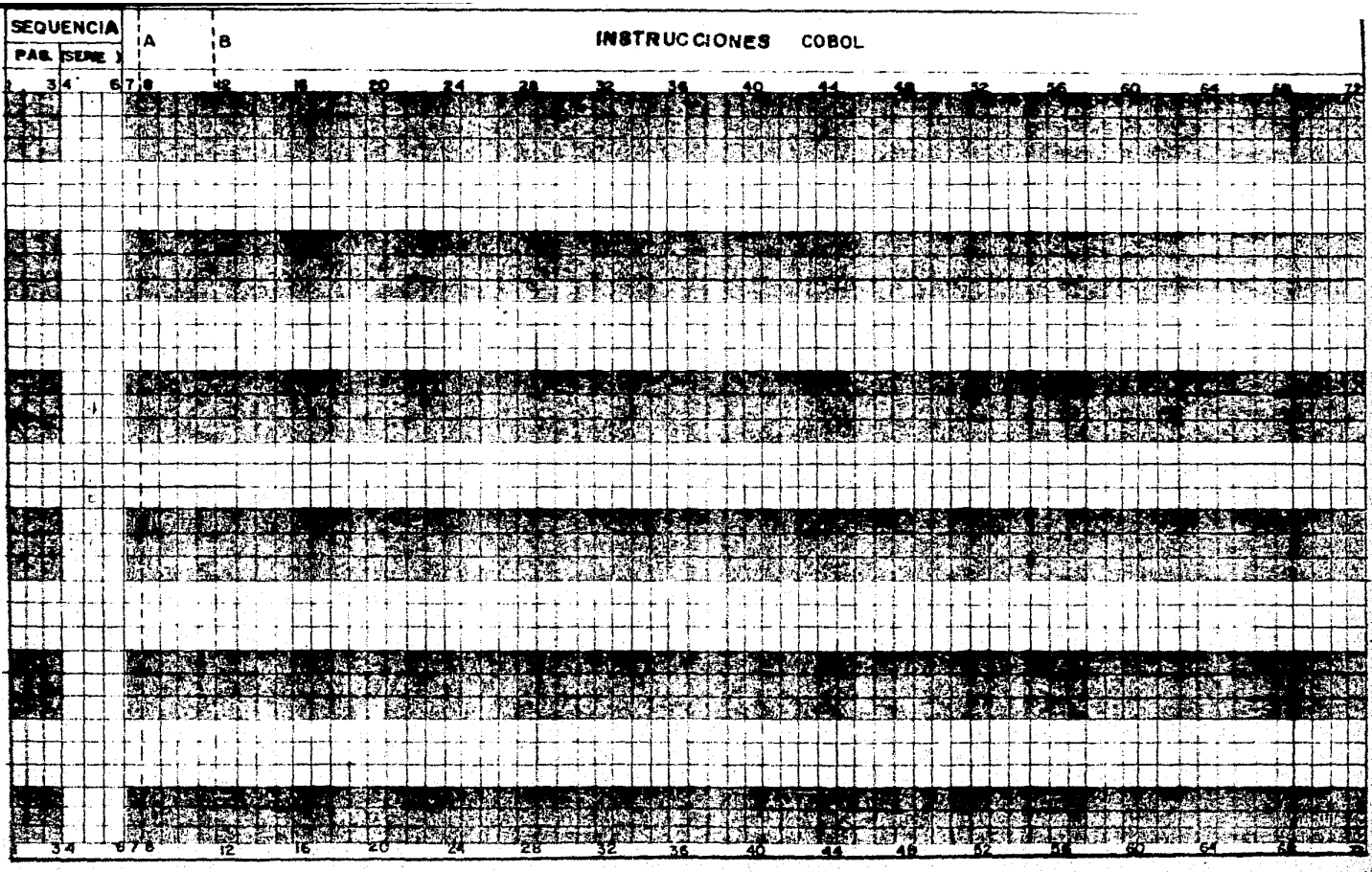
- indica la continuación de una literal no numérica de la línea anterior.

De la 8 a la 11 Se consideran los nombres de las divisiones, secciones, párrafos, etiquetas, etc.

De la 12 a la 72 Se consideran como Zona B, se emplea para la codificación de las instrucciones de que se requiere en el programa. Estas columnas son empleadas de dos formas:

a. de igual manera que las columnas 1-6
b. introduciendo el nombre de identificación del programa.

De cualquiera de estas dos formas, no interfiere en la ejecución del programa o en su compilación.


SEQUENCIA
PAG. SERIE
A
B
INSTRUCCIONES COBOL

2.3. TERMINOS

1	ACCESAR	Disponer de la información de un campo,- registro o archivo, de tal forma que se pueda consultar y/o modificar.
2	ALFANUMERICO/CARACTER	Representación de información que puede considerarse numérica o alfabética./Campo, conjunto de caracteres alfanuméricos.
3	ALGORITMO	Secuencia lógica de instrucciones que -- lleva a la solución de un problema, como características tiene: 1. Debe tener 0 ó más entradas. 2. Debe tener 1 ó más salidas. 3. Un número finito de instrucciones. 4. Debe ser efectivo y eficiente.
4	ARCHIVO	Conjunto de registros.
5	BIT	Binary Digit, 0 ó 1.
6	BYTE	Secuencia de BITS, usualmente son 8 BITS.
7	CAMPO	Sucesión de caracteres
8	CARACTER	Símbolo permito por el lenguaje, el cual puede ser: numérico, alfabético, alfanumérico, o bien también especial.
9	COBOL	Common Business-Oriented Lenguaje.

Lenguaje Común Orientado a los Negocios.

10 CODIFICACION — Transcribir un programa Cobol a hojas de codificación, acción propia de programar.

11 COMPILADOR — Programa traductor de un superlenguaje a código máquina.

12 COMPILAR — Acción de generar el programa objeto.

13 DATO — Información codificada.

14 DIAGRAMA DE FLUJO — Representación gráfica de un algoritmo.

15 DIGITO — Símbolo empleado para representar números en un sistema específico.

16 EJECUCION — Correr un programa, hacer que el computador deduzca lo especificado.

17 IDENTIFICADOR — Nombre asignado a un programa, rutina o subrutina.

18 ITERACION — Representación de un proceso (LOOP).

19 JCL — Job Control Lenguaje, lenguaje empleado en ciertos sistemas para mandar ejecutar un proceso.

20 MEMORIA — Parte de la computadora que se emplea para almacenar datos o instrucciones, la que puede ser principal o auxiliar.

21 MNEMONICO — Memory-auding, abreviatura de un nombre

que auxilia para la rápida identificación por ejemplo:
APPA - APELLIDO PATERNO.

22 OFF-LINE — Método de proceso, en el que no existe una comunicación directa entre usuario entre usuario y sistema; llámese también -- BATCH.

23 ON-LINE — Método de proceso, en el que existe una comunicación directa entre el usuario y el sistema; llámese también EN-LINEA o -- TIEMPO REAL.

24 PALABRA — Grupo de BITS, que tienen una dirección en memoria.

25 PROGRAMA — Conjunto de instrucciones, escritas en un lenguaje de computación.

26 PROGRAMACION — Acción de escribir un programa.

27 REGISTRO — Conjunto de campos.

28 RUTINA — Conjunto de instrucciones dentro de un -- programa, destinado a resolver un problema específico, dentro de aquel.

NOMENCLATURA

Con el propósito de distinguir las palabras o especificaciones - opcionales en cada una de las divisiones se establecen las si -- guientes indicaciones.

Las palabras clave o llave aparecerán subrayadas, las opcionales vendrán dentro de [ ] corchetes, en el caso de que la instrucción opcional tenga palabras clave aparecerán subrayadas, cuando exis ta más de una forma de escribirlo, se escribirán las formas posi bles dentro de { } llaves, siguiendo el mismo criterio.

3. DIVISION DE IDENTIFICACION Y DIVISION DEL MEDIO AMBIENTE.

Todo programa Cobol consta de 4 divisiones que aparecen estrictamente en el orden siguiente:

DIVISION DE IDENTIFICACION	IDENTIFICATION DIVISION.
DIVISION DEL MEDIO AMBIENTE	ENVIRONMENT DIVISION.
DIVISION DE DATOS	DATA DIVISION.
DIVISION DE PROCEDIMIENTOS	PROCEDURE DIVISION.

3.1. DIVISION DE IDENTIFICACION (IDENTIFICATION DIVISION.).

La función de esta división es proporcionar la información necesaria respecto al programador, al programa e instalaciones en que se desarrolla el programa.

Forma de codificación.

Columna 8

IDENTIFICATION DIVISION.
ID DIVISION.

PROGRAM-ID	nombre del programa.
AUTHOR.	nombre del programador.
INSTALLATION.	instalación empleada.
DATE-WRITTEN.	fecha de codificación.
DATE-COMPILED.	fecha de compilación.
SECURITY.	clave de exclusividad para uso del prog.
REMARKS.	comentarios.

Las declaraciones obligatorias para esta división son:

IDENTIFICATION DIVISION. o ID DIVISION. y PROGRAM-ID.

IDENTIFICATION DIVISION. ID DIVISION.	Es la primera división de un programa Cobol.
PROGRAM-ID.	Nombre del Programa. Proporciona el nombre de identificación del programa, fundamental para el caso de programas catalogados.
AUTHOR.	Nombre del programador. Cuando se es empleada esta declaración se pondrá el nombre de la persona que codificó el programa, o un mnemónico que lo identifique; no se debe olvidar el punto final después del nombre.
INSTALLATION.	Instalación ocupada. Cuando se es empleada esta declaración generalmente se escribe el nombre de la institución donde se hizo el programa o para la cual fue elaborado; no se debe olvidar el punto final.
DATE-WRITTEN.	Fecha de codificación.

Cuando se es empleada esta declaración, se debe anotar la fecha de codificación del programa, no debe de olvidar el punto final después de la fecha.

DATE-COMPILED. Cuando se es empleada esta declaración, en el momento de compilar el programa el computador imprimirá la fecha del día.

SECURITY. Clave de exclusividad, no es muy usual, funge como el REMARKS.

REMARKS. Comentario.
Cuando se es empleada esta declaración se escribe una descripción breve del objetivo del programa, estructura, etc. Sin embargo, existen otras formas de incluir un comentario, como el asterisco (*) en la columna 7, en cualquier parte del programa.

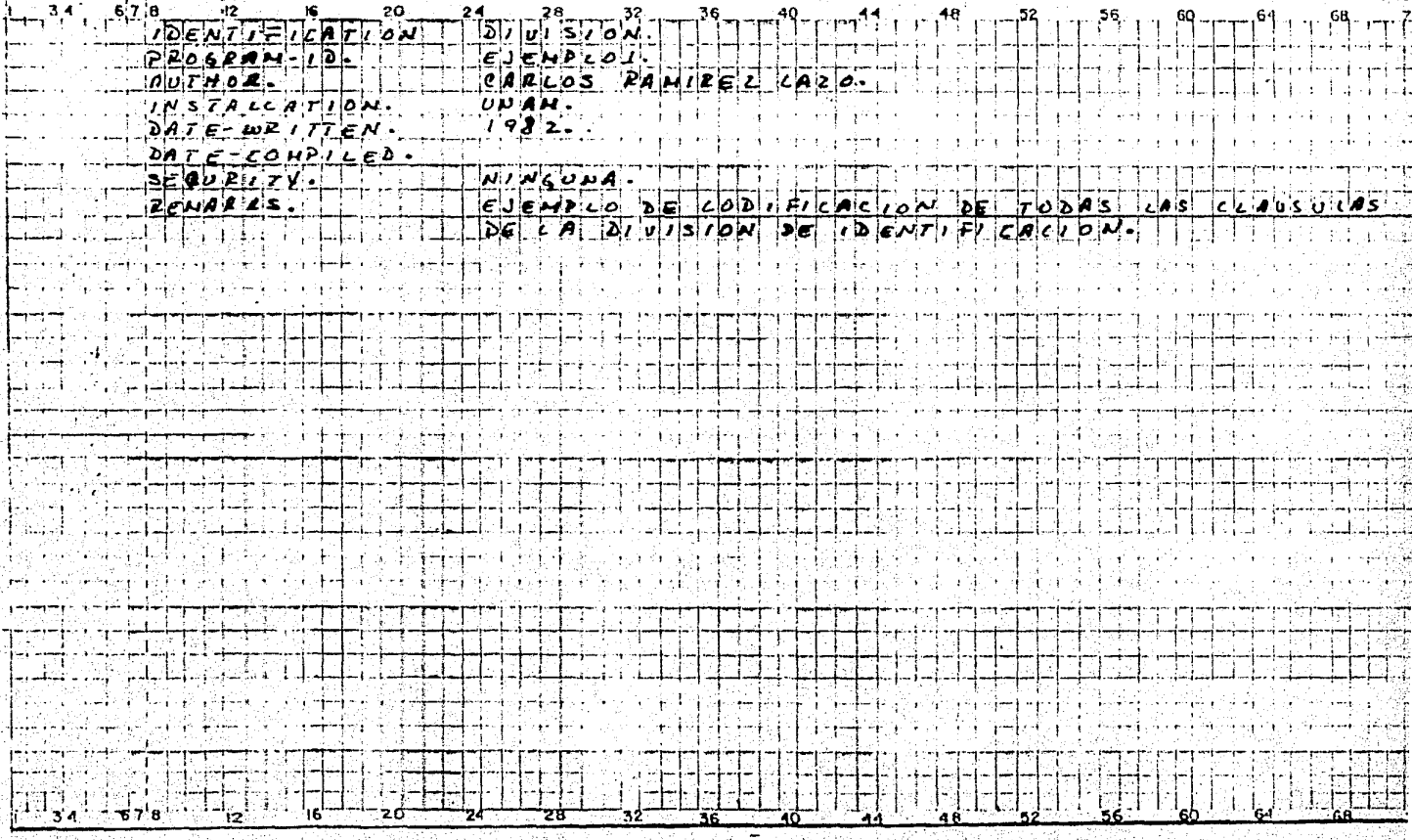
La figura dos, muestra un ejemplo de codificación de esta división.

SEQUENCIA

PAG. (SERIE)

INSTRUCCIONES COBOL

FIGURA 2



3.2. DIVISION DEL MEDIO AMBIENTE. (ENVIRONMENT DIVISION.).

La división del medio ambiente es muy importante, ya que es aquí donde se especifican las siguientes funciones:

i) la identificación del sistema que se empleará para la -- compilación y ejecución del programa.

ii) las relaciones que existen entre los archivos empleados y las unidades de entrada y salida (INPUT/OUTPUT).

Por lo que esta división cuenta con dos secciones que especifican las funciones anteriores.

. CONFIGURATION SECTION.

Contiene la información referente al sistema empleado para la compilación del programa, la información concerniente - al compilador utilizado y la información del sistema en el que se será ejecutado el programa.

Las declaraciones de esta sección son:

```
SOURCE-COMPUTER      nombre-equipo.
OBJECT-COMPUTER      nombre-equipo.
```

generalmente para todos los sistemas son iguales, salvo -- excepción de un sistema específico.

Forma de codificación.

Columna 8 12

```
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER.  nombre-equipo.
OBJECT-COMPUTER.  nombre-equipo.
```

```
. { INPUT-OUTPUT SECTION. }
  { I-O          SECTION. }
```

La función de esta sección es el establecer relaciones necesarias entre los equipos periféricos y los archivos que intervendrán en la ejecución del programa. Consta de un párrafo de control de archivos (FILE-CONTROL) y una o más declaraciones de archivos, según sean utilizados en el programa.

FORMATO.

```
COLUMNA 8
        { INPUT-OUTPUT SECTION. }
        { I-O          SECTION. }

          COLUMNA 12
          FILE-CONTROL.
              SELECT nombre-archivo ASSIGN TO dispositivo.
              SELECT nombre-archivo ASSIGN TO dispositivo.
```

Cuando la declaración SELECT es empleada, el "nombre-archivo" corresponde a la declaración del archivo, el cual se encontrará o empleará el equipo periférico declarado en el "dispositivo". En algunos sistemas se emplean los nombres de los dispositivos como PRINTER, TAPE, etc., pero en otros se ocupa una clave o file-code (FC) que tendrá relación con archivos permanentes o archivos de paso, o también existen sistemas en los que se emplean ciertas claves y relaciones.

Forma de codificación.

Columna 8 12

```
ENVIRONMENT   SECTION.
CONFIGURATION SECTION.
SOURCE-COMPUTER.      nombre-equipo.
OBJECT-COMPUTER.      nombre-equipo.
INPUT-OUTPUT  SECTION.
     FILE CONTROL
     SELECT nombre-archivo ASSIGN TO dispositivo.
```

De acuerdo al tipo de acceso del archivo, este se podrá declarar de acuerdo al siguiente formato.

FORMATO.

COLUMNA 12

```
SELECT nombre-archivo ASSIGN TO dispositivo
[;ACCESS MODE IS { SEQUENTIAL }]
                 { RAMDOM     }
[;ACTUAL KEY IS nombre- llave.]
```

Cuando se emplea esta declaración se está indicando el tipo de acceso para el archivo mencionado, el cual puede ser secuencial o RAMDOM (aleatorio, al azar, se puede elegir cualquier registro), en el cual se necesita una clave o llave que permita accesar exactamente el registro deseado. El "nombre-llave" debe ser un campo del registro a accesar.

INSTRUCCIONES COBOL

FIGURA 3

```
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. B6700.
OBJECT-COMPUTER. B6700.
INPUT-OUTPUT SECTION.
SELECT ARCHIVO-1 ASSIGN TO READER.
SELECT ARCHIVO-2 ASSIGN TO PRINTER.
*
*  ESTE EJEMPLO MUESTRA COMO SE MODIFICA
*  LA DIVISION DEL MEDIO AMBIENTE, AUNQUE
*  EN EL CASO DEL SELECT SE PODRIA CODIFICAR
*  COMO SE MUESTRA A CONTINUACION:
*
SELECT ARCHIVO-1 ASSIGN TO DISK
       ACCESS MODE IS RANDOM
       ACTUAL KEY IS CLAVE.

*  SE MUESTRA LA ASSIGNACION DE UN ARCHIVO LLAMADO
*  ARCHIVO-1 A UNA UNIDAD DE DISCO, CUYO ACCESO
*  ES DE TIPO ALEATORIO Y SU CLAVE DE ACCESO ES
*  EL CAMPO IDENTIFICADO POR "CLAVE".
```

4. DIVISION DE DATOS. (DATA DIVISION.).

El objetivo de esta tercera división de un programa COBOL, es el de proveer en forma detallada los registros de los archivos requeridos en la ejecución del programa, ya sean de entrada, salida o de nueva creación.

Es quizás la más laboriosa de las divisiones ya que requiere, de detallar todos los campos del registro, asimismo, se pueden describir áreas internas, para constantes que van a ser utilizadas, o información que pasa de un programa a otro, también describe los contenidos y formatos de todos los reportes que pueden ser generados por el proceso.

La División de Datos (DATA DIVISION).

Contiene varias secciones, por ejemplo:

La FILE SECTION, en la cual se describen los contenidos de datos externos (archivos de entrada, salida y archivos internos como el SORT).

La DATA-BASE SECTION, esta sección es ocupada cuando el programa en desarrollo tendrá como objeto accesar una base de datos.

La WORKING-STORAGE SECTION, es la sección donde se describen áreas internas de trabajo, y los controladores empleados.

La CONSTANT SECTION, esta es la sección donde se declaran las

constantes que serán manejadas, las cuales no podrán cambiar de valor durante la ejecución del programa.

La LINKAGE SECTION, esta sección contiene información que puede ser transferida de un programa a otro.

La COMMUNICATION SECTION, contiene información acerca de procesos de interfases entre terminales de teleproceso.

La REPORT SECTION, esta sección contiene la descripción de todos los reportes que serán generados en el transcurso del programa.

Las secciones de FILE SECTION (Sección de Archivos), WORKING-STORAGE SECTION (Sección de Areas de Trabajo) son las más utilizadas.

4.1. CONCEPTOS DE DATOS E INFORMACION

Si bien es cierto que cuando se trata de analizar una estadística en la que por cualquier lado, lo que se observa son números, también es cierto que el estadístico que se encarga de observar o cuantificar los fenómenos tratados en ella, partió de una cierta necesidad, identificando el contexto en el cual se desarrolla, determinando los insumos requeridos para su proceso, y de esta forma brindar resultados, esto es insumos para el proceso de análisis de otros.

Esto debe provocar una cierta inquietud, por definir de antemano el contexto y entonces poder definir lo que es dato y lo que es información.

En el caso mencionado, se ha tratado de deducir que el proceso que mencionamos es relativo, ya que el resultado de un proceso, es la entrada de otro.

Definición.- Un dato; es un factor recolectado de la observación o medida de un fenómeno.

Definición.- Información; es la interpretación adecuada y correlacionada de uno o más datos, que permite tomar una decisión.

En otras palabras, Dato "se refiere a la masa de hechos y/o cifras en bruto que se acopla a un proceso" e Información "es el estracto que se ha obtenido de esa masa y procesado para una persona dada".(1)

4.2. CONCEPTOS DE ALMACENAMIENTOS DE DATOS.

Cuando se necesita hablar de archivos y manejo de los mismos se deben tomar en cuenta dos aspectos, el aspecto físico del archivo y las características conceptuales del contenido de los registros de ese archivo.

Aspectos Físicos de un archivo.

Se debe tomar en cuenta si el archivo es manejado como de entrada (INPUT) o de salida (OUTPUT) o si el archivo toma ambas

actividades, es decir, funciona como archivo de entrada y de salida, además:

a) el modo de acceso.
   Secuencial, ramdom, index secuencial, etc.
b) la organización de información, las limitantes físicas, como: tamaño de registros, capacidad de archivo, optimización, etc.
c) los medios en los que se puede manejar el archivo (cinta, disco, tarjetas, pantallas, etc.) y el nombre de -- identificación.

Aspectos Conceptuales de un archivo.

Los aspectos conceptuales de un archivo se refieren primordialmente a cada entidad lógica del archivo propiamente dicho.

Es importante distinguir entre registro físico y registro lógico.

Para Cobol un registro físico es una unidad de información - la cual está de acuerdo en tamaño y forma de ordenamiento, - conveniendo el sistema de cómputo para el almacenamiento de datos, así también los elementos de entrada y salida.

La magnitud de un registro físico está en relación directa - con el hardware y no con la magnitud de la información contenida en un dispositivo de entrada o salida.

Uno o varios registro lógicos pueden estar contenidos en una unidad física.

El concepto de registro lógico no está restringido a archivos, puede ser aplicado a todas las secciones de la DATA DIVISION.

CONCEPTOS DE REGISTROS.

La descripción de registros está referida a un conjunto de descripciones de datos, donde se describen las características de un registro en particular.

Cada descripción de datos corresponde a un número de nivel indicado por un nombre de dato (identificador formado por una serie de cláusulas independientes si es requerido).

CONCEPTOS DE NUMEROS DE NIVEL.

El concepto de jerarquía es inherente en la estructura de un registro lógico.

Este concepto permite formar estructuras acordes a las necesidades de manejo de información, con esto se pueden crear subdivisiones y más adelante subdivisiones de las mismas, poniendo al programador en la posición de manejar la información tan detallada como lo desee, en otras palabras, el número de nivel define la interrelación de datos comprendidos en el registro.

En Cobol la jerarquía del contenido es especificada por el

uso de varios números de nivel que pueden variar en un rango de 1 al 49 (los números 77 y 88 son discutidos más adelante).

(1) PROGRAMACION LOGICA, TOMO I.
WARNIER/FLANAGAN.

```
IDENTIFICATION          DIVISION.
PROGRAM-ID.             EJEMPLO.
AUTHOR.                 CARL.
*ESTE PROGRAMA ACTUALIZA LA TABLA DE AUTORIZADOS
*UTILIZA LA INFORMACION PROPORCIONADA POR LA EMISORA.
*FECHA DE LA ULTIMA MODIFICACION.       8-OCT-1981.
*---------------------------------------------------
ENVIRONMENT             DIVISION.
*---------------------------------------------------
CONFIGURATION           SECTION.
SOURCE-COMPUTER.        HIS-SERIES-60 LEVEL-66-ASCII.
OBJECT-COMPUTER.        HIS-SERIES-60 LEVEL-66-ASCII.
INPUT-OUTPUT            SECTION.
FILE-CONTROL.
    SELECT VERO-ANIS ASSIGN TO DA
           ACCESS MODE IS SEQUENTIAL
           ORGANIZATION FREC SEQUENTIAL WITH SSF.
    SELECT ARCH-LIST ASSIGN TO DA
           ORGANIZATION FREC SEQUENTIAL WITH SCF.
    SELECT ARCH-SORT ASSIGN TO SD.
DATA                    DIVISION.
SUB-SCHEMA              SECTION.
DB ISSP WITHIN          SCHEMAS ISA.
FILE                    SECTION.
FD                      VERO-ANIS
        LABEL RECORD IS STANDARD
        DATA RECORD IS VERO-AUTO
        CODE-SET IS EBCDIC.
01                      VERO-AUTO.
  05                    VERO-AUTO-COND   PIC X(102).
  05                    VERO-AUTO-CLAU   PIC X.
  05                    VERO-AUTO-ESTA   PIC X.
  05                    VERO-AUTO-RELL   PIC X(80).
```

```
01  REGISTRO.
    05  REGI-NOMB-COMP.
        10  REGI-NOMB-APPA      PIC X(15).
        10  REGI-NOMB-APMA      PIC X(15).
        10  REGI-NOMB-NOPD      PIC X(15).
    05  REGI-OCUP-ACTO.
        10  REGI-OCUP-SUEL      PIC 9(05).
        10  REGI-OCUP-DESC      PIC X(30).
    05  REGI-TELE-FONO.
        10  REGI-TELE-TIPO      PIC 9(01).
        10  REGI-TELE-NUME.
            15  REGI-TELE-LADA  PIC 9(04).
            15  REGI-TELE-PROP  PIC 9(07).
    05  REGI-DATO-PERS.
        10  REGI-DATO-EDAD.
            15  REGI-DATO-ANIO  PIC 9(02).
            15  REGI-DATO-MES   PIC 9(02).
        10  REGI-DATO-CIUD.
            15  REGI-DATO-COLO  PIC X(30).
            15  REGI-DATO-ZOPO  PIC 9(02).
```

4.3. AREA DE ALMACENAMIENTO INTERNO PARA TRABAJO. (WORKING-STORAGE SECTION).

Dentro de un programa Cobol, es posible describir todos los formatos de los archivos empleados, ya sean de lectura, escritura, consulta, etc. Sin embargo, es necesario describir todas aquellas áreas que serán utilizadas para definir títulos, contadores, áreas de trabajo con datos, etc., ya que dichos campos no pertenecen a algún archivo es necesario que el sistema reserve áreas de memoria para tales efectos.

La WORKING-STORAGE SECTION, es la sección de la DATA DIVISION ya que se encarga de separar dichas áreas de memoria, en las que se pueden definir descripciones de registros compuestos (como: apuntadores, contadores, fechas, etc.).

Principalmente será factible reservar el área necesaria para títulos, por lo que se deberá emplear la cláusula "VALUE" que consiste en asignar el valor correspondiente a una literal.

Ejemplo.

Columna 12

```
05  TITULO-1.
 10 FILLER       PIC X (30)    VALUE SPACES.
 10 FILLER       PIC X (32)    VALUE
    "UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO"
 10 FILLER       PIC X (40)    VALUE SPACES

03  CONTADORES.
 10 SWITCH-1     PIC X         VALUE "V"
 10 FECHA        PIC 9 (6)     VALUE 101082.
```

4.4. CODIFICACION DE LA DIVISION DE DATOS. (DATA DIVISION)

Solo se describirán las secciones de "Descripción de Archivos" y la de "Almacenamiento de Información" o "Areas de -- Trabajo".

4.4.1. Sección de Archivos.

Cada archivo que fué asignado en la environment deberá ser descrito en la sección de archivos de la DATA DIVISION, contemplando los distintos niveles de acuerdo al formato del archivo.

Forma de Codificación.

OPCION 1.

Columna 8 12 25

```
DATA            DIVISION.
FILE            SECTION.
FD  nombre-archivo
    [;LABEL RECORD {IS  } {STANDARD          }.]
                   {ARE } {OMITTED           }
                          {nombre-programador}
    [;DATA RECORD {IS  }  nombre-registro-1, nom-]
                  {ARE }    bre-registro 2. . .
```

Esta opción es comúnmente empleada para describir los archivos secuenciales o los archivos de registros unitarios como las tarjetas de 80 columnas.

FD, significa File Descripción, descripción de archivo.

nombre-archivo, es el nombre que corresponde al archivo mencionado en la sección INPUT-OUTPUT SECTION.

$$;\underline{\text{LABEL}}\ \underline{\text{RECORD}}\left\{\begin{matrix}\underline{\text{IS}}\\ \underline{\text{ARE}}\end{matrix}\right\}\ \left\{\begin{matrix}\underline{\text{STANDARD}}\\ \underline{\text{OMITTED}}\\ \text{nombre-programador}\end{matrix}\right\}.$$

Cuando es empleada esta declaración, se está asignadndo una llave de acceso al archivo, que para archivos secuenciales es nula (OMITTED) o bien puede ser una llave estandard para cualquier archivo (como los archivos de impresora). Los archivos de acceso aleatorio necesitarán del "nombre-programador" que será la llave de acceso al mismo.

$$;\underline{\text{DATA}}\ \underline{\text{RECORD}}\ \left\{\begin{matrix}\underline{\text{IS}}\\ \underline{\text{ARE}}\end{matrix}\right\}\ \text{nombre-registro-1, nombre-registro-2...}$$

Cuando es empleada esta declaración, se notifica respecto - al nombre del o los registros, según se emplee la opción IS o ARE, donde "nombre-registro-1" corresponde al primer re-- gistro, "nombre-registro-2" corresponde al segundo registro, etc.

OPCION 2.

```
Columna 8   12        25

        DATA          DIVISION
        FILE          SECTION
        FD  nombre-archivo
```

```
[;BLOCK CONTAINS [entero-1 TO] entero-2
                 {CHARACTERS}
                 {RECORDS   } ]
                 {WORDS     }
[;RECORD CONTAINS entero-3 TO  entero-4
                 {CHARACTERS}
                 {WORDS     } ]

[;LABEL RECORD{IS }{STANDARD                    }]
              {ARE}{OMITTED                     }
                   {nombre-programador          }

[;LINAGE IS    {entero-5   } LINES]
              {nombre-prog.}

[;DATA RECORD{IS } nombre-reg 1, nombre regis
             {ARE}                tro 2....  ]

[;CODE-SET      IS identificador.]
```

Esta opción es empleada para describir en forma más completa el archivo asignado en la environment division, el cual puede ser un secuencial, aleatorio, etc.

```
;BLOCK CONTAINS [entero-1 TO] entero-2 [CHARACTERS]
                                       [RECORDS   ]
                                       [WORDS     ]
```

Cuando es utilizada esta declarativa, se notifica al sistema que el block de dispositivo en el que se encuentra almacenado el archivo, tiene una longitud fija o variable donde el entero-2- (o el entero-1) asignan la longitud del mismo en caracteres, registros o palabras, de preferencia al emplear palabras se deberá procurar sea siempre posible de cuadrar a caracteres.

;RECORD CONSTAINS [entero-3 TO] entero-4 { CHARACTERS / WORDS }

Esta declaración es empleada para indicar las dimensiones - del registro dentro de un archivo.

;LINAGE IS { entero-5 / nombre-prog } LINES

En el caso de describir un archivo impreso y se desee llevar un control automático de las líneas impresas, se podrá emplear esta declarativa, la cual indica el número de líneas que deberá contener el reporte por hoja. Esto se podrá indicar señalando un número fijo o el valor que sea asignado a un campo variable.

;CODE-SET

De la misma forma que el linage es empleado para los archivos impresos en papel, esta declaración cuando es utilizada indica al sistema el tipo de código empleado para la impresión del reporte.

OPCION 3.

```
Columna 8   12      25
        DATA        DIVISION
        FILE        SECTION
        FD  nombre-archivo
```

Esta opción es empleada cuando se requiere realizar una clasificación de los datos, dicha clasificación puede ser numé

rica, de mayor a menor o viceversa, para lo cual se requiere de un archivo auxiliar llamado archivo "SORT" o clasificación.

Ver apéndice en el apartado relacionado con el verbo SORT.

5. DIVISION DE PROCEDIMIENTOS.

5.1. PROGRAMACION ESTUCTURADA.

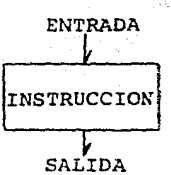
La programación estructurada, es un conjunto de técnicas de codificación, que se orientan a facilitar y optimizar las tareas de: codificación propiamente dichas, rastreo de errores, y modificación o mantenimiento de programas. Se aplican fundamentalmente a la descripción de archivos y áreas de trabajo en la DATA DIVISION, y a las diversas formas de codificar una sección, un párrafo o instrucción en la PROCEDURE DIVISION, ya que principalmente se pretende que exista un solo punto de referencia de entrada y uno solo para la salida de cualquier proceso, sin importar el nivel del mismo, como también, la ejecución de éste se ve controlada de antemano, eliminando con esto las condiciones de confusión.

Las tres estructuras lógicas que se emplean en la programación estructurada son:

1.- Ejecución secuencial de instrucciones
2.- IF-THEN-ELSE
3.- PERFORM-UNTIL.

Primera Construcción.

La primera construcción se ilustra de la siguiente forma:



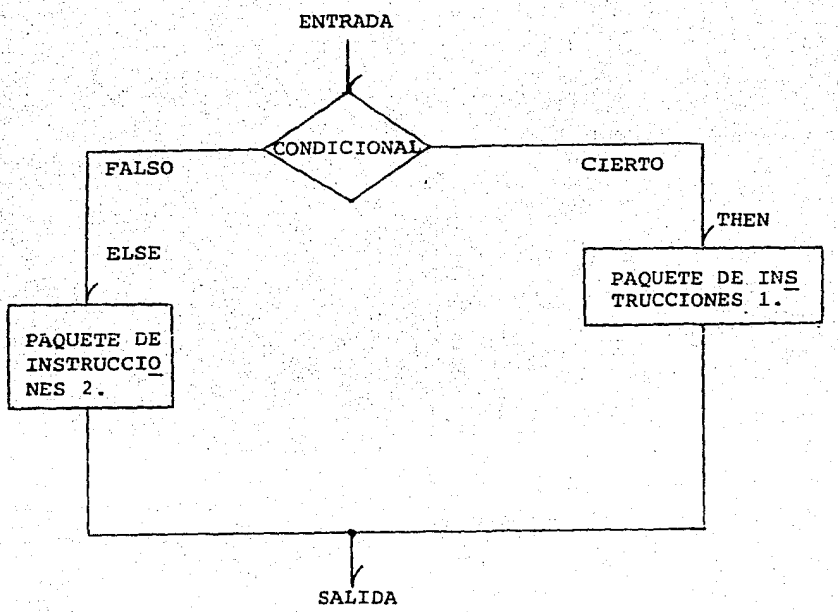
Ejemplos de este tipo de construcción son las instrucciones: MOVE, EXAMINE, DISPLAY, etc., estas instrucciones son simplemente ejecutadas según su disposición en el programa, la siguiente figura, muestra tal situación:

```
ADD HORAS-LABORADAS   TO TOTAL-HORAS.
ADD COSTO-HORAS       TO TOTAL-COSTO.
MOVE TOTAL-HORAS      TO IMP-TOTAL-HORAS.
MOVE TOTAL-COSTO      TO IMP-TOTAL-COSTO.
```

Donde cada instrucción es ejecutada por secuencia lógica.

Segunda Construcción.

La construcción IF-THEN-ELSE se ilustra de la siguiente manera:



Como se puede observar, la condicional IF, presenta dos al-- ternativas, la certeza de la condición o la falsedad, cual-- quiera que sea la respuesta, siempre existirán la o las ins- trucciones que corresponden a tal acción. El paquete 1, es un conjunto que por lo menos tiene una instrucción, mientras que el paquete 2, puede no contener instrucciones. Las si-- guientes figuras muestran tales situaciones.

```
IF HORAS-LABORADAS GREATER THAN HORAS-CONTRATADAS
THEN SUBTRACT HORAS-LABORADAS FROM HORAS-CONTRATADAS
                GIVING    HORAS-EXTRAS
PAQUETE 1       MULTIPLY HORAS-EXTRAS BY 150 GIVING
                          TOTAL-HORAS-EXTRAS
                ADD TOTAL-HORAS-EXTRAS TO SUELDO-BRUTO
     ELSE
PAQUETE 2       MOVE HORAS-LABORADAS TO SUELDO-BRUTO.
```

FIGURA # 1

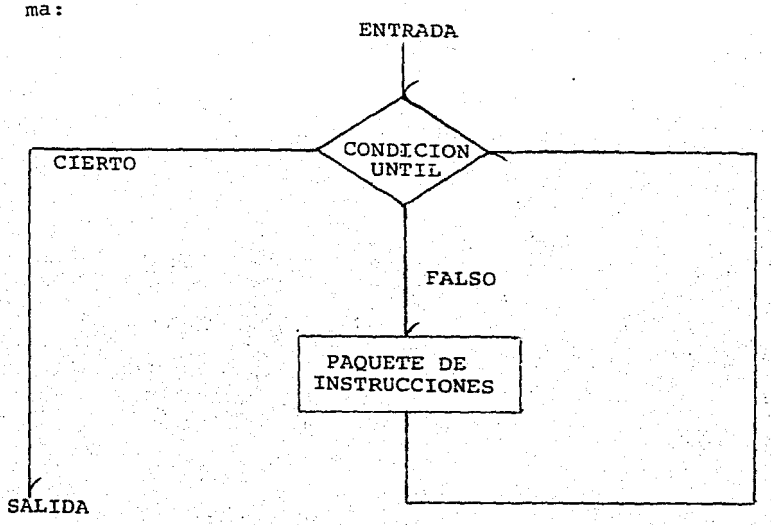
```
IF EDAD LESS THAN ANIO-CALCULADO
THEN MOVE EDAD TO NUEVO-CALCULO.
PAQUETE 1
```

FIGURA # 2

Como se observa no siempre el paquete 2 debe contener ins -- trucciones.

Tercera Construcción.

La construcción PERFORM-UNTIL se ilustra de la siguiente forma:

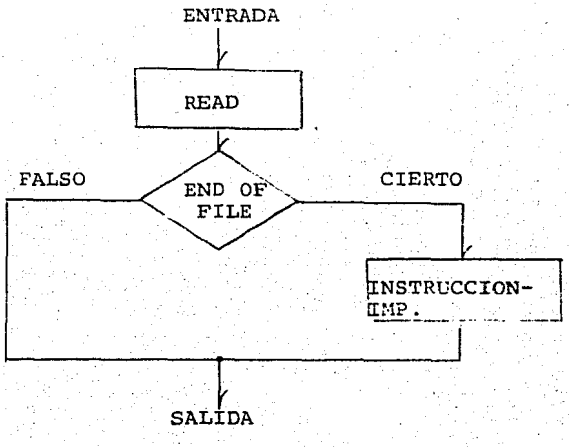


En esta construcción el paquete de instrucciones será ejecutada hasta que la condición especificada en el UNTIL del comandc perform no se cumpla.

Lo fundamental de estas estructuras es el lograr combinarlas de tal forma que no se rompa el esquema de cualquiera de --- ellas; tal es el caso de las instrucciones WRITE, READ, RE-- TURN, etc. en la que existe en forma implícita el uso de dos estructuras; por ejemplo:

```
        WRITE LINEA-DETALLE EOP PERFORM TITULOS
o
        READ ARCHIVO-GENERAL AT END MOVE 1 TO
                                    SWITH-FINAL
```

Que bien podría ilustrarse de la siguiente forma:



En la mayor parte de los programas se procura seguir la siguiente secuencia, dependiendo de la finalidad para la cual fue elaborado.

1. INICIALIZAR AREAS
2. ABRIR ARCHIVOS
3. LECTURA INICIAL DE LOS ARCHIVOS DE ENTRADA
4. PERFORM-UNTIL DEL PROCESO HASTA TERMINAR REGISTROS DE ENTRADA
5. RUTINA FINAL; CONTROLES, ETC.
6. CERRAR ARCHIVOS.

La siguiente codificación muestra tal situación:

```
S
000-RUTINA-PRINCIPAL
    OPEN INPUT ARCHIVO-ENTRADA, OUTPUT ARCHIVO-SALIDA.
```

```
    MOVE ZEROS TO VARIABLES-NUMERICAS.
    MOVE SPACES TO VARIABLES-ALFANUMERICAS.
    PERFORM 020-LEE-ARCHIVO
    MOVE DATO-NUEVO TO DATO-ANTERIOR.
    PERFORM 010-PROCESO
    CLOSE ARCHIVO-ENTRADA, ARCHIVO-SALIDA
    STOP RUN.
010-PROCESO.
    ADD DATO-1 TO DATO-TOTAL-S
    PERFORM 020-LEE-ARCHIVO
    IF DATO-NUEVO EOT EQUAL TO DATO-ANTERIOR
    THEN PERFORM 030-IMPRIME
         MOVE DATO-NUEVO TO DATO-ANTERIOR
         MOVE ZEROS TO DATO-TOTAL-S.
020-LEE-ARCHIVO.
    READ ARCHIVO-ENTRADA INTO AREA-WORKING;
    AT END, MOVE HIGH-VALUES TO DATO-NUEVO.
030-IMPRIME.
    MOVE DATO-ANTERIOR TO DATO-IMPRESION.
    MOVE DATO-TOTAL-S TO TOTAL-SALIDA.
    WRITE REG-IMPRESO AFTER ADVANCING 1 LINES.
    MOVE SPACES TO REG-IMPRESO.
```

CONSIDERACIONES RESPECTO A LA DATA DIVISION.

Tomando en cuenta que en la Data Division, se especifican todos aquellos archivos y áreas de trabajo (WORKING-STORAGE),- que son empleados en el proceso a desarrollar, se dan a continuación algunas consideraciones respecto a esta División.

1o. Usar la columna 7 para dar una descripción del objetivo de cada área empleada.

2o. Emplear números de nivel de 5 en 5 para que de esta -- forma se puedan modificar o insertar datos agrupados.

Ejemplo:

```
01      REGISTRO-ALUMNO
  05    LLAVE-DEL-REGISTRO
    10  DEPARTAMENTO     PIC XX.
    10  SECCION          PIC XX.
    10  NUM-EMPLEADO     PIC 9(8).
  05    HORAS-LABORADAS  PIC 9(3).
  05    SALARIO          PIC 9(6) V9(2).
```

3o. Codificar cada uno de los niveles empleados de la si-- guiente forma:NIVEL 05 COLUMNA 12, NIVEL 10 COLUMNA 14 NIVEL 15 COLUMNA 16, ETC.

4o. Codificar el "PIC" (PICTURE) en la columna 40.

5o. Codificar mneumónicos o "nombres-programador" que indiquen claramente el contenido o uso del campo descrito. Fundamentalmente para el caso de contadores, switches,

etc.

Ejemplo:

```
    05   NOMBRE COMPLETO          05   NOMB-COMP.
     10  APELLIDO PATERNO          10  NOMB-APPA.
     10  APELLIDO MATERNO          10  NOMB-APMA.
     10  NOMBRE PROPIO             10  NOMB-PROP.
```

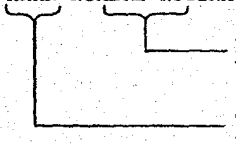
De esta forma, al querer hacer referencia a cualquier contenido en el nombre completo, bastará indicar que todos los campos pertenecientes al nombre comienzan con las 4 letras "NOMB".

CONSIDERACIONES RESPECTO A LA PROCEDURE DIVISION.

La División de Procedimientos es la que refleja propiamente el uso de las técnicas de la programación estructurada, donde se realizan todas las estructuras necesarias. Para lo -- cual se recomienda:

1o. Todos los nombre de secciones o párrafos deberán ir -- enumerados y con un nombre que sea significativo, de - acuerdo al contenido del mismo.

Ejemplo: XXXX-NOMBRE-RUTINA



2o. Todos los comandos deben codificarse en la zona B.

3o. El comando IF-THEN-ELSE, se debe codificar como se --- muestra a continuación:

```
IF CONDICION-1
THEN   paquete de instrucciones-1
ELSE
       paquete de instrucciones-2.
```

o bien:

```
IF condición-1
     paquete de instrucciones-1
ELSE
     paquete de instrucciones-2
```

4o. Codificar el comando PERFORM-UNTIL de la siguiente for

ma:

```
PERFORM   nombre-rutina-1
  UNTIL   condición-1
     OR   condición-2.
```

5o. Usar el asterisco en la columna 7 para incluir notas - en cualquier rutina, con el propósito de indicar como se efectuó dicha rutina.

CONSIDERACIONES RESPECTO A UNA PROGRAMACION EFICIENTE.

Las técnicas de reducción del tiempo de ejecución de un programa es uno de los aspectos más importantes en cualquier empresa, dado que repercute en el mejor empleo de los recursos disponibles. Para lo que se presentan las siguientes consideraciones:

1o El camino más eficiente para referirse a un campo definido con la cláusula "OCCURS", es por una literal numérica subscrita, porque el compilador (resuelve) busca - la dirección exacta del campo a compilar.

2o El uso de literales numéricas subscritas es impráctico, porque la generación de código máquina es demasiado, -- por lo que es preferible ocupar la opción "INDEXED BY".

3o Cuando se presente el caso de manejar tablas con más de un nivel de ocurrencias, es necesario efectuar una se-- rie de operaciones con tales elementos, es recomendable manejar operaciones por nivel jerárquico dentro de la - tabla, es decir, eliminar las operaciones a nivel de -- elementos que cuenten con 2 ó más indices, por ejemplo:

```
 10   VENTAS-DEPTO    OCCURS  50  TIMES
  15  PRODUCTO        PIC 9(9)  OCCURS  25  TIMES.
...
...
05     ALMACENAMIENTO-TEMPORAL
 10    ALMA-VENTAS-DEPTO.
  15   ALMA-PRODUCTO  PIC 9(9)  OCCURS  25  TIMES.
```

```
PROCESO;
 MOVE  VENTAS-DEPTO (S1)  TO ALMA-VENTAS-DEPTO.
 COMPUTE VENTA-TOTAL = ALMA-PRODUCTO(1) + ALMA-PRODUCTO(2)
                     + ALMA-PRODUCTO(5) + ALMA-PRODUCTO(10)
                     + ALMA-PRODUCTO(25)
 ADD  1  TO  S1.
```

Cuando S1 está inicializado con el valor de 1.
o bien, con la misma estructura de datos.

```
PROCESO;
        PERFORM  020-CALCULOS
          VARING  S1  FROM 1 BY 1
           UNTIL  S1  GREATER THAN 50.
020-CALCULOS
    MOVE  VENTAS-DEPTO(S1)  TO  ALMA-VENTAS-DEPTO.
    COMPUTE VENTAS-TOTAL(S1) = ALMA-PRODUCTO(1)+ ...
                               ... + ALMA-PRODUCTO(25).
```

Las dos estructuras anteriores son preferibles a:

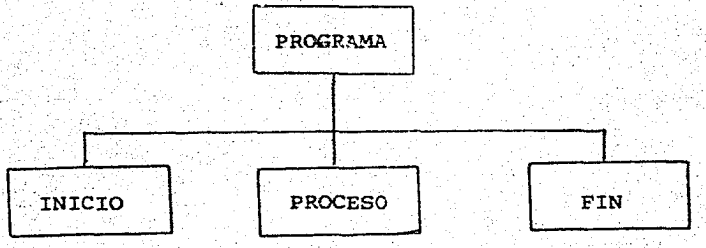
```
PROCESO:
    COMPUTE  VENTAS-TOTALES =  PRODUCTO (S1, 1) +
                               PRODUCTO (S1, 2) +...
                            ...+ PRODUCTO (S1, 25).
```

Donde se variará el valor de S1, de 1 hasta 50.

5.2. VTOC (VISUAL TABLE OF CONTENS).

Una de las técnicas empleadas actualmente para la estructuración y jerarquización de rutinas o funciones de un programa, es el uso del VTOC (TABLA VISUAL DE CONTENIDOS), que permite identificar las funciones en las que se puede dividir un proceso, de mayor a menor importancia. De tal forma que las -- funciones de más alto nivel son aquellas que reflejan la estructura general del proceso, y las rutinas del más bajo nivel, son aquellas que se reflejan en instrucciones mismas, - ya que son funciones que no se pueden dividir.

El siguiente diagrama muestra una de las situaciones más comunes:



En esta ilustración puede observarse que se respetan las indicaciones dadas por la programación estructurada, ya que:

LA RUTINA:		CONTIENE
	INICIO	INICIACION DE RUTINA
		INICIALIZACION DE AREAS DE WORKING
		INICIALIZACION DE CONTADORES Y SWITCHS
		APERTURA DE ARCHIVOS

LECTURA DEL PRIMER REGISTRO.

PROCESO RUTINAS O FUNCIONES NECESARIAS

FIN IMPRESIONES FINALES, TOTALES, ETC.
CERRAR ARCHIVOS.

Existen dos maneras de enumerar las funciones en VTOC, las - cuales obedecen a:

1.- La numeración es de arriba hacia abajo y de izquierda a derecha.

2.- Cuando una rutina se emplea en distintas partes del -- diagrama, entonces;

2.1.- Siempre aparece con el número que se le asigna - en su primer ocurrencia.

2.2.- A excepción de la primera ocurrencia, todos los rectángulos empleados para esta rutina deben --- traer dobles barras a los lados.

NUMERACION POR NIVELES.

Este tipo de numeración obedece a que las rutinas se enume-- ran, de acuerdo al nivel que pertenecen.

1.- Se emplea una cadena de 4 posiciones.

2.- Se comienza con el nivel superior, enumerando con un - 0000.

3.- El segundo nivel será el nivel de las decenas, es de-- cir; 0010, 0020, ... etc.

4.- El tercero será: 0100, 0110, 0120 ...etc., o bien; 0011

0012 ... etc., 0021, 0022 ...etc., según las divisiones del proceso.

5.- Se sigue el criterio del punto 4, para los demás niveles.

NUMERACION DE SECCIONES.

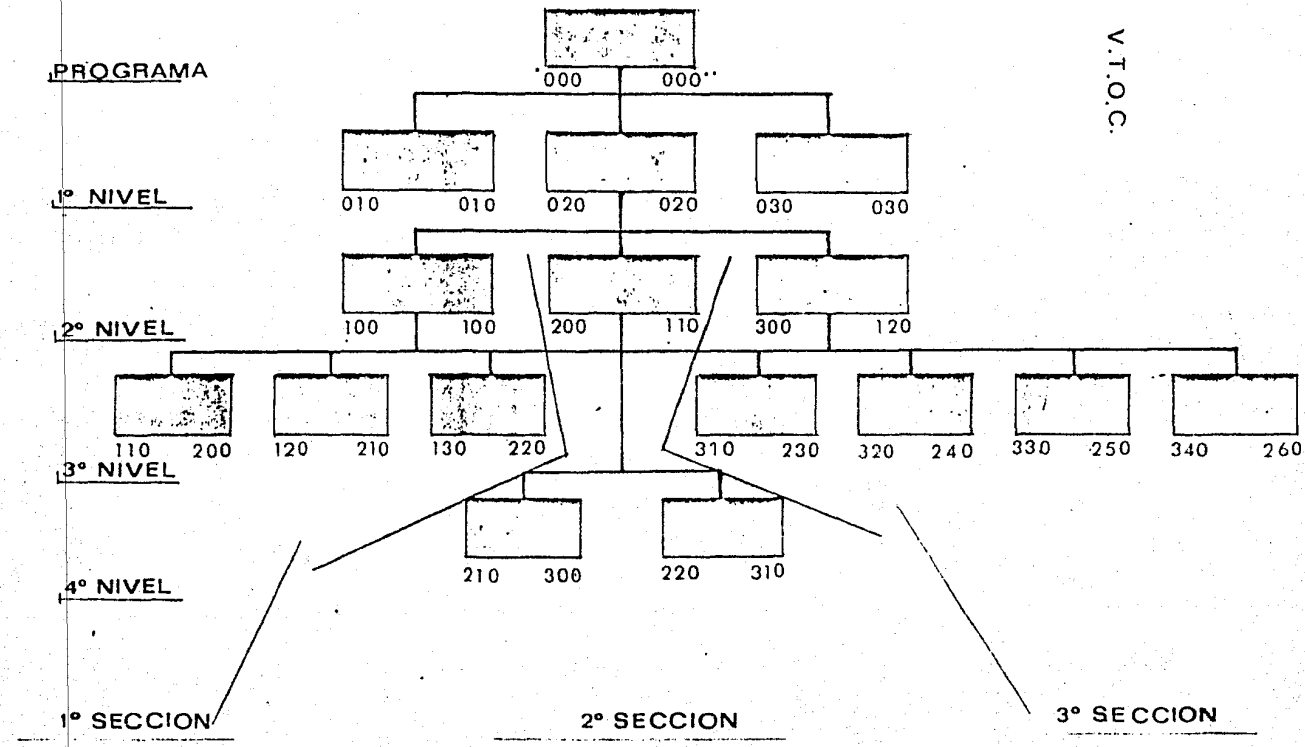
1o. Se ocupa una cadena de 4 caracteres.

2o. La primera rutina se enumera con el número 0000.

3o. Todas las rutinas que descienden de la rutina de proceso se enumerarán con los números 0100, 0200, 0300, etc, o bien, 1000, 2000, etc., según la magnitud del proceso.

4o. Todas las subrutinas que descienden de una rutina del punto anterior, serán numeradas de acuerdo al criterio de numeración por niveles, sólo que siempre comenzarán con el número de sección correspondiente, esto es: 0100, 0200, ... etc. ó 1000, 2000, ... etc.

El siguiente diagrama muestra los dos tipos de numeración. Los números a la izquierda son la numeración de secciones y los números a la derecha son la numeración por niveles.



· Numeración por secciones.
·· Numeración por niveles.

5.3. HIPO (HIERARCHY PLUS INPUT-PROCESS-OUTPUT).

Uno de los tópicos de mayor importancia hoy en día en las - cuestiones de procesamiento electrónico de datos, tanto en sus funciones de desarrollo, implantación y mantenimiento - es la "Documentación de Procesos". En relación a este punto, existen otras técnicas de documentación, distintas a la aquí presentada, tales como, desarrollos narrados, diagra-- gramas y diagramas de flujo.

HIPO, es una técnica de documentación de procesos, que proporciona las siguientes ventajas:

a.- Una descripción por estructuras del proceso, fácilmente entendibles.

b.- Presentar en forma sencilla, estructuras complejas.

c.- Dar una descripción tan detallada como se desee.

d.- Proporcionar una descripción de las entradas y - salidas involucradas en el proceso.

El empleo del HIPO se encuentra en relación directa con el uso del VTOC, ya que, mientras el VTOC indica la estructura general del proceso, por medio de funciones, el HIPO se encarga de describir cada una de estas funciones, sin importar el nivel jerárquico en el que se encuentre.

Los elementos con que se desarrolla un HIPO son:

1.- Hoja de trabajo HIPO (figura 1).

2.- Símbolos de trabajo HIPO (figura 2).

El segundo contiene todos los símbolos empleados y permitidos en HIPO, mientras que la hoja "HIPO", permite llevar un control de cada una de las funciones, mencionadas en el VTOC del proceso.

DESCRIPCION DE LA HOJA "HIPO"

Para la descripción de ésta, se debe consultar la figura 1, en la que cada elemento al que se hace referencia se encuentra enumerado.

1.- SISTEMA. Nombre del sistema al cual pertenece el proceso que se está desarrollando.

2.- SUBSISTEMA. Nombre del subsistema al cual pertenece el proceso que se está desarrollando.

3.- NIVEL. Referencia a la función que se está desarrollando.

4.- IDENTIFICACION. Nombre de la función, según punto anterior.

5.- DEPARTAMENTO. Nombre o identificación del Departamento encargado de desarrollar el análisis.

6.- RESPONSABLE. Nombre de la persona que desarrolló el --- HIPO.

7.- NOTA. Aclaraciones respecto al proceso en cuestión.

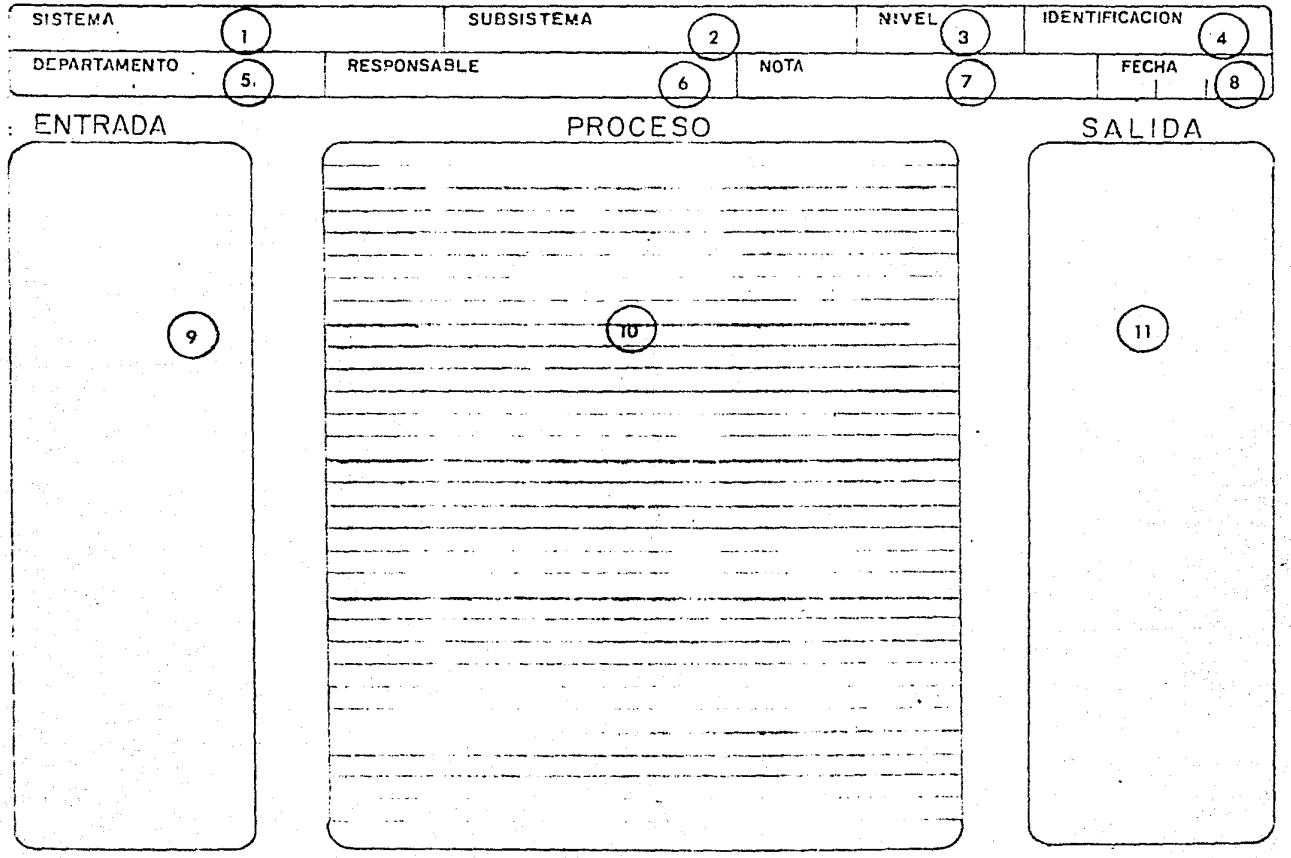
8.- FECHA. Fecha de desarrollo.

9.- ENTRADA. Descripción de los archivos de entrada al proceso.

10.- PROCESO. Desgloce detallado de cada proceso/función.

11.- SALIDA. Descripción de los archivos de salida del proceso.

FIGURA 1


SISTEMA
1
SUBSISTEMA
2
NIVEL
3
IDENTIFICACION
4
DEPARTAMENTO
5
RESPONSABLE
6
NOTA
7
FECHA
8
ENTRADA
PROCESO
SALIDA
9
10
11

DESCRIPCION DE LA PLANTILLA "HIPO".

La plantilla "HIPO" contiene todos los símbolos permitidos en HIPO, y estos son:

- FLECHAS DE CONTROL	Indican la transferencia o procedencia del control del proceso. Indican la generación o intervención de determinados archivos.
- DATOS	Indican contenidos de información.
- DOCUMENTO	Indica un archivo en papel generado de un proceso.
- DISPLAY	Indica la intervención de una terminal de teleproceso.
- TARJETA	Indica la intervención de un archivo en tarjetas.
- DISCO MAGNETICO CINTA MAGNETICA	Indica la intervención de un archivo en disco magnético.
- CONECTORES	Los conectores son elementos de control de transferencia de un proceso a otro.
- F/P	FUERA/PAGINA. El proceso se en-

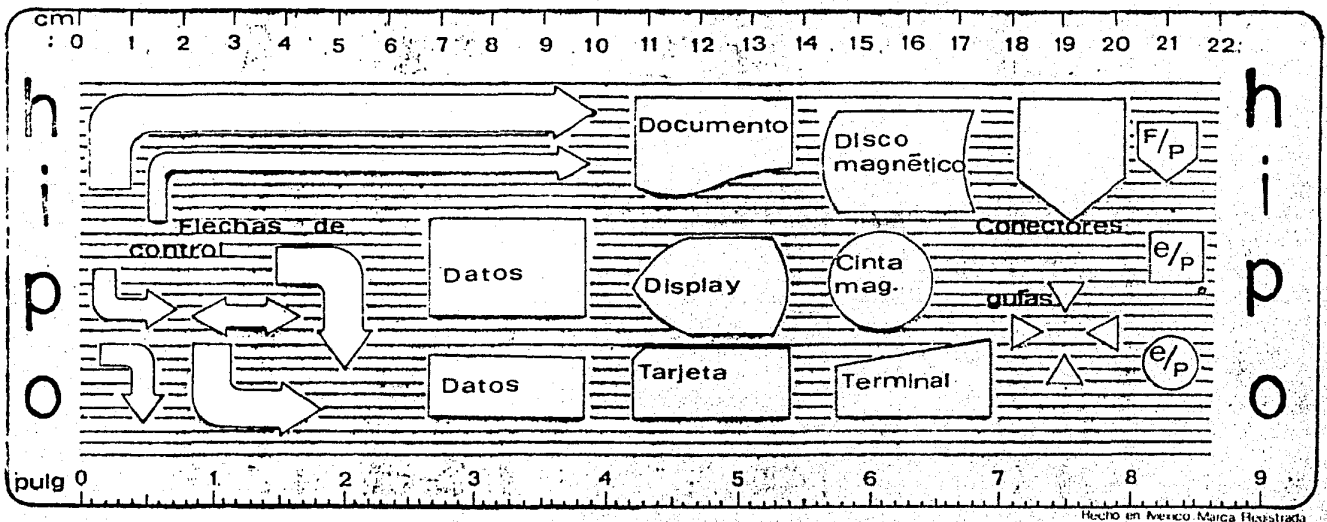
cuentra en otra página.

- E/P — EN/PAGINA. El proceso se encuentra en la misma página.

- GUIAS — Indican la terminación de una flecha delgada. (Generalmente flechas de control de condicionales).

La figura 2 muestra una plantilla "HIPO".

FIGURA 2



PLANTILLA 'HIPO'

APLICACION DEL "HIPO".

Una vez identificados los dos elementos del "Hipo", lo que procede es explicar cómo se conjuntan para su aplicación.

Considerando que el hipo tiene como finalidad identificar - los archivos que intervienen en un proceso, la hoja hipo -- tiene tres áreas distintas, destinadas a:

1.- Identificar los archivos que sirven de entrada de un -- proceso.
2.- Identificar los archivos que se generan en un determina do proceso.
3.- Identificar los archivos que se modificaron en un deter minado proceso.
4.- Identificar las condiciones de entrada o de salida de - un proceso.
5.- Describir el proceso, tan detalladamente como sea necesario.

De acuerdo a las siguientes consideraciones:

1.- Procurar ocupar una hoja por función.
2.- Distinguir las flechas de control de proceso y archivos de las de control de condiciones.
3.- Cuando una función identifique la ejecución de otras -- funciones, la descripción de éstas deberá enmarcarse en un cuadro, dentro de la zona de proceso.
4.- Cuando se desee mostrar el contenido de una área interna, de memoria, se debe ocupar el símbolo de "Datos".

5.- Cuando un proceso (o función) no sea posible de describir en una sola hoja, se deben ocupar los conectores -- F/P.

6.- Cuando un proceso (o función) continúa de una hoja anterior, se ocupa un conector E/P.

7.- Los conectores E/P que proceden del punto 6 y los F/P - del punto 5, deben indicar:

- El número del conector del que procede, y
- Número del conector al que antecede y, a una lado de la fecha:
- Número de nivel de la función que se describe.

8.- Cuando el resultado o el requerimiento de un proceso es de archivos, entonces la flecha de control debe ser blanca.

9.- En caso de que el resultado de una proceso sea ceder el control a un procedimiento ajeno al que se está describiendo, entonces la flecha de control debe ser de color negro.

5.4. PROGRAMACION WARNIER.

Al igual que la programación estructurada provee de un estilo de programación enmarcado dentro de tres estructuras lógicas, la programación Warnier proporciona otro estilo de - desarrollar programas.

La idea fundamental de esta modalidad en programación es, - contemplar cualquier programa bajo los conceptos de la teoría de conjuntos, es decir; el programa a resolver, constituye un conjunto que incluye a otros subconjuntos del pro-- grama, cada subconjunto debe descomponerse a su vez en otros subconjuntos. Por ejemplo: considere el archivo cuyos re-- gistros presentan el siguiente formato:

| NO. DE DEPARTAMENTO | NO. DE OFICINA | MOVIMIENTOS | EGRESOS |

Con lo que se deben efectuar las siguientes operaciones:

- Se deben acumular los movimientos y egresos de una - misma oficina, para obtener el total Oficina.
- Se deben acumular los totales de las oficinas de un mismo departamento, para obtener el total Departamento.
- Se deben acumular los totales de los departamentos - para obtener el total general.

En este caso el programa trata un cierto número de departamentos. Cada departamento comprende varias oficinas. Cada oficina varios movimientos y egresos, donde cada subconjun-

to pertenece a un conjunto superior.

Por lo que, para solucionar este problema, la parte del programa que trata de un departamento se ejecuta tantas veces como departamentos existan. Dentro de cada departamento encontramos subconjuntos que son ejecutados una sola ocasión, como la impresión de la línea de total, sin embargo, el subconjunto oficina se deberá ejecutar tantas ocasiones como oficinas existan.

Cuando un subconjunto de operaciones del programa está compuesto por instrucciones que satisfacen el objetivo de dicho subconjunto, entonces:

- Este subconjunto no puede descomponerse
- Todas las instrucciones que lo constituyen se ejecutan bajo las mismas condiciones.

Asimismo, cuando un subconjunto está compuesto por estas características, entonces se le llama secuencia lógica.

Entre dos instrucciones de una misma secuencia nunca puede haber bifurcaciones ni uniones; las instrucciones que se encuentran a uno y otro lado de una bifurcación o de una unión no se ejecutan ni bajo las mismas condiciones ni el mismo número de veces.

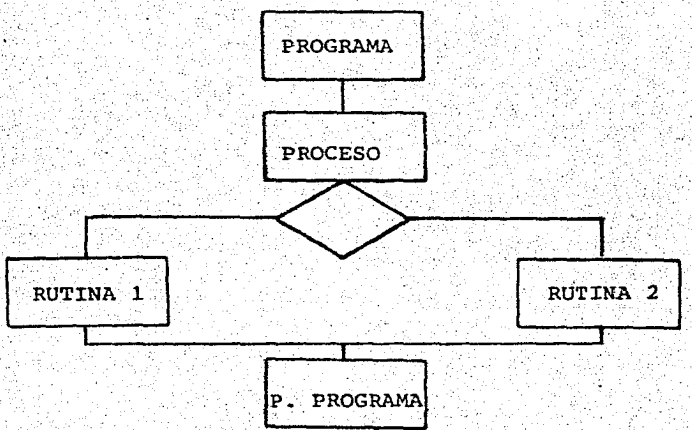
Una secuencia lógica está siempre comprendida:

- Entre una bifurcación y otra bifurcación
- O entre una bifurcación y una unión
- O entre una unión y una bifurcación

Cuya descomposición real es:

PRIMER NIVEL PROGRAMA	SEGUNDO NIVEL DEPTO.	TERCER NIVEL OFICINA
Programa { principio programa	{ principio depto.	{ principio oficina
Departamento	oficina	movimiento y egreso
fin programa	fin depto.	fin oficina

Otro tipo de estructura es la alternativa que presenta el - siguiente esquema:



Con la siguiente descomposición:

PROGRAMA {
- PRINCIPIO (1 VEZ)
- PROCESO (n VECES) {
  - PRINCIPIO (1 VEZ)
  - SI CONDICION -1 RUTINA 1
  - SI NO CONDICION -1RUTINA 2
  - FIN (1 VEZ)
- FIN (1 VEZ)

- O entre dos uniones.

Para organizar un programa en secuencias lógicas se debe -- descomponer el enunciado en subconjuntos, partiendo por el nivel más elevado, para lo cual se deben dar respuesta a:

. ¿A qué conjunto pertenece tal subconjunto y cuántas veces debe ser ejecutado?

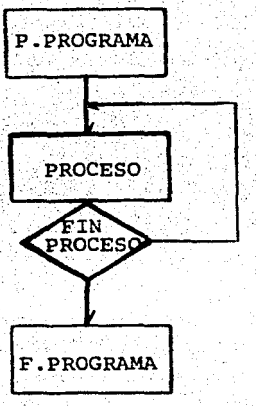
Una vez ordenado el programa por secuencias lógicas, entonces se debe responder:

. ¿Cuándo debe ser ejecutado dicho subconjunto?

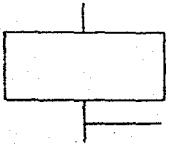
Lo que permite identificar que generalmente existirán rutinas que se deben ejecutar una sola vez, ya sea al principio y/o al final. Por lo anterior, se observa que la estructura general del programa es:

PROGRAMA (CONJUNTO)	PRINCIPIO	(subconjunto a ejecutar 1 vez)
	PROCESO	(subconjunto a ejecutar n veces)
	FIN	(subconjunto a ejecutar 1 vez)

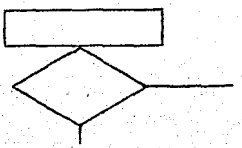
. Y su diagrama es:



Nótese que cada subconjunto está representado por

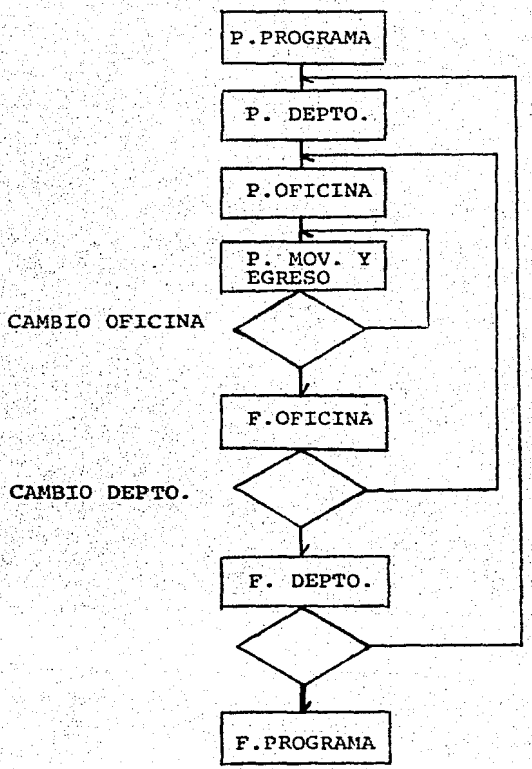


Para todo subconjunto seguido de una unión.



Para todo subconjunto terminado por una bifurcación.

Para el caso expuesto se tiene la siguiente situación:



6. COMANDOS COBOL.

INSTRUCCION "OPEN".

El comando o instrucción open (abrir) permite el acceso a cualquier archivo, excepto a aquellos que son de uso interno como el archivo de "SORT", esto es, el primer paso a efectuar para accesar un archivo de entrada o de salida es abrirlo.

El formato del comando open es:

```
          ┌ INPUT        nombre-archivo-1, nombre-archivo-2...
          │ OUTPUT       nombre-archivo-3, nombre-archivo-4...
          │ I-O
OPEN      │ INPUT-OUTPUT } nombre-archivo-5...
          │ O-I          ┘
          └ EXTEND       nombre-archivo-6
```

En esta opción la palabra INPUT, indica que el nombre-archivo-1, nombre-archivo-2, son archivos declarados como de entrada al proceso, por lo que deberá concordar esta asignación con el tipo de periférico asignado en la cláusula SELECT. De igual manera la palabra OUTPUT, indica que el nombre-archivo-3, nombre-archivo-4 son archivos asignados a la salida del proceso.

Las palabras I-O, INPUT-OUTPUT, O-I, indican la presencia de un archivo que tiene las características de los dos tipos anteriores, es decir, puede ser empleado como archivo de entrada y/o archivo de salida, tal es el caso del uso de un disco magnético, este tipo de archivos deberán ser declarados de acceso RANDOM.

INSTRUCCION "MOVE".

El comando MOVE transfiere datos de una área de memoria a otra,- ya sean internas o externas, siguiendo las reglas de edición que se aclaran posteriormente.

El formato del comando MOVE es:

Formato-1

```
MOVE  nombre-campo-1 TO nombre-campo-2.
```

Formato-2

```
MOVE  CORRESPONDING / CORR  identificador-1 TO identificador-2
```

Formato-3

```
MOVE  nombre-campo-3 TO nombre-campo-4, nombre-campo-5...
```

El formato-1 es la forma más sencilla de transferir infomación, el nombre-campo-1 y nombre-campo-2, pueden ser campos sencillos o bien registros completos.

El formato-2 presenta la alternativa de mover los contenidos de un identificador a otros, donde los nombres de los primeros son iguales a los segundos.

El formato-3 permite llevar la información del nombre-campo-3 a más de un campo receptor.

Las características del OPEN EXTEND, es que es una opción que se emplea fundamentalmente en "reprocesos" en los que no es conveniente regrabar información por lo que es necesario posicionarse en el primer registro disponible.

Cuando un comando OPEN INPUT es ejecutado, el CPU busca en la asignación I/O o disco directorio el archivo específico, si el archivo es encontrado, es abierto y está dispuesto al uso de declaraciones. Si el archivo no está en la asignación o directorio, el operador recibe un mensaje y es suspendido el programa hasta que:

a) El archivo aparece en tablas o en directorio.
b) El archivo es un archivo opcional y el operador indica que el archivo no está presente. (i.e, the of message).
c) El operador indica otro archivo como iniciador.
d) El programa es descontinuado.

INSTRUCCION "CLOSE'.

El comando CLOSE es empleado para indicar que el proceso de un - archivo ha sido terminado. El archivo debe haber sido abierto - previamente para que la cláusula CLOSE pueda ser ejecutada.

El formato de la cláusula CLOSE es:

CLOSE nombre-archivo-1, nombre-archivo-2... $\left\{\underline{\text{LOCK}}\right\}$

Los archivos en tarjetas o en papel generalmente no es necesario cerrar, sin embargo los archivos que hayan requerido de otro tipo de periférico deberán ser cerrados con la finalidad de evitar que sean empleados en otro proceso o bien que reciban basura. La opción LOCK es empleada para dar mayor seguridad a los archivos en cinta o disco.

INSTRUCCION "WRITE".

El comando WRITE (escribir) permite la creación de un registro - lógico en un archivo declarado como de salida o de entrada/salida básicamente transfiere la información para grabar, tomándola de las áreas de memoria y llevándola a los archivos específicos.

Este comando tiene el siguiente formato:

```
WRITE  nombre-registro  [FROM identificador-1]

[ {AFTER  } ADVANCING { {identificador-2} [LINES] }                 ]
[ {BEFORE }           { {entero         } [LINE ] }                 ]
                      { {mnemónico      }         }
                      { {palabra-clave  }         }

[ : AT {END-OF-PAGE} instrucción-imperativa                         ]
       {EOP        }
```

Cuando se desea escribir un registro lógico en un archivo secuencial, basta con indicarlo con la alternativa "WRITE nombre-registro" (como cintas), 'FROM' está indicando al sistema que para escribir el registro necesita mover previamente la información del contenido del identificador-1 (el cual debe tener el formato del archivo a escribir).

La alternativa es { AFTER / BEFORE } ADVANCING...

Indica al computador si antes o después de imprimir una línea debe avanzar un cierto número de líneas en blanco (identificador-2,

entero), o se debe saltar de página (mnemónico), el cual se indi ca dependiendo del sistema en cuestión.

Si se desea llevar un control sobre las líneas impresas en una - hoja y saltar de hoja para continuar la impresión, la alternativa

{ END-OF-FILE / EOF } permite tal acción.

INSTRUCCION "STOP".

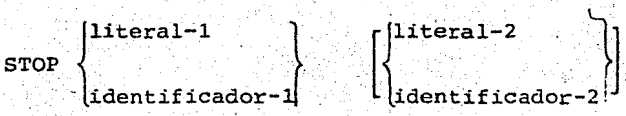
El comando STOP causa terminación o suspensión temporal del programa objeto.

El comando STOP presenta 2 formatos:

Formato-1.

STOP RUN

Formato-2.



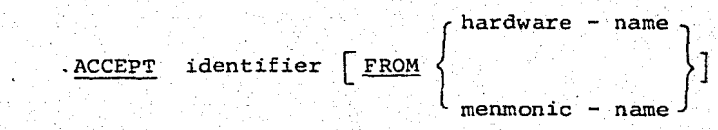
Cuando el comando STOP RUN es ejecutado, todo archivo abierto es cerrado, todas las áreas de memoria son borradas, STOP RUN no es usada para altas temporales en un programa. El uso del STOP literal-1, STOP identificador-1 ocasiona una suspensión temporal e inicia un mensaje, para activarlo nuevamente se ocupa START de la consola.

INSTRUCCION "ACCEPT".

La función del comando ACCEPT es permitir la entrada de cadenas de datos por la consola.

El formato del comando ACCEPT es:

Formato-1.



El comando ACEPT genera un mensaje en la consola el cual indica el nombre-dato a introducir. El programa es entonces suspendido hasta que el operador proporciona el dato. El identificador puede ser un elemento único o un grupo de contenidos además que un index-data-intem, index-name, EVENT, LOCK, o CP.

El grupo de contenidos puede ser DISPLAY o DISPLAY-1 usado y no más de 256 caracteres pueden ser aceptados en una ocasión.

INSTRUCCION "DISPLAY"

El comando DISPLAY proporciona para la impresión de bajos volumenes de datos, mensaje de errores e instrucciones por consola al operador.

```
DISPLAY { literal-1       } [{ literal-2       }] [UPON { MNEMONIC-NAME  }]
        { identificador-2 }  { identificador-2 }        { nombre-mnemonic }
```

Cada literal puede ser una constante figurativa, excepto ALL. Si una constante figurativa es especificada en uno de los operandos, sólo una única ocurrencia de la constante figurativa.

Cuando el display consiste de operandos-múltiples, el dato comprendido en el primer operando es desplegado.

DISPLAY nombre-dato(s) UPON Hardware-name

INSTRUCCION "GO TO".

El comando GO TO es una cláusula de control de transferencia de una parte de la PROCEDURE DIVISION a otra.

El formato para el comando GO TO presenta las dos siguientes opciones.

Formato 1

GO TO nombre-procedimiento-1

Formato 2

GO TO procedimiento-1 [,procedimiento-2 ... procedimiento-n

DEPENDING ON { fórmula / idenfificador }

Cada nombre-procedimiento es el nombre de un párrafo o una sección, o un párrafo calificado por un nombre sección.

INSTRUCCION "PERFORM".

El comando PERFORM, permite ejecutar cualquier rutina que se encuentre dentro de la estructura del programa, pasando el control a la rutina de que se trate y volviendo a la instrucción inmediata posterior, de donde fue mandada a ejecutar, esto es, pasa, el control explícitamente a uno o más procedimientos y volviendo el control implícitamente cuando se terminó la ejecución del proceso.

En este momento se presenta sólo una opción del manejo de esta instrucción.

PERFORM nombre-párrafo.

NOTA. Más adelante se analiza este comando con todas sus opciones.

INSTRUCCION "IF".

Generalmente dentro de un programa Cobol, es necesario conocer - si se cumplen ciertas condiciones o no, debidas al proceso mismo, existen instrucciones que de alguna forma controlan el cumplimien to de tal situación. La tabla que a continuación se presenta -- muestra tales instrucciones.

INSTRUCCION IMPERATIVA	CONDICIONAL
ADD	
SUBTRACT	ON SIZE ERROR
MULTIPLY	
DIVIDE	
COMPUTE	
GO TO	DEPENDING ON
	AT END
READ	INVALID KEY
WRITE	AT END OF PAGE
	INVALID KEY
PERFORM	UNTIL *

Sin embargo, en ocasiones el saber si es cierta o falsa determinada condición sin utilizar un verbo de la tabla, determina el - flujo total del proceso dependiendo de si es cierta o falsa.

El comando IF, permite conocer dichas condiciones, y, modificar- si es necesario el flujo del proceso, el formato de comando IF,es

*Se verá en un capítulo posterior.

```
IF condición { instrucción-1 } [ ELSE { instrucción-2 } ]
             { NEXT SENTENCE }       { NEXT SENTENCE }
```

La instrucción-1 será ejecutada solo si el valor de la condición es verdadero, de otra manera se ejecutará la instrucción-2. La - Opción NEXT SENTENCE permite salir de la instrucción IF, ya que ésta debe terminar con un punto, de no hacerlo así se entenderá que todas las instrucciones posteriores también se deben ejecu-- tar.

Respecto a la condición, existen diferentes formas de establecer la, a continuación se describen:

CONDICION DE RELACION

Formato.

```
{identificador-1}                                {identificador-2}
{literal-1      } condición de relación          {literal-2      }
{exp. aritmética}                                {exp. aritmética}
```

La condición de relación puede tener una las siguientes formas:

```
IS [NOT] { GREATER THAN }
         { LESS    THAN }
         { EQUEAL  TO   }
```

CONDICION DE SIGNO

```
{identificador  }      [NOT] { POSITIVE }
{               } IS         { NEGATIVE }
{exp. aritmética}            { ZERO     }
```

CONDICION DE CLASE

```
identificador  IS [NOT] { NUMERICAL  }
                        { ALPHABETIC }
```

CONDICION DE NIVEL

Cuando la condición es propiamente el nivel 88 de un campo definido en Working-Storage, se deberá entender que la condición se sustituirá por la especificación del nivel mencionado.

Cualquier condición se podrá componer con la ayuda de los operadores lógicos AND (Y) y OR (O).

6.1. INSTRUCCIONES ARITMETICAS.

INSTRUCCION ADD, conduce a que dos o más contenidos sean sumados y el resultado sea almacenado en un campo específico.

El formato de comando ADD es el siguiente:

Formato-1

```
ADD { identificador-1 } [ ,identificador-2 ]
    { literal-1       } [ ,literal-2     ]  ...TO identificador [ROUNDED]
    [ ,identificador-n [ROUNDED] ] ... [ON SIZE ERROR instrucción imperativa]
```

Formato-2

```
ADD { identificador-1 } { identificador-2 } [ ,identificador-3 ]
    { literal-1       } { literal-2       } [ ,literal-3     ]  ...
    GIVING identificador-n [ROUNDED] [ ,identificador-n [ROUNDED] ] ...
    [ON SIZE ERROR instrucción-imperativa]
```

En el formato-1 los contenidos de los identificadores o literales (1, 2) son sumados al contenido del identificador-n, dejando el resultado de esta operación en el campo del mismo identificador.

El formato-2 permite sumar el contenido de 1 ó más campos - (identificadores o literales), acumulando el total en el -- campo asignado por la opción GIVING.

La opción ROUNDED, permite que el resultado obtenido sea redondeado.

La opción ON SIZE ERROR, indicará por medio de la instrucción imperativa que acción ejecutar cuando en la operación efectuada exista un error.

INSTRUCCION "SUBTRACT".

La instrucción SUBTRACT, efectúa la diferencia de uno ó más campos numéricos y referencia los valores de uno ó más campos de resultados, básicamente hay dos formatos de esta instrucción.

Formato 1

```
SUBTRACT { identificador-1 } [ ,identificador-2 ]
         { literal-1       } [ ,literal-2       ] FROM identificador-n [ROUNDED]
  [identificador-n [ROUNDED] ... [;ON SIZE ERROR instrucción-imperativa]]
```

Formato 2

```
SUBTRACT { identificador-1 } [ ,identificador-2 ] FROM identificador-n
         { literal-1       } [ ,literal-2       ]
      GIVING identificador-n [ROUNDED] [identificador-z [ROUNDED]]
      [;ON SIZE ERROR instrucción-imperativa]
```

Tanto en el formato-1 como el formato-2 los contenidos numéricos de literal-1,, literal-2... o los de los identificadores, son restados del valor del contenido del identificador n, la diferencia entre uno y otro formato es que: en el formato-1 el resultado de la operación es almacenado en el campo del identificador-n, mientras que, en el formato-2 el resultado es referenciado al campo del identificador-n.

La opción ROUNDED, redondea los resultados obtenidos.

La opción ON SIZE ERROR permite ejecutar alguna acción cuando el resultado obtenido sea incorrecto.

INSTRUCCION "MULPIPLY".

El comando MULTIPLY ejecuta la multiplicación de campos y - también referencia el resultado a un campo específico. Existen dos formatos para esta instrucción:

Formato-1

```
MULTIPLY { identificador-1 } BY identificador-2 [ROUNDED]
         { literal-1       }
        [;identificador-3 [ROUNDED]]
        [;ON SIZE ERROR instrucción-imperativa]
```

Formato-2

```
MULTIPLY { identificador-1 } BY identificador-2
         { literal-1       }
        GIVING identificador-n [ROUNDED]
       [;ON SIZE ERROR instrucción imperativa]
```

En el formato-1 el valor numérico del contenido del identificador-1 o la literal-1 es multiplicada y almacenada por - el campo del identificador-2.

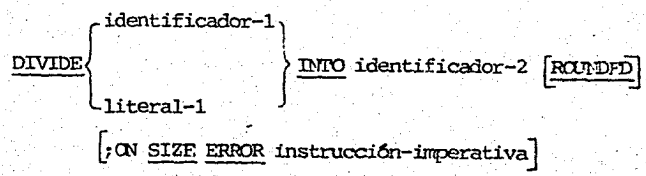
En el formato-2 el valor resultante de la operación es almacenada en el campo del identificador-n.

La opción ROUNDED, redondea los resultados obtenidos y la - opción del ON SIZE ERROR ejecuta una instrucción específica al existir un error al efectuar la operación.

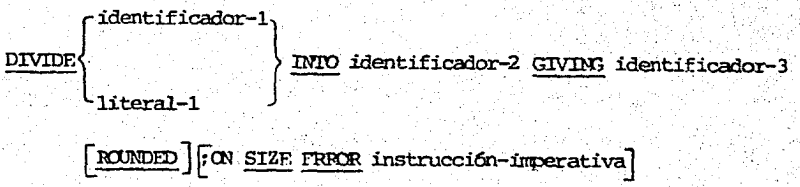
INSTRUCCION "DIVIDE".

Este comando efectúa el cociente de dos campos numéricos. - Los formatos más usuales son:

Formato-1



Formato-2



Lo que básicamente se ejecuta es una división del identificador-1 o literal-1 entre el identificador-2, en el formato-1, el resultado es almacenado en campo del identificador-2, en el formato, la opción GIVING asigna el resultado al - campo del identificador-3.

La opción ROUNDED, redondea los resultados.

La opción ON SIZE ERROR, permite ejecutar una instrucción - específica cuando existe un error en los cálculos.

INSTRUCCION "COMPUTE".

La instrucción COMPUTE, se emplea cuando es necesario ejecutar un cálculo complicado que no es posible desarrollar con las instrucciones elementales (ADD, DIVIDE, etc.).

El formato de la instrucción es:

COMPUTE identificador-1 [ROUNDED] identificador-2 [ROUNDED]...
= expresión-aritmética [;ON SIZE ERROR instrucción-imperativa]

Esta instrucción consiste fundamentalmente en asignar el valor obtenido al desarrollar la expresión aritmética o los valores de los identificadores.

El orden en que se ejecutan las operaciones en la expresión aritmética es:

1o. Todo lo que se encuentre entre paréntesis ( )
2o. Exponenciación **
3o. Multiplicación y/o división *, /
4o. Suma y/o resta +, -

6.2. INSTRUCCIONES COMPLEMENTARIAS.

INSTRUCCION "EXAMINE".

El comando EXAMINE se puede emplear como contador o reemplazador de ocurrencias dadas de caracteres, en un campo específico.

Formato-1

```
EXAMINE  identificador-1  TALLYING { UNTIL FIRST } literal-1
                                    { ALL       }
                                    { LEADING   }
        [REPLACING BY  literal-2]
```

Formato-2

```
EXAMINE  identificador-1  REPLACING { ALL       } literal-1
                                   { LEADING   }
                                   { UNTIL FIRST }
        BY  literal-4
```

Siendo la literal-1 numérica o no, el análisis de éstas comienza por el caracter más a la izquierda, siguiendo hacia la derecha y analizando caracter por caracter.

La opción de TALLYING, requiere que se declare en WORKING-STORAGE SECTION, un contador especial llamado TALLY, el cual deberá ser definido como numérico entero, las características que se presentan en el formato-1 es que el contador contiene:

a.- Cuando UNTIL FIRST es empleada, el número de ca-

racteres distintos a literal-1, después de encon trada ésta.

b.- Cuando se emplea ALL, el número de ocurrencias - de literal-1.

c.- Cuando LEADING, el número de ocurrencias de lite ral-1 antes de encontrar la última.

Cualquiera de las opciones que emplee la opción REPLACING, deberá contemplar las características anteriores, además de que:

a.- Cuando la opción ALL es empleada, entonces literal-2 o literal-4 son sustituídas por literal-1 o literal-3.

b.- Cuando la opción LEADING se emplea, la sustitución de literal-2 o literal-4 termina cuando: se encuentra un caracter igual a literal-1 o literal-3 o se encuentra el límite derecho de la palabra.

c.- Cuando la opción UNTIL FIRST, la sustitución de literal-2 o literal-4 termina al encontrar la -- primera ocurrencia de literal-4 o literal-3

d.- Cuando la opción FIRST se emplea, se sustituye - la primera ocurrencia de literal-2 o literal-4 - por literal-1 o literal-3.

INSTRUCCION "EXIT".

El comando EXIT provee un punto común para una serie de procedimientos.

Formato.

EXIT.

El comando EXIT, debe aparecer siempre solo y después de un nombre de párrafo.

INSTRUCCION "INITIALIZE".

El comando INITIALIZE permite inicializar ciertas áreas de referencia o trabajo, moviendo determinados valores a los - campos a inicializar, dependiendo de la naturaleza de éstos.

A los campos alfanuméricos mueve espacios y a los numéricos ceros.

Formato.

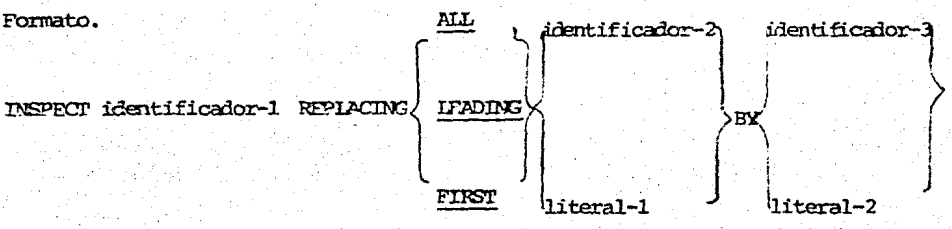
INITIALIZE identificador

Si el identificador es un area que contiene campos de dis-- tinta naturaleza, se puede interpretar este verbo como el - equivalente a una serie de MOVE.

INSTRUCCION "INSPECT".

La instrucción INSPECT porporciona la facilidad de rempla-- zar cadenas de caracteres, de un campo específico, pero a - diferencia de la instrucción EXAMINE, lo efectúa en grupos de caracteres.

Formato.



Cuando la opción ALL es especificada cada ocurrencia de literal-1 encontrada en el contenido del identificador-1 es - remplazado por literal-2.

Cuando la opción LEADING se ocupa, cada ocurrencia contigua de literal-1 encontrada en el contenido del identificador-1 es remplazo por literal-2.

Cuando la opción FIRST se emplea, la primera ocurrencia de literal-1 en el identificador-1 será sustituído por literal 2.

7. APENDICE.

Existen algunas otras instrucciones adicionales a las presentadas anteriormente, las cuales se ha preferido presentar con un poco más de detalle, dada la importancia que revisten. Asimismo, dentro de los temas de almacenamiento de datos existe un apartado especial para la descripción, uso y manejo de "TABLAS DE INFORMACION".

7.1. TABLAS.

Uno de los temas más importantes cuando se habla de almacenar información, es la organización que los datos presentan, ya sean distintos contenidos cada campo o iguales, sin embargo, cuando son iguales o semejantes, se podría pensar en ordenarlos dentro de un arreglo que conserve tantas ocurrencias del campo como sean necesarias, sin que se presenten las situaciones de los siguientes ejemplos:

Ejemplo 1.

Considérense las calificaciones obtenidas por un grupo de - alumnos ordenadas por número de alumno.

```
05   CALIFICACION-1        PIC 99.99.
05   CALIFICACION-2        PIC 99V99.
05   CALIFICACION-3        PIC 99V99.
 }      }                     }    }
05   CALIFICACION-N        PIC 99V99.
                  <-- CAMPO QUE IDENTIFICA LA
                      N-ESIMA CALIFICACION.
```

En este caso es necesario codificar tantos campos de "CALI-FICACION-Y" como sean requeridos.

Ejemplo 2.

Considérense los ingresos del personal de cierta fábrica, - por departamento, en donde se requiere que esta información esté ordenada por departamento y oficina:

```
05  DEPARTAMENTO-1.
    10  OFICINA1-1          PIC 9(06)V99.
    10  OFICINA1-2          PIC 9(06)V99.
     ?        ?                    ?
    10  OFICINA1-N          PIC 9(06)V99.
05  DEPARTAMENTO-2.
    10  OFICINA2-1          PIC 9(06)V99.
    10  OFICINA2-2          PIC 9(06)V99.
     ?        ?                    ?
    10  OFICINA2-N          PIC 9(06)V99.
     ?                             ?

05  DEPARTAMENTO-N.
    10  OFICINAN-1          PIC 9(06)V99.
    10  OFICINAN-2          PIC 9(06)V99.
     ?        ?                    ?
    10  OFICINAN-N          PIC 9(06)V99.
```

En este caso se hace necesario referencias tanto el departamento como la oficina de un modo fijo, sin poder variar, -- por lo que, es necesario accesar cada "OFICINA-N-1" de modo particular, así también la codificación.

Se ha querido presentar con estos ejemplos tres aspectos - fundamentalmente:

1. El inconveniente de la codificación.
2. El acceso a cada elemento debe ser fijo por ele-- mento.
3. Que se puede categorizar la información en los ni veles que se juzgue necesario.

LA CLAUSULA OCCURS.

Se ha mencionado que se requiere de un cierto número de ocu rrencias, tantas como el campo a ocupar pueda presentarse.- La cláusula OCCURS especifica el número de ocasiones que un campo se repetirá, excluyendo los niveles de información FD, 01, 77, 88.

El formato de la cláusula OCCURS es:

OCCURS entero TIMES

El entero indica el número de ocurrencias del campo.

Ejemplo.

```
05  CALIFICACION-1      PIC 99V99.
05  CALIFICACION-2      PIC 99V99.
    :                    }      }
05  CALIFICACION-50     PIC 99V99.

        CON LA CLAUSULA OCCURS

05  CALIFICACION OCCURS 50 TIMES PIC 99V99.
```

Donde se puede accesar cualquier calificación, declarando - un NOMBRE-INDICE en WORDKING-STORAGE, de tal forma que se - puede accesar la quinta información de la siguiente manera:

CALIFICACION (5) o CALIFICACION (INDICE)

Cuando INDICE tiene asignado el valor de 5.

REDEFINICION DE DATOS.

En ocasiones se dispone de datos que requiere sean utilizados de acuerdo a formatos diferentes al original, por ejemplo:

```
05 FECHA          PIC 9(6)   ó
05 FECHA-1.
   10 DIA-FECHA        PIC 9(2).
   10 MES-FECHA        PIC 9(2).
   10 AÑO-FECHA        PIC 9(2).  ó

05 FECHA-2.
   10 DIA-FECHA        PIC 9(2).
   10 MES-AÑO          PIC 9(4).
```

Lo cual implicaría al ocupar el verbo MOVE en la PROCEDURE, para disponer de la información en cualquiera de las tres - presentaciones, lo que sería:

```
MOVE  FECHA    TO   FECHA-1.  o
MOVE  FECHA    TO   FECHA-2.  o
MOVE  FECHA-1  TO   FECHA-2.  o
MOVE  FECHA    TO   FECHA-1,  FECHA-2.
```

Lo que permite la cláusula REDEFINES, al emplearla con el - campo FECHA-1 o FECHA-2 en ambas, de la siguiente forma:

05	FECHA	
05	FECHA1 REDEFINES	FECHA.
10	FECHA1-DIA	PIC 99.
10	FECHA1-MES	PIC 99.
10	FECHA1-ANO	PIC 99.
05	FECHA2 REDEFINES	FECHA.
10	FECHA2-DIA	PIC 99.
10	FECHA2-AYN	PIC 9(04).

Lo cual permite contar con información en distintos forma-- tos con el sólo hecho de recibir información en fecha.

La cláusula REDEFINES se ocupa también acompañada de la --- cláusula OCCRUS, ya que esta última no permite la cláusula "VALUE" en sus elementos. La forma de emplearla es la si-- guiente:

```
05  MESES.
    10  FILLER PIC X(36) VALUE
        "ENEFEBMARABRMAYJUNJUKAGOSEPOCTNOVDIC".
05  EL-MES  REDEFINES MESES
    10  MES  OC  12  TIMES  PIC  X(3).
```

De esta forma se indica que existen 12 elementos de 3 posicio nes alfanuméricas cada una, de lo cual, al referirse al ele

mento cuarto (ABR), se hará de la forma que aquí se muestra:

MES (4) indica que se tomarán las letras del filler de meses,

ENE	FEB	MAR	ABR	MAY
1	2	3	4	5

Que son ABR.

Conjuntando los ejemplos anteriores, se puede relacionar el nombre del mes con el número de mes obtenido en la fecha.

TABLAS DE DOS O MAS DIMENSIONES.

Cuando se requiere manejar más de una variable, como en el ejemplo 2, se puede diseñar una "MATRIZ" de dos o más dimensiones, empleando nuevamente la cláusula OCCURS, de la siguiente forma:

Supóngase que se tienen 10 departamentos y 15 oficinas por departamento, entonces:

```
05    DEPARTAMENTO  OCCURS 10 TIMES.
  10  OFICINA  OCCURS 15 TIMES PIC 9(06)V99.
```

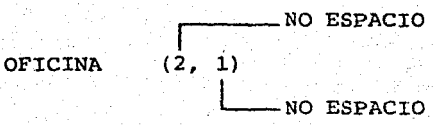
De tal forma que; OFICINA (2,1) indica, la oficina uno del departamento dos, lo que equivaldría a:

OFICINA-2-1

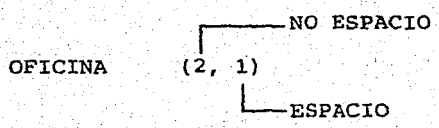
La codificación de este tipo de tablas sigue los siguientes

puntos:

. No se dejan espacios dentro del paréntesis ni a la dere-- cha ni a la izquierda:

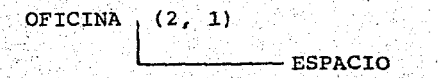


. Se debe dejar al menos un espacio en blanco entre la coma y el siguiente índice:



No así entre el índice que antecede a la coma

. Se debe dejar al menos un espacio entre el nombre de campo y el inicio del paréntesis:



. No es permitido los índices negativos o ceros

OFICINA (-2,1) u

OFICINA (2, 0).

Ejemplo ilustrativo:

```
05  RENGLON    OCCURS  3  TIMES.
 10 COLUMNA    OCCURS  4  TIMES.
  15ELEMENTO    PIC  9(3)  vg.
```

De tal forma que si se desea conocer o accesar el elemento (3,3), bastará con:

```
ELEMENTO (3,3)
         |└COLUMNA
         └──RENGLON
```

o

```
ELEMENTO (IND-RENGLON, IND-COLUMNA)
```

Donde IND-RENGLON y el IND_COLUMNA fueron definidos en WORKING STORAGE SECTION.

TABLAS DE LONGITUD VARIABLE

Ahora bien, qué pasa si el número de localidades que se necesita es un número desconocido implica que se requiere de una tabla de dimensión variable.

Dicha tabla se crea de igual forma, sólo que se necesita un elemento más, un campo en WORKING-STORAGE, el que deberá contener el valor asignado a la dimensión de la tabla.

Para lo cual se emplea el siguiente formato:

```
{OC      } entero-1 TO entero-2 TIMES [DEPENDING ON identificador-1]
{OCCURS  }

[[{ASCENDING } KEY is identificador-2 [identificador-3] ...]]
  {DESCENDING}

[INDEXED BY nombre-índice- [nombre-índice-2]...]
```

Donde entero-1 es un número entero mayor que cero y menor al entero-2. Entero-1 indica el número de elementos mínimos y entero-2 el número de elementos máximo, identificador-1, es el campo en WORKING STORAGE SECTION, que contiene la información respecto a la ocurrencia.

Ejemplo:

Correcto.

```
77  NUM-ELEMENTOS  PIC  99  VALUE  80.
01  TABLA
    05 ELEMENTOS OC 1 TO 100 TIMES DEPENDING NUM-ELEMENTOS
```

Incorrecto.

```
01  TABLA
    05 ELEMENTOS OC 1 TO 100 TIMES DEPENDING NUM-ELEMENTOS
       15  NUM-ELEMENTOS PIC 99.
       15  DEMAS-DATOS.
```

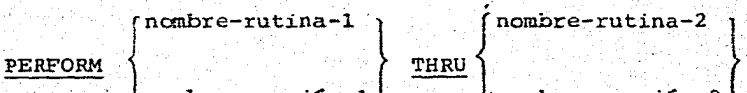
7.2. INSTRUCCIONES "PERFORM","SEARCH", "SORT".

INSTRUCCION "PERFORM".

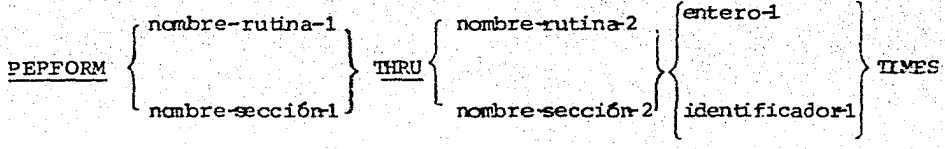
Como se indicó anteriormente el comando PERFORM manda ejecutar una rutina específica desde una determinada parte del programa Cobol, y regresando el control de ejecución a la instrucción inmediata posterior al PERFORM.

Este comando muestra varias formas de hacer uso del mismo, permitiendo ejecutar la rutina para situaciones diversas, o bien, un número específico de ocasiones o más aún, hasta el cumplimiento de alguna condición.

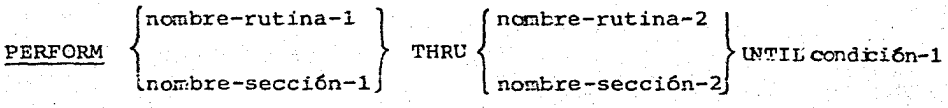
Formato-1.



Formato-2.



Formato-3.



Formato 4.

```
PERFORM { nombre-rutina-1    } [ THRU { nombre-rutina-2    } ]
        { nombre-sección-1   }        { nombre-sección-2   }

 VARYING identificador-1 FROM { literal-1      }
                              { identificador-2}
 BY      identificador-3 UNTIL condición-1
[AFTER   identificador-4 FROM { literal-2      }
                              { identificador-5}
 BY      identificador-6 UNTIL condición-2]
[AFTER   identificador-7 FROM { literal-3      }
                              { identificador-6}
 BY      identificador-9 UNTIL condición-3]
```

En cualquiera de los formatos, el nombre-rutina-1 indica cuál es el conjunto de instrucciones a ejecutar, aunque bien todas estas instrucciones se encuentren en una rutina o en un conjunto de rutinas, cuando se cumple el nombre-sección-1.

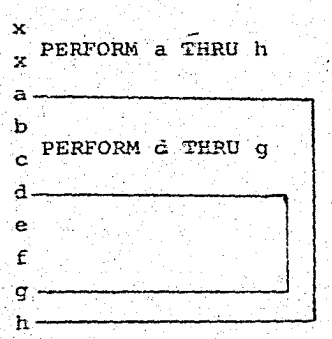
El formato-1 es la forma más sencilla de emplear el PERFORM, la opción THRU indica que existe una rutina final de instrucciones, que generalmente es de la siguiente forma:

```
nombre-rutina-2          nombre-sección-2        SECTION.
EXIT.              o     nombre-rutina.
                         EXIT.
```

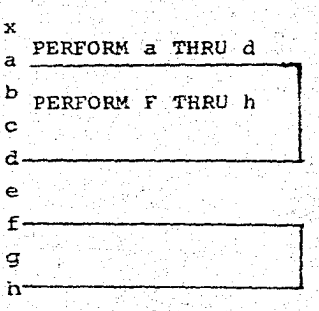
Lo que está indicando que se ha terminado el conjunto de\ -- instrucciones y se vuelve el control al punto original.

El comando PERFORM, bien puede ser una instrucción dentro - de una serie de instrucciones previamente mandadas a ejecutar con lo que se debe prever las siguientes situaciones:

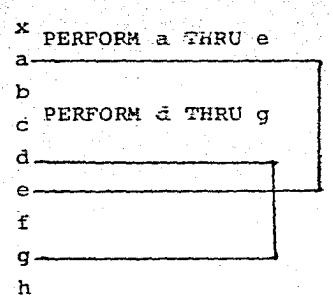
ESQUEMA-1



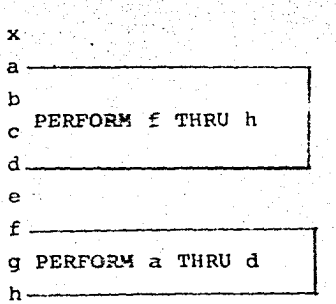
ESQUEMA-2



ESQUEMA-3



ESQUEMA-4



Las situaciones que muestran los esquemas 1 y 2 se puede de

cir que son situaciones normales, en las que se presenta - mandar ejecutar un PERFORM, donde el inicio y fin de uno no interfiere con el inicio y fin de otro.

Sin embargo, las situaciones que muestran los esquemas 3 y 4 se pueden llamar irregulares, ya que el esquema-3 provoca una ruptura del flujo normal del THRU, impidiendo que se -- ejecute el primero o segundo PERFORM, ya que la etiqueta -- "d" forma parte de las instrucciones del segundo PERFORM, - por lo que al encontrarla daría por terminado el primer PERFORM y el segundo nunca terminaría. El esquema-4 provocaría caer en un LOOP interminable, a menos de que se llevasen controles muy bien cuidados.

Prácticamente, todos los formatos se realizan de la misma - forma que el formato-1, salvo que:

a. El formato-2, se ejecutará un número determinado de ocasiones.
b. El formato-3, se ejecutará hasta que se cumpla la condición-1.
c. El formato-4, permite que se ejecuten las rutinas mencionadas, variando el identificador-1 que bien podría ser - un contador declarado en WORKING-STORAGE, de acuerdo al - valor del identificador-2, a partir de un valor inicial para el identificador-3, hasta cumplirse la condición-1.

7.3. COMANDO "SORT".

Generalmente se requiere que además de la organización lógica que presenta la información de un archivo, los registros del mismo sean clasificados anterior o posteriormente al proceso para el cual son empleados, esto es, cuando se requiere que los registros de un archivo ingresen al proceso ya clasificado, o que el nuevo archivo esté ya organizado.- Es cuando se requiere emplear el comando SORT.

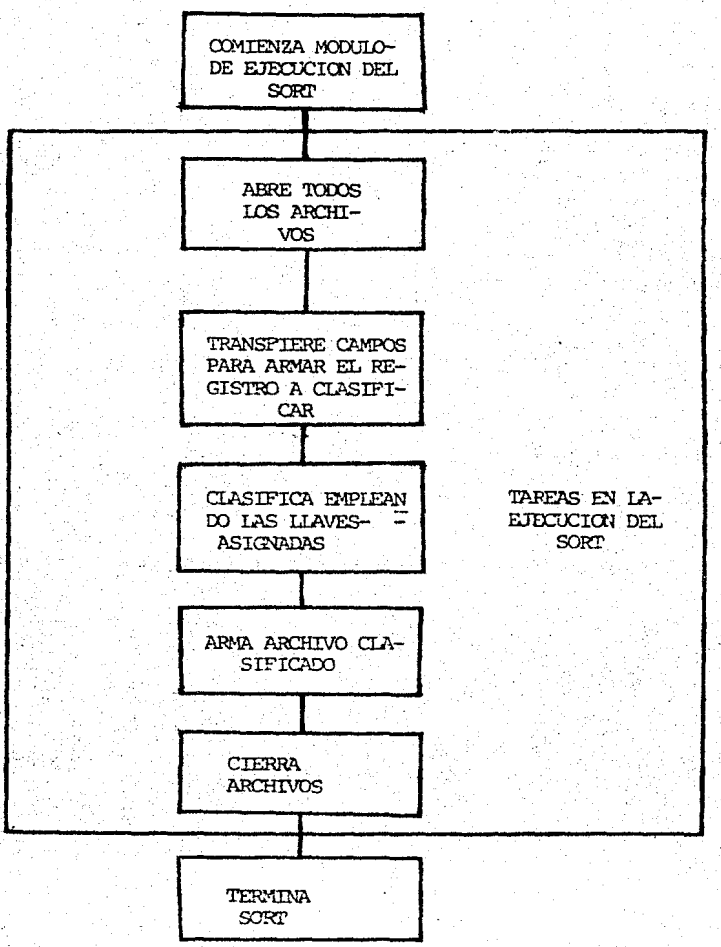
La utilización de este comando implica hacer ciertas especificaciones desde la división del medio ambiente (ENVIRONMENT DIVISION) en la cláusula SELECT, de la siguiente forma:

```
SELECT nombre-archivo-sort  ASSING TO { nombre-dispositivo }
                                      { file-code         }
```

En la DIVISION DE DATOS (DATA DIVISION)

```
SD  nombre-archivo-sort
    DATA RECORD IS  nombre-registro-sort.
01  nombre-registro-sort.
```

Como se mencionó anteriormente, el comando SORT emplea archivos internos de trabajo, asimismo, genera la ejecución automática de instrucciones comunes de cobol, el diagrama número 1 muestra el comportamiento de esta instrucción.


COMIENZA MODULO-
DE EJECUCION DEL
SORT
ABRE TODOS
LOS ARCHI-
VOS
TRANSPIERE CAMPOS
PARA ARMAR EL RE-
GISTRO A CLASIFI-
CAR
CLASIFICA EMPLEAN
DO LAS LLAVES-
ASIGNADAS
ARMA ARCHIVO CLA-
SIFICADO
CIERRA
ARCHIVOS
TERMINA
SORT
TAREAS EN LA-
EJECUCION DEL
SORT

Formato

```
SORT  nombre-archivo-sort

ON { ASCENDING  } KEY   nombre-llave-1
   { DESCENDING }

ON { ASCENDING  } KEY   nombre-llave-2
   { DESCENDING }

USING  nombre-archivo-1
INPUT  PROCEDURE IS  nombre-sección-1  THRU nombre-sección-2

GIVING nombre-archivo-2
OUTPUT PROCEDURE IS nombre-sección-3 THRU nombre-sección-4
```

El nombre-archivo-sort es el nombre asignado al archivo a - sortear o clasificar. Los nombres-llave son campos del archivo de sort. La opción ASCENDING indica que la clasificación deberá ejecutarse en orden ascendente, mientras que la opción DESCENDING indica sea ejecutada en forma descendente de acuerdo al contenido de la llave indicada.

Como se podrá observar, este comando se compone de dos opciones obligatorias que son:

```
USING nombre-archivo-1
INPUT PROCEDURE IS nombre-sección-1  THRU nombre-sección-2
```

GIVING nombre-archivo-2

OUTPUT PROCEDURE IS nombre-sección-2 THRU nombre-sección-4

El nombre-archivo-1, es el archivo del cual los datos serán transferidos al archivo de sort, el cual tendrá el formato del registro del nombre-archivo-2, para obtenerlo como resultado de la clasificación.

Cuando se emplea la opción del INPUT PROCEDURE junto con el OUTPUT PROCEDURE, se le está indicando al computador que la ejecución de la clasificación está sujeta a la aplicación de un proceso, para lo cual se seguirán las siguientes especificaciones.

INPUT PROCEDURE.

Indica cuál es el proceso previo a la clasificación y se requiere del comando RELEASE, que equivale a escribir el registro a clasificar, el formato es el siguiente:

RELEASE nombre-registro-sort FROM identificador

La explicación de este comando es idéntica al verbo WRITE, salvo que ésta esta orientada a la impresión del archivo -- sort.

OUTPUT PROCEDURE.

Indica cuál es el proceso posterior a la clasificación y se

requiere del comando RETURN, que equivale a leer un registro clasificado, el formato es el siguiente:

RETURN nombre-archivo-sort RECORD INTO identificador

AT END instrucción-imperativa

La explicación de este comando es idéntica al verbo READ, - salvo que ésta esta orientada a la lectura del archivo sort.

7.4. COMANDO "SEARCH".

Cuando se establece que en el proceso se requiere de una tabla, se puede emplear para: el ordenamiento más claro y codificación más rápida de las áreas de memoria, o bien porque ésta será empleada para efectuar búsquedas de información.

Cuando una tabla es definida con la opción del

```
OCCURS  entero  TIMES

ASCENDING
                KEY  nombre-llave
DESCENDING

INDEXED BY  campo-indexado.
```

Se establece que será empleada para búsqueda de elementos, donde el campo-indexado es elemento del campo de la ocurrencia.

Ejemplo:

El comando SEARCH establece la búsqueda de algún elemento de esta tabla, y el formato que presenta es:
búsqueda secuencial:

```
SEARCH nombre-tabla
  AT END  instrucción-imperativa-1
                    instrucción-imperativa-2
WHEN condición-1
                    NEXT SENTENCE
```

La búsqueda secuencial hace como su nombre lo indica, una - búsqueda secuencial desde el primer elemento, haciendo referencia al campo de la tabla y al campo-indexado.

EJEMPLOS.

8.1. DESCRIPCION DEL PROBLEMA.

Una de las actividades primordiales de una compañía de seguros es proteger al asegurado de la ocurrencia de un riesgo, que en caso del ramo de vida, es brindar al asegurado la confianza de que en la ausencia de él sus asegurados contarán con un capital que garantice o procure cierta situación.

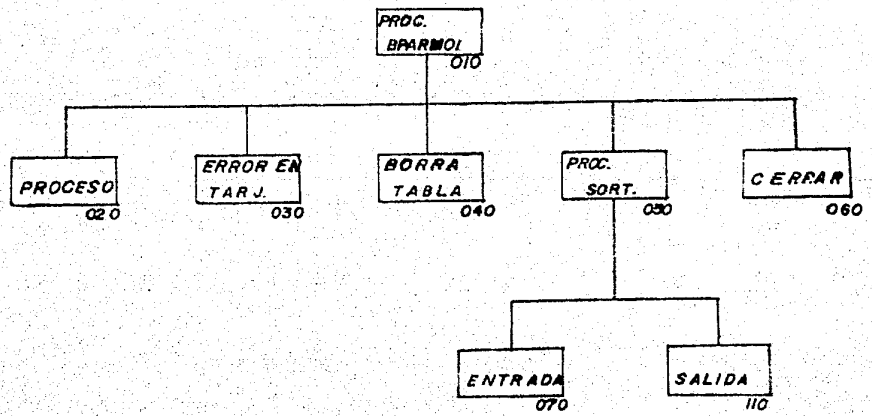
En el caso del seguro de daños, es proteger al asegurado de los daños a los que están expuestos sus bienes, sin embargo, no debemos olvidar que el seguro funciona bajo condiciones de incertidumbre y "buena fe", pero cuidado, esto no quiere decir que las compañías de seguros no cuenten con elementos de control que les permiten seleccionar sus "RIESGOS".

En el ramo de automóviles, la "Asociación Mexicana de Instituciones de Seguros" (AMIS), proporciona la información de los autos reportados por el Registro Federal de Vehículos, como robados, para que de esta manera, cuenten con elementos de control y no asegurar algún auto que fue robado anteriormente.

En el caso que se presenta, la compañía en cuestión recibe mensualmente la información de la AMIS, la que le sirve para actualizar sus bancos de información (en este caso, una base de datos, por lo que en el ejemplo, fueron codificadas

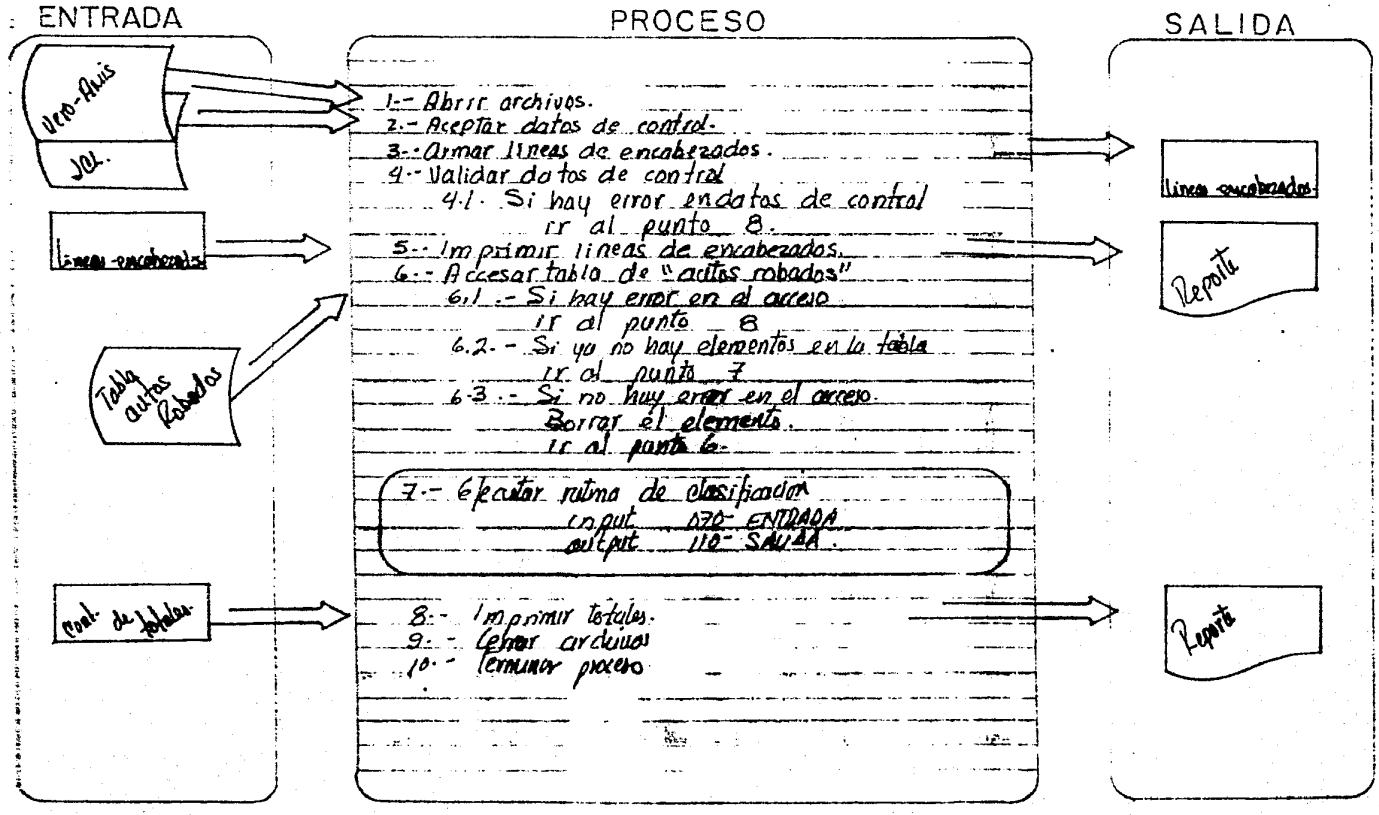
instrucciones de MDIV — Data Magnament IV —) y generar el reporte de control correspondiente.

PROC:
BPARMØI.



V.T.O.C.

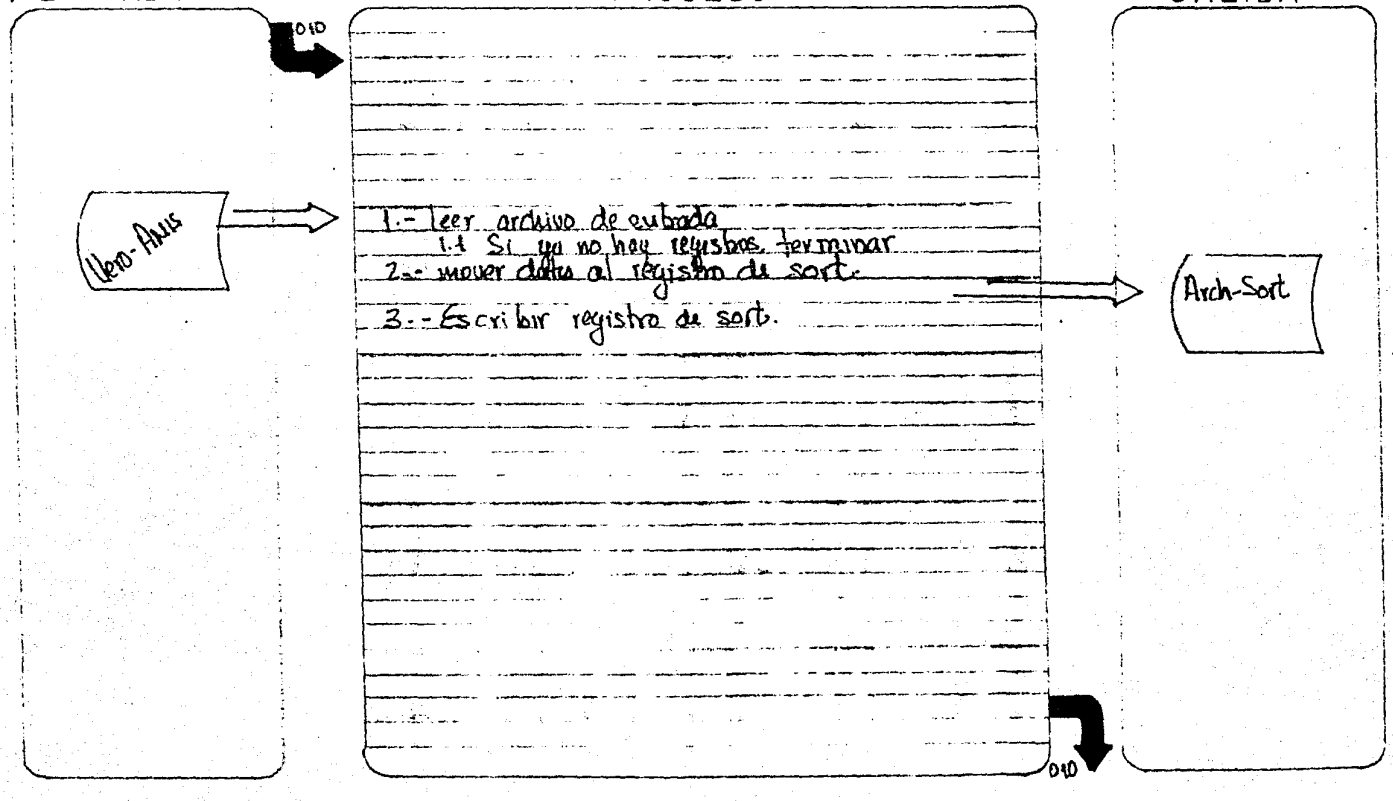
TESIS	PROBLEMA - 1	NIVEL 010	IDENTIFICACION BPARN01
DEPARTAMENTO	RESPONSABLE Carlos Alberto Ramirez Lazo.	NOTA	FECHA 20 / IX / 81



DEPARTAMENTO	RESPONSABLE	NOTA	FECHA
TESIS	CARLOS ALBERTO RAMIREZ LAZO		20 IX 81

PROBLEMA 1 | 070 | IDENTIFICACION: ENTRADA.

ENTRADA | PROCESO | SALIDA



DEPARTAMENTO	RESPONSABLE	NOTA	110	SALIDA
	CARLOS ALBERTO RAMIREZ LAZO			FECHA 20 | IX | 82

ENTRADA

Arch sort.

PROCESO

1.- Leer archivo de sort.
2.- Inicializa areas de tablas.
3.- Grabar nuevo elemento en tablas.
4.- Sumar 1 a los contadores
5.- Armar linea de detalle.
6.- Si contador-1 > 11
Escribir linea de detalle.
7.- Si lineas contadas > 59
- ejecuta 150- titulos

SALIDA

Tabla autos Robados

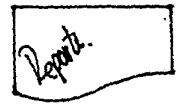
[redacted]	PROBLEMA 1		150	TITULOS
DEPARTAMENTO	RESPONSABLE Carlos Alberto Ramirez Lazo	NOTA		FECHA 20 IX 82

ENTRADA

PROCESO

1.- Escribir titulos

SALIDA



EJEMPLO.

NIVEL																
01	R E G I S T R O															
05	N O M B R E					OCUPACION			TELEFONO			DATOS PERSONALES				
10	APELLIDO PATERNO		APELLIDO MATERNO		NOMBRE	SUELDO		DESCRIPCION PUESTO	TIPO DE TELEFONO	No.		EDAD		CIUDAD		
15						PESOS	CENTAVOS			LADA	No. BIS	AÑOS	MESES	COLONIA		Z. P.

```
05  VERO-AUTO-MARC      PIC X(03).
05  VERO-AUTO-MODE      PIC X(02).
05  VERO-AUTO-USER      PIC X(03).
FD  ARCH-LIST
    LABEL RECORD IS STANDARD
    CODE-SET     IS GBCD
    LINAGE       IS 59 LINES
    DATA RECORD IS LINEA.
01  LINEA               PIC X(132).
SD  ARCH-SORT.
01  SORT-CLAV           PIC 9(08).
WORKING-STORAGE SECTION.
    COPY  DBWKPGC.
01  LINEAS-ENCABEZADO.
  05  LINEA-ENCA1.
    10  FILLER          PIC X(06) VALUE SPACES.
    10  FILLER          PIC X(08) VALUE "S.I.S.A.".
    10  FILLER          PIC X(29) VALUE SPACES.
    10  FILLER          PIC X(31) VALUE "
                                    ".
    10  FILLER          PIC X(18) VALUE SPACES.
    10  FILLER          PIC X(11) VALUE "PAGINA NO.".
    10  LINE-A1PA       PIC Z29   VALUE ZEROS.
    10  FILLER          PIC X(16) VALUE SPACES.
  05  LINEA-ENCA2.
    10  FILLER          PIC X(06) VALUE SPACES.
    10  FILLER          PIC X(10) VALUE
"PROCESO: BPARNO1".
    10  FILLER          PIC X(34) VALUE SPACES.
    10  FILLER          PIC X(23) VALUE
"REPORTE  NETOS  ROBADOS".
    10  FILLER          PIC X(33) VALUE SPACES.
```

Cont.

```
10  FILLER          PIC X(06) VALUE "FECHA".
10  LINE-A1D1      PIC 99    VALUE ZEROS.
10  FILLER          PIC X     VALUE "/".
10  LINE-A1MG      PIC 99    VALUE ZEROS.
10  FILLER          PIC X     VALUE "/".
10  LINE-A1AN      PIC 99    VALUE ZEROS.
10  FILLER          PIC X(06) VALUE SPACES.

01  WS-CAMPOS-VARIOS.
05  WS-CONTADOR    PIC 9(06) VALUE ZEROS.
05  WS-CONTADOR1   PIC 9(06) VALUE ZEROS.
```

Cont.

```
05  TABLA-MESES.
    10  FILLER          PIC X(10) VALUE "ENERO".
    10  FILLER          PIC X(10) VALUE "FEBRERO".
    10  FILLER          PIC X(10) VALUE "MARZO".
    10  FILLER          PIC X(10) VALUE "ABRIL".
    10  FILLER          PIC X(10) VALUE "MAYO".
    10  FILLER          PIC X(10) VALUE "JUNIO".
    10  FILLER          PIC X(10) VALUE "JULIO".
    10  FILLER          PIC X(10) VALUE "AGOSTO".
    10  FILLER          PIC X(10) VALUE "SEPTIEMBRE".
    10  FILLER          PIC X(10) VALUE "OCTUBRE".
    10  FILLER          PIC X(10) VALUE "NOVIEMBRE".
    10  FILLER          PIC X(10) VALUE "DICIEMBRE".
03  MESES-TABLA REDEFINES TABLA-MESES.
    10  MES             PIC X(10) OCCURS 12 TIMES.
05  WS-TABLA-DIAS.
    10  FILLER          PIC X(02) VALUE "31".
    10  WS-DIAS-FEB     PIC XX.
    10  FILLER          PIC X(20) VALUE
    "312831303130313130313031".
05  WS-DIAS-TAB REDEFINES WS-TABLA-DIAS.
    10  WS-DIAS-TABLA   OCCURS 12 TIMES PIC 99.
```

Cont.

```
*
 PROCEDURE          DIVISION.
*
 010-PROCESO-BPARNOL    SECTION.
 020-PROCESO.
     OPEN  INPUT   VERO-ANIS
           OUTPUT  ARCH-LIST.
     CALL "SSBRLE" USING W-DRIVACY.
     READY TARTARIF USAGE-MODE IS UPDATE.
     ACCEPT FECHA-NOX FROM DATE.
     ACCEPT CAMPOS-CONTROL.
     MOVE   FECHA-ANNO     TO LINE-ARAN.
     MOVE   FECHA-MES      TO LINE-ARME.
     MOVE   FECHA-DIA      TO LINE-ARDI.
     ADD    1              TO WS-PASIVA.
     MOVE   WS-PASIVA      TO LINE-AIDA.
     IF     FECHA-TARJ NOT NUMERIC
       OR   FECHA-TARJ EQUAL TO ZEROS
            MOVE 1         TO WS-ERROR
            GO TO 030-ERROR-EN-TARJ
     ELSE
            MOVE "28"      TO WS-DIAS-FEB.
     IF     FECHA-MES      EQUAL TO ZEROS
       OR   FECHA-MES      GREATER 12
            MOVE 1         TO WS-ERROR
            GO TO 030-ERROR-EN-TARJ.
     IF     FECHA-ANO NOT EQUAL TO ZEROS
            DIVIDE 4 INTO FECHA-ANO
            GIVING        WS-COCIENTE
            REMAINDER     WS-RESIDUO
            IF WS-RESIDUO IS EQUAL TO ZEROS
               MOVE "29" TO WS-DIAS-FEB
            ELSE
```

Cont.

```
                NEXT SENTENCE
            ELSE
                MOVE 1 TO WS-ERROR
                GO TO 030-ERROR-EN-TARJ.
            IF  FECHA-DIA IS EQUAL TO ZEROS
                OR FECHA-DIA IS GREATER THAN
                WS-DIAS-TABLA(FECHA-MES)
                MOVE 1 TO WS-ERROR.
      * EN CASO DE EXISTIR ERROR EN ALGUNO DE LOS PARAMETROS DE CON-
      * TROL SE DEBERA GENERAR EL MENSAJE CORRESPONDIENTE.
        030-ERROR-EN-TARJ.
            IF  IDENTIFICACION NOT EQUAL TO "BPARUO1"
                OR WS-ERROR EQUAL TO 1
                DISPLAY "LA CLAVE UTILIZADA ES " IDENTIFICACION
                DISPLAY "LA FECHA ES " FECHA-TARJ
                WRITE LINEA FROM LINEA-ENCA1 AFTER PAGE
                WRITE LINEA FROM LINEA-ENCA2 AFTER ADVANCING 2
                WRITE LINEA FROM LINEA-ERROR AFTER ADVANCING 4
                GO TO 060-CERRAR.
            MOVE    MES(FECHA-MES) TO LINE-ABME.
            MOVE    FECHA-ANO      TO LINE-ABAN.
            WRITE LINEA FROM LINEA-ENCA1 AFTER PAGE.
            WRITE LINEA FROM LINEA-ENCA2 AFTER ADVANCING 1.
            WRITE LINEA FROM LINEA-ENCA3 AFTER ADVANCING 2.
            WRITE LINEA FROM LINEA-ENCA4 AFTER ADVANCING 2
            WRITE LINEA FROM LINEA-BLANCA AFTER ADVANCING 1.
        040-BORRA-TABLA.
            MOVE "TABTARIF" TO PARM-1.
            IF WS-CONTADOR EQUAL TO ZEROS
                FIND FIRST TPRO WITHIN TABTARIF
            ELSE
                FIND NEXT TPRO WITHIN TABTARIF.
```

Cont.

```
           IF DB-STATUS EQUAL TO ZEROS
              ADD 1 TO WS-CONTADOR
              GET TPRO.
           IF DB-STATUS NOT EQUAL TO ZEROS
              IF DB-STATUS EQUAL TO "0502100"
                 DISPLAY "REG. TPRO LEIDOS " WS-CONTADOR
                 DISPLAY "REG. BORRADOS " WS-BORRADOS
                 GO TO 050-PROCESO-SORT
              ELSE
                 DISPLAY "PROCESO CANCELADO"
                 DISPLAY "DB-STATUS " DB-STATUS
                 GO TO 060-CERRAR.
           IF TPRO-NUME-TABL GREATER THAN 620
              AND TPRO-NUME-TABL LESS THAN 641
              AND TPRO-CODI-DEPU EQUAL TO 1
                 DISPLAY TPRO
                 ERASE TPRO
              IF DB-STATUS NOT EQUAL TO ZEROS
                 DISPLAY "PROCESO CANCELADO"
                 DISPLAY "DB-STATUS" DB-STATUS
                 GO TO 060-CERRAR.
           GO TO 040-BORRA-TABLA.
050-PROCESO-SORT.
           MOVE ZEROS TO WS-CONTADOR WS-BORRADOS.
           SORT ARCH-SORT ON ASCENDING SORT-CLAV
                INPUT PROCEDURE IS 070-ENTRADA
                                THRU 090-SALIDA
                OUTPUT PROCEDURE IS 110-SALIDA
                                 THRU 130-FIN-SA.
060-CERRAR.
           MOVE WS-CONTADOR TO LINE-AFIT.
           WRITE LINEA FROM LINEA-FINAL AFTER ADVANCING 4 LINES
                 EOP PERFORM 130-TITULOS
```

Cont.

```
                THRU 160-EXIT.
        DISPLAY "TPRO GRABADOS " WS-CONTADOR
        CLOSE VERO-AMIS.
        CLOSE ARCH-LIST.
        FINISH TABTARIF.
        STOP RUN.
070-ENTRADA          SECTION.
080-ENTRA.
        READ VERO-AMIS AT END GO TO 090-FIN-ENT.
        MOVE VERO-AUTO-REV TO SORT-LLAV.
        RELEASE SORT-LLAV.
        GO TO 080-ENTRA.
090-FIN-ENT          SECTION.
100-FIN-E.
        EXIT.
110-SALIDA           SECTION.
        RETURN ARCH-SORT AT END PERFORM 130-DLT-LINEA
                                THRU 190-EXIT
                                GO TO 170-FIN-SA.
        ADD 1 TO WS-CONTADOR WS-CONTADOR1
        INITIALIZE TPRO
        MOVE SORT-LLAV TO TPRO-LLAV
        MOVE FECHA-HOY TO TPRO-DATA-VAR1.
        STORE TPRO.
        IF DB-STATUS NOT EQUAL TO ZEROS
            DISPLAY "PROCESO CANCELADO"
            DISPLAY "DB-STATUS " DB-STATUS
            GO TO 170-FIN-SA.
        IF WS-CONTADOR GREATER THAN 11
            MOVE 1 TO WS-CONTADOR 1
            WRITE LINEA FROM LINEA-DETALLE AFTER ADVANCING 1 LINES
                  EOP PERFORM 150-TITULOS
                  THRU 160-EXIT.
```

Cont.

```
           MOVE SPACES TO LINEA-DETALLE.
           MOVE SORT-CLAV TO LINE-ARAV(WS-CONTADOR1).
           MOVE SPACES    TO ESPACIO(WS-CONTADOR1).
           GO TO 120-SAC.
       130-ULT-LINEA.
           IF LINEA-DETALLE NOT EQUAL TO SPACES
              WRITE LINEA FROM LINEA-DETALLE
                    AFTER ADVANCING 1 LINES
              GO PERFORM 150-TITULOS
                         THRU 160-EXIT.
       140-EXIT.
           EXIT.
       150-TITULOS.
           ADD 1 TO WS-PAGINA.
           MOVE WS-PAGINA TO LINE-ALPA.
           WRITE LINEA FROM LINEA-ENCA1
           WRITE LINEA FROM                    2   AFTER ADVANCING PAGE.
           WRITE LINEA FROM                    3
           WRITE LINEA FROM                    4       1.
           WRITE LINEA FROM   -BLANCA          1       1.
       160-EXIT.                               2       2.
           EXIT.
       170-FIN-SA      SECTION.
       180-FINS.
           EXIT.
```

Cont.

```
0010     IDENTIFICATION     DIVISION.
0020     PROGRAM-ID.        BPARM01.
0030     AUTHOR.            CARL.
0040    *ESTE PROGRAMA ACTUALIZA LA TABLA DE AUTOS ROBADOS
0050    *UTILIZA LA INFORMACION PROPORCIONADA POR LA AMIS.
0060    *FECHA DE LA ULTIMA MODIFICACION.      8-OCT-1981.
0061    *---------------------------------------------------------------
0070     ENVIRONMENT        DIVISION.
0071    *---------------------------------------------------------------
0080     CONFIGURATION      SECTION.
0090     SOURCE-COMPUTER.   HIS-SERIES-60 LEVEL-66-ASCII.
0100     OBJECT-COMPUTER.   HIS-SERIES-60 LEVEL-66-ASCII.
0110     INPUT-OUTPUT       SECTION.
0120     FILE-CONTROL.
0130         SELECT VERO-AMIS ASSIGN TO Y1
0140                ACCESS MODE IS SEQUENTIAL
0150                ORGANIZATION GFRC  SEQUENTIAL WITH SSF.
0160         SELECT ARCH-LIST ASSIGN TO R1
0170                ORGANIZATION GFRC  SEQUENTIAL WITH SSF.
0180         SELECT ARCH-SORT ASSIGN TO S5.
0190     DATA               DIVISION.
0200     SUB-SCHEMA         SECTION.
0210     BD SSB9 WITHIN     SCHEMASISA.
0220     FILE               SECTION.
0230     FD                 VERO-AMIS
0240         LABEL RECORD IS STANDARD
0250         DATA RECORD IS VERO-AUTO
0260         CODE-SET IS GBCD.
0270     01                 VERO-AUTO.
0280       05               VERO-AUTO-COMP PIC X(02).
0290       05               VERO-AUTO-CLAV PIC X.
0300       05               VERO-AUTO-ESTA PIC X.
0310       05               VERO-AUTO-RFV  PIC 9(08).
0320       05               VERO-AUTO-MARC PIC X(03).
0330       05               VERO-AUTO-MODE PIC X(02).
0340       05               VERO-AUTO-NUSE PIC X(05).
0350     FD                 ARCH-LIST
0360         LABEL RECORD IS OMITTED
0370         CODE-SET   IS GBCD
0380         LINAGE         IS 59 LINES
0390         DATA RECORD IS LINEA.
0400     01                 LINEA          PIC X(132).
0410     SD                 ARCH-SORT.
0420     01                 SORT-LLAV      PIC 9(08).
0430     WORKING-STORAGE    SECTION.
0440         COPY DBWKPGGC.
0450     01                 LINEAS-ENCABEZADO.
0460       05               LINEA-ENCA1.
0470         10             FILLER         PIC X(06) VALUE SPACES.
0480         10             FILLER         PIC X(06) VALUE "S.I.S.A.".
0490         10             FILLER         PIC X(29) VALUE SPACES.
0500         10             FILLER         PIC X(51) VALUE
0510         10             FILLER         PIC X(18) VALUE SPACES.
0520         10             FILLER         PIC X(11) VALUE "PAGINA NO ".
0530         10             LINE-A1PA      PIC ZZ9   VALUE ZEROS.
```

```
0540         10              FILLER          PIC X(05) VALUE SPACES.
0550       05                LINEA-ENCA2.
0560         10              FILLER          PIC X(06) VALUE SPACES.
0570         10              FILLER          PIC X(16) VALUE
0580         "PROCESO  BPARM01".
0590         10              FILLER          PIC X(34) VALUE SPACES.
0600         10              FILLER          PIC X(23) VALUE
0610         "REPORTE AUTOS ROBADOS".
0620         10              FILLER          PIC X(33) VALUE SPACES.
0630         10              FILLER          PIC X(06) VALUE "FECHA".
0640         10              LINE-A2DI       PIC 99    VALUE ZEROS.
0650         10              FILLER          PIC X     VALUE "/".
0660         10              LINE-A2ME       PIC 99    VALUE ZEROS.
0670         10              FILLER          PIC X     VALUE "/".
0680         10              LINE-A2AN       PIC 99    VALUE ZEROS.
0690         10              FILLER          PIC X(06) VALUE SPACES.
0700       05                LINEA-ENCA3.
0710         10              FILLER          PIC X(50) VALUE SPACES.
0720         10              FILLER          PIC X(14) VALUE "DEL MES DE
0730         10              LINE-A3ME       PIC X(10) VALUE SPACES.
0740         10              FILLER          PIC X(08) VALUE "  DE 19".
0750         10              LINE-A3AN       PIC 99    VALUE ZEROS.
0760         10              FILLER          PIC X(50) VALUE SPACES.
0770       05                LINEA-ENCA4.
0780         10              FILLER          PIC X(46) VALUE SPACES.
0790         10              FILLER          PIC X(41) VALUE
0800         "- REGISTROS FEDERALES DE VEHICULOS -" .
0810         10              FILLER          PIC X(45) VALUE SPACES.
0820       05                LINEA-DETALLE.
0830         10              FILLER          PIC X(06) VALUE SPACES.
0840         10              AUTOMOVIL       OCCURS 11 TIMES.
0850           15            LINE-ADAU       PIC 9(08).
0860           15            ESPACIO         PIC X(03).
0870         10              FILLER          PIC X(05) VALUE SPACES.
0880       05                LINEA-FINAL.
0890         10              FILLER          PIC X(06) VALUE SPACES.
0900         10              FILLER          PIC X(46) VALUE
0910         "TOTAL DE AUTOS REPORTADOS POR LA AMIS     ".
0920         10              LINE-AFIT       PIC ZZZ,ZZZ9 VALUE ZEROS.
0930         10              FILLER          PIC X(73) VALUE SPACES.
0940       05                LINEA-ERROR.
0950         10              FILLER          PIC X(42) VALUE SPACES.
0960         10              FILLER          PIC X(48) VALUE
0970         "**** FAVOR DE VERIFICAR DATOS CONTROL ****" .
0980         10              FILLER          PIC X(42) VALUE SPACES.
0990       05                LINEA-BLANCA    PIC X(132)VALUE SPACES.
1000     01                  WS-CAMPOS-VARIOS.
1010       05                WS-CONTADOR     PIC 9(06) VALUE ZEROS.
1020       05                WS-CONTADOR1    PIC 99    VALUE ZEROS.
1030       05                WS-BORRADOS     PIC 9(06) VALUE ZEROS.
1040       05                WS-PAGINA       PIC 9(03) VALUE ZEROS.
1050       05                WS-ERROR        PIC 9     VALUE ZEROS.
1060       05                FECHA-HOY.
1070         10              FECHA-HANO      PIC 99    VALUE ZEROS.
1080         10              FECHA-HMES      PIC 99    VALUE ZEROS.
1090         10              FECHA-HDIA      PIC 99    VALUE ZEROS.
```

```
01100      05                CAMPOS-CONTROL.
01110        10              IDENTIFICACION PIC X(07) VALUE SPACES.
01120        10              FECHA-TARJ.
01130          15            FECHA-ANO      PIC 99    VALUE ZEROS.
01140          15            FECHA-MES      PIC 99    VALUE ZEROS.
01150          15            FECHA-DIA      PIC 99    VALUE ZEROS.
01160      05                TABLA-MESES.
01170        10              FILLER         PIC X(10) VALUE " ENERO ".
01180        10              FILLER         PIC X(10) VALUE " FEBRERO".
01190        10              FILLER         PIC X(10) VALUE " MARZO ".
01200        10              FILLER         PIC X(10) VALUE " ABRIL ".
01210        10              FILLER         PIC X(10) VALUE " MAYO ".
01220        10              FILLER         PIC X(10) VALUE " JUNIO ".
01230        10              FILLER         PIC X(10) VALUE " JULIO ".
01240        10              FILLER         PIC X(10) VALUE " AGOSTO".
01250        10              FILLER         PIC X(10) VALUE "SEPTIEMBRE".
01260        10              FILLER         PIC X(10) VALUE " OCTUBRE".
01270        10              FILLER         PIC X(10) VALUE "NOVIEMBRE".
01280        10              FILLER         PIC X(10) VALUE "DICIEMBRE".
01290      05                MESES-TABLA REDEFINES TABLA-MESES.
01300        10              MES            PIC X(10)OCCURS 12 TIMES.
01310      05                WS-TABLA-DIAS.
01320        10              FILLER         PIC XX    VALUE  31 .
01330        10              WS-DIAS-FEB    PIC XX.
01340        10              FILLER         PIC X(20) VALUE
01350         "31303130313130313031".
01360      05                WS-TAB-DIAS REDEFINES WS-TABLA-DIAS.
01370        10              WS-DIAS-TABLA  OCCURS 12 TIMES PIC 99.
01380      05                WS-DIVERSOS.
01390        10              WS-COCIENTE    PIC 99 COMP-4 VALUE ZEROS.
01400        10              WS-RESIDUO     PIC 99 COMP-4 VALUE ZEROS.
01410      05                LLAVE.
01420        10              CODI-USER      PIC 9     VALUE 1.
01430        10              NUME-TABL      PIC 9(05) VALUE 00528.
01440        10              NUME-RFV       PIC 9(10) VALUE ZEROS.
01450      05                TABLA-ANALIZADA.
01460        10              FILLER         PIC X.
01470        10              NUMERO         PIC 9(05).
01480        10              FILLER         PIC 9(10).
01481    *-----------------------------------------------------------*
01490    PROCEDURE           DIVISION.
01491    *-----------------------------------------------------------*
01500        COPY  DBDE9GGC.
01510    010-PROCESO-BPARM01  SECTION.
01520    020-PROCESO.
01530        OPEN   INPUT  VERO-AMIS
01540               OUTPUT ARCH-LIST.
01550        CALL "SSB9KE" USING W-PRIVACY.
01560        READY TABTAIF USAGE-MODE IS UPDATE.
01570        ACCEPT FECHA-HOY FROM DATE.
01580        ACCEPT CAMPOS-CONTROL.
01590        MOVE  FECHA-HANO  TO  LINE-A2AN.
01600        MOVE  FECHA-HMES  TO  LINE-A2ME.
01610        MOVE  FECHA-HDIA  TO  LINE-A2DI.
01620        ADD   1           TO  WS-PAGINA.
```

```
1630         MOVE  WS-PAGINA   TO  LINE-A1PA.
1640         IF    FECHA-TARJ NOT  NUMERIC
1650           OR  FECHA-TARJ EQUAL TO ZEROS
1660               MOVE  1      TO  WS-ERROR
1670               GO TO 030-ERROR-EN-TARJ
1680         ELSE
1690               MOVE "28"    TO  WS-DIAS-FEB.
1700         IF    FECHA-MES    EQUAL TO ZEROS
1710           OR  FECHA-MES    GREATER 12
1720                MOVE  1      TO  WS-ERROR
1730                GO TO 030-ERROR-EN-TARJ.
1740         IF FECHA-ANO  NOT EQUAL TO ZEROS
1750                DIVIDE 4  INTO  FECHA-ANO
1760                GIVING    WS-COCIENTE
1770                REMAINDER WS-RESIDUO
1780                IF WS-RESIDUO IS EQUAL TO ZEROS
1790                   MOVE "29" TO WS-DIAS-FEB
1800                ELSE
1810                   NEXT SENTENCE
1820         ELSE
1830                MOVE 1 TO WS-ERROR
1840                GO TO 030-ERROR-EN-TARJ.
1850         IF     FECHA-DIA  IS EQUAL TO ZEROS.
1860           OR   FECHA-DIA IS GREATER THAN
1870                WS-DIAS-TABLA (FECHA-MES)
1880                MOVE 1 TO WS-ERROR.
1881       * EN CASO DE EXISTIR ERROR EN ALGUNO DE LOS PARAMETROS DE CON-
1882       * TROL SE DEBERA GENERAR EL MENSAJE CORRESPONDIENTE.
1890        030-ERROR-EN-TARJ.
1900           IF    IDENTIFICACION NOT EQUAL TO "BPARM01"
1910             OR  WS-ERROR EQUAL TO 1
1920                 DISPLAY "LA CLAVE UTILIZADA ES" IDENTIFICACION
1930                 DISPLAY "LA FECHA ES          " FECHA-TARJ
1940                 WRITE   LINEA   FROM LINEA-ENCA1 AFTER PAGE
1950                 WRITE   LINEA   FROM LINEA-ENCA2 AFTER ADVANCING
1960                 WRITE   LINEA   FROM LINEA-ERROR AFTER ADVANCING 4
1970                 GO TO   060-CERRAR.
1980              MOVE   MES(FECHA-MES) TO   LINE-A3ME.
1990              MOVE   FECHA-ANO      TO   LINE-A3AN.
2000              WRITE  LINEA  FROM  LINEA-ENCA1 AFTER PAGE.
2010              WRITE  LINEA  FROM  LINEA-ENCA2 AFTER ADVANCING 1.
2020              WRITE  LINEA  FROM  LINEA-ENCA3 AFTER ADVANCING 1.
2030              WRITE  LINEA  FROM  LINEA-ENCA4 AFTER ADVANCING 2.
2040              WRITE  LINEA  FROM  LINEA-BLANCA AFTER ADVANCING 1.
2050        040-BORRA-TABLA.
2060            MOVE "TABTARIF" TO PARM-1.
2070            IF WS-CONTADOR EQUAL TO ZEROS.
2080               FIND FIRST TPRO WITHIN TABTARIF
2090            ELSE
2100               FIND NEXT  TPRO WITHIN TABTARIF.
2110            IF DB-STATUS EQUAL TO ZEROS
2120               ADD 1 TO WS-CONTADOR
2130               GET TPRO.
2140            IF DB-STATUS NOT EQUAL TO ZEROS
2150               IF DB-STATUS EQUAL TO "0502100"
```

```
02160                DISPLAY "REG. TPRO LEIDOS"  WS-CONTADOR
02170                DISPLAY "REG. -ORRADOS    "  WS-BORRADOS
02180                GO TO 050-PROCESO-SORT
02190             ELSE
02200                DISPLAY "PROCESO CANCELADO."
02210                DISPLAY "DB-STATUS  " DB-STATUS
02220                GO TO 060-CERRAR.
02230          IF TPRO-NUME-TABL EQUAL TO 528
02240             ERASE TPRO
02250             ADD 1 TO WS-BORRADOS
02260             IF DB-STATUS NOT EQUAL TO ZEROS
02270                DISPLAY "PROCESO CANCELADO"
02280                DISPLAY "EL DB-STATUS ES  " DB-STATUS
02290                GO TO 060-CERRAR
02300             ELSE
02310                GO TO 040-BORRA-TABLA.
02320          IF TPRO-NUME-TABL GREATER THAN 520
02330             AND TPRO-NUME-TABL LESS THAN 541
02340             AND TPRO-CODI-DEPU EQUAL TO 1
02350                 DISPLAY TPRO
02360                 ERASE TPRO
02370             IF DB-STATUS NOT EQUAL TO ZEROS
02380                DISPLAY "PROCESO CANCELADO"
02390                DISPLAY "EL DB-STATUS ES  " DB-STATUS
02400                GO TO 060-CERRAR.
02410          GO TO 040-BORRA-TABLA.
02420       050-PROCESO-SORT.
02430          MOVE ZEROS TO WS-CONTADOR WS-BORRADOS.
02440          SORT ARCH-SORT ON ASCENDING SORT-LLAV
02450               INPUT PROCEDURE IS 070-ENTRADA
02460                          THRU    090-FIN-ENT
02470             OUTPUT PROCEDURE IS 110-SALIDA
02480                          THRU     170-FIN-SA.
02490       060-CERRAR.
02500          MOVE WS-CONTADOR TO LINE-AFIT.
02510          WRITE LINEA FROM LINEA-FINAL AFTER ADVANCING 4 LINES
02520                EOP PERFORM 150-TITULOS
02530                       THRU 160-EXIT.
02540          DISPLAY "TPRO GRABADOS "  EX-BORRADOS
02550          CLOSE VERO-AMIS.
02560          CLOSE ARCH-LIST.
02570          FINISH TABTARIF
02580          STOP RUN.
02590       070-ENTRADA        SECTION.
02600       080-ENTRA.
02610          READ VERO-AMIS AT EN GO TO 090-FIN ENT.
02620          MOVE VERO-AUTO-RFV  TO  SORT-LLAV.
02630          RELEASE SORT-LLAV.
02640          GO TO 080-ENTRA.
02650       090-FIN-ENT        SECTION.
02660       100-FIN-E.
02670          EXIT.
02680       110-SALIDA         SECTION.
02690       120-SAL.
```

```
02700          RETURN ARCH-SORT AT END PERFORM 130-ULT-LINEA
02710                                      THRU 140-EXIT
02720                                   GO TO 170-FIN-SA.
02730          ADD 1 TO WS-CONTADOR  WS-CONTADOR1
02740          INITIALIZE TPRO
02750          MOVE SORT-LLAV TO NUME-RFV
02760          MOVE LLAVE TO  TPRO-LLAV
02770          MOVE FECHA-HOY TO TPRO-DATA-VARI.
02780          STORE TPRO.
02790          ADD 1 TO WS-BORRADOS
02800          IF DB-STATUS NOT EQUAL TO ZEROS
02810             DISPLAY "PROCESO CANCELADO"
02820             DISPLAY "DB-STATUS OBTENIDO ES "  DB-STATUS
02830             GO TO 170-FIN-SA.
02840          IF WS-CONTADOR1 GREATER THAN 11
02850             MOVE 1 TO WS-CONTADOR1
02860             WRITE LINEA  FROM LINEA-DETALLE AFTER ADVANCING 1 LINES
02870                   EOP PERFORM 150-TITULOS
02880                          THRU 160-EXIT
02890             MOVE SPACES TO LINEA-DETALLE.
02900          MOVE SORT-LLAV TO LINE-ADAU(WS-CONTADOR1) .
02910          MOVE SPACES    TO ESPACIO (WS-CONTADOR1).
02920          GO TO 120-SAL.
02930      130-ULT-LINEA.
02940          IF LINEA-DETALLE NOT EQUAL TO SPACES
02950             WRITE LINEA FROM LINEA-DETALLE
02960                   AFTER ADVANCING 1 LINES
02970                   EOP PERFORM 150-TITULOS
02980                          THRU 160-EXIT.
02990      140-EXIT.
03000          EXIT.
03010      150-TITULOS.
03020          ADD 1 TO WS-PAGINA.
03030          MOVE WS-PAGINA TO LINE-ALPA.
03040          WRITE LINEA  FROM LINEA-ENCA1 AFTER PAGE.
03050          WRITE LINEA  FROM LINEA-ENCA2 AFTER ADVANCING 1.
03060          WRITE LINEA  FROM LINEA-ENCA3 AFTER ADVANCING 1.
03070          WRITE LINEA  FROM LINEA-ENCA4 AFTER ADVANCING 1.
03080          WRITE LINEA  FROM LINEA-BLANCA AFTER ADVANCING 2.
03090      160-EXIT.
03100          EXIT.
03110      170-FIN-SA        SECTION.
03120      180-FINS.
03130          EXIT.
```

PROCESO, SPARNOT — REPORTE AUTOS ROBADOS — FECHA 10/10/81

DEL MES DE SEPTIEMBRE DE 1981

- REGISTROS FEDERALES DE VEHICULOS -

00014110	00031651	00116416	00116741	00125426	00126395	00127582	00130646	00135753	00135837	00135984
00136666	00140435	00142299	00144970	00147306	00147440	00159309	00166320	00169200	00171414	00172735
00172953	00181703	00184700	00188355	00195080	00195195	00197008	00198813	00200568	00203228	00205366
00206420	00208809	00210362	00213660	00214227	00216247	00219816	00219828	00224944	00227407	00228019
00228058	00228141	00228362	00231012	00231137	00232925	00233985	00236054	00236888	00237499	00237532
00237537	00237931	00238205	00242101	00242222	00244829	00244303	00245788	00246813	00248531	00248972
00249601	00249742	00250597	00253567	00253582	00254554	00254617	00254746	00255777	00257261	00259045
00259057	00259118	00261292	00261842	00264162	00264801	00266188	00266951	00267843	00267890	00269622
00272114	00272772	00272859	00272988	00274064	00276630	00277630	00279047	00280564	00281044	00281411
00281617	00283199	00284280	00284864	00286255	00288868	00290730	00290819	00291056	00292178	00292828
00295280	00295455	00295514	00297169	00298900	00298443	00301211	00301456	00301662	00302390	00303626
00303783	00304130	00305228	00306570	00307129	00308707	00308725	00308830	00308965	00309280	00309335
00310240	00311049	00311566	00312212	00313628	00314642	00316206	00317175	00317277	00318395	00318585
00318613	00318701	00318711	00318996	00319387	00319422	00320184	00320525	00320745	00320972	00321422
00322427	00322661	00322855	00323420	00323948	00325766	00326209	00326243	00326414	00326486	00327085
00332022	00332361	00333269	00333272	00333835	00333872	00333966	00334059	00334078	00334107	00334518
00334683	00334984	00334990	00335017	00336331	00337168	00337242	00338603	00338853	00339617	00340334
00340475	00340623	00342451	00343578	00345562	00345654	00345840	00345979	00346543	00346684	00346820
00348514	00348760	00349943	00351669	00351692	00352043	00352299	00352370	00352375	00352513	00352604
00352714	00352728	00352906	00352963	00352991	00352997	00353167	00353590	00354186	00354200	00354771
00357066	00357294	00358450	00359109	00359741	00359952	00360243	00360286	00360366	00361188	00363091
00363575	00363694	00364604	00364818	00365086	00365469	00365568	00365673	00365787	00365882	00365919
00366742	00368372	00369047	00369075	00369184	00369745	00370348	00370378	00371086	00371180	00371458
00371618	00371649	00371692	00379136	00372544	00372922	00373515	00374237	00374555	00375744	00376733
00377053	00377492	00379096	00380163	00380188	00380235	00380277	00380279	00380435	00380454	00380873
00380917	00382373	00384191	00384519	00384549	00384720	00385514	00385846	00385863	00386374	00387253
00387797	00388572	00388650	00388768	00389755	00389930	00389978	00390264	00390414	00390471	00391298
00392034	00392687	00392764	00392855	00392985	00393095	00393104	00393249	00393249	00393345	00393346
00393464	00394179	00394434	00394848	00396974	00397666	00397682	00397942	00397981	00397989	00398180
00398309	00398310	00398428	00398549	00398911	00398945	00399878	00400095	00400606	00401279	00401425
00401855	00403057	00403320	00403583	00403599	00403605	00403623	00403714	00403732	00403755	00403895
00404273	00404813	00404895	00405301	00405385	00406696	00406726	00406728	00406919	00406963	00407221
00407261	00407340	00407385	00407416	00407936	00408863	00409122	00410242	00410478	00410496	00410642
00410737	00410738	00410753	00410981	00411000	00411115	00411170	00411753	00411932	00412125	00412134
00412149	00412188	00412981	00413347	00413566	00413580	00413915	00414334	00414428	00414429	00414472
00414799	00414860	00415132	00415157	00415329	00415386	00415388	00415678	00415866	00416545	00418013
00418048	00418049	00418320	00419514	00419529	00420024	00421272	00421880	00422177	00422306	00422987
00423122	00423251	00423856	00424888	00424947	00424978	00425041	00425059	00425080	00425111	00425260

[redacted]

DEL MES DE SEPTIEMBRE DE 1981

- REGISTROS FEDERALES DE VEHICULOS -

FECHA 10/10/81

00425455	00425598	00425645	00425691	00425705	00425735	00425797	00425843	00425973	00426426	00427043
00427371	00427564	00427642	00427769	00428053	00428082	00428492	00428804	00429437	00429541	00429879
00430516	00430624	00430716	00432547	00433026	00433033	00433219	00433330	00433610	00433611	00433648
00433890	00434038	00434055	00434062	00434085	00434089	00434213	00434256	00434381	00434395	00434409
00434501	00434519	00434575	00434610	00434627	00434934	00434963	00435048	00435135	00435400	00435410
00435494	00435713	00435714	00435748	00435843	00436264	00436313	00436503	00436628	00438743	00438760
00439094	00439719	00439796	00440263	00440435	00440703	00440720	00440765	00441334	00441336	00442024
00442332	00442923	00443082	00443093	00443095	00443110	00443132	00443133	00443143	00443201	00443247
00443314	00443328	00443376	00443381	00443878	00444082	00445095	00445107	00445183	00445261	00445374
00445403	00445473	00445584	00445767	00446076	00446220	00446304	00446341	00446684	00447052	00447150
00447152	00447159	00447194	00447218	00447247	00447571	00447640	00447763	00447955	00448106	00448201
00448374	00448511	00448564	00448814	00449257	00449280	00449290	00449302	00449338	00449341	00449358
00449425	00449521	00449847	00449848	00449850	00449968	00450004	00450116	00450118	00450208	00450233
00451149	00451156	00451197	00451209	00451435	00451474	00451506	00451612	00451730	00452015	00452271
00452364	00452705	00453492	00454859	00454949	00455041	00455466	00455486	00455525	00455558	00455649
00455686	00455690	00455746	00455967	00455968	00456088	00456214	00456243	00456298	00456347	00456363
00456415	00456610	00456661	00456810	00456962	00456998	00457276	00457554	00457566	00457613	00457753
00457743	00458014	00458491	00458631	00458727	00458822	00459888	00460003	00460054	00460121	00460169
00460209	00460266	00460413	00460442	00460606	00460638	00461138	00461449	00461450	00461999	00462198
00462216	00462399	00462436	00462571	00462657	00462689	00462798	00462806	00463007	00463358	00463855
00464621	00464739	00464780	00464782	00465072	00465157	00465287	00465766	00465772	00465842	00465906
00465926	00465956	00465990	00465995	00465997	00466027	00466176	00466187	00466216	00466294	00466303
00466579	00466589	00467169	00467211	00467249	00467252	00467300	00467405	00467459	00467463	00467714
00467843	00467856	00467967	00468052	00468123	00468652	00468868	00469015	00469048	00469088	00469104
00469142	00469508	00469535	00469563	00469583	00469596	00469689	00469707	00469726	00469736	00469745
00469749	00469759	00469785	00469851	00469852	00469860	00469881	00469932	00469943	00469945	00469955
00469956	00469964	00469971	00469983	00470043	00470201	00470306	00470365	00470472	00470545	00470589
00470638	00470686	00470760	00470858	00470868	00471008	00471010	00471041	00471059	00471062	00471089
00471111	00471275	00471309	00471324	00471341	00472162	00472507	00473328	00473400	00473481	00473612
00473652	00473678	00473699	00474142	00474206	00474237	00474281	00474316	00474325	00474326	00474475
00474497	00474541	00474546	00474550	00474559	00474589	00474624	00474636	00474648	00474671	00474691
00474804	0047834	00474920	00474930	00474967	00475554	00475835	00476256	00476684	00476685	00476862
00477224	00477364	00477373	00477405	00477416	00477530	00477602	00477642	00477648	00477690	00477737
00477752	00477753	00477769	00477785	00477970	00478265	00478919	00478990	00479153	00479407	00479518
00479689	00479761	00479827	00479875	00480413	00480437	00480538	00480605	00480768	00481216	00481691
00481856	00481933	00482009	00482081	00483438	00483611	00483939	00484115	00485041	00485723	00485779
00485789	00485815	00485867	00485930	00485944	00485955	00485981	00485987	00486071	00486072	00486115
00486124	00486149	00486200	00486228	00486238	00486389	00486407	00486551	00486560	00486583	00486632
00486641	00486701	00486853	00486904	00486941	00486945	00486983	00486992	00487006	00487232	00487245

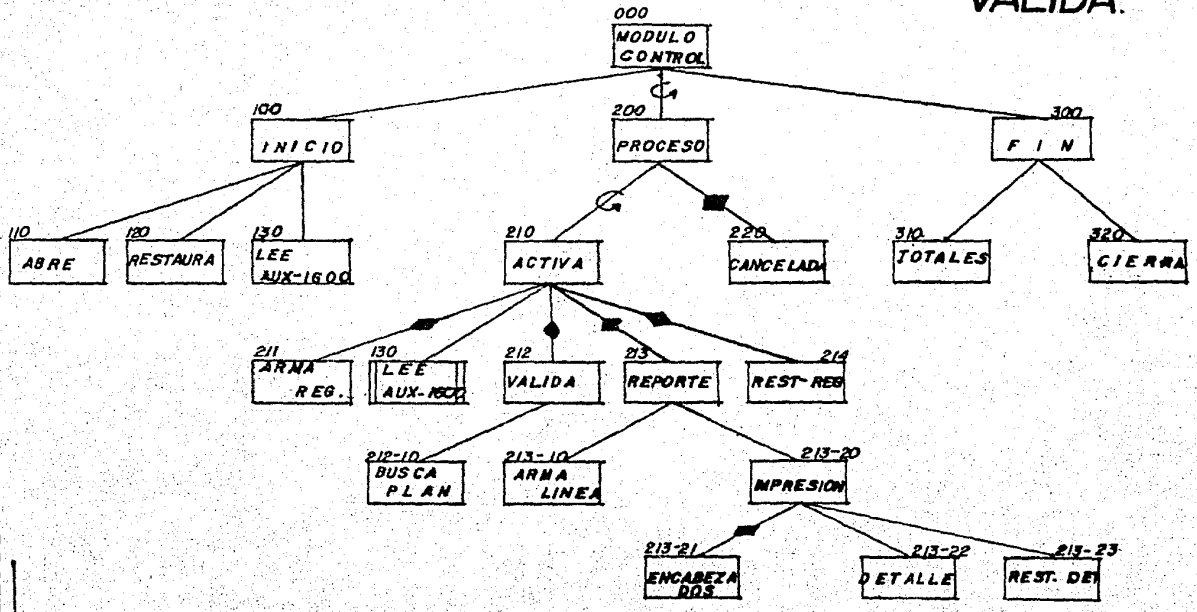
8.2. DESCRIPCION DEL PROBLEMA.

Uno de los problemas a los que se enfrenta cualquier compañía o institución, una vez que ha implantado nuevos sistemas de trabajo, como: incluir un computador en el desarrollo de las "viejas" actividades, es la conversión de lo ya existente a los nuevos estandares de procedimientos, almacenamiento etc. Más aun cuando los cambios intervienen con los insumos básicos de un proceso de cobranzas.

El ejemplo en cuestión valida los datos de un archivo empleado en un viejo sistema de cobranzas en una compañía de seguros, para contar con la información más confiable en la conversión de las pólizas de vida, contratadas antes de la automatización de la captura y generación de las pólizas de vida.

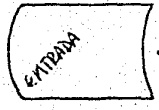
La característica del programa que se expone es la descripción y manejo de tablas indexadas.

VALIDA.


000
MODULO CONTROL
100
INICIO
200
PROCESO
300
FIN
110
ABRE
120
RESTAURA
130
LEE AUX-1600
210
ACTIVA
220
CANCELADA
310
TOTALES
320
CIERRA
211
ARMA REG.
130
LEE AUX-1600
212
VALIDA
213
REPORTE
214
REST-REG
212-10
BUSCA PLAN
213-10
ARMA LINEA
213-20
IMPRESION
213-21
ENCABEZADOS
213-22
DETALLE
213-23
REST. DET

TESIS	PROBLEMA 2	000	MODULO CONTROL.
DEPARTAMENTO	RESPONSABLE Carlos Alberto Ramirez Lazo.	NOTA	FECHA 20 | IX | 82

ENTRADA



PROCESO

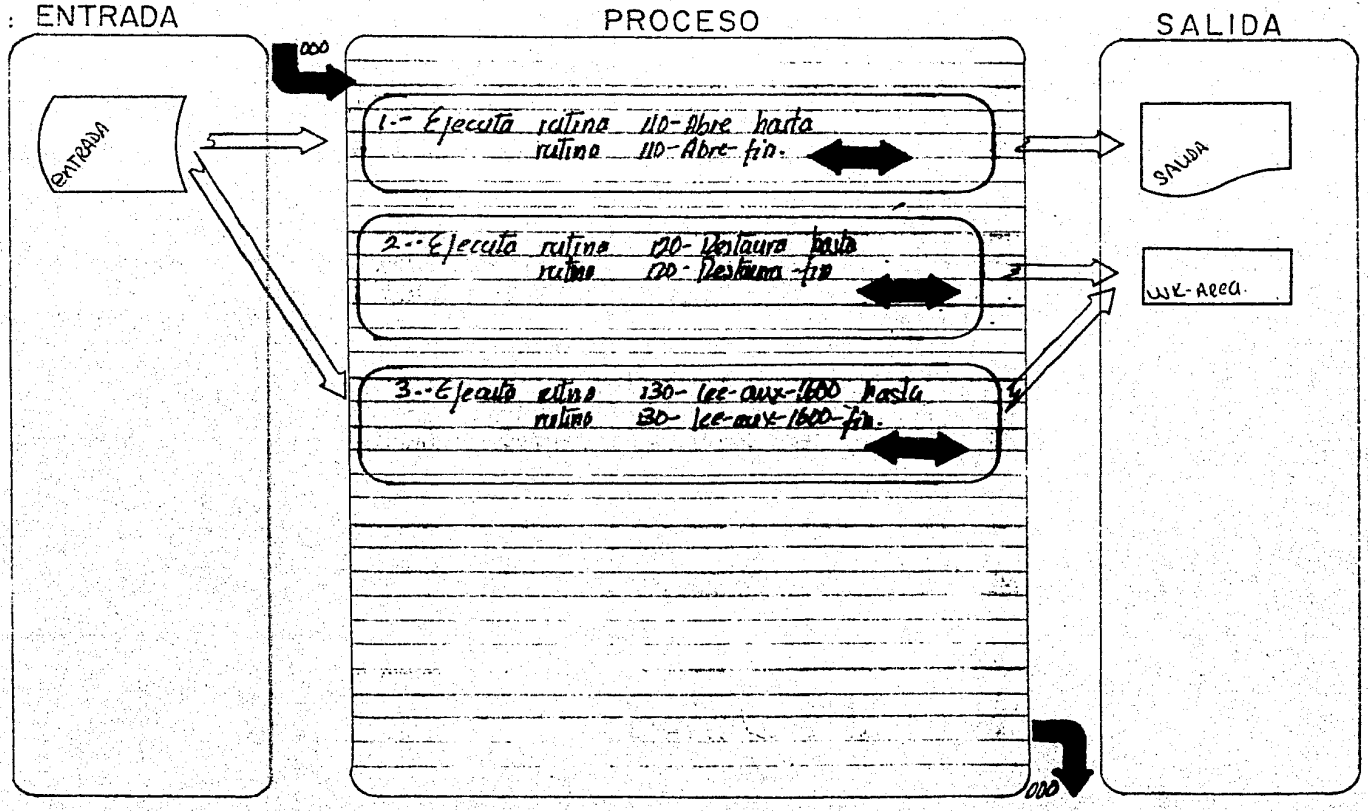
1.- Ejecutar rutina 100-INICIO hasta rutina 100-INICIO-FIN.

2.- Ejecutar rutina 200-PROCESO hasta rutina 200-PROCESO-FIN. hasta que no halla registros.

3.- Ejecutar rutina 300-FIN hasta salida 300-FIN-FIN.

SALIDA

DEPARTAMENTO	RESPONSABLE	NOTA	FECHA
TESIS PROBLEMA 2	Carlos Alberto Ramirez Lazo	IDD	INICIO 20 | IX | 82



TESIS	PROBLEMA-2	110	ABRE
DEPARTAMENTO	RESPONSABLE Carlos Alberto Ramírez Lazo	NOTA	FECHA 20 | IX | 82

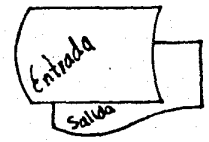
ENTRADA

PROCESO

SALIDA

100

1-- Abrir archivos; Entrada y Salida



100

DEPARTAMENTO	RESPONSABLE	NOTA	FECHA
PROBLEMA 2	Carlos Alberto Ramirez Lazo	120	20 | IX | 81

RESTRUIDA

ENTRADA

PROCESO

SALIDA

100

1.- Inicializa areas internas
(mover zeros a campos numéricos,
mover espacios a campos alfanumericos).

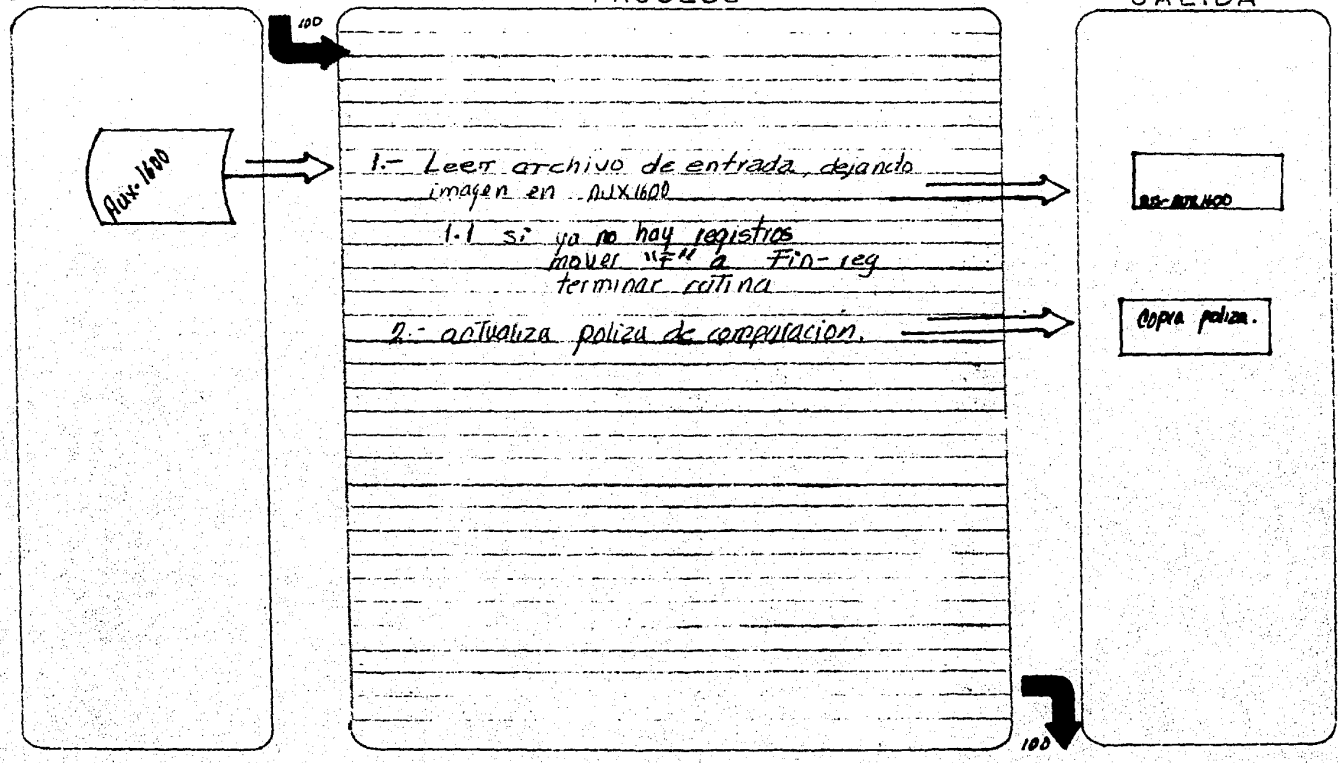
2.- Aceptar fecha del computador.

WK-AREA.

100

TESIS	PROBLEMA 2	130	Lee-AUX-1600
DEPARTAMENTO	RESPONSABLE Carlos Alberto Ramirez Lazo	NOTA	FECHA 20 / x / 81

ENTRADA | PROCESO | SALIDA



DEPARTAMENTO	RESPONSABLE	NOTA	200	PROCESO
	Carlos Alberto Ramirez Lazo.		FECHA	20 / 12 / 82

ENTRADA

PROCESO

SALIDA

000

1.- Si EDO-AUX DISTINTO A 04
ejecutar rutina 210-ACTUALIZA hasta
rutina 210-ACTUALIZA-FIN.

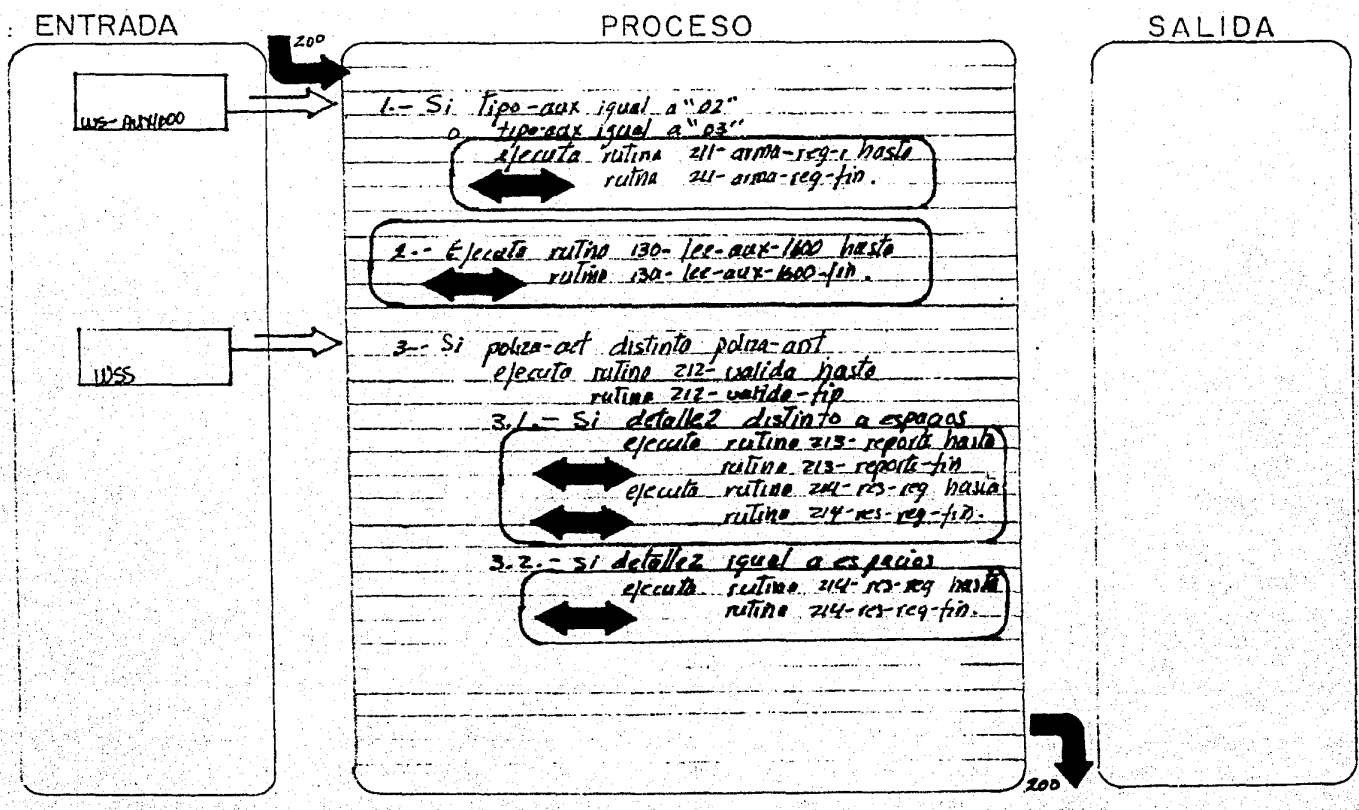
2.- Si EDO-AUX IGUAL A 04
ejecutar rutina 220- cancelada hasta
rutina 220- cancelada-fin.

WS-AREA

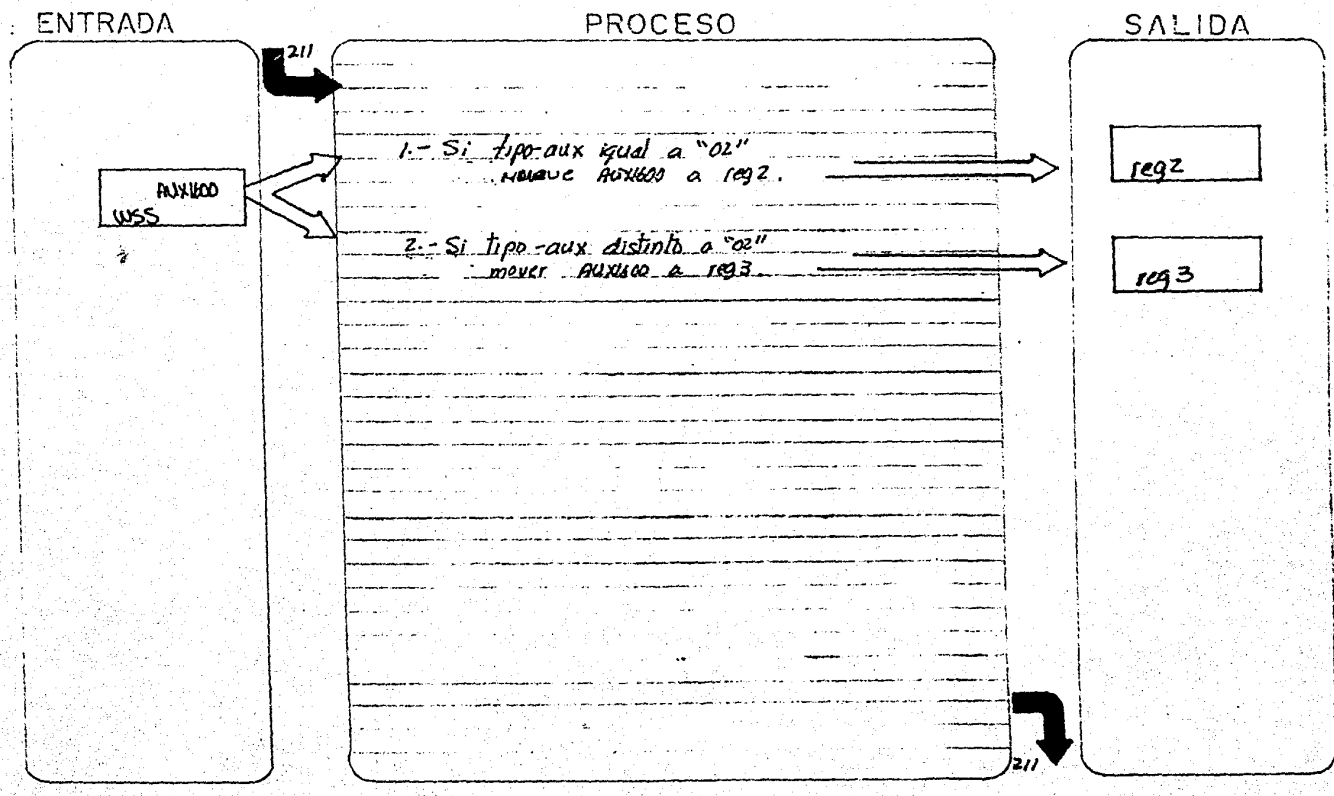
SALIDA

000

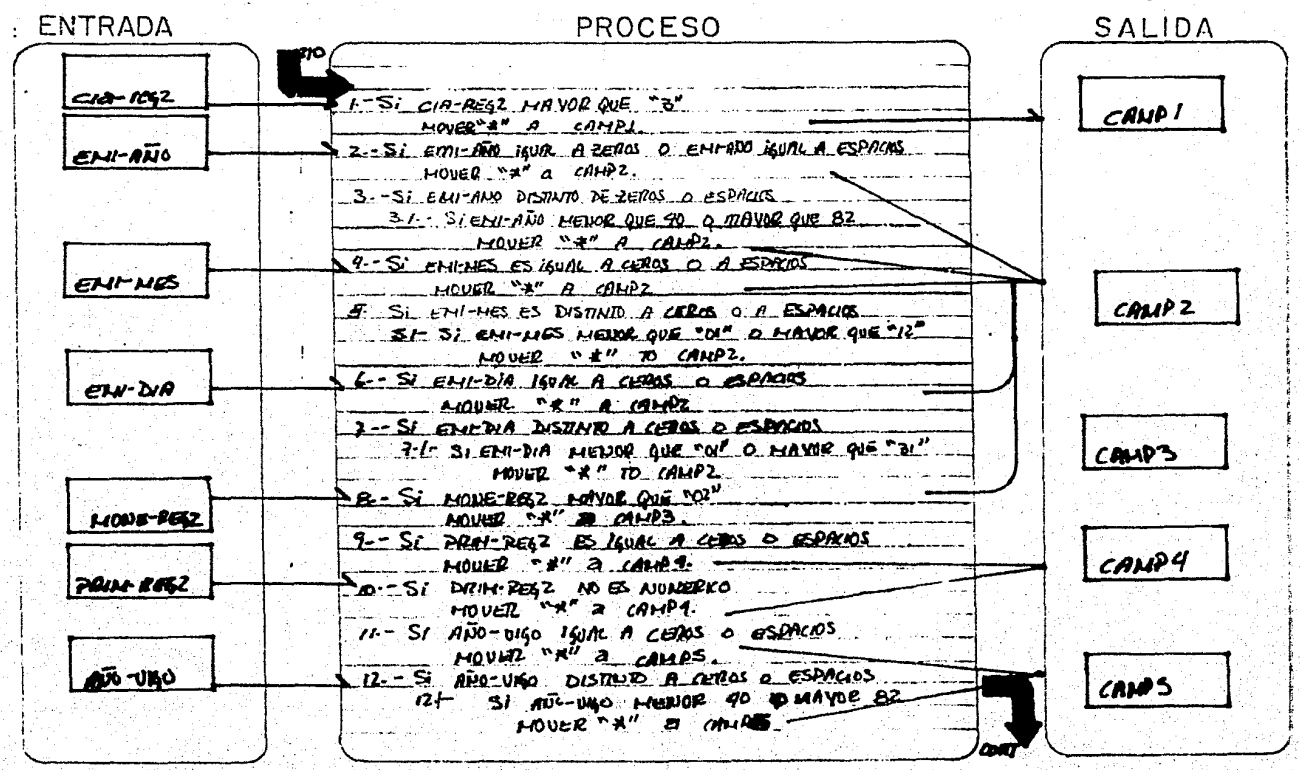
TESIS	PROBLEMA 2	210	ACTIVA
DEPARTAMENTO	RESPONSABLE Carlos Alberto Ramirez Lazo	NOTA	FECHA 20 IX 82



DEPARTAMENTO	RESPONSABLE	NOTA	FECHA
TESIS PROGRAMA 211 ARMA-REG	Carlos Alberto Ramirez Lazo		20 IX 81

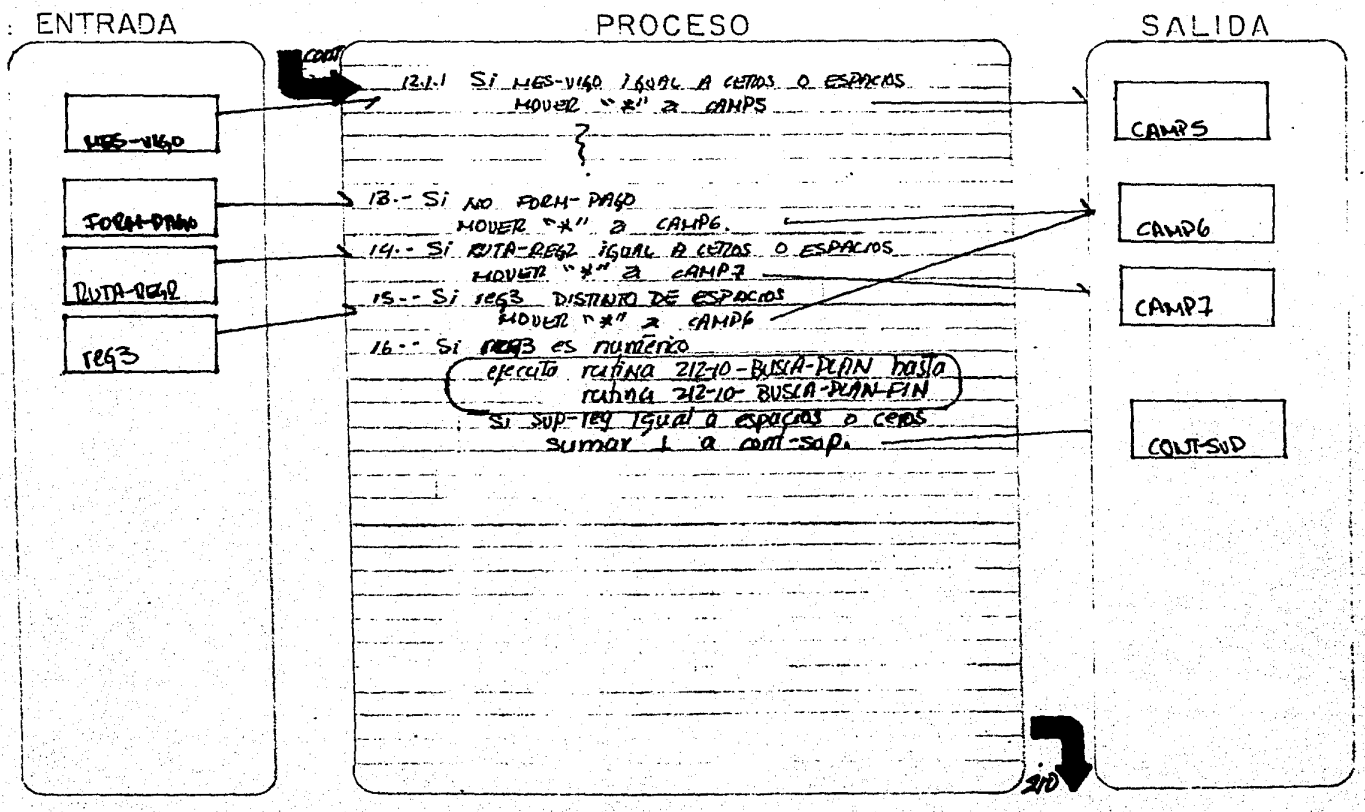


TESIS	PROBLEMA 2	212	VALIDO
DEPARTAMENTO	RESPONSABLE Carlos Alberto Ramírez Cozo	NOTA	FECHA 20 | IX | 82



TESIS | PROBLEMA 2 | 212 | VALIDA

DEPARTAMENTO	RESPONSABLE	NOTA	FECHA
	Carlos Alberto Ramírez Lazo	continuacion / aqui se manda la rutina con SEARCH	20 / x / 82



DEPARTAMENTO	RESPONSABLE	NOTA	FECHA
PLANS PROGRAMAS 212-10 BUSCA-PLAN	Carlos Alberto Ramirez Lozo	aqui se ocupa el search	20 IX 82

ENTRADA

"tabla"

PROCESO

0.- Inicializa indice en 1

1.- Busca Plan, variando indice
Si no hay plan ejecuta 212-11-ESO-HAY
cuando el plan (indice) sea igual a
plan-reg3 termina.

SALIDA

indice

DEPARTAMENTO	RESPONSABLE	NOTA	FECHA
TESIS PROBLEMA 2	Carlos Alberto Ramirez Lazo	300 FIN	20 IX '92

ENTRADA

PROCESO

000

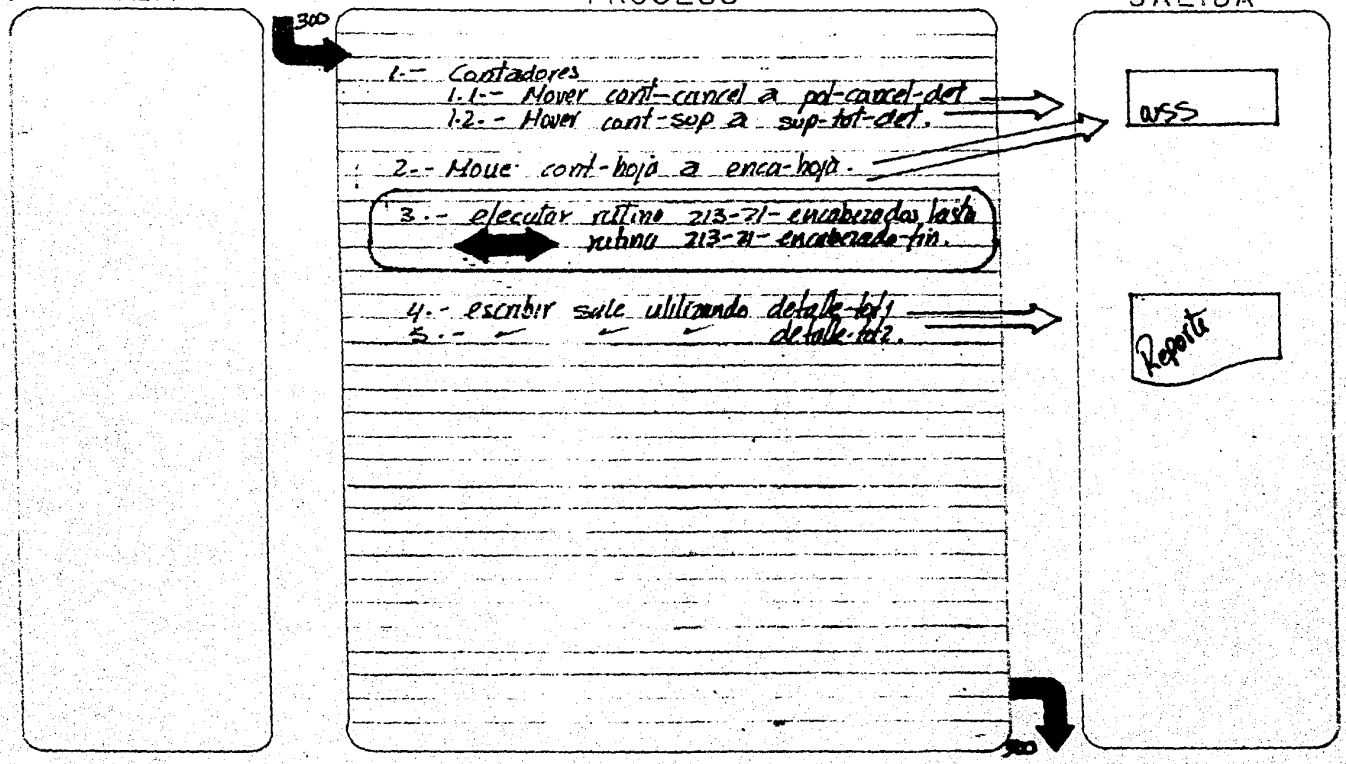
1.- Ejecuta rutina 310-totales hasta rutina 310-totales-fin.

2.- Ejecuta rutina 320-Cierre hasta rutina 310-Cierre-fin.

SALIDA

000

SISTEMA	SUBSISTEMA	NIVEL	IDENTIFICACION
TESIS	PROBLEMA 2	310	TOTALES
DEPARTAMENTO	**RESPONSABLE**	**NOTA**	**FECHA**
	Carlos Alberto Ramirez Lazo		20 'IX '81



TESIS	PROBIENAZ	320	CIERRA
DEPARTAMENTO	RESPONSABLE: Carlos Alberto Ramírez Lazo	NOTA: ultimo mitin	FECHA: 20 IX 82

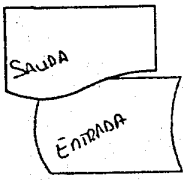
ENTRADA

PROCESO

SALIDA

300

1.- Cerrar archivos de entrada y Salida.



303

```
IDENTIFICATION DIVISION.
PROGRAM-ID. VALIDA.
AUTHOR. CARLOS RANIREZ.
*****************************************************
* ESTE PROGRAMA EMITE UN REPORTE DE LA VALIDACION DE LOS *
* DATOS DEL ARCHIVO AUX-1600 CON EL FIN DE PREPARAR LOS *
* DETALLES PARA LA CONVERSION DE LAS POLIZAS DE VIDA. *
*****************************************************
INSTALLATION.
*-------------------------------------------------------*
ENVIRONMENT DIVISION.
*-------------------------------------------------------*
CONFIGURATION SECTION.
SOURCE-COMPUTER. 600-WITH EIS.
OBJECT-COMPUTER. 600-WITH EIS.
SPECIAL-NAMES.
COMPILE ERRORS.
SETLNE IS DATER-600.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT ENTRADA ASSIGN TO T1.
    SELECT SALIDA ASSIGN TO T4 FOR LISTING.
I-O-CONTROL.
    APPLY SYSTEM STANDARD ON
    ENTRADA,
    SALIDA.
```

```
*-----------------------------------------------------------*
DATA                    DIVISION.
*-----------------------------------------------------------*
FD  ENTRADA
    LABEL RECORD STANDARD
    VALUE OF ID "1600-AUX".
01  ENTRA                  PIC X(286).
FD  SALIDA
    LABEL RECORD STANDARD.
01  SALE                   PIC X(136).
WORKING-STORAGE   SECTION.
*-----------------------------------------------------------*
* CONTADORES SENALES Y ACUMULADORES                         *
*-----------------------------------------------------------*
01  CONTADORES.
    05 CONT-LIN   PIC 99 VALUE 51.
    05 CONT-HOJA PIC 9(4) VALUE ZEROES.
01  CONTADORES-CONTROL.
    05 CONT-SUP          PIC 9(6).
    05 CONT-CANCEL       PIC 9(6).
01  CAMPOS.
    05 POL-ACT           PIC X(6).
    05 POL-ANT           PIC X(6).
01  SWITCHES.
    05 FIN-SEG           PIC X.
01  AUX1600.
    05 FILLER            PIC X(7).
    05 TIA-AUX           PIC X.
    05 POL-AUX           PIC X(6).
    05 FILLER            PIC X(6).
    05 TIPO-AUX          PIC X(2).
    05 FILLER            PIC X(193).
    05 EDO-AUX           PIC X(2).
```

Cont.

```
05  FILLER                  PIC X(75).
01  FECHA.
    05  MES-FECH           PIC 99.
    05  DIA-FECH           PIC 99.
    05  ANO-FECH           PIC 99.
    05  FILLER             PIC X(6).
01  REG2.
    ~
01  REG3.
    ~
01  DETALLE1.
    ~
01  DETALLE-TOT1.
    ~
01  DETALLE-TOT2.
    ~
```

Cont.

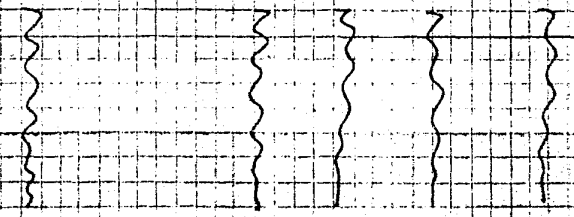
```
PROCEDURE                    DIVISION.
*-------------------------------------------------------------*
000-MODULO-CONTROL.
*-------------------------------------------------------------*
    PERFORM 100-INICIO  THRU 100-INICIO-FIN.
    PERFORM 200-PROCESO THRU 200-PROCESO-FIN
      UNTIL FIN-REG EQUAL "F".
    PERFORM 300-FIN     THRU 300-FIN-FIN.
    STOP RUN.
000-MODULO-CONTROL-FIN.
    EXIT.
*-------------------------------------------------------------*
100-INICIO.
*-------------------------------------------------------------*
    PERFORM 110-ABRE          THRU 110-ABRE-FIN.
    PERFORM 120-RESTAURA      THRU 120-RESTAURA-FIN.
    PERFORM 130-LEE-AUX-1600  THRU 130-LEE-AUX-1600-FIN.
100-FINICIO-FIN.
    EXIT.
110-ABRE.
    OPEN INPUT  ENTRADA
         OUTPUT SALIDA.
110-ABRE-FIN.
    EXIT.
120-RESTAURA.
    MOVE SPACES TO FIN-REG
                   CAMPOS
                   REG2
                   REG3
                   DETALLE1
                   DETALLE2
```

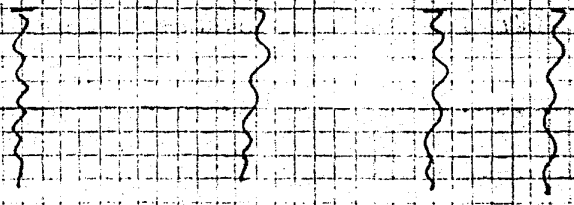
Cont.

PAG. (SERIE)

```
01  TABLA.
    02  FI    PIC XXX VALUE "010".
    02  FI    PIC XXX VALUE "011".
    02  FI    PIC XXX VALUE "012".
    02  FI    PIC XXX VALUE "013".
    02  FI    PIC XXX VALUE "014".

01  RETABLA  REDEFINES TABLA.
    02 TAB-PLAN OCCURS 271 INDEXED BY IND-TAB.
       03  EL-PLAN PIC XXX.
```



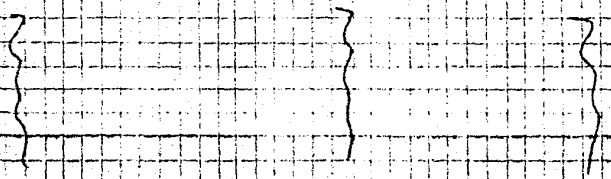


Cont.

```
                                SWITCHES
                                AUX1600.
           MOVE ZEROES TO CONTODORES-CONTROL.
           ACCEPT FECHA FROM DATER-600.
       120-REGTAURA-FIN.
           EXIT.
       130-LEE-AUX-1600.
           READ ENTRADA INTO AUX1600
                AT END MOVE "F" TO FIN-RES
                       MOVE ALL "2" TO POL-AUX.
           MOVE POL-AUX TO POL-ACT.
       130-LEE-AUX-1600-FIN.
           EXIT.
       200-PROCESO.
           IF EDO-AUX NOT EQUAL "04"
                MOVE POL-ACT TO POL-ANT
                PERFORM 210-ACTIVA THRU
                        210-ACTIVA-FIN
                    UNTIL POL-ACT NOT EQUAL POL-ANT
           ELSE
                PERFORM 220-CANCELADAS THRU
                        220-CANCELADA-FIN.
       200-PROCESO-FIN.
           EXIT.
```

Cont.

```
212-ID-BUSCA-PLAN.
    SET IND-TAB TO 1.
    SEARCH TAB-PLAN VARYING IND-TAB AT END GO TO 212-ID-NO-HAY
    WHEN EL-PLAN(IND-TAB) EQUAL TO PLAN-REG3 NEXT SENTENCE.
212-ID-BUSCA-PLAN-FIN.
    EXIT.
```



```
320-CIERRA.
    CLOSE ENTRADA SALIDA.
320-CIERRA-FIN.
    EXIT.
```

Cont.

```
0010:IDENTIFICATION DIVISION.
0020:PROGRAMA-ID.      VALIDA.
0030:AUTHOR.           CARLOS RAMIREZ
0040       *************************************************************
0050       * ESTE PROGRAMA EMITE UN REPORTE DE LA VALIDACION DE LOS    *
0060       * DATOS DEL ARCHIVO AUX-1600 CON EL FIN DE PREPARAR LOS     *
0070       * DETALLES PARA LA CONVERSION DE LAS POLIZAS DE VIDA        *
0080       *************************************************************
0090:INSTALLATION.
0100       *-----------------------------------------------------------*
0110:ENVIRONMENT DIVISION.
0120       *-----------------------------------------------------------*
0130:CONFIGURATION SECTION.
0140.SOURCE-COMPUTER. 6000 WITH EIS.
0150:OBJECT-COMPUTER. 6000 WITH EIS.
0160.SPECIAL-NAMES.
0170.COMPILE ERRORS.
0180.GETIME IS DATER-600.
0190.INPUT-OUTPUT SECTION.
0200.FILE-CONTROL.
0210.     SELECT ENTRADA ASSIGN TO T1.
0220.     SELECT SALIDA  ASSIGN TO T4 FOR LISTING.
0230.I-O-CONTROL.
0240.     APPLY SYSTEM STANDARD ON
0250.     ENTRADA
0260.     SALIDA.
0270       *-----------------------------------------------------------*
0280.DATA DIVISION.
0290       *-----------------------------------------------------------*
0300.FILE SECTION.
0310.FD   ENTRADA
0320.     LABEL RECORD STANDARD
0330.     VALUE OF ID "AUX-1600".
0340.01.ENTRA                PIC X(244).
0350.FD   SALIDA
0360.     LABEL RECORD STANDARD.
0370.01.SALE                 PIC X(136).
0380.WORKING-STORAGE SECTION.
0390       *-----------------------------------------------------------*
0400       * CONTADORES SENALES Y ACUMULADORES
0410       *-----------------------------------------------------------*
0420.01   CONTADORES.
0430.     05 CONT-LIN         PIC 99 VALUE 51.
0440.     05 CONT-HOJA        PIC 9(4) VALUE ZEROES.
0450.01   CONTADORES-CONTROL.
0460.     05 CONT-SUP         PIC 9(6).
0470.     05 CONT-CANCEL      PIC 9(6).
0480.01   CAMPOS.
0490.     05  POL-ACT         PIC X(6).
0500.     05  POL-ANT         PIC X(6).
0510.01   SWITCHES.
0520.     05  FIN-REG         PIC X.
0530.01   AUX1600.
0540.     05 FILLER           PIC X(7).
0550.     05 CIA-AUX          PIC X.
```

```
0560,     05 POL-AUX          PIC X(6),
0570,     05 FILLER           PIC X(6),
0580,     05 TIPO-AUX         PIC X(2),
0590,     05 FILLER           PIC X(145),
0600,     05 EDO-AUX          PIC X(2),
0610,     05 FILLER           PIC X(75),
0620,01   FECHA,
0630,     05 MES-FECH         PIC 99,
0640,     05 DIA-FECH         PIC 99,
0650,     05 ANO-FECH         PIC 99,
0660,     05 FILLER           PIC X(6),
0670,01   REG2,
0680,     05 FILLER           PIC X(7),
0690,     05 CIA-REG2         PIC X,
0700,     05 POL-REG2         PIC X(6),
0710,     05 EMI-REG2,
0720,        10 EMI-ANO       PIC X(2),
0730,        10 EMI-MES       PIC X(2),
0740,        10 EMI-DIA       PIC X(2),
0750,     05 TIPO-REG2        PIC X(2),
0760,     05 FILLER           PIC X,
0770,     05 MONE-REG2        PIC X(2),
0780,     05 FILLER           PIC X(62),
0790,     05 PRIM-REG2        PIC X(10),
0800,     05 FILLER           PIC X(6),
0810,     05 VIGO-REG2,
0820,        10 ANO-VIGO      PIC X(2),
0830,        10 MES-VIGO      PIC X(2),
0840,        10 DIA-VIGO      PIC X(2),
0850,     05 FILLER           PIC X(6),
0860,     05 PAGO-REG2        PIC X(2),
0870,        88 FORM-PAGO     VALUES "01" "03" "06" "12"
0880,                                "81" "83" "86" "82",
0890,     05 FILLER           PIC X(71),
0900,     05 RUTA-REG2        PIC X(5),
0910,     05 FILLER           PIC X(51),
0920,01   REG3,
0930,     05 FILLER           PIC X(7),
0940,     05 CIA-REG3         PIC X,
0950,     05 POL-REG3         PIC X(6),
0960,     05 FECH-REG3        PIC X(6),
0970,     05 TIPO-REG3        PIC X(2),
0980,     05 FILLER           PIC X(65),
0990,     05 COM-REG3         PIC X(10),
1000,     05 FILLER           PIC X(23),
1010,     05 AGTE-REG3        PIC X(5),
1020,     05 FILLER           PIC X(45),
1030,     05 PLAN-REG3        PIC X(3),
1040,     05 SUP-REG3         PIC X(5),
1050,     05 FILLER           PIC X(66),
1060,01   DETALLE1,
1070,     05 FILLER  PIC X(5),
1080,     05 CIA-DET          PIC X,
1090,     05 FILLER           PIC X(9),
1100,     05 POL-DET          PIC X(6),
```

```
1110.    05 FILLER              PIC X(5).
1120.    05 EMI-DET             PIC X(6).
1130.    05 FILLER              PIC X(9).
1140.    05 MONE-DET            PIC X(2).
1150.    05 FILLER              PIC X(6).
1160.    05 PRIM-DET            PIC X(10).
1170.    05 FILLER              PIC X(4).
1180.    05 VIGO-DET            PIC X(6).
1190.    05 FILLER              PIC X(8).
1200.    05 PAGO-DET            PIC X(2).
1210.    05 FILLER              PIC X(8).
1220.    05 RUTA-DET            PIC X(5).
1230.    05 FILLER              PIC X(5).
1240.    05 COM-DET             PIC X(10).
1250.    05 FILLER              PIC X(4).
1260.    05 AGTE-DET            PIC X(5).
1270.    05 FILLER              PIC X(8).
1280.    05 PLAN-DET            PIC X(3).
1290.    05 FILLER              PIC X(9).
1300.01 DETALLE2.
1310.    05 FILLER              PIC X(4).
1320.    05 CAMP1               PIC X.
1330.    05 FILLER              PIC X(23).
1340.    05 CAMP2               PIC X.
1350.    05 FILLER              PIC X(11).
1360.    05 CAMP3               PIC X.
1370.    05 FILLER              PIC X(11).
1380.    05 CAMP4               PIC X.
1390.    05 FILLER              PIC X(11).
1400.    05 CAMP5               PIC X.
1410.    05 FILLER              PIC X(11).
1420.    05 CAMP6               PIC X.
1430.    05 FILLER              PIC X(11).
1440.    05 CAMP7               PIC X.
1450.    05 FILLER              PIC X(12).
1460.    05 CAMP8               PIC X.
1470.    05 FILLER              PIC X(10).
1480.    05 CAMP9               PIC X.
1490.    05 FILLER              PIC X(11).
1500.    05 CAMP10              PIC X.
1510.    05 FILLER              PIC X(6).
1520.01 DETALLE-TOT1.
1530.    05 FILLER              PIC X(32) VALUE SPACES.
1540.    05 FILLER              PIC X(21) VALUE "POLIZAS CANCELADAS=".
1550.    05 POL-CANCEL-DET      PIC 9(8)  VALUE ZEROES.
1560.    05 FILLER              PIC X(75) VALUE SPACES.
1570.01 DETALLE-TOT2.
1580.    05 FILLER              PIC X(38) VALUE SPACES.
1590.    05 FILLER              PIC X(15) VALUE "SUPERVISORES=" .
1600.    05 SUP-TOT-DET  PIC 9(8) VALUE ZEROES.
1610.    05 FILLER              PIC X(75) VALUE SPACES.
1620.01 TABLA.
1630.    02 F1                  PIC XXX VALUE "010".
1640.    02 F1                  PIC XXX VALUE "011".
```

```
1650.    02 FI                    PIC XXX VALUE "012".
1660.    02 FI                    PIC XXX VALUE "013".
1670.    02 FI                    PIC XXX VALUE "014".
1680.    02 FI                    PIC XXX VALUE "015".
1690.    02 FI                    PIC XXX VALUE "016".
1700.    02 FI                    PIC XXX VALUE "017".
1710.    02 FI                    PIC XXX VALUE "018".
1720.    02 FI                    PIC XXX VALUE "019".
1730.    02 FI                    PIC XXX VALUE "020".
1740.    02 FI                    PIC XXX VALUE "025".
1750.    02 FI                    PIC XXX VALUE "065".
1760.    02 FI                    PIC XXX VALUE "100".
1770.    02 FI                    PIC XXX VALUE "101".
1780.    02 FI                    PIC XXX VALUE "120".
1790.    02 FI                    PIC XXX VALUE "127".
1800.    02 FI                    PIC XXX VALUE "128".
1810.    02 FI                    PIC XXX VALUE "129".
1820.    02 FI                    PIC XXX VALUE "130".
1830.    02 FI                    PIC XXX VALUE "131".
1840.    02 FI                    PIC XXX VALUE "140".
1850.    02 FI                    PIC XXX VALUE "141".
1860.    02 FI                    PIC XXX VALUE "142".
1870.    02 FI                    PIC XXX VALUE "145".
1880.    02 FI                    PIC XXX VALUE "154".
1890.    02 FI                    PIC XXX VALUE "160".
1900.    02 FI                    PIC XXX VALUE "161".
1910.    02 FI                    PIC XXX VALUE "162".
1920.    02 FI                    PIC XXX VALUE "163".
1930.    02 FI                    PIC XXX VALUE "164".
1940.    02 FI                    PIC XXX VALUE "165".
1950.    02 FI                    PIC XXX VALUE "166".
1960.    02 FI                    PIC XXX VALUE "167".
1970.    02 FI                    PIC XXX VALUE "168".
1980.    02 FI                    PIC XXX VALUE "169".
1990.    02 FI                    PIC XXX VALUE "170".
2000.    02 FI                    PIC XXX VALUE "171".
2010.    02 FI                    PIC XXX VALUE "172".
2020.    02 FI                    PIC XXX VALUE "173".
2030     02 FI                    PIC XXX VALUE "175".
2040     02 FI                    PIC XXX VALUE "176".
2050     02 FI                    PIC XXX VALUE "181".
2060     02 FI                    PIC XXX VALUE "182".
2070     02 FI                    PIC XXX VALUE "183".
2080     02 FI                    PIC XXX VALUE "184".
2090     02 FI                    PIC XXX VALUE "190".
2100     02 FI                    PIC XXX VALUE "191".
2110     02 FI                    PIC XXX VALUE "192".
2120     02 FI                    PIC XXX VALUE "193".
2130     02 FI                    PIC XXX VALUE "212".
2140     02 FI                    PIC XXX VALUE "213".
2150     02 FI                    PIC XXX VALUE "214".
2160     02 FI                    PIC XXX VALUE "215".
2170     02 FI                    PIC XXX VALUE "240".
2180     02 FI                    PIC XXX VALUE "281".
```

```
2190.    02 F1                    PIC XXX VALUE "282".
2200.    02 F1                    PIC XXX VALUE "283".
2210.    02 F1                    PIC XXX VALUE "284".
2220.    02 F1                    PIC XXX VALUE "290".
2230.    02 F1                    PIC XXX VALUE "291".
2240.    02 F1                    PIC XXX VALUE "303".
2250.    02 F1                    PIC XXX VALUE "305".
2260.    02 F1                    PIC XXX VALUE "306".
2270.    02 F1                    PIC XXX VALUE "307".
2280.    02 F1                    PIC XXX VALUE "308".
2290.    02 F1                    PIC XXX VALUE "309".
2300.    02 F1                    PIC XXX VALUE "313".
2310.    02 F1                    PIC XXX VALUE "314".
2320.    02 F1                    PIC XXX VALUE "315".
2330.    02 F1                    PIC XXX VALUE "317".
2340.    02 F1                    PIC XXX VALUE "318".
2350.    02 F1                    PIC XXX VALUE "321".
2360.    02 F1                    PIC XXX VALUE "323".
2370.    02 F1                    PIC XXX VALUE "324".
2380.    02 F1                    PIC XXX VALUE "325".
2390.    02 F1                    PIC XXX VALUE "331".
2400.    02 F1                    PIC XXX VALUE "332".
2410.    02 F1                    PIC XXX VALUE "333".
2420.    02 F1                    PIC XXX VALUE "334".
2430.    02 F1                    PIC XXX VALUE "335".
2440.    02 F1                    PIC XXX VALUE "341".
2450.    02 F1                    PIC XXX VALUE "342".
2460.    02 F1                    PIC XXX VALUE "343".
2470.    02 F1                    PIC XXX VALUE "344".
2480.    02 F1                    PIC XXX VALUE "352".
2490.    02 F1                    PIC XXX VALUE "353".
2500.    02 F1                    PIC XXX VALUE "354".
2510.    02 F1                    PIC XXX VALUE "355".
2520.    02 F1                    PIC XXX VALUE "361".
2530.    02 F1                    PIC XXX VALUE "363".
2540.    02 F1                    PIC XXX VALUE "367".
2550.    02 F1                    PIC XXX VALUE "370".
2560.    02 F1                    PIC XXX VALUE "373".
2570.    02 F1                    PIC XXX VALUE "377".
2580.    02 F1                    PIC XXX VALUE "380".
2590.    02 F1                    PIC XXX VALUE "381".
2600.    02 F1                    PIC XXX VALUE "382".
2610.    02 F1                    PIC XXX VALUE "383".
2620.    02 F1                    PIC XXX VALUE "384".
2630.    02 F1                    PIC XXX VALUE "385".
2640.    02 F1                    PIC XXX VALUE "386".
2650.    02 F1                    PIC XXX VALUE "391".
2660.    02 F1                    PIC XXX VALUE "392".
2670.    02 F1                    PIC XXX VALUE "403".
2680.    02 F1                    PIC XXX VALUE "404".
2690.    02 F1                    PIC XXX VALUE "405".
2700.    02 F1                    PIC XXX VALUE "409".
2710.    02 F1                    PIC XXX VALUE "410".
2720.    02 F1                    PIC XXX VALUE "411".
```

```
2730.    02 F1                PIC XXX VALUE "412".
2740.    02 F1                PIC XXX VALUE "413".
2750.    02 F1                PIC XXX VALUE "414".
2760.    02 F1                PIC XXX VALUE "415".
2770.    02 F1                PIC XXX VALUE "416".
2780.    02 F1                PIC XXX VALUE "421".
2790.    02 F1                PIC XXX VALUE "422".
2800.    02 F1                PIC XXX VALUE "441".
2810.    02 F1                PIC XXX VALUE "442".
2820.    02 F1                PIC XXX VALUE "443".
2830.    02 F1                PIC XXX VALUE "444".
2840.    02 F1                PIC XXX VALUE "450".
2850.    02 F1                PIC XXX VALUE "451".
2860.    02 F1                PIC XXX VALUE "500".
2870.    02 F1                PIC XXX VALUE "510".
2880.    02 F1                PIC XXX VALUE "511".
2890.    02 F1                PIC XXX VALUE "512".
2900.    02 F1                PIC XXX VALUE "513".
2910.    02 F1                PIC XXX VALUE "514".
2920.    02 F1                PIC XXX VALUE "515".
2930.    02 F1                PIC XXX VALUE "520".
2940.    02 F1                PIC XXX VALUE "525".
2950.    02 F1                PIC XXX VALUE "526".
2960.    02 F1                PIC XXX VALUE "527".
2970.    02 F1                PIC XXX VALUE "540".
2980.    02 F1                PIC XXX VALUE "541".
2990.    02 F1                PIC XXX VALUE "543".
3000.    02 F1                PIC XXX VALUE "550".
3010.    02 F1                PIC XXX VALUE "551".
3020.    02 F1                PIC XXX VALUE "552".
3030.    02 F1                PIC XXX VALUE "553".
3040.    02 F1                PIC XXX VALUE "555".
3050.    02 F1                PIC XXX VALUE "560".
3060.    02 F1                PIC XXX VALUE "565".
3070.    02 F1                PIC XXX VALUE "566".
3080.    02 F1                PIC XXX VALUE "568".
3090.    02 F1                PIC XXX VALUE "571".
3100.    02 F1                PIC XXX VALUE "573".
3110.    02 F1                PIC XXX VALUE "574".
3120.    02 F1                PIC XXX VALUE "575".
3130.    02 F1                PIC XXX VALUE "576".
3140.    02 F1                PIC XXX VALUE "581".
3150.    02 F1                PIC XXX VALUE "583".
3160.    02 F1                PIC XXX VALUE "590".
3170.    02 F1                PIC XXX VALUE "591".
3180.    02 F1                PIC XXX VALUE "603".
3190.    02 F1                PIC XXX VALUE "660".
3200.    02 F1                PIC XXX VALUE "661".
3210.    02 F1                PIC XXX VALUE "662".
3220.    02 F1                PIC XXX VALUE "663".
3230.    02 F1                PIC XXX VALUE "664".
3240.    02 F1                PIC XXX VALUE "670".
3250.    02 F1                PIC XXX VALUE "671".
3260.    02 F1                PIC XXX VALUE "680".
```

```
3270.    02 F1                    PIC XXX VALUE "681".
3280.    02 F1                    PIC XXX VALUE "682".
3290.    02 F1                    PIC XXX VALUE "683".
3300.    02 F1                    PIC XXX VALUE "684".
3310.    02 F1                    PIC XXX VALUE "685".
3320.    02 F1                    PIC XXX VALUE "686".
3330.    02 F1                    PIC XXX VALUE "687".
3340.    02 F1                    PIC XXX VALUE "701".
3350.    02 F1                    PIC XXX VALUE "702".
3360.    02 F1                    PIC XXX VALUE "703".
3370.    02 F1                    PIC XXX VALUE "704".
3380.    02 F1                    PIC XXX VALUE "705".
3390.    02 F1                    PIC XXX VALUE "706".
3400.    02 F1                    PIC XXX VALUE "707".
3410.    02 F1                    PIC XXX VALUE "708".
3420.    02 F1                    PIC XXX VALUE "709".
3430.    02 F1                    PIC XXX VALUE "710".
3440.    02 F1                    PIC XXX VALUE "711".
3450.    02 F1                    PIC XXX VALUE "712".
3460.    02 F1                    PIC XXX VALUE "713".
3470.    02 F1                    PIC XXX VALUE "714".
3480.    02 F1                    PIC XXX VALUE "715".
3490.    02 F1                    PIC XXX VALUE "716".
3500.    02 F1                    PIC XXX VALUE "717".
3510.    02 F1                    PIC XXX VALUE "718".
3520.    02 F1                    PIC XXX VALUE "719".
3530.    02 F1                    PIC XXX VALUE "720".
3540.    02 F1                    PIC XXX VALUE "765".
3550.    02 F1                    PIC XXX VALUE "810".
3560.    02 F1                    PIC XXX VALUE "811".
3570.    02 F1                    PIC XXX VALUE "815".
3580.    02 F1                    PIC XXX VALUE "820".
3590.    02 F1                    PIC XXX VALUE "850".
3600.    02 F1                    PIC XXX VALUE "860".
3610.    02 F1                    PIC XXX VALUE "861".
3620.    02 F1                    PIC XXX VALUE "862".
3630.    02 F1                    PIC XXX VALUE "863".
3640.    02 F1                    PIC XXX VALUE "864".
3650.    02 F1                    PIC XXX VALUE "865".
3660.    02 F1                    PIC XXX VALUE "866".
3670.    02 F1                    PIC XXX VALUE "867".
3680.    02 F1                    PIC XXX VALUE "868".
3690.    02 F1                    PIC XXX VALUE "869".
3700.    02 F1                    PIC XXX VALUE "870".
3710.    02 F1                    PIC XXX VALUE "871".
3720.    02 F1                    PIC XXX VALUE "872".
3730.    02 F1                    PIC XXX VALUE "873".
3740.    02 F1                    PIC XXX VALUE "874".
3750.    02 F1                    PIC XXX VALUE "875".
3760.    02 F1                    PIC XXX VALUE "876".
3770.    02 F1                    PIC XXX VALUE "877".
3780.    02 F1                    PIC XXX VALUE "878".
3790.    02 F1                    PIC XXX VALUE "879".
3800.    02 F1                    PIC XXX VALUE "880".
```

```
3610,    02 FI                  PIC XXX VALUE "881",
3620,    02 FI                  PIC XXX VALUE "882",
3630,    02 FI                  PIC XXX VALUE "883",
3640,    02 FI                  PIC XXX VALUE "884",
3650,    02 FI                  PIC XXX VALUE "885",
3660,    02 FI                  PIC XXX VALUE "886",
3670,    02 FI                  PIC XXX VALUE "887",
3680,    02 FI                  PIC XXX VALUE "888",
3690,    02 FI                  PIC XXX VALUE "889",
3700,    02 FI                  PIC XXX VALUE "890",
3710,    02 FI                  PIC XXX VALUE "891",
3720,    02 FI                  PIC XXX VALUE "892",
3730,    02 FI                  PIC XXX VALUE "895",
3740,    02 FI                  PIC XXX VALUE "896",
3750,    02 FI                  PIC XXX VALUE "899",
3760,    02 FI                  PIC XXX VALUE "901",
3770,    02 FI                  PIC XXX VALUE "903",
3780,    02 FI                  PIC XXX VALUE "905",
3790,    02 FI                  PIC XXX VALUE "906",
4000,    02 FI                  PIC XXX VALUE "907",
4010,    02 FI                  PIC XXX VALUE "908",
4020,    02 FI                  PIC XXX VALUE "911",
4030,    02 FI                  PIC XXX VALUE "912",
4040,    02 FI                  PIC XXX VALUE "913",
4050,    02 FI                  PIC XXX VALUE "914",
4060,    02 FI                  PIC XXX VALUE "915",
4070,    02 FI                  PIC XXX VALUE "916",
4080,    02 FI                  PIC XXX VALUE "917",
4090,    02 FI                  PIC XXX VALUE "918",
4100,    02 FI                  PIC XXX VALUE "919",
4110,    02 FI                  PIC XXX VALUE "920",
4120,    02 FI                  PIC XXX VALUE "921",
4130,    02 FI                  PIC XXX VALUE "922",
4140,    02 FI                  PIC XXX VALUE "923",
4150,    02 FI                  PIC XXX VALUE "924",
4160,    02 FI                  PIC XXX VALUE "925",
4170,    02 FI                  PIC XXX VALUE "926",
4180,    02 FI                  PIC XXX VALUE "927",
4190,    02 FI                  PIC XXX VALUE "941",
4200,    02 FI                  PIC XXX VALUE "942",
4210,    02 FI                  PIC XXX VALUE "943",
4220,    02 FI                  PIC XXX VALUE "944",
4230,    02 FI                  PIC XXX VALUE "950",
4240,    02 FI                  PIC XXX VALUE "951",
4250,    02 FI                  PIC XXX VALUE "960",
4260,    02 FI                  PIC XXX VALUE "961",
4270,    02 FI                  PIC XXX VALUE "962",
4280,    02 FI                  PIC XXX VALUE "963",
4290,    02 FI                  PIC XXX VALUE "964",
4300,    02 FI                  PIC XXX VALUE "970",
4310,    02 FI                  PIC XXX VALUE "980",
4320,    02 FI                  PIC XXX VALUE "990",
4340.01 RETABLA                 REDEFINES TABLA.
```

```
4350.      02 TAB-PLAN              OCCURS 271 INDEXED BY IND-TAB.
4360.        03 EL-PLAN             PIC XXX.
4370.       *-------------------------------------------------------*
4380.       *                        ENCABEZADOS                    *
4390.       *-------------------------------------------------------*
4400.01    ENCABEZADO-1.
4410.      05 ENCA-CODI-USUA         PIC X(4) VALUE SPACES.
4420.      05 FILLER                 PIC X(35) VALUE SPACES.
4430.      05 FILLER                 PIC X(54) VALUE
4440.      05 FILLER                 PIC X(28) VALUE SPACES.
4450.      05 FILLER PIC X(7) VALUE "PAGINA".
4460.      05 ENCA-HOJA              PIC ZZZ9  VALUE ZEROES.
4470.01    ENCABEZADO-2.
4480.      05 FILLER                 PIC X(5)  VALUE "PROG".
4490.      05 ENCA-PROGR             PIC X(7)  VALUE "VALIDA".
4500.      05 FILLER                 PIC X(32) VALUE SPACES.
4510.      05 FILLER                 PIC X(52) VALUE "VALIDACION DE DATOS DEL AU
4520.           "X-1600 CONVERSION POLIZAS".
4530.      05 FILLER                 PIC X(20) VALUE SPACES.
4540.      05 FILLER                 PIC X(7)  VALUE"FECHA".
4550.      05 ENCA-DIA-CORR1         PIC 99    VALUE ZEROS.
4560.      05 FILLER                 PIC X     VALUE "/".
4570.      05 ENCA-MES-CORR1         PIC 99    VALUE ZEROS.
4580.      05 FILLER                 PIC X     VALUE "/".
4590.      05 ENCA-ANO-CORR1         PIC 99    VALUE ZEROS.
4600.01    ENCABEZADO-3.
4610.      05 FILLER                 PIC X     VALUE SPACES.
4620.      05 FILLER                 PIC X(131)VALUE ALL "-".
4630.      05 FILLER                 PIC X(4)  VALUE ALL "-".
4640.01    ENCABEZADO-4
4650.      05 FILLER PIC X(132) VALUE  " CIA      POLIZA      EMISION
4660.       -     "   MONEDA     IMPO PRIM     VIGOR   FORM PAGO       RUTA
4670.       -     "       COMISION      AGRENTE   PLAN".
4680.       *-------------------------------------------------------*
4690.PROCEDURE DIVISION.
4700.       *-------------------------------------------------------*
4710.000-MODULO-CONTROL.
4720.       *-------------------------------------------------------*
4730.      PERFORM 100-INICIO  THRU 100-INICIO-FIN.
4740.      PERFORM 200-PROCESO THRU 200-PROCESO-FIN
4750.        UNTIL FIN-REG EQUAL "F".
4760.      PERFORM 300-FIN     THRU 300-FIN-FIN.
4770.      STOP RUN.
4780.000-MODULO-CONTROL-FIN.
4790.      EXIT.
4800       *-------------------------------------------------------*
4810.100-INICIO.
4820       *-------------------------------------------------------*
4830.      PERFORM  110-ABRE THRU 110-ABRE-FIN.
4840.      PERFORM 120-RESTAURA       THRU 120-RESTAURA-FIN.
4850.      PERFORM 130-LEE-AUX-1600   THRU 130-LEE-AUX-1600-FIN.
4860.100-INICIO-FIN.
4870.      EXIT.
4880       *-------------------------------------------------------*
4890.110-ABRE
4900       *-------------------------------------------------------*
```

```
4910.     OPEN INPUT ENTRADA
4920.          OUTPUT SALIDA.
4930.110-ABRE-FIN.
4940.     EXIT.
4950       *-----------------------------------------------------------*
4960.120-RESTAURA.
4970       *-----------------------------------------------------------*
4980.     MOVE SPACES TO FIN-REG
4990.                    CAMPOS
5000.                    REG2
5010.                    REG3
5020.                    DETALLE1
5030.                    DETALLE2
5040.                    SWITCHES
5050.                    AUX1600.
5060.     MOVE ZEROES TO CONTADORES-CONTROL.
5070.     ACCEPT FECHA FROM DATER-600.
5080.120-RESTAURA-FIN.
5090.     EXIT.
5100       *-----------------------------------------------------------*
5110.1300-LEE-AUX-1600.
5120       *-----------------------------------------------------------*
5130.     READ ENTRADA INTO AUX1600
5140.          AT END MOVE "F" TO FIN-REG
5150.                 MOVE ALL "Z" TO POL-AUX.
5160.     MOVE POL-AUX TO POL-ACT.
5170.130-LEE-AUX-1600-FIN.
5180.     EXIT.
5190       *-----------------------------------------------------------*
5200.200-PROCESO
5210       *-----------------------------------------------------------*
5220.     IF EDO-AUX NOT EQUAL"04"
5230.         MOVE POL-ACT TO POL-ANT
5240.         PERFORM 210-ACTIVA THRU
5250.                 210-ACTIVA-FIN
5260.                 UNTIL POL-ACT NOT EQUAL POL-ANT
5270.     ELSE
5280.         PERFORM 220-CANCELADA  THRU
5290.                 220-CANCELADA-FIN.
5300.200-PROCESO-FIN.
5310.     EXIT.
5320       *-----------------------------------------------------------*
5330.210-ACTIVA.
5340       *-----------------------------------------------------------*
5350.     IF TIPO-AUX EQUAL "02" OR TIPO-AUX  EQUAL "03"
5360.          PERFORM 211-ARMA-REG THRU 211-ARMA-REG-FIN.
5370.     PERFORM 130-LEE-AUX-1600 THRU 130-LEE-AUX-1600-FIN.
5380.     IF POL-ACT NOT EQUAL POL-ANT
5390.           PERFORM 212-VALIDA THRU 212-VALIDA-FIN
5400.           IF DETALLE2 NOT EQUAL SPACES
5410.               PERFORM 213-REPORTE THRU 213-REPORTE-FIN
5420.               PERFORM 214-RES-REG THRU 214-RES-REG-FIN
5430.           ELSE
5440.               PERFORM 214-RES-REG THRU 214-RES-REG-FIN.
5450.210-ACTIVA-FIN.
```

```
5460.     EXIT.
5470       *-----------------------------------------------------------*
5480.211-ARMA-REG.
5490       *-----------------------------------------------------------*
5500.     IF TIPO-AUX EQUAL "02"
5510.         MOVE AUX1600 TO REG2
5520.     ELSE
5530.         MOVE AUX1600 TO REG3.
5540.211-ARMA-REG-FIN.
5550.     EXIT.
5560       *-----------------------------------------------------------*
5570.212-VALIDA.
5580       *-----------------------------------------------------------*
5590.     IF CIA-REG2 CREATER THAN "3"
5600.          MOVE "*"TO CAMP1.
5610.     IF EMI-ANO EQUAL ZEROES OR EMI-ANO EQUAL SPACES
5620.         MOVE "*" TO CAMP2
5630.     ELSE
5640.         IF EMI-ANO LESS THAN "40" OR EMI-ANO GREATER THAN "82"
5650.              MOVE "*"TO CAMP2.
5660.             IF EMI-MES EQUAL ZEROES OR EMI-MES EQUAL SPACES
5670.                 MOVE" *"TO CAMP2
5680.             ELSE
5690.                  IF EMI-MES LESS THAN "01" OR
5700.                     EMI-MES GREATER THAN "12"
5710.                       MOVE "*"TO CAMP2.
5720.                      IF EMI-DIA EQUAL ZEROES OR
5730.                         EMI-DIA EQUAL SPACES
5740.                             MOVE "*" TO CAMP2
5750.                      ELSE
5760.                          IF EMI-DIA LESS THAN "01" OR
5770.                             EMI-DIA GREATER THAN "31"
5780.                               MOVE "*" TO CAMP2.
5790.     IF MONE-REG2 GREATER THAN "02"
5800.          MOVE "*"TO CAMP3.
5810.     IF PRIM-REG2 EQUAL ZEROES OR PRIM-REG2 EQUAL SPACES
5820.          MOVE "*" TO CAMP4.
5830.     IF PRIM-REG2 NOT NUMERIC
5840.          MOVE "*" TO CAMP4.
5850.     IF ANO-VIGO EQUAL ZEROES OR ANO-VIGO EQUAL SPACES
5860.          MOVE "*" TO CAMP5
5870.     ELSE
5880.         IF ANO-VIGO  "40" OR ANO-VIGO   "82"
5890.              MOVE "*" TO CAMP5.
5900.             IF MES-VIGO EQUAL ZEROES OR MES-VIGO EQUAL SPACES
5910.                 MOVE "*" TO CAMP5
5920.             ELSE
5930.                 IF MES-VIGO   "01" OR MES-VIGO   "12"
5940.                      MOVE "*"TO CAMP5.
5950.                     IF DIA-VIGO EQUAL ZEROES OR
5960.                        DIA-VIGO EQUAL SPACES
5970.                        MOVE "*" TO CAMP5
5980.                     ELSE
5990.                        IF   DIA-VIGO   "01"OR DIA-VIGO   "31"
6000                             MOVE "*" TO CAMP5.
```

```
6010.     IF NOT FORM-PAGO
6020.          MOVE "*" TO CAMP6.
6030.     IF RUTA-REG2 EQUAL ZEROES OR RUTA-REG2 EQUAL SPACES
6040.          MOVE "*" TO CAMP7.
6050.     IF REG3 NOT EQUAL SPACES
6060.         IF COM-REG3 NOT NUMERIC
6070.              MOVE "*" TO CAMP8
6080.         ELSA
6090.              IF AGTE-REG3 NOT NUMERIC
6100.                   MOVE "*" TO CAMP9
6110.              ELSE
6120.                   PERFORM 212-10-BUSCA-PLAN THRU
6130.                           212-10-BUSCA-PLAN-FIN
6140.                           IF SUP-REG3 EQUAL SPACES OR ZEROES
6150.                               ADD 1 TO CONT-SUP
6160.     ELSE
6170.         NEXT SENTENCE.
6180.212-VALIDA-FIN.
6190.     EXIT.
6200       *---------------------------------------------------------
6210.212-10-BUSCA-PLAN.
6220       *---------------------------------------------------------
6230.     SET IND-TAB TO 1
6240.     SEARCH TAB-PLAN VARYING IND-TAB AT END GO TO 212-11-NO-HAY
6250.     WHEN EL-PLAN (IND-TAB) EQUAL PLAN-REG3 NEXT SENTENCE.
6260.212-10-BUSCA-PLAN-FIN.
6270.     EXIT.
6280       *---------------------------------------------------------
6290.212-11-NO-HAY
6300       *---------------------------------------------------------
6310.     MOVE "*" TO CAMP10.
6320.     GO TO 212-10-BUSCA-PLAN-FIN.
6330.212-11-NO-HAY-FIN.
6340.     EXIT.
6350.      *---------------------------------------------------------
6360.213-REPORTE.
6370       *---------------------------------------------------------
6380.     PERFORM 213-10-ARMA-LINEA THRU 213-10-ARMA-LINEA-FIN.
6390.     PERFORM 213-20-IMPRESION THRU 213-20-IMPRESION-FIN.
6400.213-REPORTE-FIN.
6410.     EXIT.
6420       *---------------------------------------------------------
6430.213-10-ARMA-LINEA.
6440       *---------------------------------------------------------
6450.     MOVE CIA-REG2 TO CIA-DET.
6460.     MOVE POL-REG2 TO POL-DET.
6470.     MOVE EMI-REG2 TO EMI-DET.
6480.     MOVE MONE-REG2 TO MONE-DET.
6490.     MOVE PRIM-REG2 TO PRIM-DET.
6500.     MOVE VIGO-REG2 TO VIGO-DET.
6510.     MOVE PAGO-REG2 TO PAGO-DET.
6520.     MOVE RUTA-REG2 TO RUTA-DET.
6530.     MOVE COM-REG3  TO COM-DET.
6540.     MOVE AGTE-REG3 TO AGTE-DET.
6550.     MOVE PLAN-REG3 TO PLAN-DET.
```

```
6560.213-10-ARMA-LINEA-FIN.
6570.     EXIT.
6580       *-----------------------------------------------------------
6590.213-20-IMPRESION.
6600       *-----------------------------------------------------------
6610.     IF CONT-LIN GREATER THAN 49
6620.          ADD 1 TO CONT-HOJA
6630.          MOVE CONT-HOJA TO ENCA-HOJA
6640.          PERFORM 213-21-ENCABEZADO THRU 213-21-ENCABEZADO-FIN
6650.          MOVE 0 TO CONT-LIN.
6660.     PERFORM 213-22-DETALLE THRU 213-22-DETALLE-FIN.
6670.     PERFORM 213-23-RESTAURA-DET THRU 213-23-RESTAURA-DET-FIN.
6680.213-20-IMPRESION-FIN.
6690.     EXIT.
6700       *-----------------------------------------------------------
6710.213-21-ENCABEZADO.
6720       *-----------------------------------------------------------
6730.     MOVE DIA-FECH TO ENCA-DIA-CORR1.
6740.     MOVE MES-FECH TO ENCA-MES-CORR1.
6750.     MOVE ANO-FECH TO ENCA-ANO-CORR1.
6760.     IF CONT-HOJA NOT EQUAL 1
6770.        MOVE SPACES TO ENCA-PROGR.
6780.     WRITE SALE FROM ENCABEZADO-1 AFTER ADVANCING TO TOP OF PAGE.
6790.     WRITE SALE FROM ENCABEZADO-2 AFTER 2.
6800.     WRITE SALE FROM ENCABEZADO-3 AFTER 2.
6810.     WRITE SALE FROM ENCABEZADO-4 AFTER 1.
6820.     WRITE SALE FROM ENCABEZADO-3 AFTER 1.
6830.213-21-ENCABEZADO-FIN.
6840.     EXIT.
6850       *-----------------------------------------------------------
6860.213-22-DETALLE.
6870       *-----------------------------------------------------------
6880.     WRITE SALE FROM DETALLE1 AFTER 1.
6890.     WRITE SALE FROM DETALLE 2 AFTER 1.
6900       ADD 2 TO CONT-LIN.
6910.213-22-DETALLE-FIN.
6920.     EXIT.
6930       *-----------------------------------------------------------
6940.213-23-RESTAURA-DET.
6950       *-----------------------------------------------------------
6960.     MOVE SPACES TO DETALLE 1 DETALLE2.
6970.213-23-RESTAURA-DET-FIN.
6980.     EXIT.
6990       *-----------------------------------------------------------
7000.214-RES-REG.
7010       *-----------------------------------------------------------
7020.     MOVE SPACES TO REG2 REG3.
7030.214-RES-REG-FIN.
7040.     EXIT.
7050       *-----------------------------------------------------------
7060.220-CANCELADA.
7070       *-----------------------------------------------------------
7080.     ADD 1 TO CONT-CANCEL.
7090.     MOVE POL-ACT TO POL-ANT.
7100.     PERFORM 130-LEE-AUX-1600 THRU 130-LEE-AUX 1600-FIN
```

```
7110,              UNTIL POL-ACT NOT EQUAL POL-ANT.
7120,220-CANCELADA-FIN.
7130,     EXIT.
7140       *-----------------------------------------------------------
7150,300-FIN.
7160       *-----------------------------------------------------------
7170,     PERFORM 310-TOTALES THRU
7180,             310-TOTALES-FIN.
7190,     PERFORM 320-CIERRA THRU
7200,             320-CIERRA-FIN.
7210,300-FIN-FIN.
7220,     EXIT.
7230       *-----------------------------------------------------------
7240,310-TOTALES.
7250       *-----------------------------------------------------------
7260,     MOVE CONT-CANCEL TO POL-CANCEL-DET.
7270,     MOVE CONT-SUP TO SUP-TOT-DET.
7280,     ADD 1 TO CONT-HOJA.
7290,     MOVE CONT-HOJA TO ENCA-HOJA.
7300,     PERFORM 213-21-ENCABEZADO THRU
7310,             213-21-ENCABEZADO-FIN.
7320,     WRITE SALE FROM DETALLE-TOT1 AFTER 5.
7330,     WRITE SALE FROM DETALLE-TOT2 AFTER 5.
7340,310-TOTALES-FIN.
7350,     EXIT.
7360       *-----------------------------------------------------------
7370,320-CIERRA.
7380       *-----------------------------------------------------------
7390,     CLOSE ENTRADA SALIDA.
7400,320-CIERRA-FIN.
7410,     EXIT.
7420$;PRMFL;C*,W,S,SISABT/INTERFASE/OBJ/INDAVAL
7430$;FILE;K*,NULL
```

PROG VALIDA

VALIDACION DE DATOS DEL AUX-1600 CONVERSION POLIZAS

FECHA 19/05/82

CIA	POLIZA	EMISION	MONEDA	IMPO PRIM	VIGOR	FORM PAGO	RUTA	COMISION	AGENTE	PLAN
0	005908	820531	01	0000042234	820531	12	NM178	0000000000	IE700	101
0	013924	820426	01	0000014443	820426	12	NM129	0000000000	IE700	101
0	015351	811213	01	0000002727	811213	06	GR100	0000000000	IE700	101
0	015603	820112	01	0000143343	820112	12	GR700	0000000000	B1101	344
0	015640	820201	01	0007976650	820201	12	04101	0000079516	04101	812
0	015890	820221	01	0000053856	820221	12	GR500	0000000000	D8700	344
0	018338	820326	01	0000006133	820326	12	NM166	0000000000	IE700	101
0	019200	820228	01	0712570420	820228	12	GR100	0000000000	00208	812
0	020602	820305	01	0000007443	820305	03	GR400	0000000000	B1101	101
0	020894	820130	01	0000090878	820120	12	GR100	0000000000	B1101	344
0	021098	820312	01	0000002302	820312	12	GR400	0000000000	IE700	101
0	021489	820524	01	0000033805	820524	12	GR500	0000000000	IE700	335
0	022649	811220	01	0000130472	811220	06	GR100	0000000000	IE700	101
0	023126	820410	01	0000012147	820410	03	GR100	0000000000	IE700	101
0	023522	820529	01	0000001814	820529	12	GR400	0000000000	B1101	101
0	024571	811205	01	0000226873	811205	12	NM443	0000000000	DX700	355
0	024834	820118	01	0000581375	820118	12	GR100	0000000000	B1101	355
0	025091	811230	01	0000068854	811230	06	GR100	0000000000	B1101	355
0	025147	820322	01	0000688325	820322	03	GR103	0000000000	B1101	355
0	025233	820405	01	0000300001	820405	12	GR100	0000000000	IE700	355
0	025433	820504	01	0000173154	820504	12	NM016	0000000000	IE700	355
0	025449	820508	01	0000114857	820508	12	GR100	0000000000	IE700	355
0	026768	820330	01	0000025270	820330	03	GR100	0000000000	IE700	355
0	027541	820427	01	0000047968	820427	12	GR100	0000000000	D5700	315
0	028088	820115	01	0000045058	820115	06	GR100	0000000000	IE700	355

PROG

FECHA 19/05/82

CIA	POLIZA	EMISION	MONEDA	IMPO PRIM	VIGOR	FORM PAGO	RUTA	COMISION	AGENTE	PLAN
0	028552	820403	01	0000023905	820403	03	GR100	0000000000	IE700	355
0	028674	810226	01	0000050740	810226	01	GM002	0000000000	B1101	354
0	029233	820124	01	0000454213	820124	12	GR100	0000000000	B1101	335
0	029316	820501	01	0002790594	820501	01	10723	0000027905	00420	812
0	029319	820501	01	0011798049	820501	01	10723	0000117980	00420	812
0	029622	820327	01	0000166460	820327	12	GR100	0000000000	IE700	315
0	029800	820519	01	0000004903	820519	01	GR100	0000000000	DN700	525
0	031128	820310	01	0000152961	820310	12	NM106	0000000000	C3200	355
0	031337	801225	01	0003990170	821225	01	GR500	0000000000	00003	387
0	031598	811213	01	0000131940	811213	12	GR100	0000000000	B1101	101
0	031713	811228	01	0000166282	811228	12	GR300	0000000000	IE700	335
0	031944	820131	01	0000129382	820131	12	NM106	0000000000	C3200	355
0	032426	820329	01	0000245112	820329	12	GR100	0000000000	IE700	315
0	032676	820509	01	0000192038	820509	12	GR100	0000000000	IE700	355
0	033028	811201	01	0000110413	811201	06	NM202	0000000000	B1101	309
0	033335	820512	01	0000030465	820512	01	GM002	0000000000	DN700	354
0	033998	820416	01	0000071806	820416	12	GR100	0000000718	00208	812
0	034539	820207	01	0000167977	820207	12	NM152	0000000000	IE700	335
0	034644	820224	01	0000166282	820224	12	GR206	0000000000	B1101	335
0	035059	820415	01	0000057885	820415	12	NM176	0000000000	IE700	335
0	035060	820415	01	0000057885	820415	12	NM176	0000000000	IE700	335
0	035129	820423	01	0000775016	820423	12	GR100	0000000000	IE700	334
0	035130	820423	01	0002290601	820423	12	GR700	0000000000	B1101	334
0	035202	820506	01	0000240520	820506	12	NM171	0000000000	IE700	334
0	036681	811208	01	0000425630	811208	12	GR100	0000000000	IE700	355

9. BIBLIOGRAFIA.

HONEYWELL
1979 SERIES 60(LEVEL 66), SOFTWARE
COBOL 74, REFERENCE MANUAL.

HONEYWELL
1980 SERIES 60 (LEVEL 66), SOFTWARE
COBOL 68, REFERENCE MANUAL

BOURROUGHS
1979 REFERENCE MANUAL, COBOL.

PEDDICORR, RICHARD G.
1980 UNDERSTANDING COBOL
CHICAGO. ALFRED PUBLISHING. CO.

WARNIER, J. D.
FLANAGAN, B. M.
1976 PROGRAMACION LOGICA, TOMO I
CONSTRUCCION DE PROGRAMAS.
BARCELONA. ED. TECNICOS ASOCIADOS S.A.
HONEYWELL BULL.

CHAI, WINCHUNG A.
CHAI, HENRY W.
1976 PROGRAMING STANDARD COBOL.
NEW YORK. ACADEMIC PRESS, INC.