

INDICE

Archivos de Control	3
Archivos de Información	7
Archivos para manejo de información	22
Archivo para dependencias	15
Archivo descriptorpedido	28
Archivo facturaseextra	10
Archivo Fcontrol	5
Archivo Fdisnopedido	22
Archivo Fdisnompedido	25
Archivo Ffactura	13
Archvo Fproveedor	19
Archivo Nopedido	7
Procedure Altapedido	105
" Asignadescriptor	58
" Abrearchivos	70
" Asignapresupuesto	97
" Asignaextra	98
" Buscados	101
" Bascasobretodo	102
" CERRARCHIVOS	70
" Creadirectorio	50
" Consiguedisponible	54
" Crearchivos	69
" Checapresupuesto	77
" Cargador	74
" Cierrarchivos	70
" Cargaproveedor	85
" Checanúmero	42
" Checa	41
" Checatipo	41
" Checafecha	42
" Escribe	33
" Encabezado	35
" Error	36
" Emitelistadospedido	106
" Emitelistadosrecepción	107
" Estadosdecuenta	115 - 117
" Ejecución	123
" Fechac	34
" Filtra 1	37
" Guardanombrespedido	103
" Informa	119
" Impresión	110

BIBLIOTECA



CENTRO UNIVERSITARIO
DE INVESTIGACIONES Y
BIBLIOTECA

INFOBILA

Procedure	Inserta	45
"	Insertaendiretorio	52
"	Intersecta	61
"	Leetarjeta	36
"	Localiza y compara	55
"	Localizando	100
"	Llenalista	67
"	Pedido	89 - 93
"	Recepción	111
"	Sacanombres y revisa	48
"	Send	33
"	Sacafecha	34
"	Sacaproveedor	87
"	Sacaproveedorcompleto	88
"	Validando	92 - 95

DESCRIPCION Y CARACTERISTICAS DE LOS ARCHIVOS
DENTRO DEL SISTEMA DE ADQUISICIONES

Los archivos dentro del sistema de adquisiciones automatizado, fueron diseñados de tal forma que captan, alojen y devuelvan información usando el mínimo de recursos de máquina.

En su estructura interna, el Banco de Información está formado por dos tipos de archivos:

A) Archivos permanentes

B) Archivos temporales

Archivos permanentes.- Son aquellos que forman el Banco de Datos después de que la información ha sido procesada. Estos archivos se encuentran a su vez divididos en:

a) Archivos de control

b) Archivos de información

c) Directorios o archivos para manejo de la información

Archivos temporales.- Son aquellos que contienen almacenadas las imágenes de las tarjetas de datos. Como las tarjetas son recibidas dentro del sistema por comandos de revisión específicos para cada programa, las imágenes de tarjetas correctas quedan guardadas en archivos que tienen como prefijo la palabra RETENIDOS; y como sufijo el número secuencial de archivo que le corresponde. En caso de ser tarjetas con errores, quedan guardadas en archivos con prefijo ERROR; y como sufijo el número de archivo.

No. de Adquisición: _____
Vols.: _____
Fecha: _____

REMISION DE OBRAS Y TARJETAS
A LA BIBLIOTECA

El encargado de la Biblioteca

El jefe del Departamento de Procesos Técnicos

ARCHIVOS

PERMANENTES

CONTROL

INFORMACION

MANEJO DE
INFORMACION

TEMPORALES

ERRORES

RETENIDOS

FCONTROL

NO PEDIDO

FFACTURA

FACTURASEXTRA

DEPENDENCIAS

FPROVEEDOR

FDISNO PEDIDO

FDESCRIPTOR PEDIDO

FDISPNOMPEDIDO

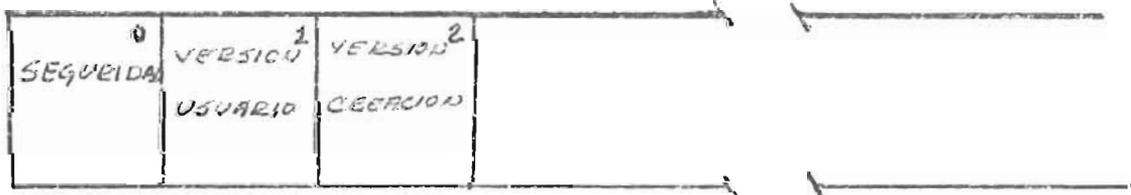
a) ARCHIVOS DE CONTROL.-

Cada uno de los archivos que forman el Banco tienen reservado el registro cero o primer registro lógico del archivo, como registro de CONTROL. Las primeras tres palabras de éste contienen:

PALABRA 0.- Un identificador alfabético, para indicar el estado del archivo. Si el archivo es usado y por cualquier motivo el programa no acaba normalmente, queda almacenada la palabra "USADO"; en caso contrario la palabra "CORRECT"

PALABRA 1.- Contiene la versión del archivo, consistente en la fecha, la hora y clave del último usuario del archivo

PALABRA 2.- Contiene la versión de creación del archivo, con fecha, hora y clave del usuario que creó el archivo



REGISTRO CERO DE TODOS LOS ARCHIVOS

47	35	31	33	31	47	25	19	15	11	7	3
46	42	38	34	30	26	22	18	14	10	6	2
45	37	33	29	25	21	17	13	9	5	1	
44	40	36	32	28	24	20	16	12	8	4	0

PALABRA CERO

"CORRECT"

Quando el archivo fué cerrado correctamente

"USADO"

Mientras no se cierra el archivo

47	35	31	33	31	47	25	19	15	11	7	3
46	42	38	34	30	26	22	18	14	10	6	2
45	41	37	33	29	25	21	17	13	9	5	1
44	40	36	32	28	24	20	16	12	8	4	0

PALABRA UNO

Versión del usuario

47	35	31	33	31	47	25	19	15	11	7	3
46	42	38	34	30	26	22	18	14	10	6	2
45	41	37	33	29	25	21	17	13	9	5	1
44	40	36	32	28	24	20	16	12	8	4	0

PALABRA DOS

Versión de creación

Para controlar las versiones de todos los archivos del sistema, se cuenta con un archivo de control "FCONTROL".

ARCHIVO FCONTROL

Este archivo tiene registros lógicos de 30 palabras.

En el registro cero, las tres primeras palabras son para control y a partir de la palabra diez, es almacenada la versión de los archivos fijos del sistema la cual debe coincidir con la versión que cada archivo tiene en su registro cero.

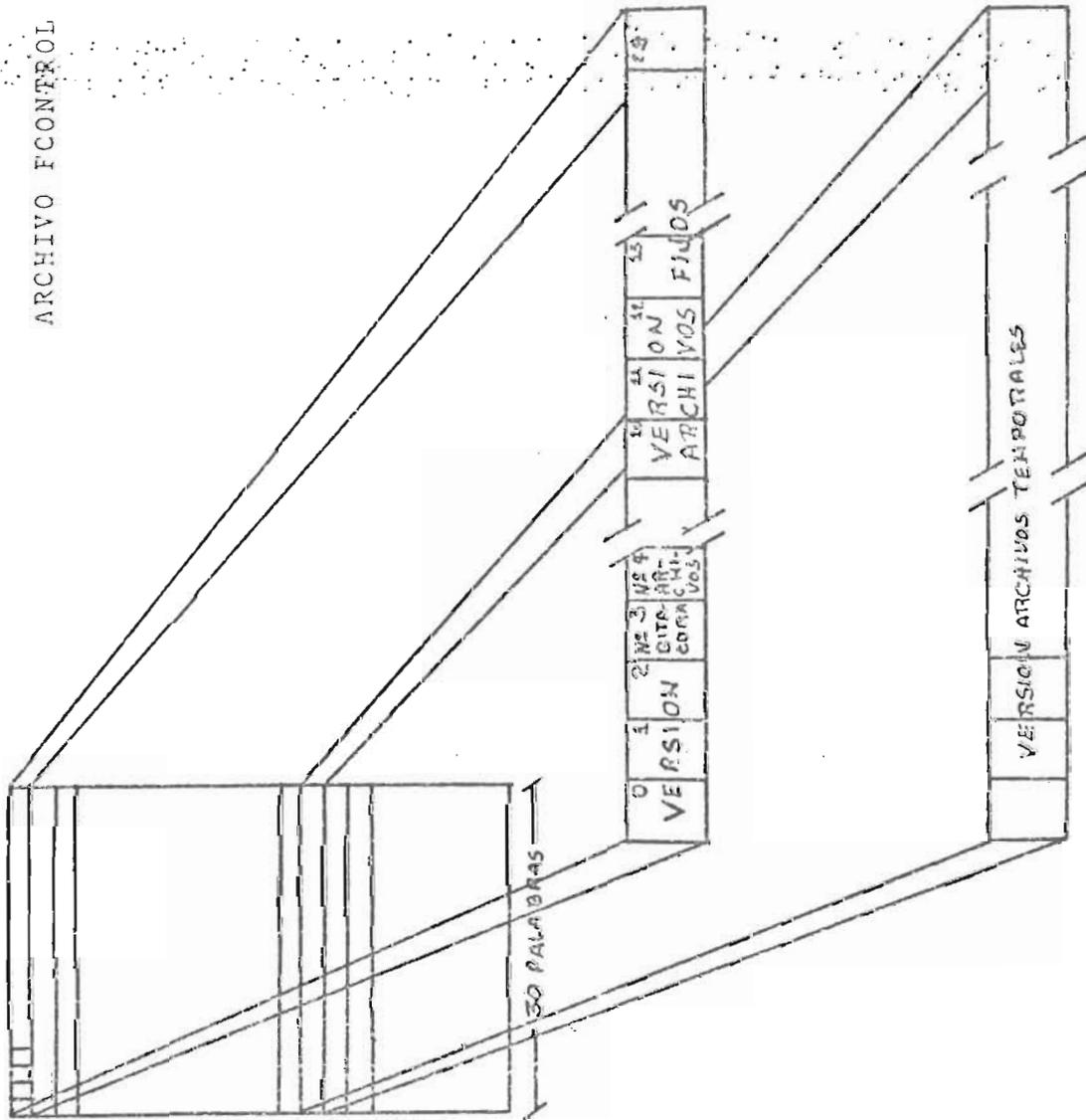
Además se guardan datos complementarios para el control del mismo sistema como número de corrida para emitir la bitácora y número del último archivo temporal almacenado.

Los registros posteriores contienen las versiones de los archivos temporales del sistema por número de archivo. La versión se localiza por medio de las funciones:

f_1 = Número de archivo div. 30 Para localizar el registro

f_2 = Número de archivo mod. 30 Para localizar la palabra

ARCHIVO FCONTROL



b) ARCHIVOS DE INFORMACION

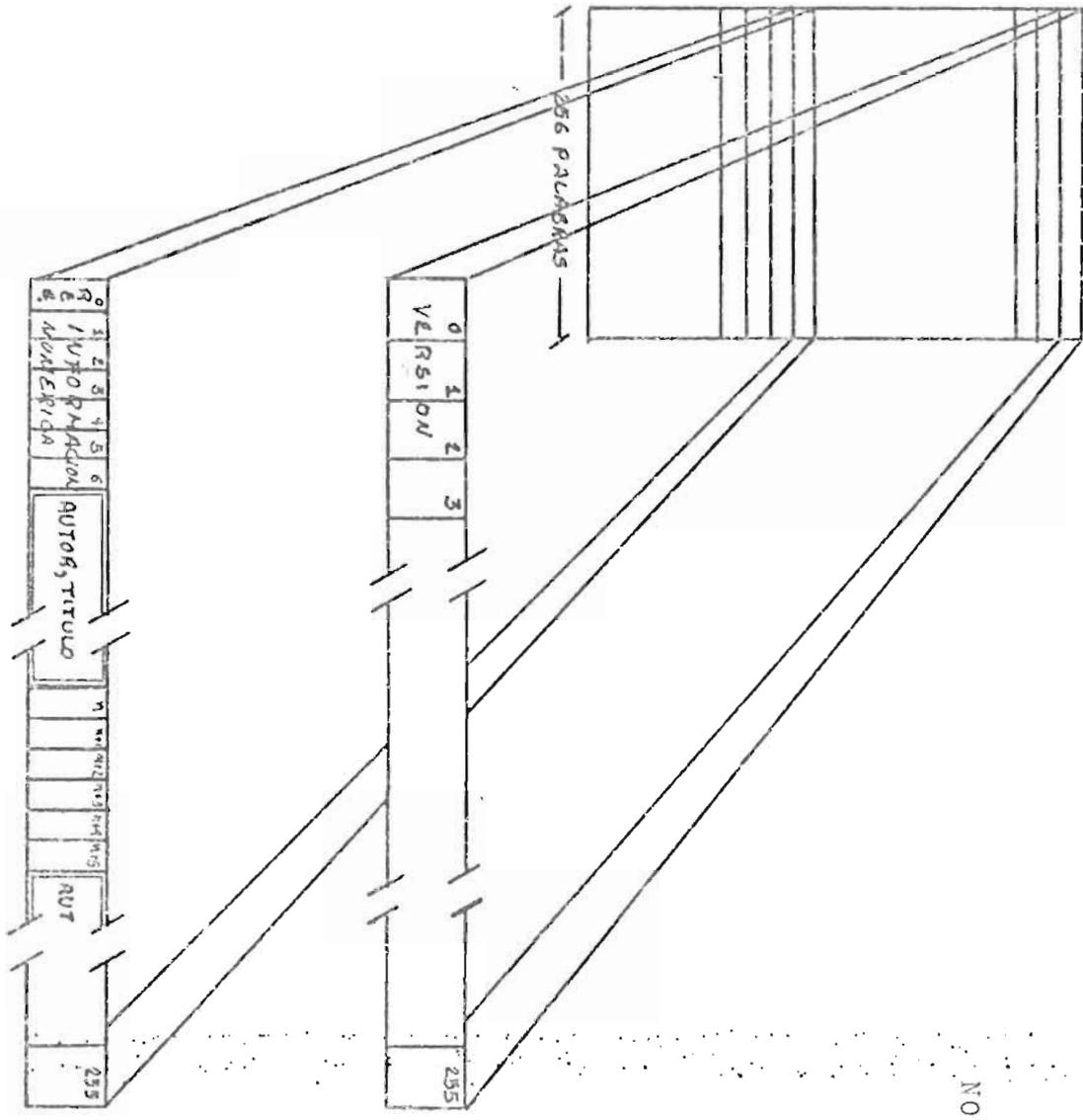
Archivos de información son aquellos que tienen "directamente" almacenada la información después de que ésta ha sido procesada. Los datos son almacenados de diversas formas, dependiendo del tipo de identificador que se requiera, así cuando el identificador es numérico, se guarda en los bits necesarios para su alojamiento, si es un código es convertido a un número y almacenado en los bits necesarios, el identificador alfabético es almacenado usando como unidad básica la palabra máquina (seis caracteres), usando únicamente el número de palabras necesarias para almacenar el identificador por medio de técnica de memoria dinámica (*).

ARCHIVO NOPEDIDO

En NOPEDIDO se almacena toda la información referente a un pedido tanto en el momento en que esté solicitado, como una vez surtido. Cada registro lógico del archivo tiene una longitud de 256 palabras pero internamente es trabajado por medio de memoria dinámica y cada registro o nodo de información ocupa únicamente el número de palabras necesario para alojar la información del pedido.

Cada nodo contiene una parte fija de seis palabras en las cuales almacenan los datos numéricos del pedido y las direcciones para localizar los identificadores variables. Enseguida están los identificadores alfabéticos que forman la parte variable del nodo. Cada identificador alfabético tiene en los primeros 8 bits de la palabra, la longitud del identificador.

Cada nodo de éstos, es referenciado desde otros directorios del sistema. Así, la dirección hacia la parte fija del nodo está alojada en archivo Dispnpedido, la dirección hacia cada uno de los identificadores está guardada en el campo de llave en el nodo correspondiente del directorio doble.



NO PEDIDO

2

47	43	39	35	31	27	23	19	15	11	7	3
46	42	38	34	30	26	22	18	14	10	6	2
45	41	37	33	29	25	21	17	13	9	5	1
44	40	36	32	28	24	20	16	12	8	4	0

DPRECIO=PRECIO DE LOS LIBROS
 DCOSTO=COSTO REAL DE LOS LIBROS.

3

47	43	39	35	31	27	23	19	15	11	7	3
46	42	38	34	30	26	22	18	14	10	6	2
45	41	37	33	29	25	21	17	13	9	5	1
44	40	36	32	28	24	20	16	12	8	4	0

DPEDICION=EDICION
 DPAÑOEDICION=AÑO DE EDICION
 DFECHAFACT=FECHA DE LA FACT.

4

47	43	39	35	31	27	23	19	15	11	7	3
46	42	38	34	30	26	22	18	14	10	6	2
45	41	37	33	29	25	21	17	13	9	5	1
44	40	36	32	28	24	20	16	12	8	4	0

DFTITULO=TITULO
 DFSUBTITULO=SUBTITULO

5

47	43	39	35	31	27	23	19	15	11	7	3
46	42	38	34	30	26	22	18	14	10	6	2
45	41	37	33	29	25	21	17	13	9	5	1
44	40	36	32	28	24	20	16	12	8	4	0

DFAUTOR=AUTOR
 DFAPELLIDO=APELLIDO DEL AUTOR.

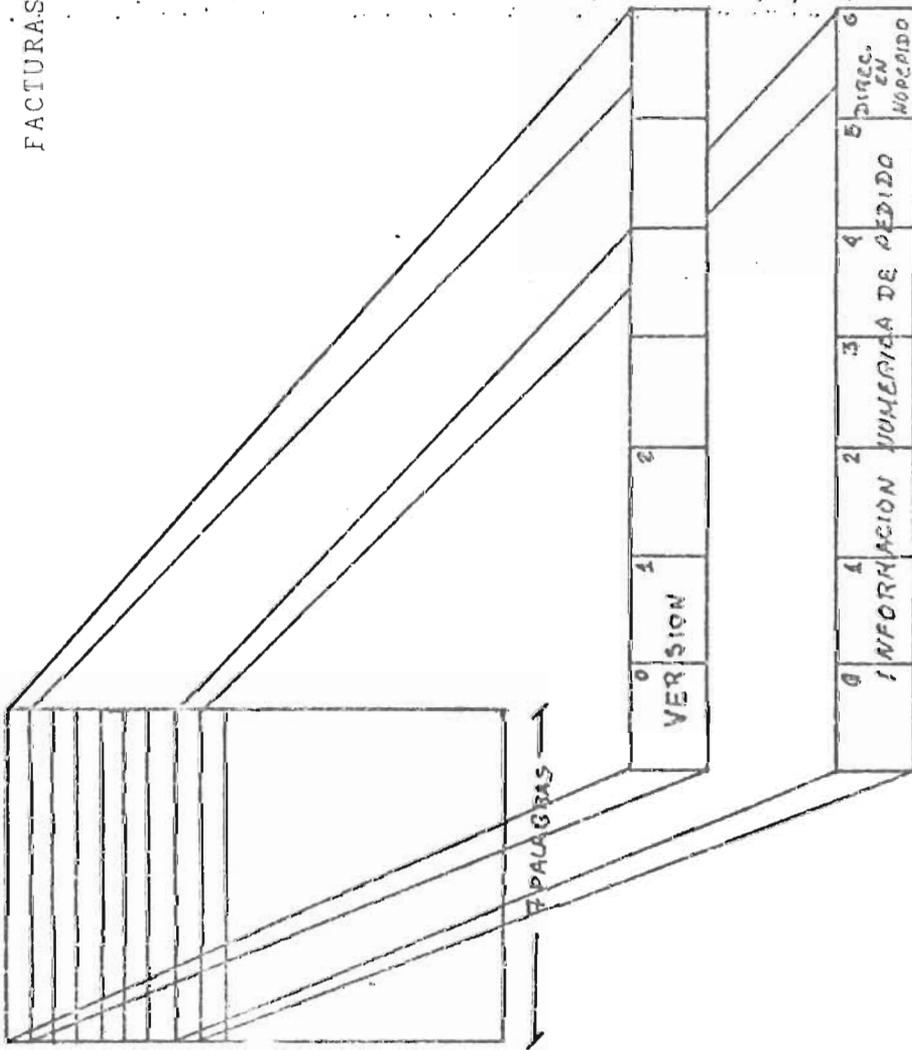
6

47	43	39	35	31	27	23	19	15	11	7	3
46	42	38	34	30	26	22	18	14	10	6	2
45	41	37	33	29	25	21	17	13	9	5	1
44	40	36	32	28	24	20	16	12	8	4	0

DLUGAR=LUGAR EN

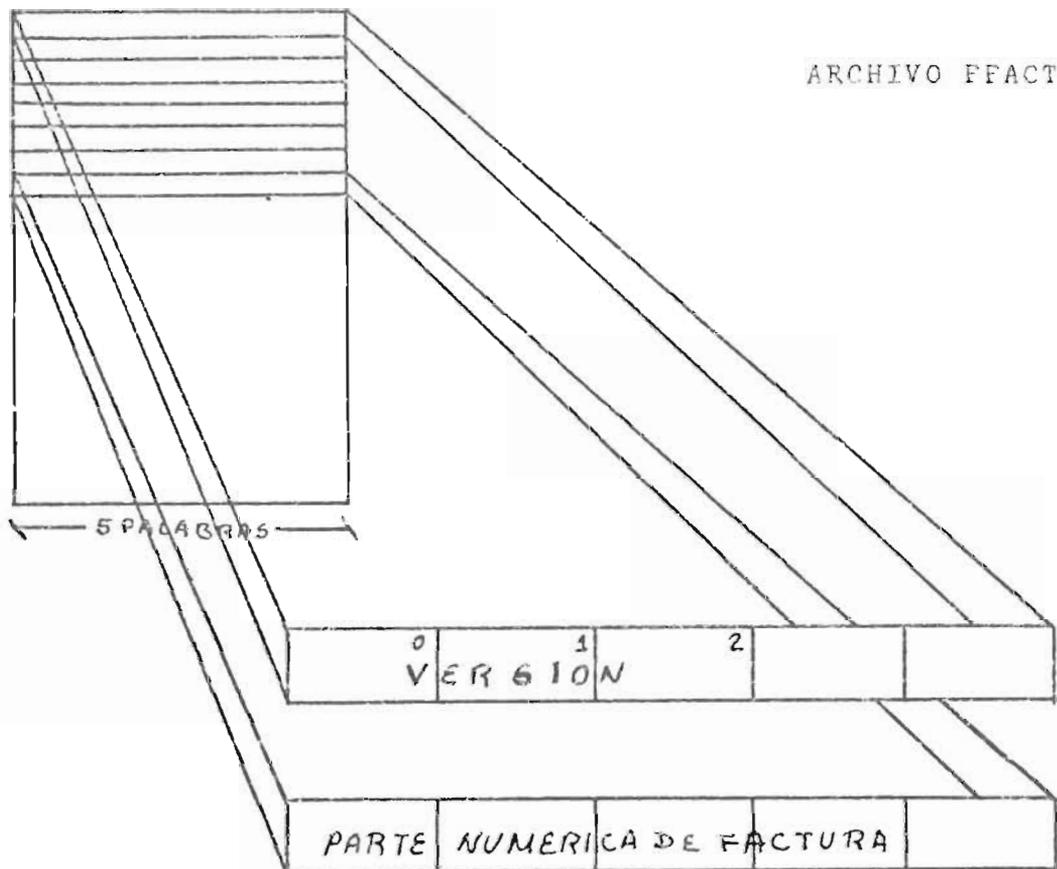
INDICACIONES
 PARA EL
 CATALOGO

FACTURASEXTRA



ARCHIVO FFACTURA

Es un archivo secuencial donde cada nodo, contiene los datos correspondientes a las facturas captadas dentro del sistema. Cada factura puede contener más de un pedido. Cada registro lógico es de cinco palabras.



7
ARCHIVO DEPENDENCIAS

Archivo secuencial de longitud fija de 10 palabras por registro, donde cada nodo contiene los datos presupuestales de las dependencias.

El acceso a éste archivo se hace en forma aleatoria localizando cada registro por el número de dependencia. Forma "RANDOM" de acceso a disco.

DESCRIPCION DEL REGISTRO FISICO POR PALABRA

0

17	15	33	35	31	27	23	19	15	11	7	3
16	12	28	34	30	26	22	18	14	10	6	2
REG-	11	27	33	29	25	21	17	13	9	5	1
DEP	10	26	32	28	24	20	16	12	8	4	0

REGDEP=NUMERO DEL REGISTRO

1

17	15	33	35	31	27	23	19	15	11	7	3
16	12	28	34	30	26	22	18	14	10	6	2
PREVIO	11	27	33	29	25	21	17	13	9	5	1
COMPROMETIDO	10	26	32	28	24	20	16	12	8	4	0

PREVIO=GASTOS POR COMPROBAR

COMPROMETIDO=EFECTIVO APARTADO PARA COMPRAS

47	43	39	35	31	27	23	19	15	11	7	3
46	42	38	34	30	26	22	18	14	10	6	2
45	41	37	33	29	25	21	17	13	9	5	1
44	40	36	32	28	24	20	16	12	8	4	0

EJERCIDO=EJERCIDO

DISPONIBLE=DISPONIBLE

47	43	39	35	31	27	23	19	15	11	7	3
46	42	38	34	30	26	22	18	14	10	6	2
45	41	37	33	29	25	21	17	13	9	5	1
44	40	36	32	28	24	20	16	12	8	4	0

PREVIOANT=PREVIO AL MES ANTE
RIOR

COMPROMETIDOANT=COMPROMETIDO
AL MES ANTERIOR

47	43	39	35	31	27	23	19	15	11	7	3
46	42	38	34	30	26	22	18	14	10	6	2
45	41	37	33	29	25	21	17	13	9	5	1
44	40	36	32	28	24	20	16	12	8	4	0

EJERCIDOANT=EJERCIDO AL MES
ANTERIOR

DISPONIBLEANT=DISPONIBLE AL
MES ANTERIOR

47	43	39	35	31	27	23	19	15	11	7	3
46	42	38	34	30	26	22	18	14	10	6	2
45	41	37	33	29	25	21	17	13	9	5	1
44	40	36	32	28	24	20	16	12	8	4	0

ASIGNADO=PRESUPUESTO ASIGNA-
DO

EXTRA=ASIGNACION EXTRA

47	43	39	35	31	27	23	19	15	11	7	3
46	42	38	34	30	26	22	18	14	10	6	2
45	41	37	33	29	25	21	17	13	9	5	1
44	40	36	32	28	24	20	16	12	8	4	0

CONGELADO=PRESUPUESTO CONGE-
LADO ANTE EGRESOS

CONGELADOANT=PRESUPUESTO CON-
GELADO ANTE EGRE--
SOS HASTA EL MES -
ANTERIOR

7

17	13	21	33	31	27	23	19	3	11	7	3
16	12	21	31	30	25	22	18	14	10	6	2
15	11	21	33	27	23	21	17	13	9	5	1
14	10	26	32	28	24	20	16	12	8	4	0

LIBROADQ=LIBROS ADQUIRIDOS

LIBROSPEDIDOS= LIBROS PEDIDOS

8

17	13	21	33	31	27	23	19	3	11	7	3
16	12	21	31	30	26	22	18	14	10	6	2
15	11	21	33	27	23	21	17	13	9	5	1
14	10	26	32	28	24	20	16	12	8	4	0

INVENTARIO=INVENTARIO

SOBREGIRO=SOBREGIRO

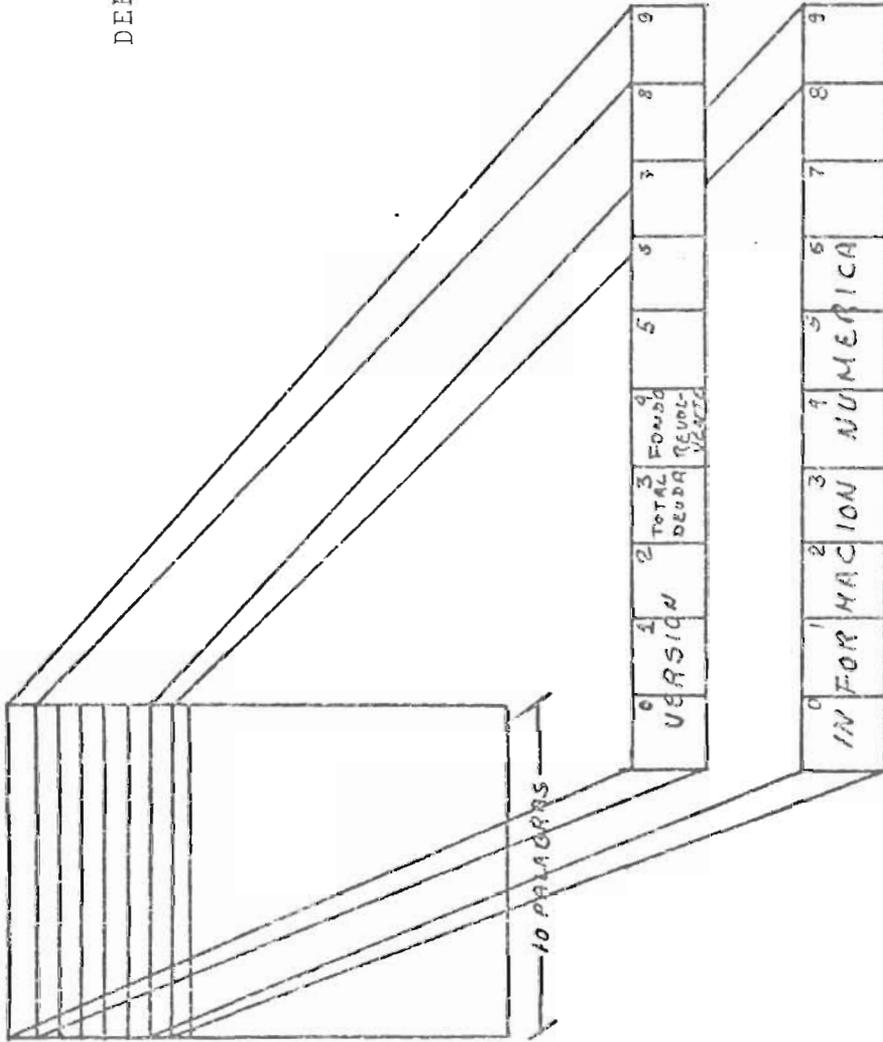
9

17	13	21	33	31	27	23	19	3	11	7	3
16	12	21	31	30	26	22	18	14	10	6	2
15	11	21	33	27	23	21	17	13	9	5	1
14	10	26	32	28	24	20	16	12	8	4	0

PERDIDA=PERDIDA

TOTAL=TOTAL
 TOTAL QUE SE PUEDE
 EN UN MOMENTO DADO

DEPENDENCIA



ARCHIVO PROVEEDOR

Este archivo guarda toda la información referente a los proveedores. Cada registro lógico tiene una longitud de 256 palabras, se maneja con la técnica de memoria dinámica de la misma forma que el archivo NOPEDIDO, en la parte fija del nodo se guardan las direcciones que dentro del mismo archivo localizan las palabras donde están alojados los identificadores de plaza, dirección, población y ciudad.

En el registro cero de este archivo además de las palabras de control se implementó un directorio para localizar a los proveedores por número, donde cada media palabra contiene la dirección dentro del archivo que aloja al nodo variable que contiene la información del proveedor, en la misma media palabra, se aloja la moneda en que se paga al proveedor y el número de días que tiene autorizado para surtir un pedido.

DESCRIPCION DE NODOS EN REGISTRO 0

47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99																															
D CALLE												D POBLACION												D PLAZA												D CIUDAD												D NUM -												D DE -												D PROV											

DCALLE= CALLE

DPLAZA=PLAZA

DNUMPROV=NUMERO DEL

DPOBLACION=POBLACION

DCIUDAD=CIUDAD

PROVEEDOR

DESCRIPCION DE PARTE FIJA DEL NODO

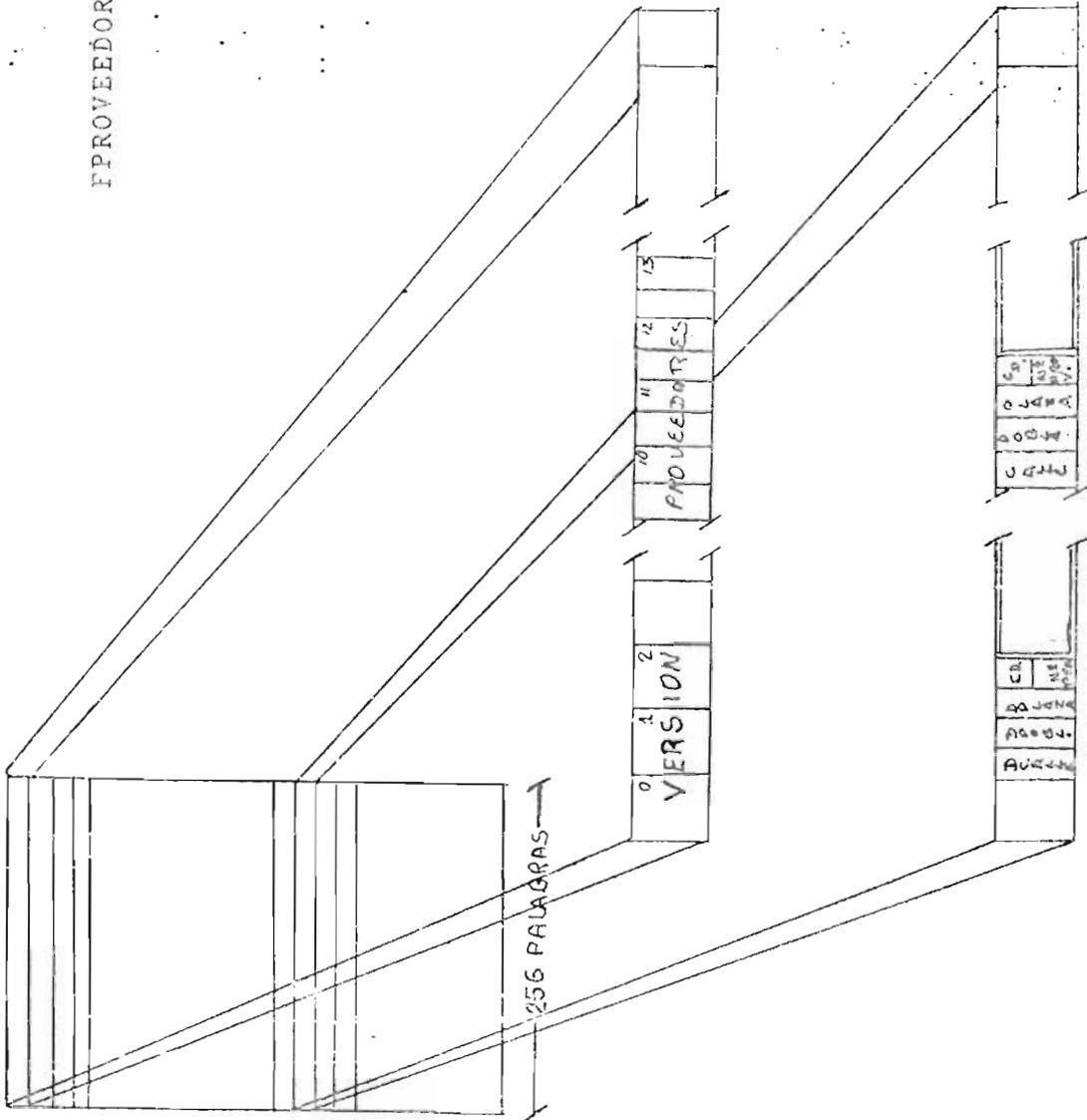
57	53	59	55	21	27	25	17	15	7	9	3
NOMBRE-				TIEM-							
56	52	58	54	20	26	24	16	14	6	8	2
PROV-				PROV-				TIEM-			
55	51	57	53	19	25	23	15	13	5	7	1
PROV				PROV				PROV			
54	50	56	52	18	24	22	14	12	4	6	0
PROV				PROV							

NOMBREPROV=NOMBRE DEL PROVEEDOR

TIEMPOPROV=TIEMPO PERMITIDO PARA ENTREGA DE LOS LIBROS

MONEDAPROV=MONEDA

FPROVEEDOR



ARCHIVOS PARA MANEJO DE INFORMACION

Los archivos para manejar la información son aquellos que se usan para agrupar y localizar la información dentro del Banco de Datos. Estos archivos son los directorios usados para localizar los pedidos, tanto por número de orden como por nombre de autor o título. También los archivos que contienen las listas inventadas que forman grupos de información por identificador o llave.

D I R E C T O R I O S

ARCHIVO FDISNOPEDIDO

Es un archivo que tiene 256 palabras de registro lógico. En el registro cero contiene las tres palabras de protección y en la palabra cuatro, un indicador al registro que tiene pedidos aún en proceso de adquisición de cada período.

En los demás registros, los primeros 24 bits de la palabra cero se guarda el número de registro y cada media palabra contiene la dirección del nodo de pedido en el ARCHIVO NOPEDIDO. En la misma, se indica si el pedido ya ha sido procesado completamente o está dado de baja-

Cada nodo se localiza por las siguientes funciones de dispersión:

- Para localizar el registro

$$f_1 = (I \text{ DIV } 511 - \text{REAL} (I \text{ MOD } 511 \text{ EQL } 0) + 1)$$

- Para localizar la palabra

$$f_2 = ((I - 511 \times (f (I) - 1)) \text{ DIV } 2$$

DESCRIPCION DE LOS NODOS

0

57	53	51	55	47	47	55	7	5	7	7	3
56	52	50	54	46	46	54	6	4	6	6	2
55	51	49	53	45	45	53	5	3	5	5	1
54	50	48	52	44	44	52	4	2	4	4	0

REG. INFORMACION

53	53	53	53	53	53	53	53	53	53	53	53
52	52	52	52	52	52	52	52	52	52	52	52
51	51	51	51	51	51	51	51	51	51	51	51
50	50	50	50	50	50	50	50	50	50	50	50

INFORMACION INFORMACION

1 a la 255

Para la recuperación de Identificadores Alfanuméricos, dentro del sistema se implementó el Directorio Doble con zona de nodos índice, zona de nodos de sobreflujo que puede abarcar la zona de nodos índice cuando el sobreflujo se agota. (Ver procesos para la implementación del directorio doble).

ARCHIVO FDISPNOMPEDIDO

Es el archivo que aloja al directorio doble para la localización de identificadores alfanuméricos, tiene 240 palabras por registro lógico. El registro cero contiene las palabras de seguridad. Los siguientes registros contienen las listas de los identificadores, la palabra cero de cada registro contiene:

- Número de registro
- Número de coliciones en el registro
- Primer nodo de la lista de nodos disponibles en el registro.

41	45	33	36	31	21	23	1	3	11	7	3
46	47	22	34	30	20	22	12	7	10	6	2
45	24	27	23	19	15	DE	17	3	5	1	
49	40	36	32	28	24	20	16	2	1	4	0

Cada nodo de información está formado por:

- Ligas anteriores y posteriores dentro del mismo registro
- Ligas a la lista invertida que forma el grupo de identificadores iguales. (CLUSTERING)
- Longitud del nodo almacenado
- Dirección a la primera palabra del nodo de pedido en el archivo (NOPEPIDO)
- Indicador de cambio de línea

PRIMERO = PRIMER DE LA
LISTA DE DESCRIPCIÓN

ULTIMO = ULTIMO DE LA LISTA DE DESCRIPCIÓN

DLE = LONGITUD DEL NOMBRE

DONDE =

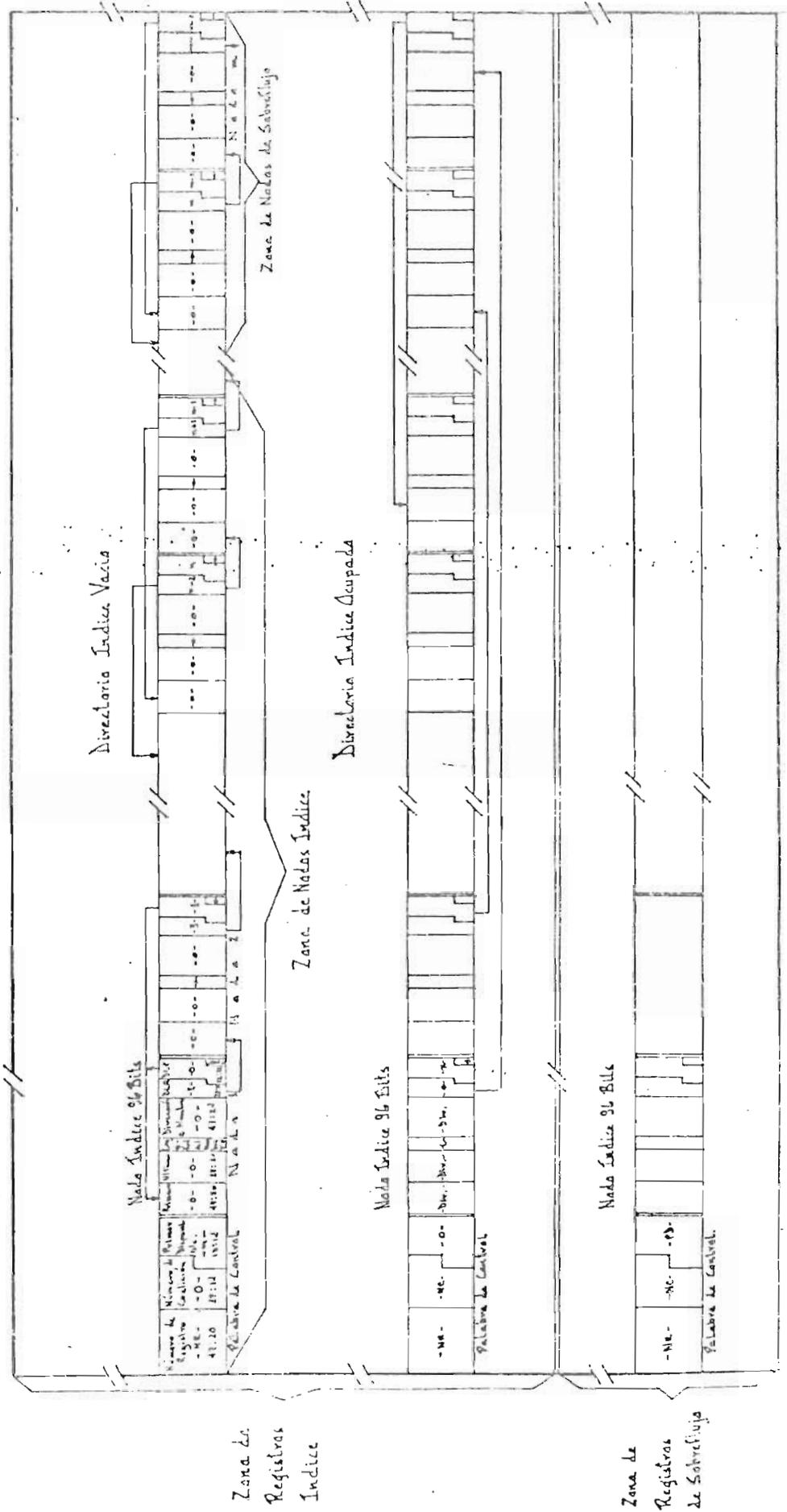
DLA = LIGA ANTERIOR DE W

DLP = LIGA POSTERIOR

0
NUMERO DE COLOCACION
CUANDO HAY CAMBIO DE
LINEA

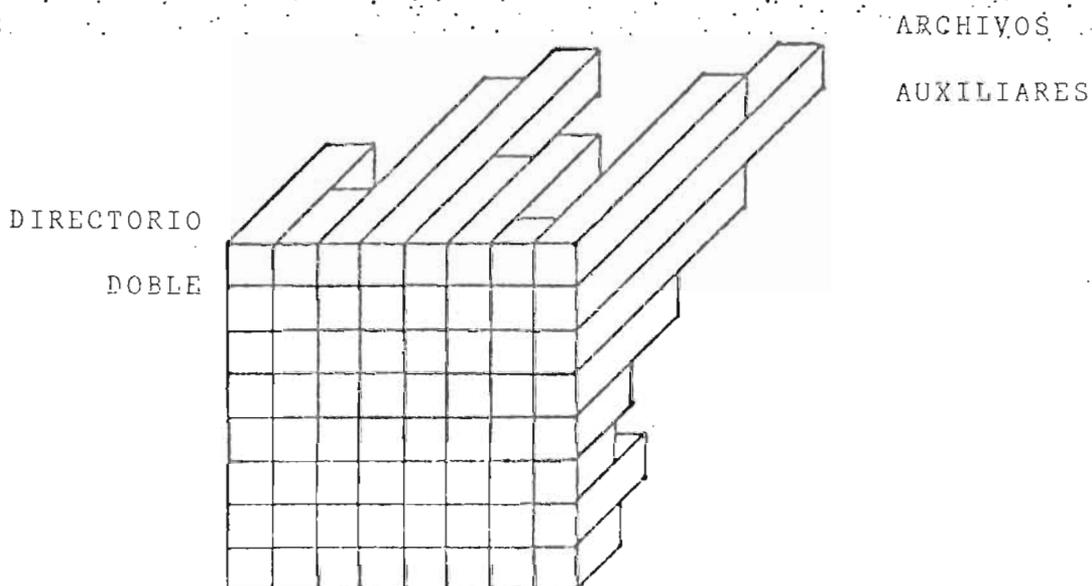
DIRECTORIO

AUTOR / TITULO.



ARCHIVO DESCRIPTOR PEDIDO

Es un archivo que contiene las listas invertidas para alojar los diferentes grupos de nodos que contienen al mismo identificador alfabético (clusters), es decir, convierte al directorio doble en un archivo de tres dimensiones, donde la tercera dimensión es variable según el número de veces que un identificador está almacenado en el archivo.

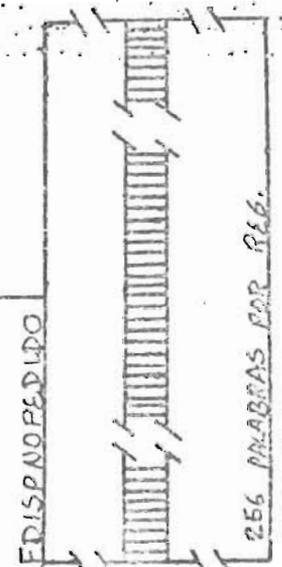
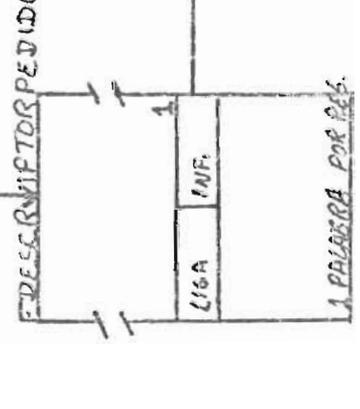
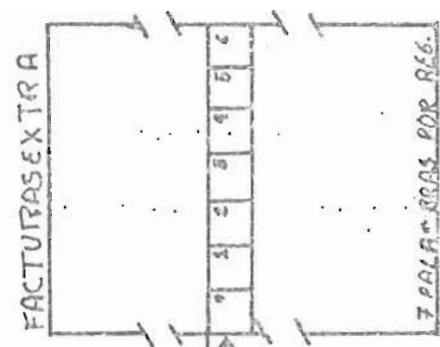
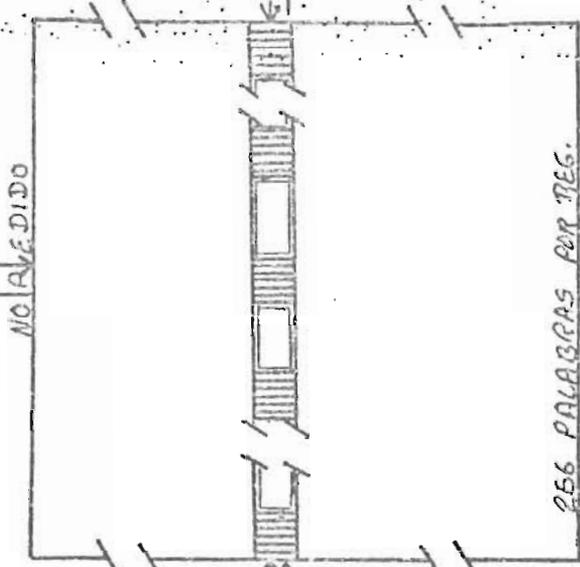
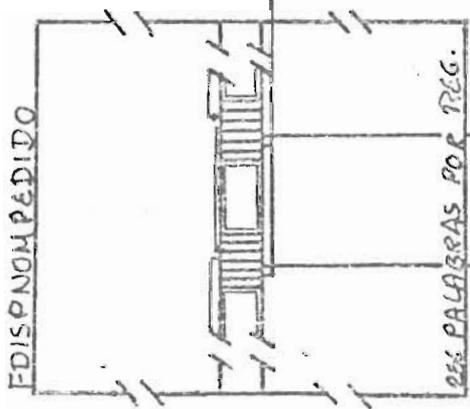
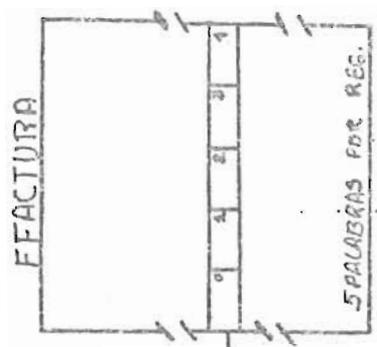
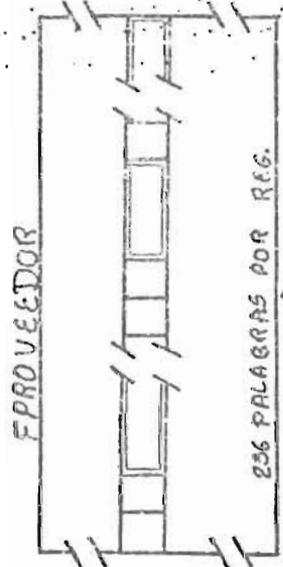
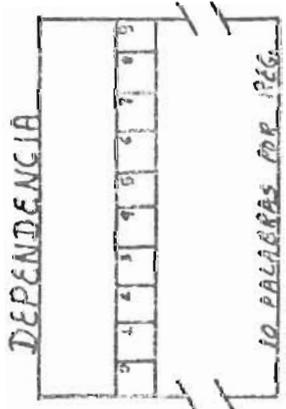


ARCHIVO EN TRES DIMENSIONES

Este archivo tiene registros lógicos de una palabra y cada palabra contiene 24 bits para la zona de liga y 24 bits para almacenar la llave.

47	53	31	35	34	47	23	14	1	41	4	3
16	12	38	31	33	30	22	5	1	1	0	4
15	17	17	33	21	3	50	1	3	3	5	2
44	50	34	32	38	29	30	14	2	3	4	0

DESCUIGA
IWA (I)



DESCRIPCION Y CARACTERISTICAS DE LAS
RUTINAS DENTRO DEL PROGRAMA

La programación del Sistema/Bibliotecas se hizo en forma general y estructurada de tal manera que se pueden agregar módulos para ampliar el programa, sin alterar los que ya funcionan dentro de él.

Los módulos correspondientes a los ^eprovesos necesarios dentro del proyecto de adquisiciones centralizadas, están completamente programados. El programa tiene la base para la creación de los módulos necesarios para la ampliación del mismo y así:

Brindar apoyo al Departamento Técnico de catalogación y clasificación, servicio de información a usuarios, control de estadísticas del acervo bibliográfico, adquisiciones, presupuestos, etc.

En general, brindar a una biblioteca tan grande como la de la UNAM, el apoyo automatizado que fuera requiriendo.

El programa se divide en:

- 1) Rutinas de ^fIntraestructura del programa.
- 2) Rutinas específicas para cada comando del programa.

Las rutinas de infraestructura son aquellas usadas para crear la base del programa, con ellas se capta información, se escribe, se controlan y manejan los archivos. Estas rutinas se programaron en forma general para servir de base a todos los módulos del sistema y son:

a) Procesos auxiliares.-

- Lee datos
- Revisa los datos
- Se comunica con el usuario
- Escribe
- Emite mensajes de error

b) Procesos de manejo de memoria.-

Implementan las diferentes técnicas de manejo de memoria necesarios para almacenar y recuperar la información dentro del Banco de Datos.

c) Procesos para la protección de Archivos

Abre, crea y cierra los archivos.

Rutinas específicas para cada comando del programa.

Cada uno de los comandos del programa, es un módulo ~~del mismo~~ que corre independientemente de los otros siempre apoyado en las rutinas de Infraestructura del mismo.

PROCESOS AUXILIARES
PROCESOS PARA IMPRESION
PROCEDURE ESCRIBE

Rutina por medio de la cual se imprime en papel o terminal, todos los resultados del programa. Existe un contador de líneas (LINEAS) que se incrementa cada vez que se escribe y cuando éste llega al valor máximo de líneas que se pueden imprimir por hoja (NLINEAS), cambia de hoja numerándolas (TIT); si los resultados son por impresora.

En cambio, si los resultados se están imprimiendo por terminal, suspende la impresión emitiendo una pregunta para seguir adelante.

Parámetros

ARCHIVO	Nombre del archivo de impresión
LINEAS	Contador de líneas
TIT	Contador de hojas

PROCEDURE SEND

Siempre inicia el contador de líneas en cero y llama al procedure ESCRIBE, con los parámetros necesarios para imprimir por terminal. Este proceso es usado para la comunicación entre el programa y el usuario por teletipo.

PROCEDURE FECHAC

Proceso que escribe la fecha de máquina en forma compacta: -
DD/MM/AA o en forma extendida MIÉRCOLES 15 de DIC. de 1976
fecha mes año

Parámetros

- F Valor de la fecha
- F=0 Escribe la fecha del día
- F>0 Calcula la fecha del número de días que han transcurrido a partir del 1o. de enero de 1700 (Puede ser también un valor almacenado de fecha de máquina en otra ocasión).
- B Variable lógica que devuelve la fecha compacta cuando es verdadera
- P Es el apuntador al arreglo donde se desea escribir la fecha

PROCEDURE SACFECHA

Las fechas que son captadas con el PROCEDURE CHECAFECHA son desempacadas y enviadas a impresión por medio de ésta rutina, siempre las deja en el apuntador al arreglo LINEA, de la forma DD/MM/AA

PROCEDURE ENCABEZADO

Genera los encabezados para cada uno de los programas que son corridos dentro del sistema

Parámetros

A	Número del título que lleva el encabezado
B	Número de la dependencia a la cual se va a entregar el listado. Si B=0, se omite el nombre de la dependencia
IMPRESOR	Nombre del archivo de impresión
LINEAS	Contador de líneas
TIT	Contador de hojas

PROCEDURE CENTRA CENTRA

Proceso que ajusta los textos dentro de un espacio dado para dejar margen uniforme a los lados del mismo, separando por palabras completas e introduciendo blancos adicionales entre ellas cuando sea necesario. En cada llamado, ajusta un renglón y deja actualizados los apuntadores para el siguiente llamado al procedure.

Parámetros

PL	Apuntador al arreglo donde se deja el texto
PAY	Apuntador a la palabra del arreglo que contiene el texto por escribir
CUAL	Longitud de líneas que se desea centrar.

- LC Largo del identificador que se va a centrar, es-
te valor regresa alterado, devolviendo el núme-
ro de caracteres que faltan por escribir
- D Devuelve el número de caracteres que fueron a--
justados

PROCEDURE ERROR

Este proceso emite el mensaje de error correspondiente, depen-
diendo del valor de ERRONEO, emite la imagen de tarjeta seña--
lando con un asterisco el lugar donde está localizado el error

Cuando el valor de erróneo es cero, emite la imagen del paque-
te de tarjetas que contienen error, copia tarjetas erróneas en
un archivo de disco para su posterior corrección.--

Parámetros

ERRONEO Número del error que se presentó

PROCESOS PARA LA LECTURA DE DATOS

PROCEDURE LEETARJETAS

Procedimiento que lee las tarjetas en Binario, traduciendo in-
mediatamente a código EBCIDIC para ser procesadas posteriormen-
te, así evita que la lectora suspenda la lectura de datos cuan-
do alguna tarjeta tiene perforado un carácter inválido.

PROCEDURE FILTRA1

Esta rutina lee todos los datos que se captan por tarjeta, verificando que:

- Todas las tarjetas pertenecientes a un paquete, estén en orden y con el número de datos esperado.
- La sintaxis de los datos sea correcta, es decir, si un identificador es numérico, revisa que realmente esté formado por números, que un código sea un código dentro del programa, emite mensajes de error en caso contrario.

Es un proceso de revisión dinámico y encuentra cada uno de los datos buscando un separador, en caso que un dato venga dividido en varias tarjetas lo junta introduciendo blanco como separador.

Parámetros

FILTROS Es un procedure consistente en un CASE COMAS - OF, donde cada instrucción del mismo, indica que verificación se necesita.

NOCOMAS Número de identificadores por revisar

TEMP Archivo donde se encuentran almacenadas las
tarjetas de datos.

EOF Etiqueta de fin de archivo.

Para formar las rutinas que se pasan como parámetro a FILTRA1,
existen una serie de procesos que verifican cada uno de los ti
pos de datos que se espera.

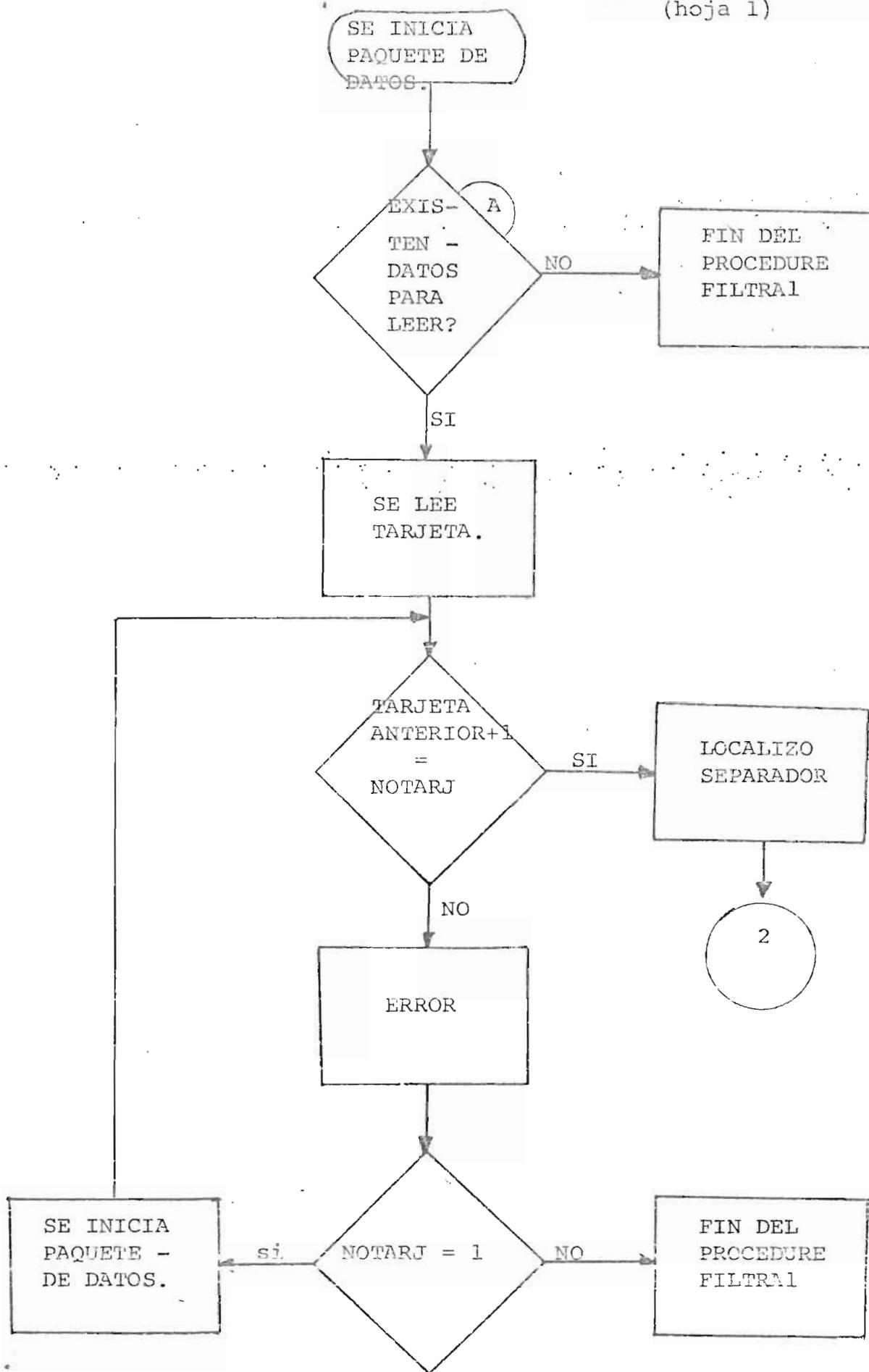
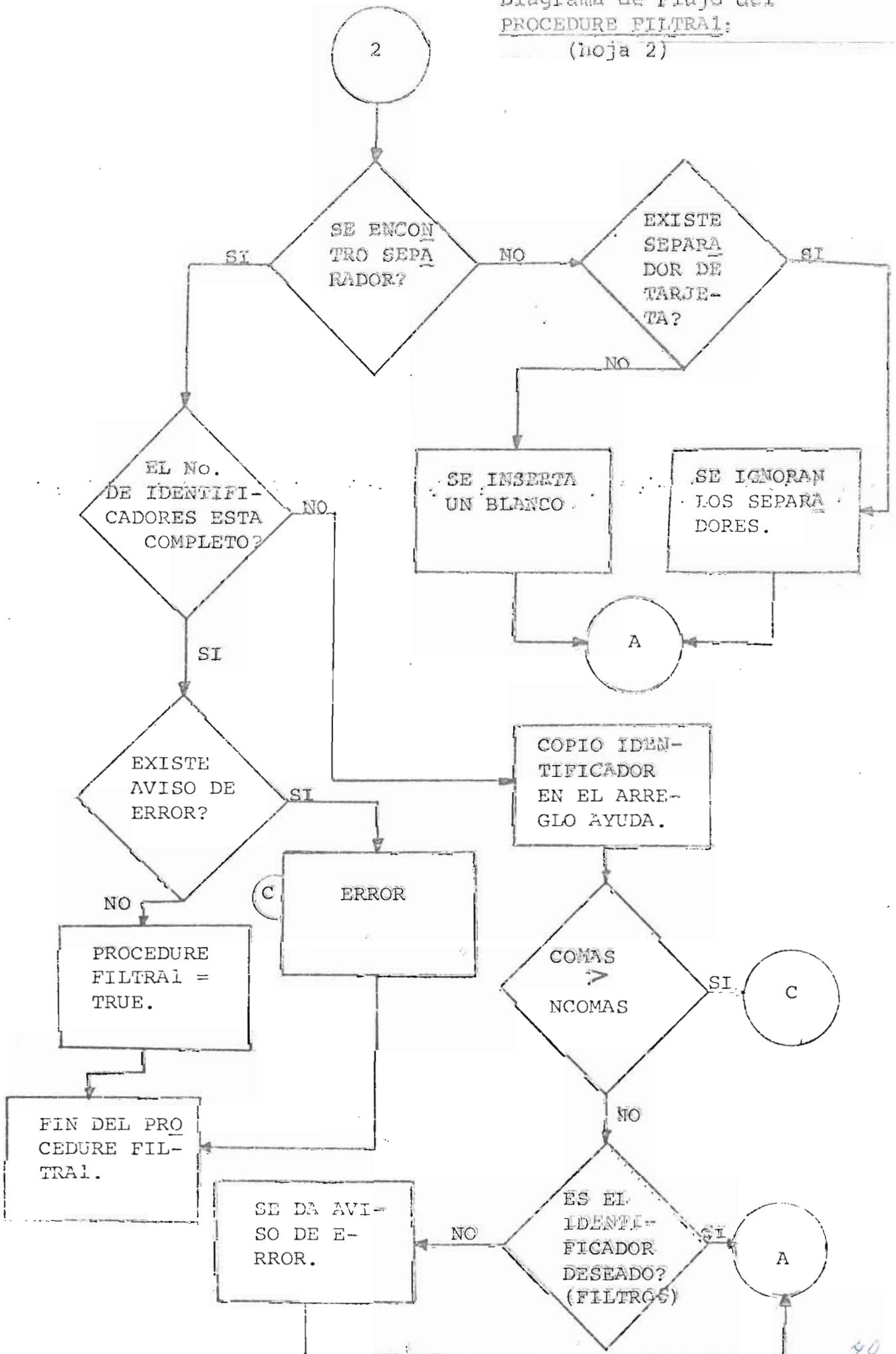


Diagrama de Flujo del
PROCEDURE FILTRAI:
 (hoja 2)



PROCEDURE CHECA

Revisa el identificador alfanumérico, cuando es más largo que la longitud permitida lo trunca y manda mensaje pero deja pasar el dato. Copia el identificador al arreglo en el cual debe quedar, omitiendo los blancos sobrantes.

Parámetros

P	Apuntador al arreglo donde se deja la cadena
LC	Largo de la cadena después de ser revisada
MAX	Número máximo de caracteres que puede contener la cadena

PROCEDURE CHECATIFO

Revisa los códigos que son aceptados en el sistema; cuando el código tiene más de seis caracteres, se localizan los primeros seis caracteres y después se compara el resto de la cadena, en caso de que ésta no coincida se repite la operación.

Parámetros

VARIABLE Devuelve el número correspondiente al código para ser almacenado dentro del Bando de Información. Si el valor de la variable es -1, significa que no encontró el código solicitado.

ARREGLO Es el nombre del "VALUE ARRAY" que contiene esos códigos.

ERRONEO Mensaje de error.

PROCEDURE CHECAFECHA

Revisa los rangos de los valores dentro de la fecha y la transforma en un número consistente en el número de días transcurridos desde el 1/I/72

PROCEDURE CHECANUMERO

Revisa que el identificador sea realmente numérico, suprime los puntos decimales y asigna el valor del identificador cuando está dentro del rango permitido.

Parámetros

MAX	Rango máximo del número que es aceptado
DESCNO	Valor numérico del identificador
ERRONEO	Mensaje de error

PROCESOS PARA CAPTAR DATOS POR TERMINAL

En general los procesos para leer por teletipo hacen una pregunta para solicitar el dato, el cual es revisado inmediatamente - por los mismos procesos de revisión usados para el PROCEDURE FILTERA 1.

Los procesos implementados para leer por teletipo son:

LECTORTIP	Para captar códigos
LECTOR	Para cadenas alfanuméricas
LECTORNUM	Para identificadores numéricos
LECTORFECHAS	Para leer fechas

Parámetros

MAX	Máximo permitido para el identificador
MENSAJE	Número de mensaje para solicitar el dato
NUMCONT	Número para prender una bandera de aviso
VARIABLE	Devuelve el valor numérico del identificador (LECTOR/TIP,LECTORNUM)
ERRONEO	Mensaje de error
EOF	Etiqueta de fin de archivo

RUTINAS PARA EL MANEJO DE ARCHIVOS CON MEMORIA DINAMICA

Los archivos implementados para el manejo de memoria dinámica, siempre son referenciados desde los directorios de manejo de información del sistema. Los archivos están formados por registros lógicos de 256 palabras de longitud y cada nodo ocupa únicamente el número de palabras necesarias para su alojamiento.

IMAGEN DE UN ARCHIVO EN EL CUAL SE ALOJA LA INFORMACION EN FORMA DINAMICA.

PROCESOS PARA ALMACENAR LA INFORMACION

INSERTA

Este proceso localiza el área de memoria disponible dentro del archivo para alojar la información. Se hace de dos maneras diferentes:

- 1) Cuando localiza un área nueva de memoria, lee el último registro usado del archivo y encuentra la primera palabra disponible en él, inserta el nodo y actualiza el apunta-

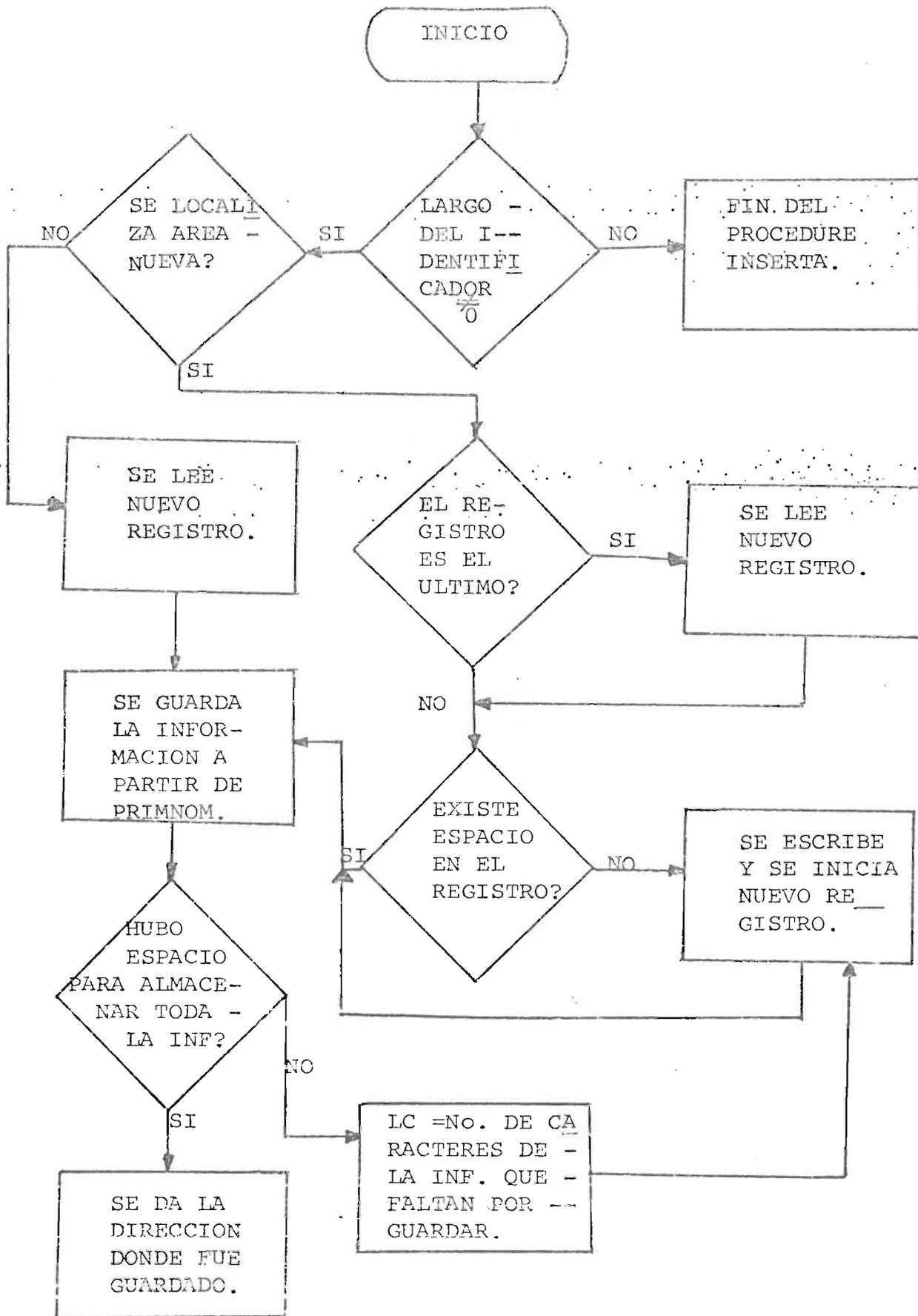
dor a la primera palabra disponible en el registro; si la longitud del identificador es mayor que el número de caracteres disponibles en el registro, almacena únicamente éstos e inicia un nuevo registro para copiar el resto de la información.

- 2) Cuando usa la misma área de memoria de un nodo que se da de baja, lee el registro en donde se encuentra el nodo anterior y almacena en el mismo espacio, al nodo nuevo. Este debe ser de menor o igual longitud que el nodo dado de baja. En ambos casos, en los primeros ocho bits de la primera palabra del nodo, se guarda la longitud del mismo.

Parámetros

B	Es una variable lógica que indica si se usa un área de memoria nueva o no. B = Verdadero, usa área nueva de memoria.
ANOM	Arreglo que contiene la cadena alfabética por almacenar.
LC	Largo de la cadena alfabética.
NUM	Dirección en la cual es alojado el nodo en el archivo.
NOMBRE	Archivo donde es alojada la información
ANOMBRE	Arreglo del archivo NOMBRE
ULTIMONOMBRE	Ultimo registro útil del archivo NOMBRE

Diagrama de Flujo del PROCEDURE INSERTA:



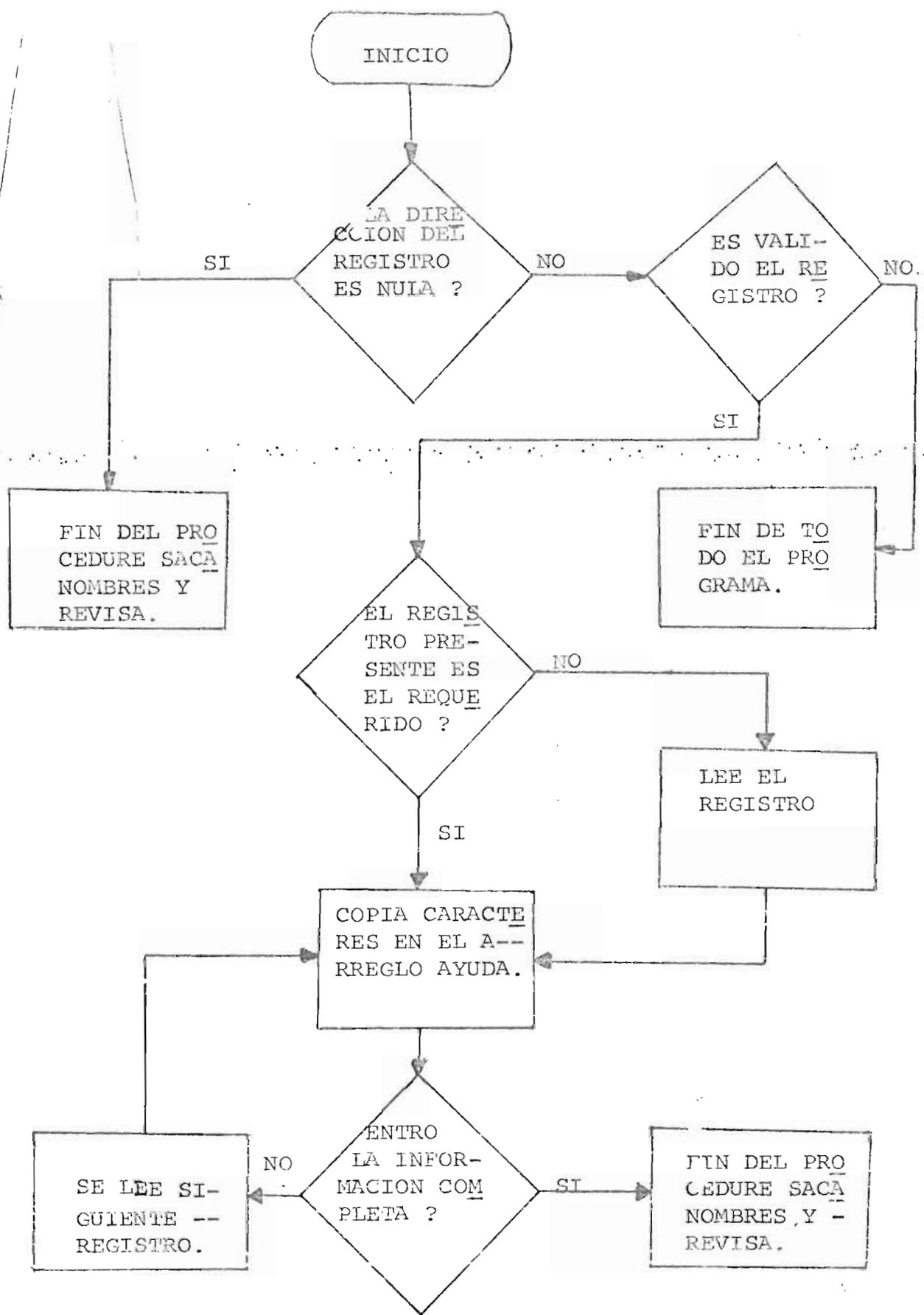
PROCESOS PARA RECUPERAR INFORMACION

PROCEDURE SACANOMBRES

Este proceso se usa en unión del procedure Revisa, el cual verifica que la dirección solicitada se encuentre dentro de los registros útiles del archivo. En caso afirmativo se copia la información del nodo a partir de la dirección solicitada y por el largo especificado en los primeros ocho bits del mismo (SACANOMBRES).

Parámetros

ANOM	Arreglo en el que se copia el identificador.
LC	Largo del identificador
NUM	Dirección donde está almacenado el nodo
NOMBRE	Archivo que contiene la información
ANOMBRE	Arreglo del archivo NOMBRE
ULTIMONOMBRE	Ultimo registro útil del archivo NOMBRE



PROCESOS PARA LA IMPLEMENTACION DEL DIRECTORIO DOBLE

Para alojar información dentro del Directorio Doble, es necesario seguir una rutina de cuatro procesos, usando en ocasiones los valores lógicos que devuelve la misma rutina.

LOCALIZA.- El valor del proceso es verdadero cuando encuentra el identificador o no existe éste.

INSERTAENDIRECTORIO.- El valor del procedure es verdadero cuando se usa un nodo nuevo dentro del directorio.

ASIGNADESCRIPTOR.- El valor es verdadero cuando un nuevo registro es usado.

LLENANODO.- Siempre que es llamado, aloja la información en el nodo del Directorio Doble.

Para dar acceso a la información se usa la rutina de LOCALIZA.

PROCEDURE CREADIRECTORIO

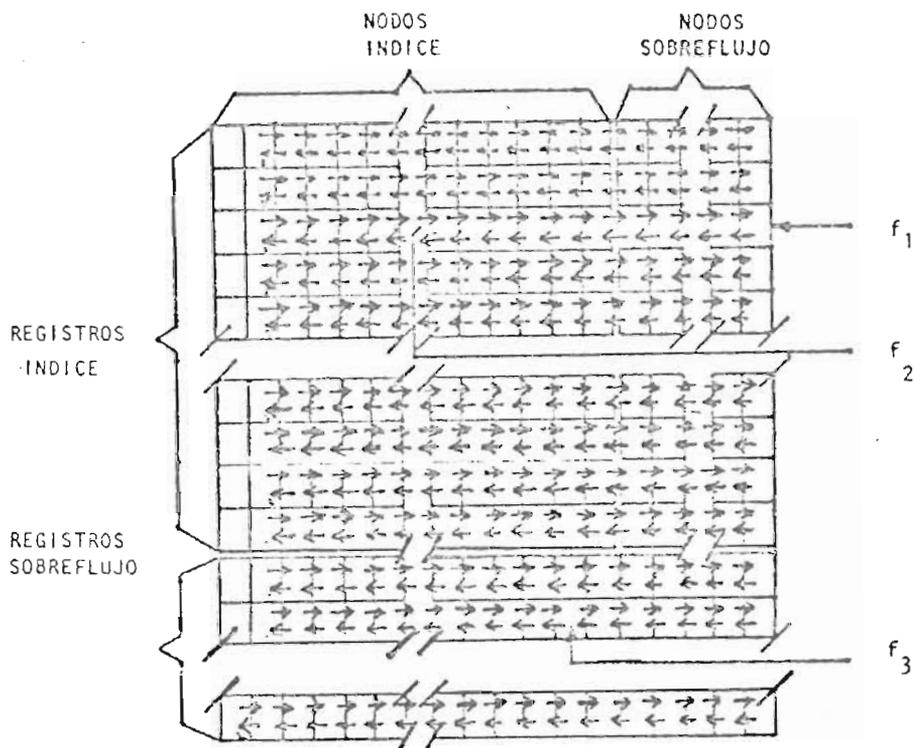
Este proceso liga todos los nodos del archivo, con una doble ligadura. La liga posterior une los nodos de mayor a menor; la liga anterior, de menor a mayor. En la palabra cero guarda el número de registro correspondiente y coloca como primer nodo -

disponible de sobreflujo, al primer nodo de la lista.

Parámetros

ARCHIVO Archivo que se usa como directorio
ARREGLO Arreglo del archivo
NOBLOQUES Número de registros que tiene el archivo.
Se calculan como la suma de los registros índice y los registros de sobreflujo.

Imagen del Directorio Doble después de ejecutado el procedure.



PROCEDURE INSERTAENDIRECTORIO

Este proceso trabaja en unión con el procedure CONSIGUEDISPONIBLE. Estos dos procesos dejan con las ligas necesarias al nodo que se va a usar.

Verifica si el nodo encontrado por el procedure LOCALIZA, es virgen en cuyo caso, únicamente desliga el nodo y lo deja como último de una lista nueva y devuelve el valor del procedure como verdadero. En caso contrario, arregla los apuntadores del nodo para continuar la lista a otro disponible para almacenar la nueva información.

Para conseguir el nuevo nodo, verifica si todavía quedan otros disponibles en el mismo registro, colocando al primero como último de la lista de la dispersión. Como primer nodo disponible, queda al que apunta la liga posterior del mismo.

En el caso que no haya más nodos libres dentro del registro, es necesario encontrar uno en el área general de sobreflujo.

Para ésto, en la palabra cero se lleva un contador de colisiones dentro del registro y aplicando la función:

INFOBILA

$f_3 = (\text{Número de coliciones} + \text{Número de Registro}) \text{ MOD SOBREFLUJO}$
localiza el registro donde se continuará la lista, este valor es guardado en el lugar correspondiente a la liga posterior del nodo y se prende un bit para indicar que se efectuó cambio de línea.

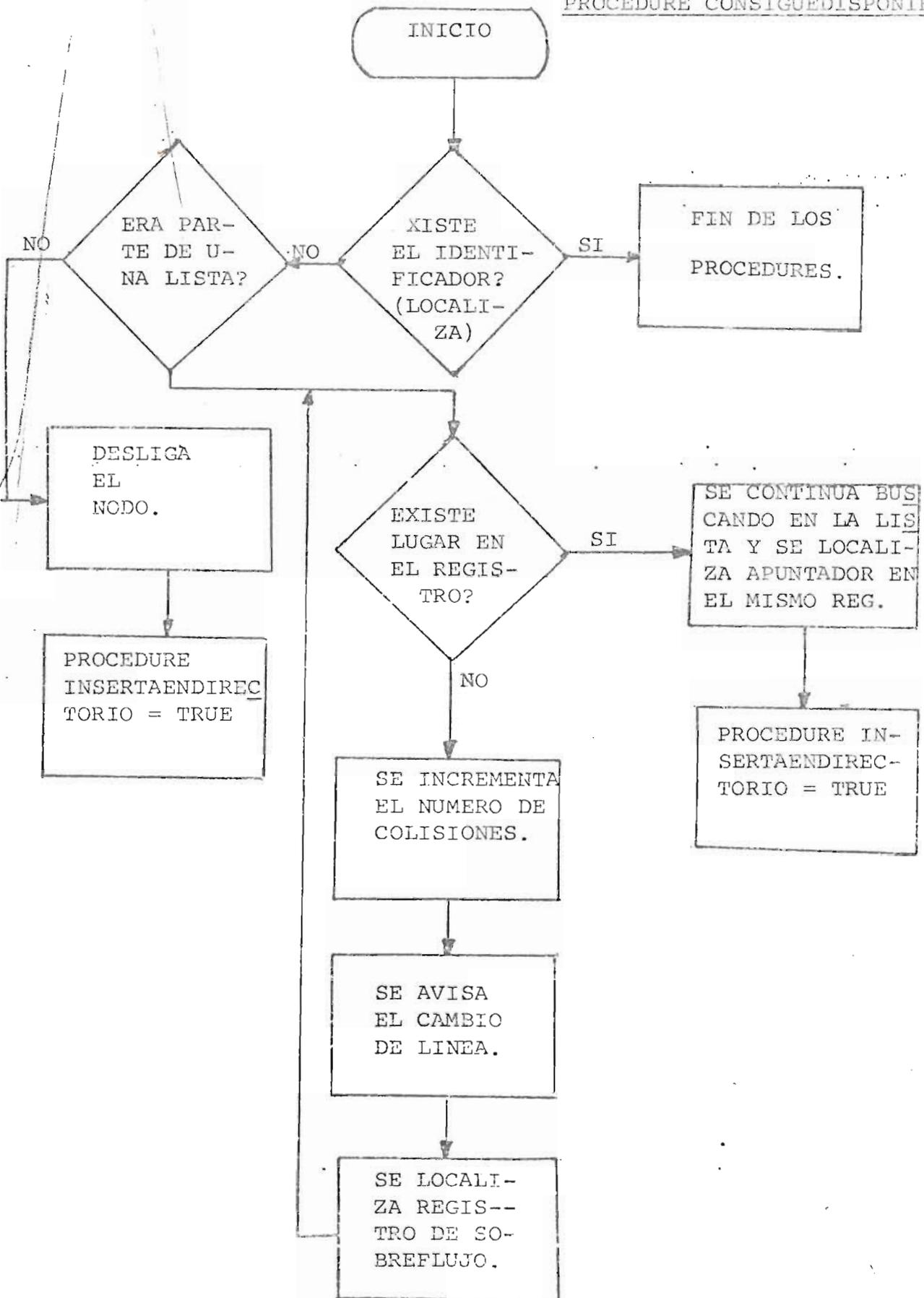
En el nuevo registro se encuentra el nodo paralelo al último de la lista y se repite el proceso para conseguir uno disponible.

Una vez localizado el nodo, el procedure devuelve el valor VERDADERO

Parámetros

BUFFER	Registro donde se localiza el nodo
APUNT	Palabra del registro donde se localiza el nodo
ARCHIVO	Nombre del archivo que se usa como directorio
ARREGLO	Arreglo del archivo
SOBREFLUJO	Número de registros de sobreflujo
MODULO	Número de nodos índice
CUANTOS	Número total de nodos índice del archivo

Diagrama de Flujo de los
PROCEDURE INSERTAENDIRECTORIO
PROCEDURE CONSIGUEDISPONIBLE.



PROCEDURE LOCALIZA

Proceso que encuentra el nodo correspondiente en el directorio doble al identificador alfabético que se desea alojar o recuperar en el Banco de Información. Para efectuar ésta rutina, se procede de la siguiente manera:

1) Aplica una transformación a la cadena alfabética para convertirla en un número (procedure DISPERSION) consistente en sumar cada 17 bits de ella en forma secuencial.

2) Encuentra el registro correspondiente en el directorio doble aplicando la función de dispersión:

$$f_1 = \text{DISPERSION DIV MODULOS}$$

Localiza la palabra en el registro con una segunda función de dispersión:

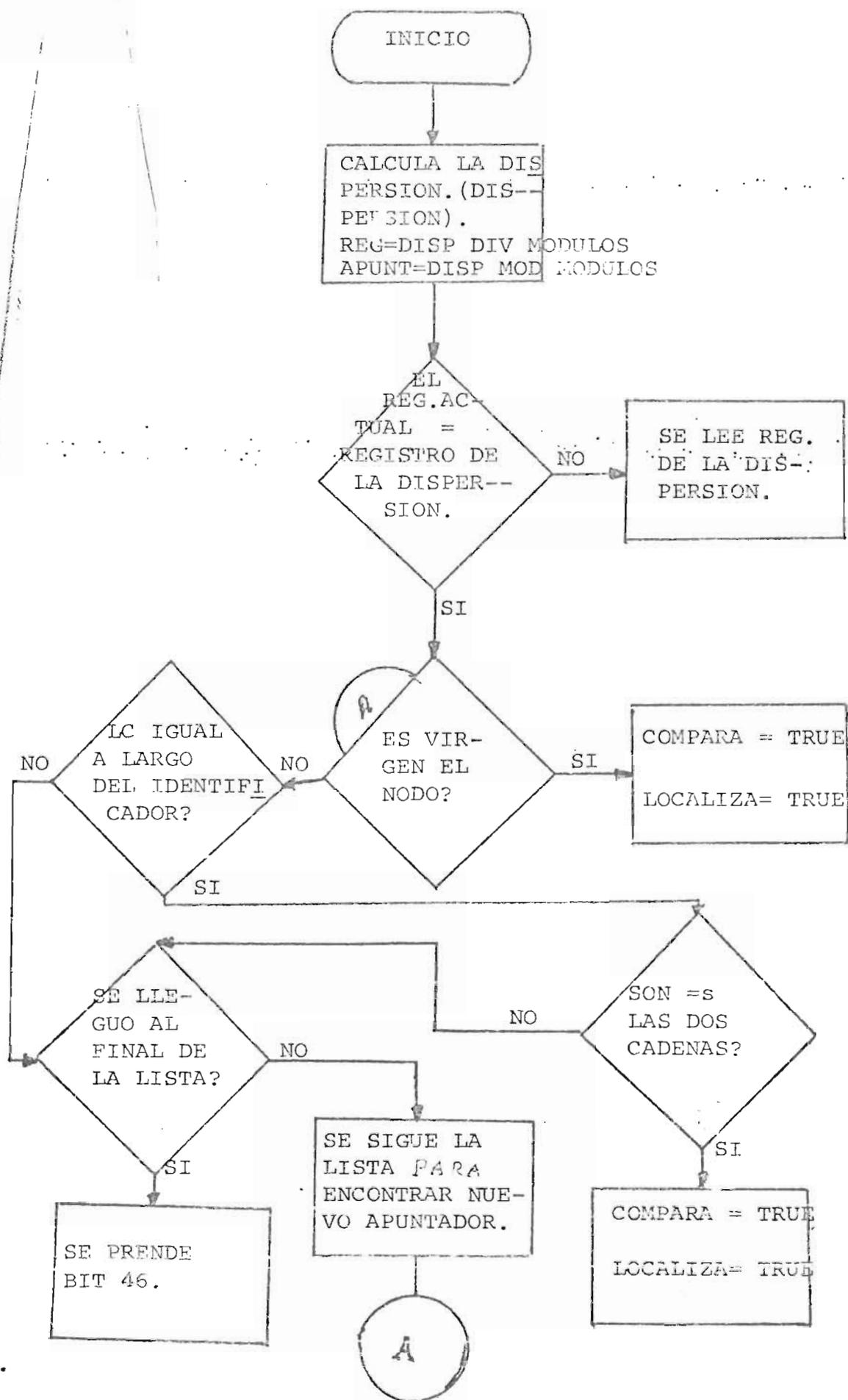
$$f_2 = \text{DISPERSION MOD MODULOS}$$

3) Una vez localizado el nodo en el directorio, si está ocupado y la longitud del identificador guardada en él es igual a la longitud de la cadena alfanumérica; compara entonces las dos cadenas y si son iguales (PROCEDURE COMPARA), el valor de LOCALIZA es verdadero y el bit 47 de apunt está prendido.

Parámetros

MODULO	Número de nodos índice por registro
ARCHIVO	Archivo del directorio doble
ARREGLO	Arreglo del archivo del directorio doble
BUFFER	Número de registro en el directorio doble.
APUNT	Número de palabra del registro en el directorio doble
ARR	Arreglo que contine el identificador alfanumérico
LC	Largo de la cadena
CUANTOS	Número de registros índice del directorio doble
NOMBRE	Archivo que tiene almacenadas las cadenas alfanuméricas
ANOMBRE	Arreglo del archivo Nombre

Diagrama de Flujo de:
PROCEDURE LOCALIZA y
PROCEDURE COMPARA.



PROCEDIMIENTOS PARA FORMAR LAS LISTAS INVERTIDAS
LOS GRUPOS DE IDENTIFICADORES ALFABETICOS

RMAN -
TERING)

PROCEDURE ASIGNADESCRIPTOR

Rutina para implementar las listas invertidas. El programa
con una característica diferente; en lugar de un solo modo de
información contenga una llave, se hace en forma de lista y ca
da nodo puede contener tantas llaves (número de nodos como -
se desee, con objeto de aprovechar el mismo espacio para
alojar más de una información. Esto resulta útil para
el que nos ocupa, pues en la localización de títulos
o temas es lógico esperar que más de un identificador igual --
sea alojado dentro del Banco de Información.
El procedure devuelve valor verdadero cuando un registro
es usado.

Parámetros

DESCRIPTORES	Archivo donde se almacena la lista invertida.	lista in
ULTIMODESCRIPTOR	Ultimo registro útil del archivo	del archi
ADESCRIPTORES	Arreglo sobre el archivo de descripciones	descrip--
NODOS	Número de llaves que se van a almacenar en un mismo nodo de información	en un mis
TU	Ultimo nodo de la lista se va a aumentar	se va a -
B	Identificador Booleano que es verdadero cuando se incrementa el valor del último descriptor	es verdade

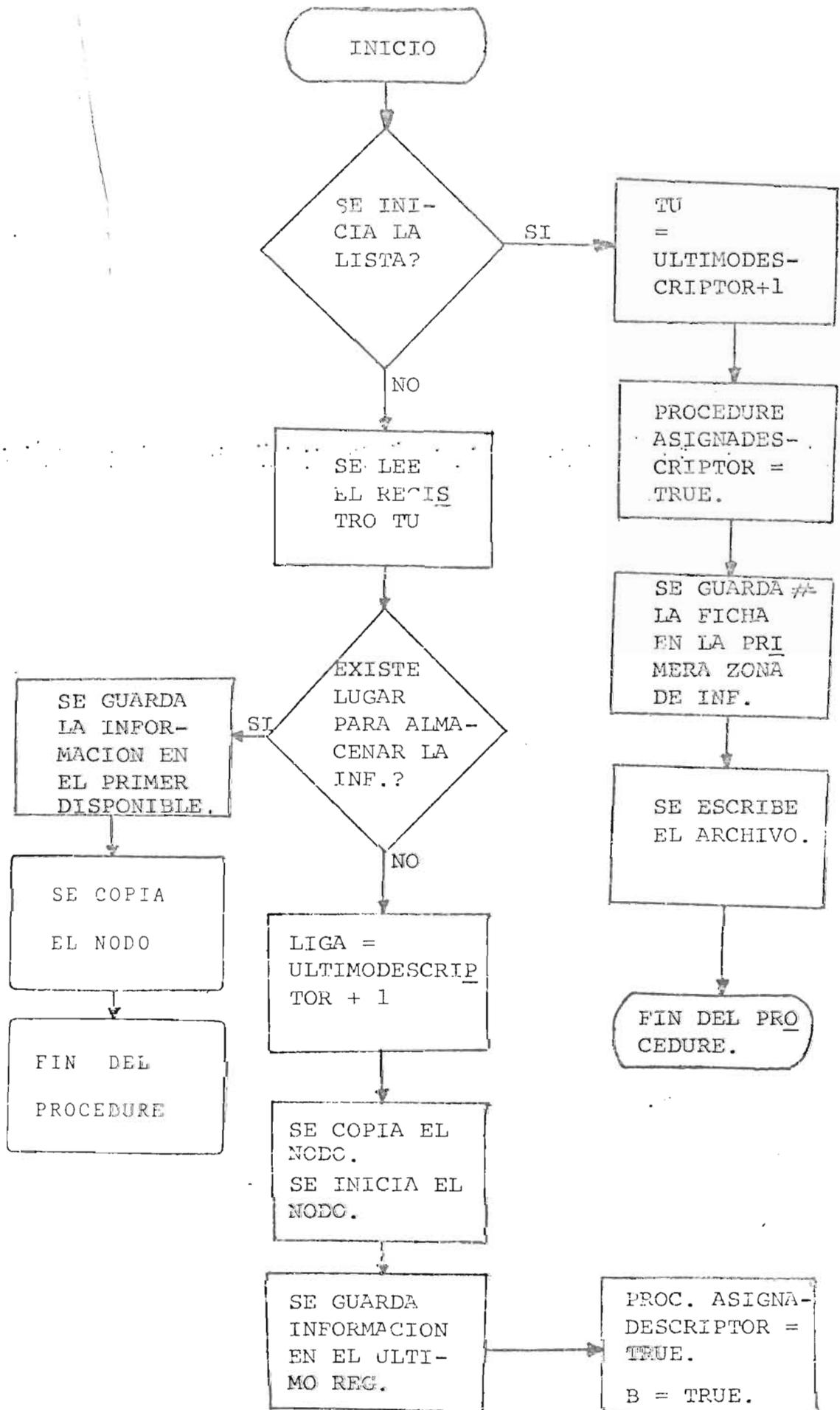
NUMFICHA

Llave que se almacena dentro del nodo

(*) FUNDAMENTAL ALGORITHMS

DONALD E. KNUTH

Diagrama de Flujo del
PROCEDURE ASIGNADESCRIPTOR:

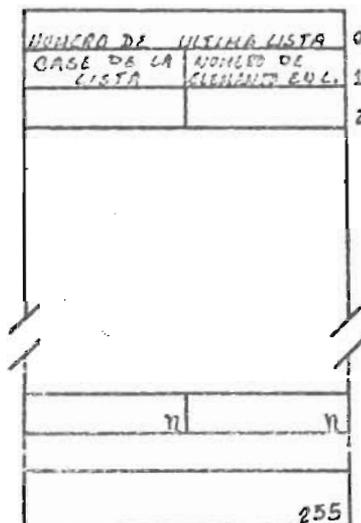


PROCEDURE INTERSECTA

Este proceso está formado por un conjunto de rutinas programadas en forma general que intersecta hasta 255 listas diferentes de identificadores numéricos.

Para la implementación de ésta rutina se usan dos arreglos de 256 palabras cada uno y un archivo de maniobra para almacenar información de 256 palabras por registro lógico.

Uno de los arreglos es usado como tabla de referencia para localizar las diferentes listas. La palabra cero, contiene el número de la última lista que se desea intersectar, las restantes 255, son divididas en dos, en la primera parte se guarda la dirección en donde empieza la lista correspondiente al número de palabra en el arreglo y en la segunda parte, el número de identificadores que forman la lista.



El otro arreglo guarda los identificadores de cada lista en forma secuencial y cuando se acaba, lo escribe en el archivo iniciando un nuevo registro.

8	9	15	25	39	26	2	3	9	32	17
19	3	7	6	8	7	6	15	11	22	36
12	1									
										

Para efectuar la intersección de las listas se hace en cuatro etapas:

PRIMERA.- Se inician los dos arreglos para recibir las listas.

SEGUNDA.- Dentro del segundo arreglo, se van guardando los identificadores numéricos de cada lista en orden creciente, reacomodándolos cuando ésto es necesario, por el método de la burbuja.

Guarda en la palabra correspondiente al número de lista en el primer arreglo o ARREGLO TABLA, los indicadores en donde empieza y cuantos identificadores contiene la lista.

TERCERA.- Una vez que todas las listas han sido armadas en el archivo, se procede a interseccionarlas. Para esto se va comparando nodo a nodo entre la lista de resultados que para empezar es la primera y la que se está revisando. En la lista que contiene el valor más pequeño se pasa a comparar el siguiente nodo. Si los dos nodos son iguales, se genera una nueva lista de resultados que después pasa a interseccionar con la siguiente lista del arreglo tabla, continúa el proceso hasta que se agotan las listas o cuando la lista de resultados es nula.

Devuelve como resultado del procedure un número de nodos que cumplieron con la intersección.

CUARTA.- En esta etapa el procedure toma un valor lógico el cual es FALSO mientras existan nodos en la lista de resultados.

Parámetros

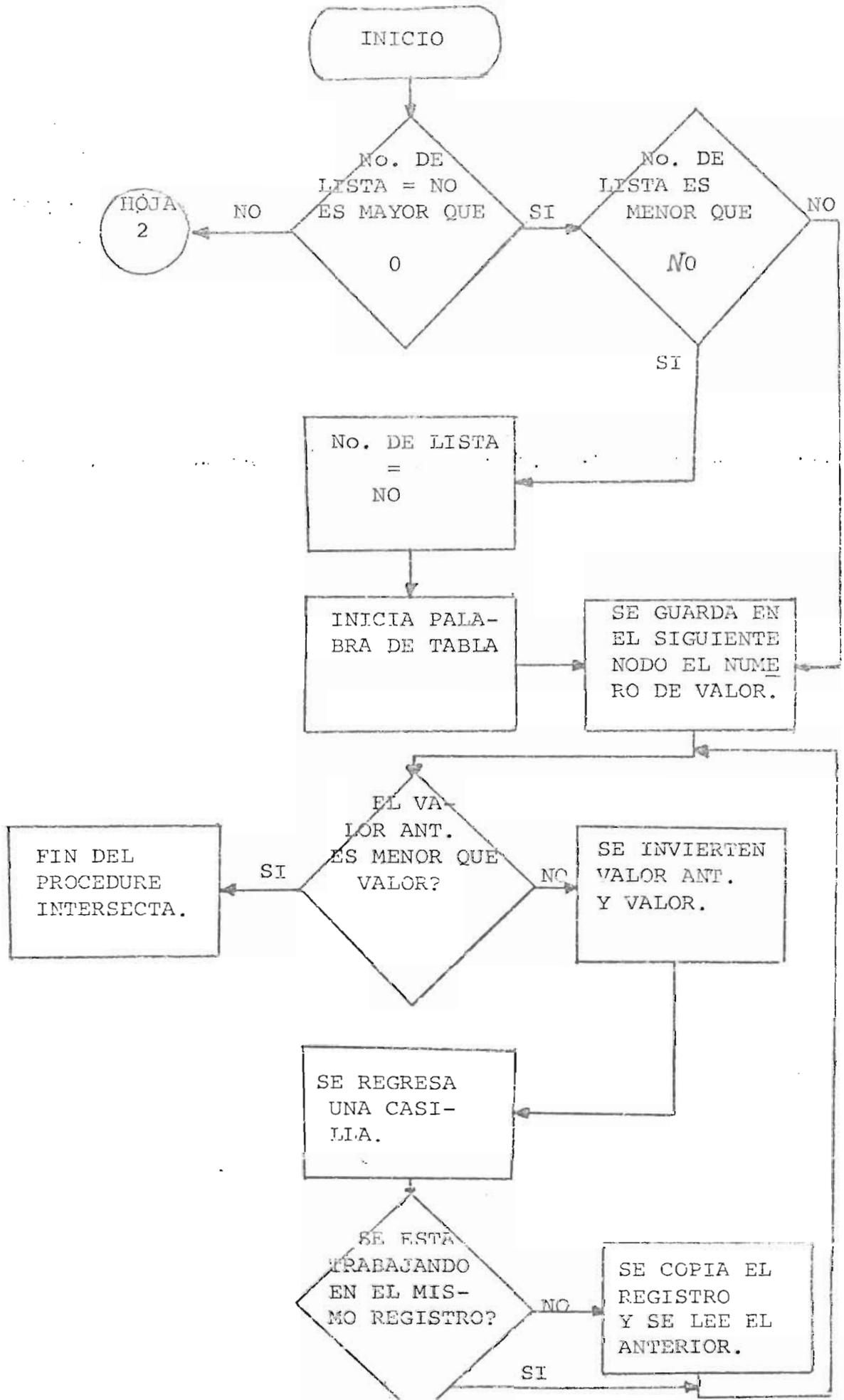
NO Dependiendo del valor le indica cuál es la etapa a --
realizar:

NO -2 Inicia los arreglos

NO= -1 Hace la intersección

NO= 0 Recorre uno a uno los nodos de la lista de --
resultados

Diagrama de Flujo del
PROCEDURE INTERSECTA:



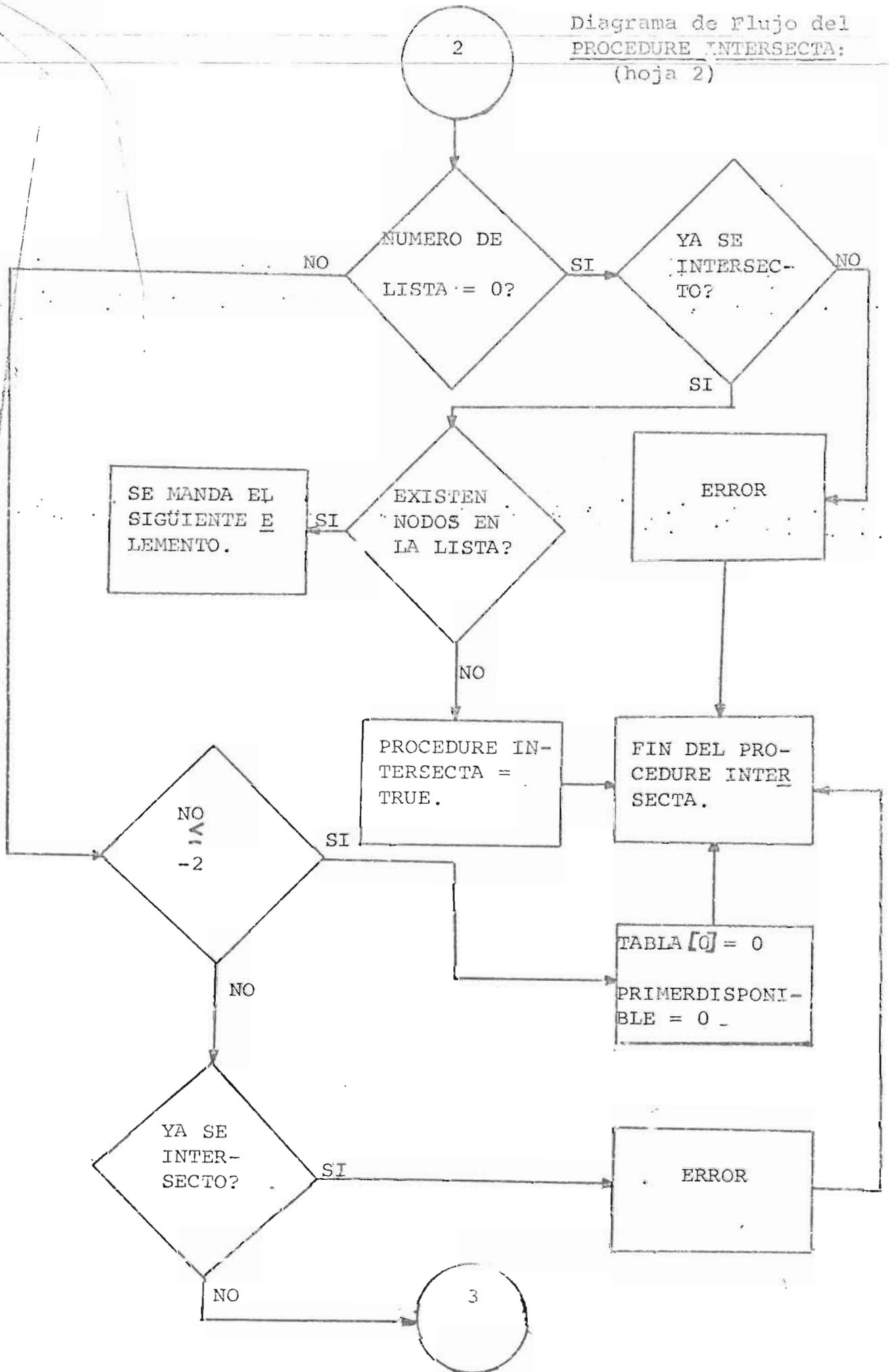
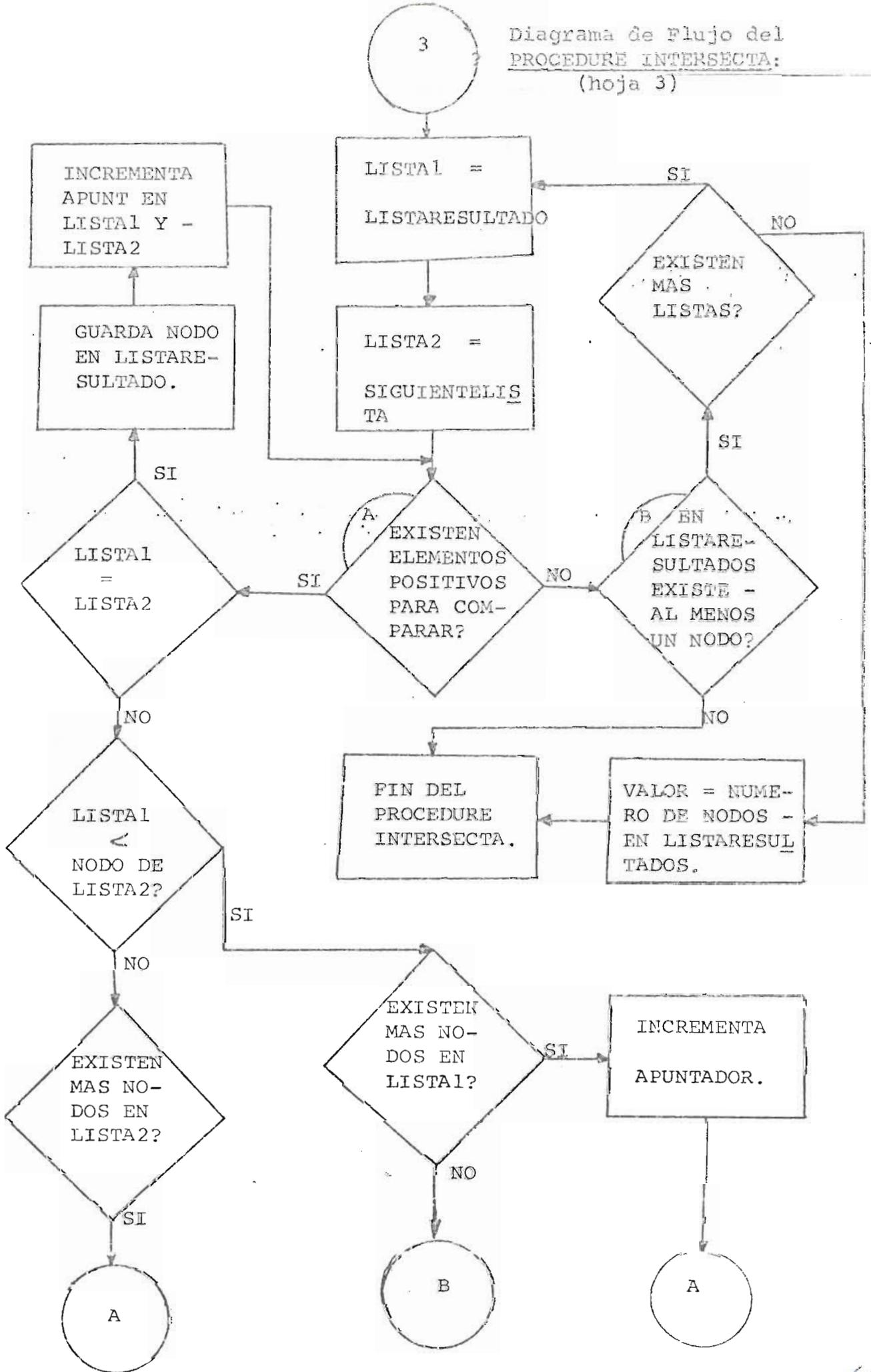


Diagrama de Flujo del
PROCEDURE INTERSECTA:
 (hoja 3)



VALOR Para $NO < 0$ el VALOR no es tomado en cuenta.
NO = 0 VALOR devuelve el valor almacenado dentro del nodo de la lista de resultado.
NO > 0 Inserta en la lista número NO el VALOR en el siguiente nodo.

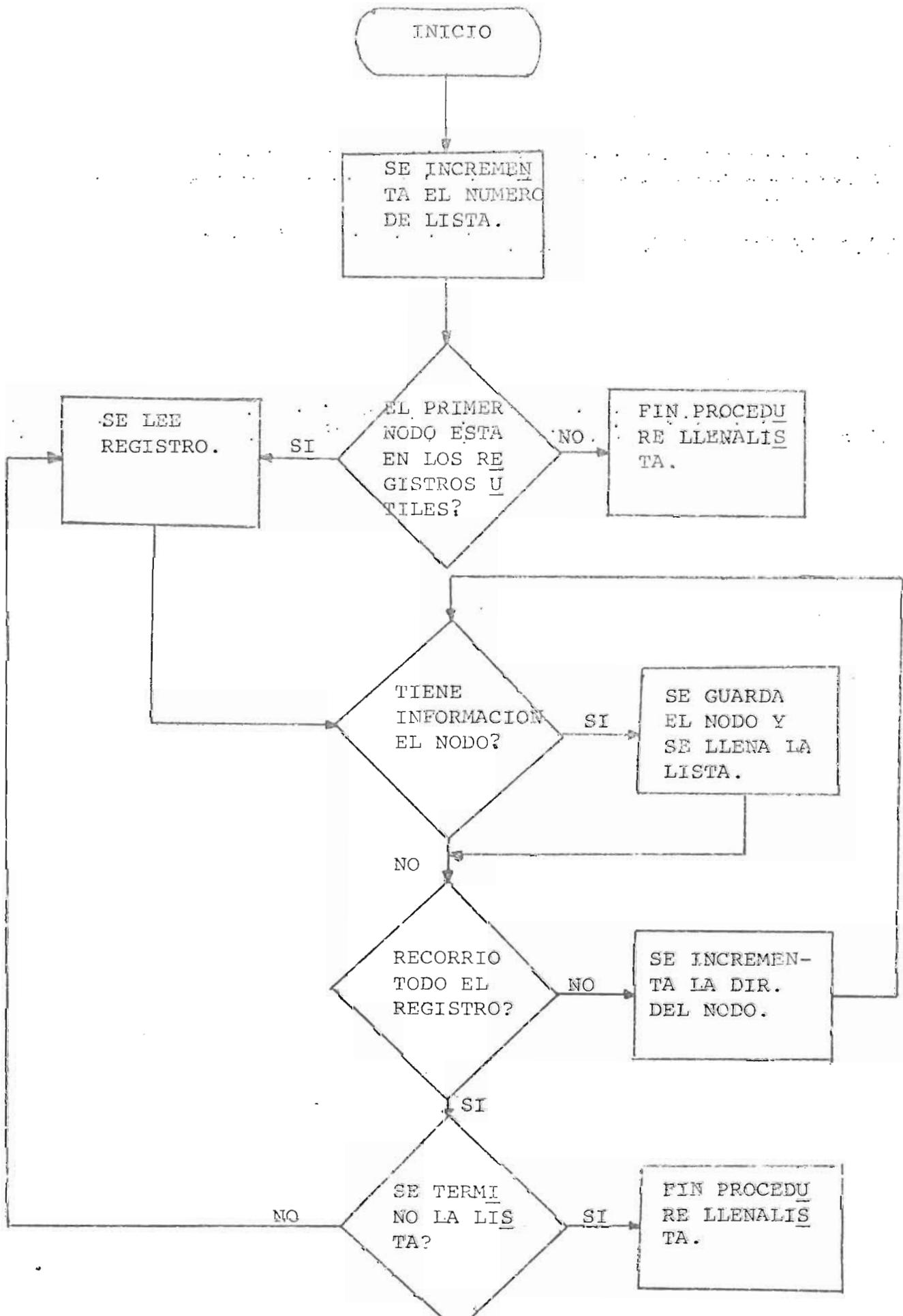
PROCEDURE LLENA LISTA

Por medio de este proceso le van llenando las listas diferentes que necesita el procedure INTERSECCION. Incrementa el número de lista y recorre el archivo descriptores, localiza los nodos de cada identificador pasándolos al procedure interseccion.

Parámetros

N	Número de lista
PRIMER	Primero nodo en la lista del archivo descriptores
ARCH	Archivo que contiene las listas de los descriptores
ARR	Arreglo sobre el archivo
EOF	Etiqueta de fin de archivo

Diagrama de Flujo del
PROCEDURE LLENALISTA;



69

PROCESOS PARA LA PROTECCION DE ARCHIVOS

Todos los archivos dentro del sistema son creados en forma específica. Una vez creado el archivo debe abrirse y cerrarse verificando que esté correcto.

PROCEDURE CREARCHIVOS

Este procedimiento inicia los archivos modificando el MUYSE para dejarlos permanentes en disco y almacena la versión de creación en la palabra dos de protección del registro.

Parámetros

- F El nombre del archivo que se desea crear
- ARR El arreglo de control del archivo
- OP El número del archivo que se está creando para:
- OP > 0 Es un archivo de datos temporal
- OP < 0 Es un archivo fijo del sistema
- OP = 0 Es el archivo de control

Existen procedimientos específicos para crear los archivos fijos del sistema, a saber:

INICIAFACTEXTRA

INICIADISPERSIONPEDIDO

INICIADISPERSIONNOMPEDIDO Se crea el directorio doble (CREADIRECTORIO)

INICIADESCRIPTORPEDIDO

INICIAPROVEEDORES

INICIADEPENDENCIAS

PROCEDURE CIERRARCHIVO

Procedure que cierra los archivos del sistema, colocando el identificador "CORRECT" y la versión de usuario dentro del campo de protección de los archivos. En la palabra correspondiente al archivo dentro de FCONTROL guarda la misma versión.

Parámetros

F Nombre del archivo
ARR Arreglo de control del archivo
OP Número del archivo (ver crearchivos)

PROCEDURE ABREARCHIVOS

Este proceso abre los archivos del sistema verificando que:

- a) El archivo existe dentro del sistema
- b) El archivo sea residente en disco
- c) El archivo contiene el identificador "CORRECT" en el registro cero
- d) La versión de último usuario y la correspondiente en FCONTROL sea la misma

Si alguna de éstas condiciones no se cumple, se emite un mensaje solicitando el archivo correspondiente y termina el programa.

Parámetros

F	Nombre del archivo
ARR	Arreglo de control del archivo
OP	Número de archivo
ULTIMO	Último registro útil del archivo

PROCESOS PARA REVISAR LAS TARJETAS QUE CONTIENEN
LOS DATOS DE LOS PROGRAMAS

Al codificar y perforar los datos para alimentar a la computadora pueden cometerse diversos errores, a saber:

- a) Romperse la secuencia de las tarjetas perforadas, ya sea por acomodarlas en desorden o que en el momento de la perforación se efectúe ésta por diferentes formatos y no por orden de tarjetas, etc.
- b) Por cambio de dígitos o letras en identificadores numéricos o códigos, que no pueden ser fácilmente reconocidos en la revisión sintáctica del identificador, por ejem:

I) Al solicitar un libro que costaba \$31.25, se perfora \$32.25 el identificador está dentro del rango permitido.

II) Un libro es solicitado para la dependencia:

3.34.611.02.519.01 y en realidad es para la dependencia 3.33.611.02.519.01. Cuando éstos dos códigos son válidos en el programa se alteran los inventarios y los presupuestos de dos dependencias.

Para prevenir éstos errores y con objeto de simplificar el proceso de revisión posterior de los resultados, se implementaron

una serie de rutinas para revisar al través de la máquina, pequeños paquetes de datos, cotejándolos con una tarjeta conocida como "TARJETA MAESTRA" que es hecha por el codificador para "AMARRAR" los datos en las demás tarjetas. De ésta forma, las tarjetas se dividen en pequeños grupos de más fácil revisión cuando son erróneas.

La tarjeta maestra para cada programa se diseñó tomando en cuenta los datos que son fundamentales para la formación de archivos fijos y su corrección dentro de ellos es muy complicada.

Ejemplo; En el programa que solicita los libros se consideró importante:

- 1) El código programático de la dependencia
- 2) El precio de los libros
- 3) El número de ejemplares solicitados

La tarjeta maestra, tiene perforados:

- a) El código programático de la dependencia a la cual corresponden los pedidos de las tarjetas que ampara.
- b) El número total de ejemplares de todos los pedidos
- c) El monto total de lo solicitado

Se revisan las tarjetas de detalle comparando los códigos programáticos de la dependencia solicitante contra el código programático perforado en la tarjeta maestra y acumulando los totales de ejemplares solicitados y los montos de los pedidos,

al terminar el paquete, coteja que las cantidades acumuladas sean iguales a los de la tarjeta maestra y permite pasar el paquete de tarjetas almacenándolas en un archivo RETENIDOS y avisa al usuario el nombre del mismo. En caso de error, se almacenan en un archivo de ERROR y emite un listado ^{de} las imágenes de tarjetas almacenadas en él indicando la secuencia de las mismas.

PROCEDURE CARGADOR

Este procedure clasifica las tarjetas por número de secuencia. Revisa si la tarjeta que está leyendo es tarjeta maestra para iniciar los contadores de paquete y guardar los datos necesarios. Si la tarjeta es de DETALLE compara y actualiza los contadores, al percatarse de que se acabó un paquete, procede a almacenarlo en el archivo de RETENIDOS o en el de ERRORES y emitir el listado correspondiente.

A éste proceso le son pasados como parámetros cuatro procedimientos que son las que hacen los diferentes cotejos:

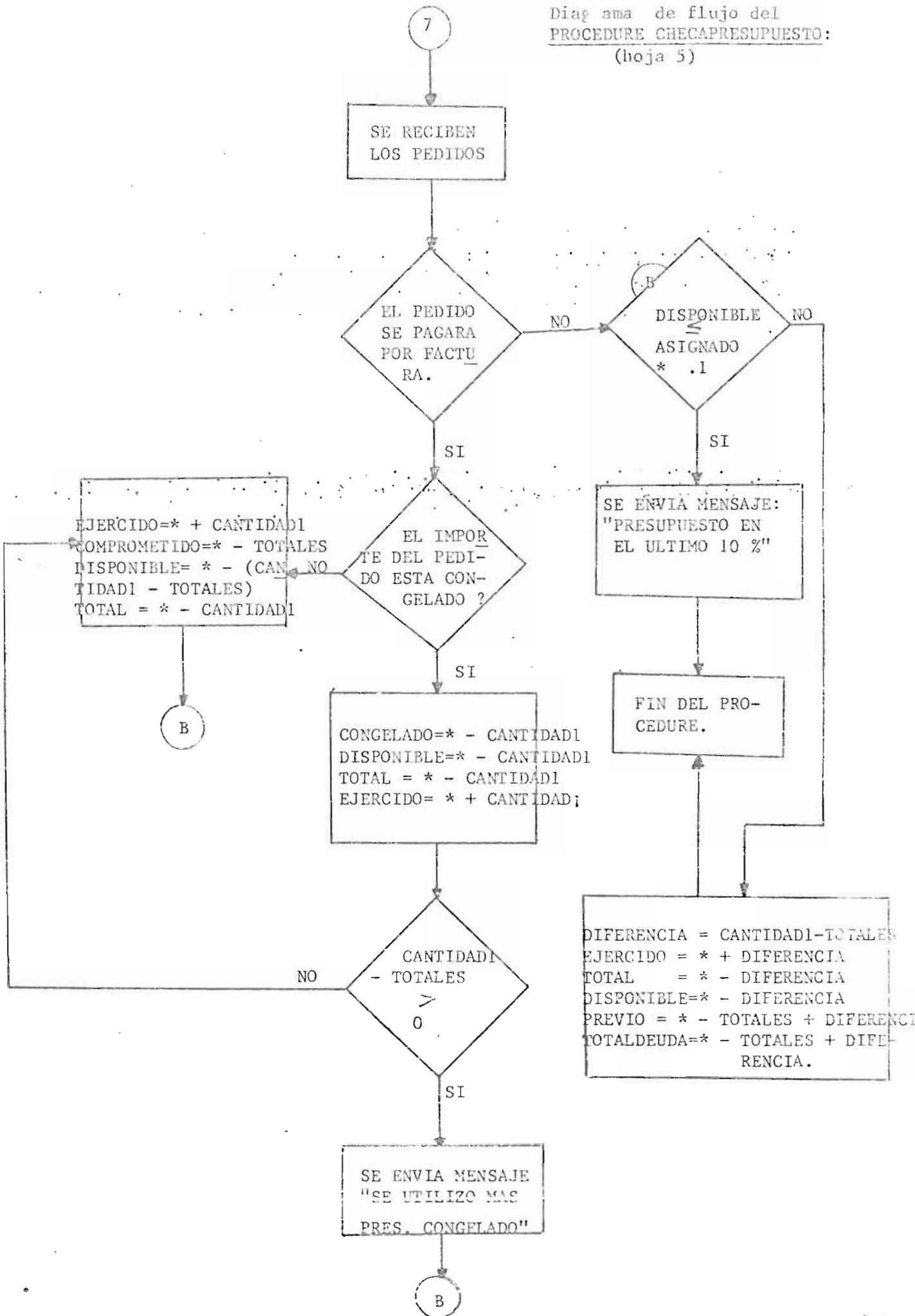
Parámetros

COMPARACOSTOS Procedure que verifica que los datos acumulados sean iguales a los de la tarjeta maestra.

INICIA Procedure que inicia los contadores en cero cada vez que un nuevo paquete es detectado.

MAESTRA Procedure que revisa a la tarjeta MAESTRA sintácticamente, avisa el inicio de un paquete y guarda los valores de "AMARRE"

REVISA Revisa sintácticamente las tarjetas de detalle y acumula los contadores



RUTINAS ESPECIFICAS PARA CADA COMANDO DEL PROGRAMA

PROCEDURE CARGA PROVEEDOR :

Los proveedores son dados de alta dentro del sistema por medio de un código numérico asignado por el Departamento de Adquisiciones.

En el archivo FPROVEEDORES, el registro cero funciona como un DIRECTORIO que contiene l s direcciones dentro del mismo archivo donde empiezan los nodos variables que contienen los datos del proveedor.

De cada proveedor se tiene:

NOMBRE

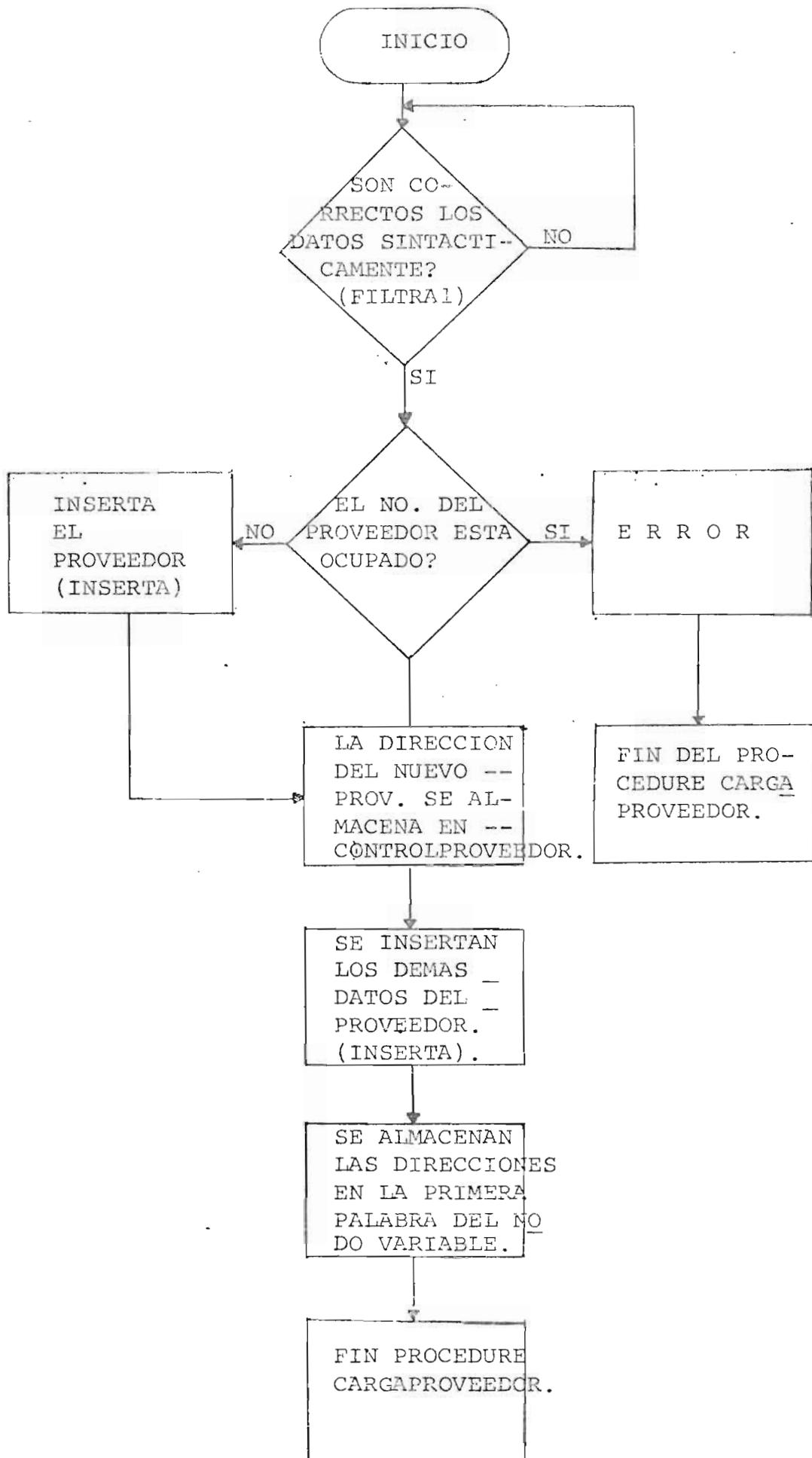
DOMICILIO

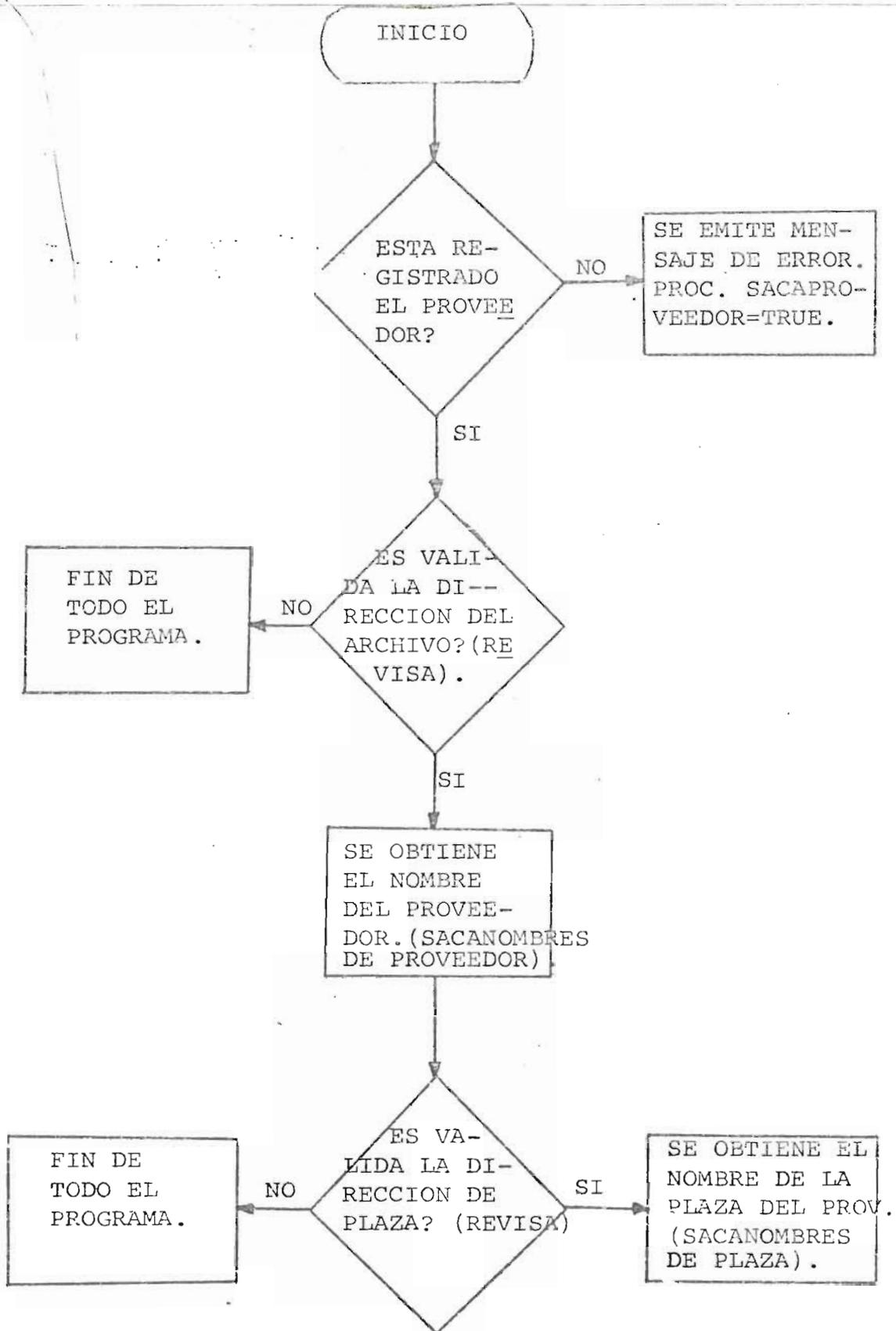
POBLACION

CIUDAD

Los procesos para dar de alta y recuperar a los proveedores, verifican en el registro cero si el proveedor está dado de alta, después proceden a guardar la información por medio del procedure INSERTA o a sacarla por medio del procedure SACANOMBRES.

Diagrama de Flujo del
PROCEDURE CARGAPROVEEDOR.

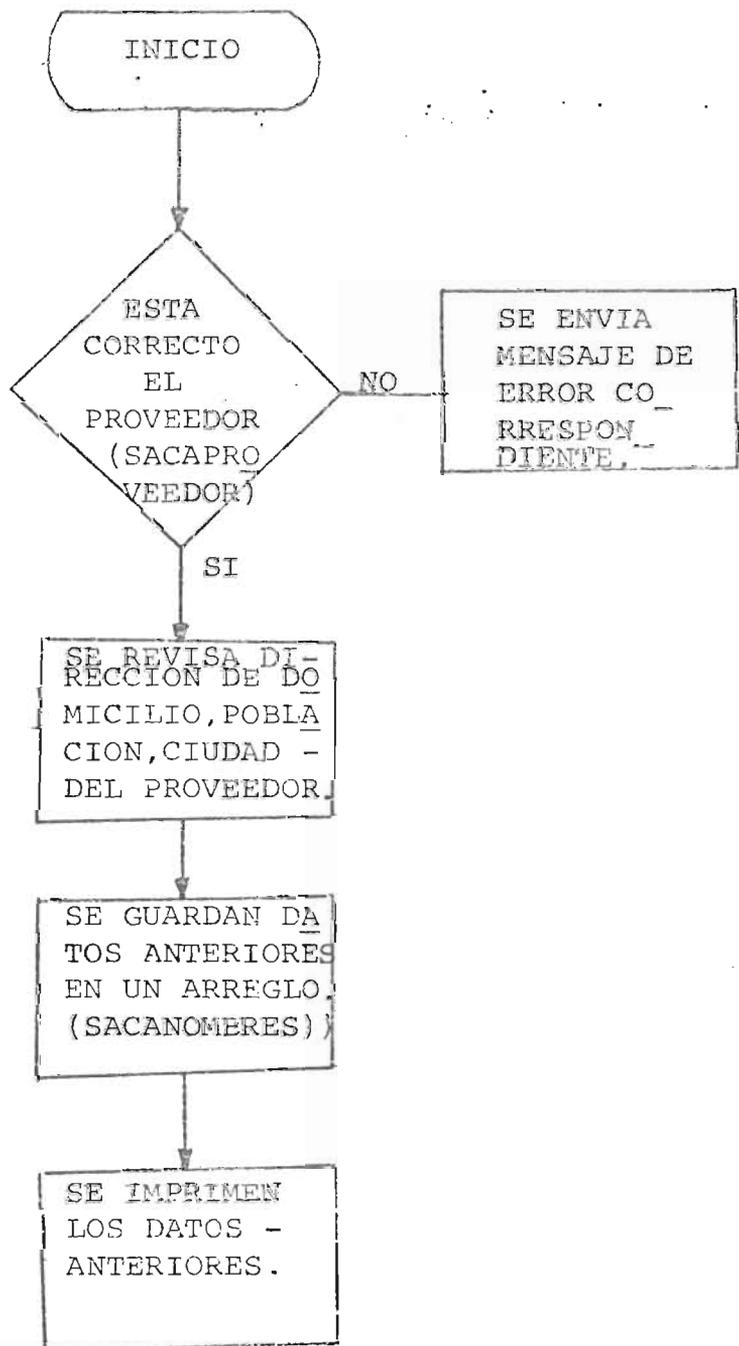




Procedure:
SACAPROVEEDOR COMPLETO.

Por medio de éste procedure se obtienen los datos complementarios del proveedor , así como la impresión de los mismos.

Diagrama de Flujo:



Para implementar el comando de PEDIDO se implementaron una serie de rutinas, las cuales son llamadas desde el PROCEDURE PEDIDO y efectúan los pasos necesarios para procesar y alojar la información.

PROCEDURE PEDIDO

Este proceso aloja los datos de los libros solicitados dentro del Banco de Información, actualiza los Directorios correspondientes y emite los listados necesarios.

El primer paso de éste proceso, así como todos los demás -- del programa, es verificar que los archivos necesarios para la ejecución del mismo estén correctos.

Revisa la palabra de seguridad (CORRECT) en cada uno de ellos y las versiones de los mismos. Los archivos usados en éste proceso son:

NOPEDIDO
FDISPNOPEDIDO
FDISPNOMPEDIDO
FDESCRIPTORPEDIDO
DEPENDENCIAS

Con los archivos correctos se procede a revisar los datos de un pedido sintácticamente, convirtiendo los códigos en valores numéricos, los identificadores numéricos en variables de punto fijo y alojando el autor y el título en los arreglos correspondientes. (FILTRA 1)

Este comando proporciona la opción a la Biblioteca Departamental, de saber si el libro ha sido solicitado por ella misma o por otra dependencia.

Para brindar ésta información dentro del Banco de Datos, primero localiza e intersecta todos los identificadores proporcionados por el usuario. Si resulta infructuosa ésta investigación, se procede a intersectar los identificadores por partes. Busca si existe la intersección por apellido del autor y título o por apellido del autor y subtítulo.

Cuando alguna intersección existe, avisa al usuario emitiendo una tarjeta de pedido con los datos correspondientes.

En caso contrario, se da curso al pedido al igual que los libros que no solicitaron verificación. Antes de alojar la información dentro de los archivos fijos, se verifica si el número de pedido no ha sido usado anteriormente, que el proveedor al cual se le solicita el libro esté registrado dentro del Banco de Información y que las disponibilidades presupuestales de la dependencia adquirente, sean suficientes. (PROCEDURE CHECAENBANCO)

Si no existen problemas de incompatibilidad con el Banco de Datos, se procesa la información (PROCEDURE ALTAPEIDIDO).

Proceso de la Información:

- 1) Localiza la dirección dentro del archivo NOPEDIDO para alojar los datos del pedido.
- 2) Almacena la parte fija del nodo de pedido consistente en los identificadores numéricos y los números correspondientes a los códigos.
- 3) Aloja cada uno de los identificadores alfanuméricos del sistema anagando su dirección en la parte fija del nodo y actualizando el archivo del Directorio Doble FDISNOMPEDIDO y el archivo de DESCRIPTORPEDIDO que forma las listas invertidas. (PROCEDURE GUARDANOMPRESPEIDIDO)

- 4) Anota en el nodo correspondiente dentro del archivo de dispersión numérica de los pedidos FDISNOPEDIDO, la dirección del nodo del pedido en el archivo NOPEDIDO.
- 5) Cuando el pedido es de proforma, copia la parte fija del nodo en un archivo de maniobra.
- 6) Emite las tarjetas de pedido para solicitar los libros y las copias correspondientes para la Biblioteca Departamental y el Departamento de Adquisiciones (PROCEDURE IMPRESION)
- 7) Después de que los pedidos de la corrida han sido procesados, se generan a partir del archivo de maniobra las formas correspondientes para solicitar los giros bancarios al Departamento de Egresos para la adquisición de libros por Proforma (PROCEDURE EMITELISTADOSPEDIDO)

PROCEDURE VALIDANDO

Esta rutina localiza y envía la información correspondiente a identificadores alfabéticos alojados dentro del Banco de Datos.

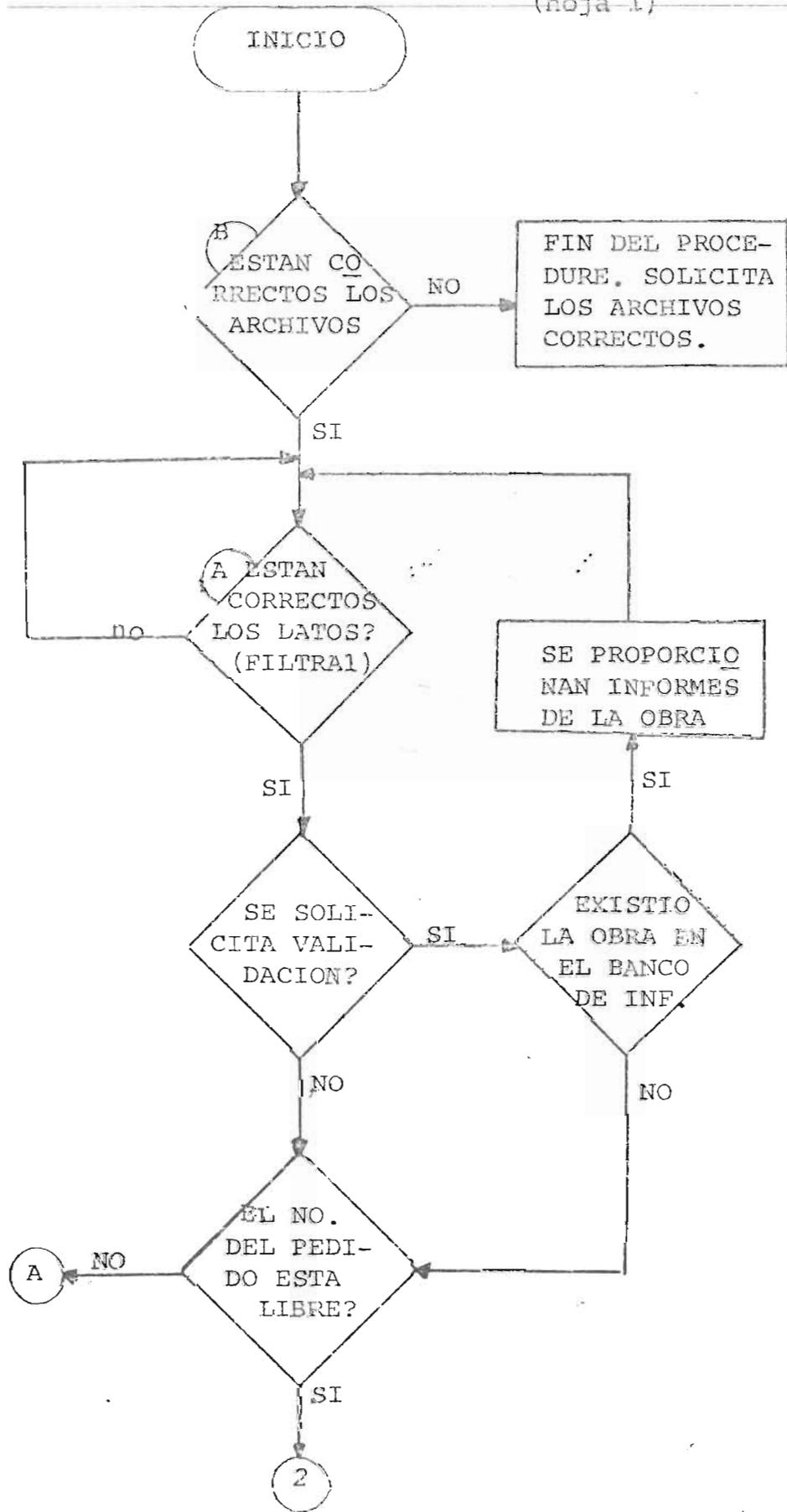
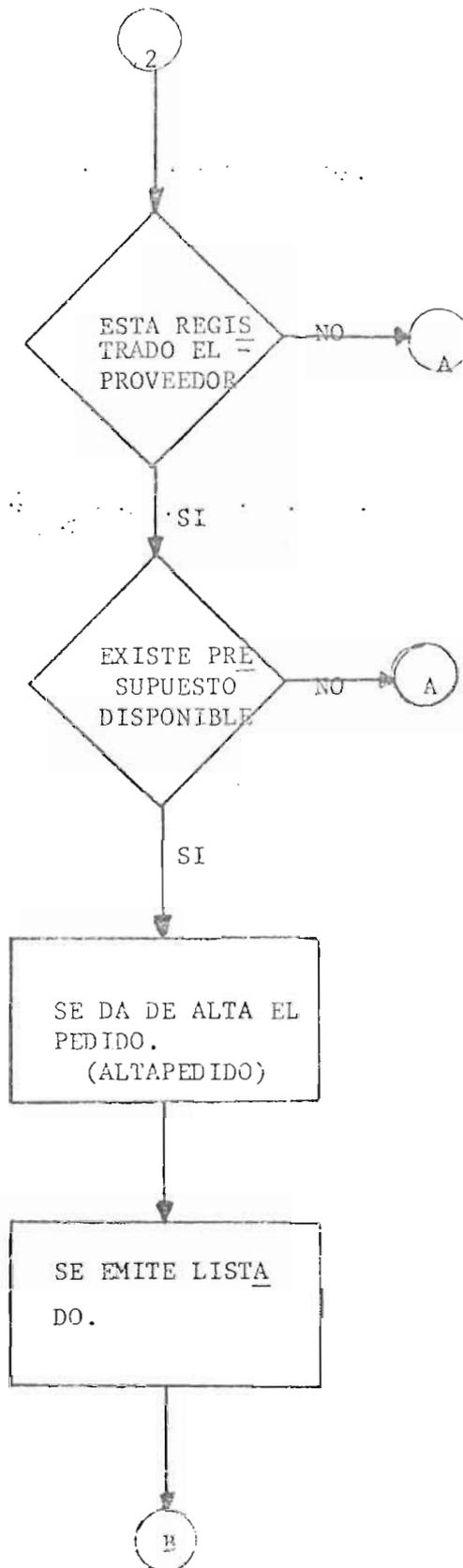


Diagrama de flujo del
PROCEDURE PEDIDO:
(hoja 2)



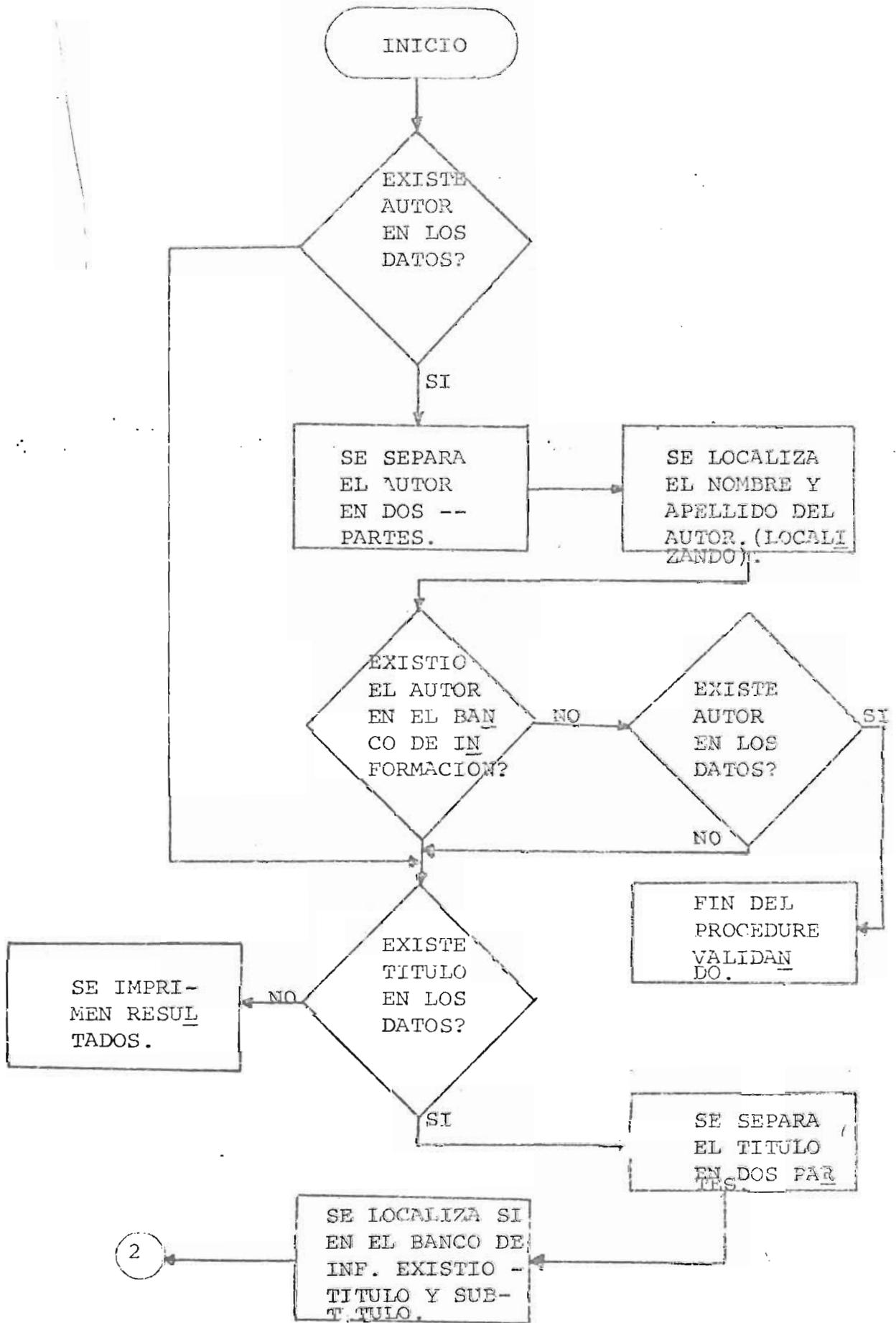
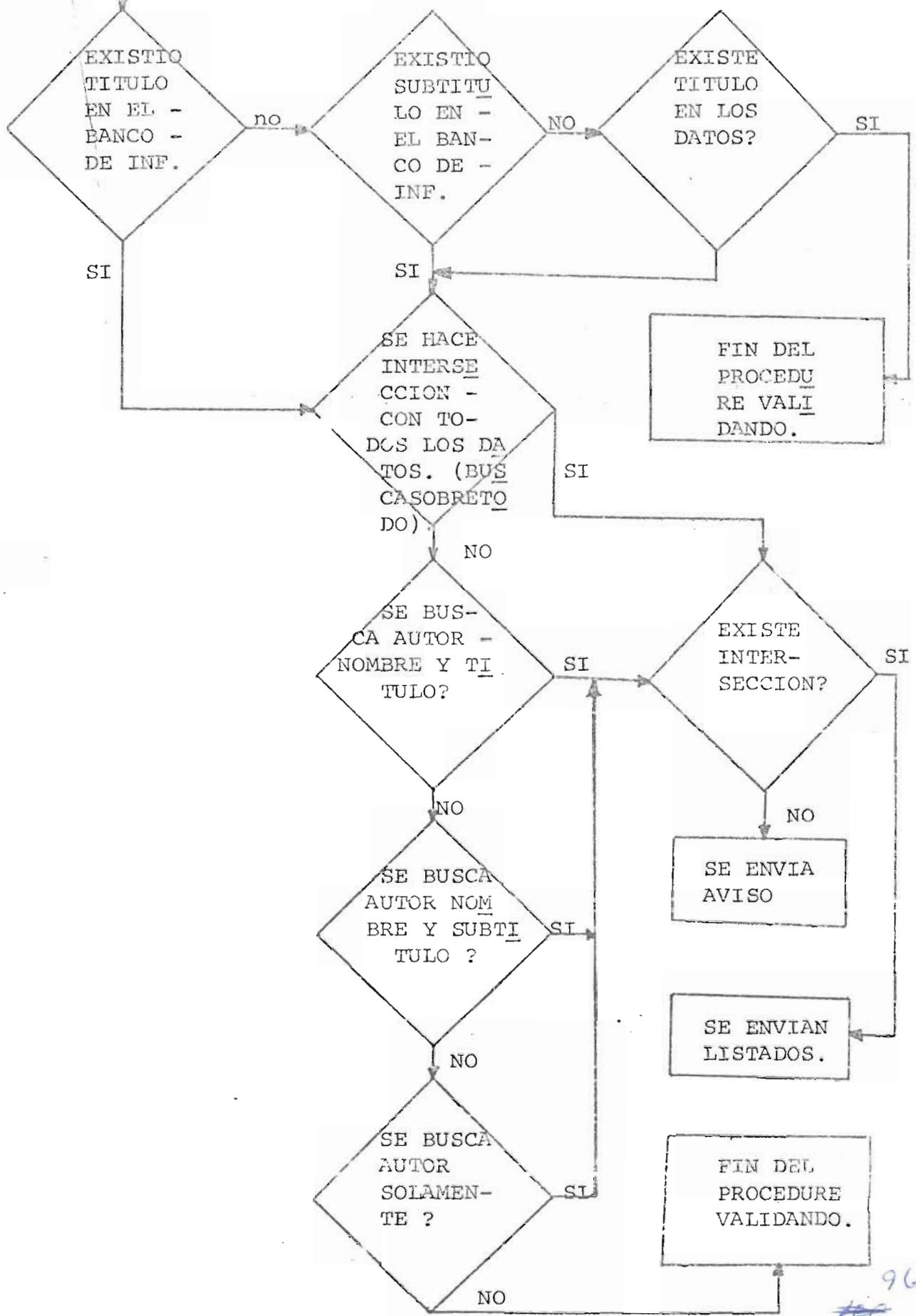
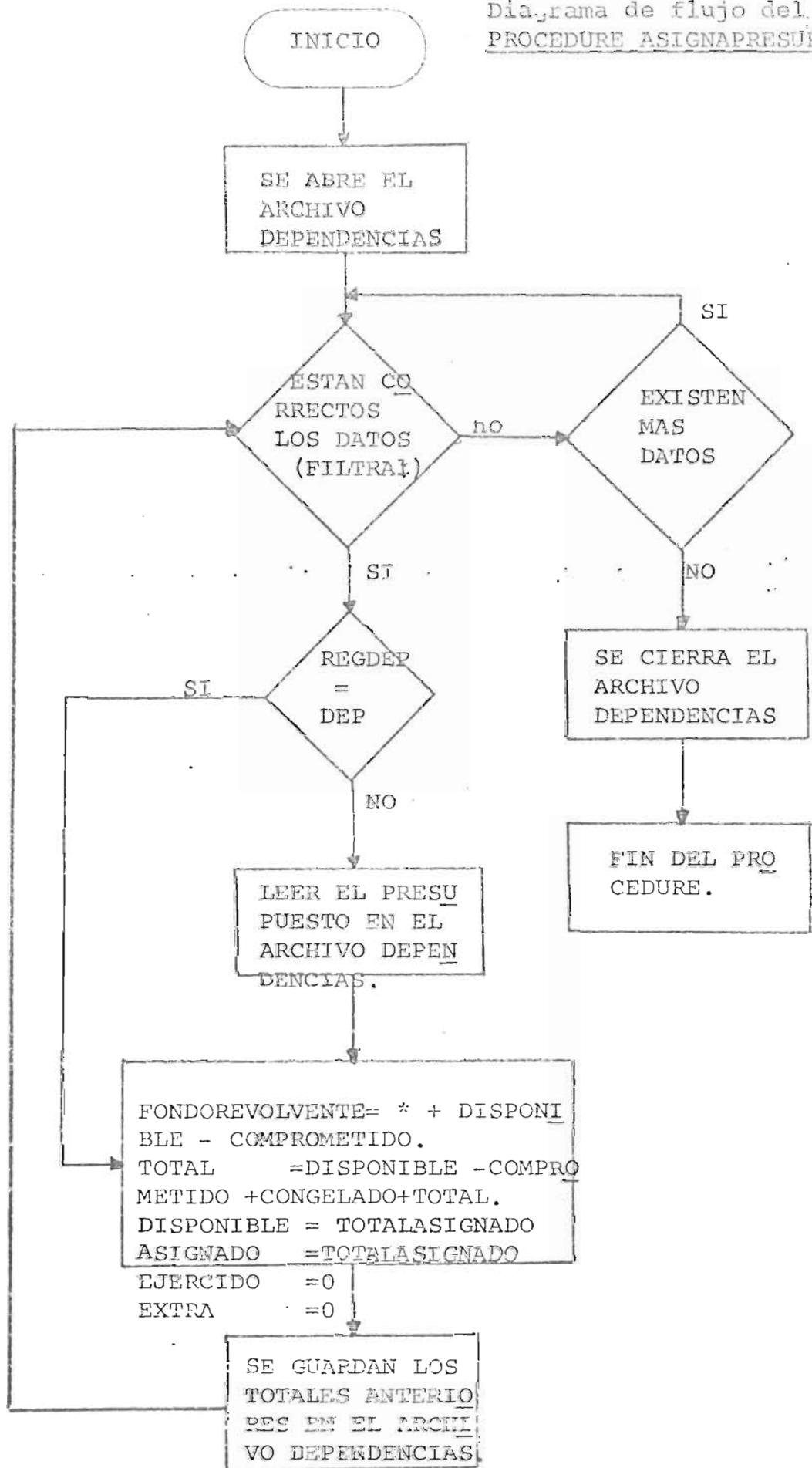


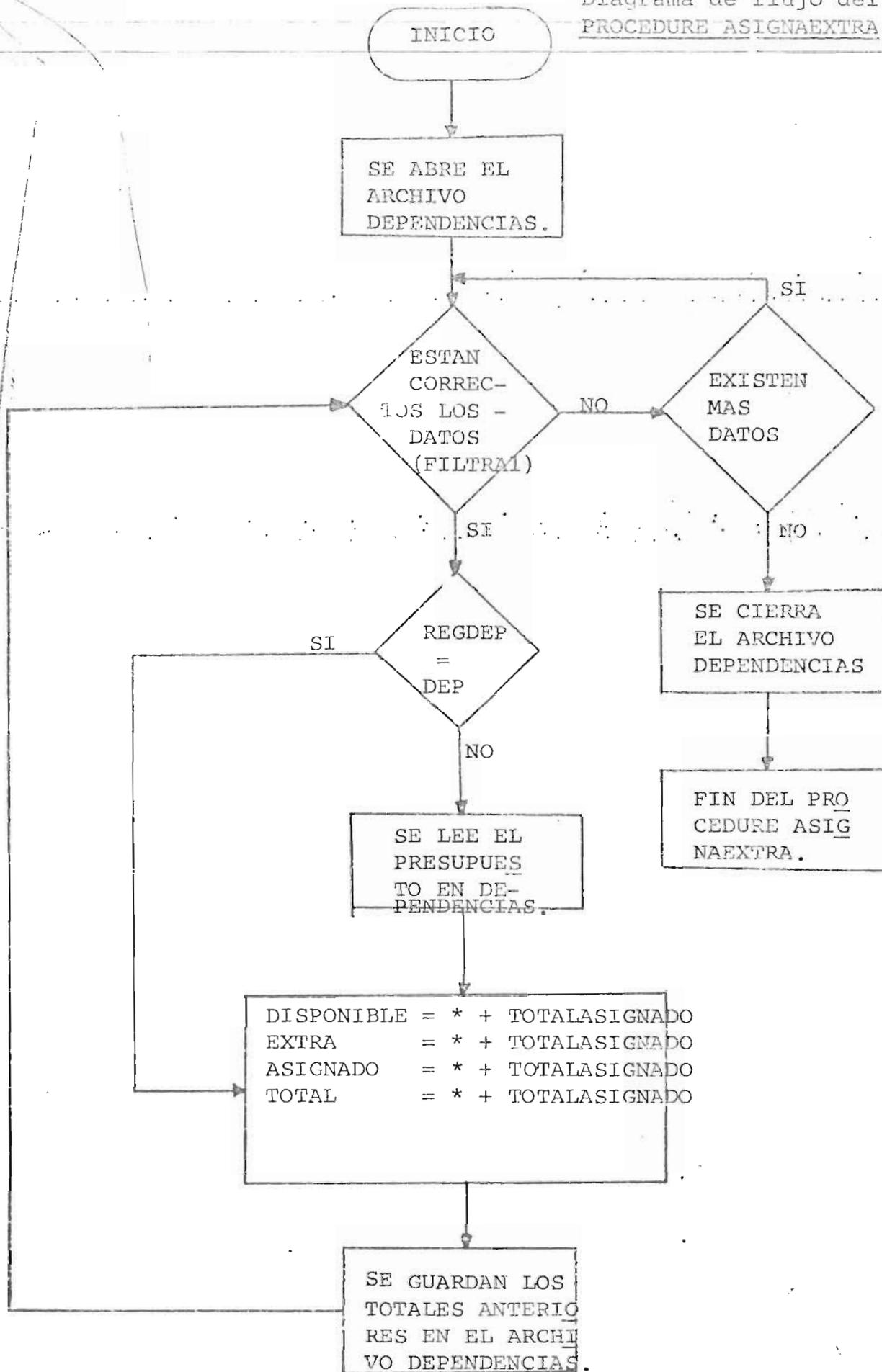
Diagrama de Flujo del PROCEDURE VALIDANDO:

(hoja 2)

2







Esta rutina es usada dentro del proceso de pedido, cuando la dependencia adquiriente así lo solicita o dentro del comando de información, cuando se desea saber la existencia de un pedido por Autor y/o título.

La indagación sobre los identificadores, se hace en etapas. Primero se separan los identificadores alfabéticos buscando el signo de puntuación correspondiente y se localizan cada uno de ellos dentro de los archivos fijos del sistema (PROCEDURE LOCALIZA)

Con los identificadores que fueron encontrados, se indagan las intersecciones que se efectúan entre ellos. En primer término, la intersección se hace con todos los elementos que el usuario proporciona pero, si ésta resulta infructuosa; el programa procura hacer la intersección por caminos más restringidos que pueden ayudar a la localización de los nodos deseados:

- 1) Intersecta apellido del autor con título
- 2) Intersecta apellido del autor con subtítulo
- 3) Título y subtítulo

Cuando ninguna de las intersecciones resultó positiva, se da por mala la búsqueda.

D agrama de Flujo del
PROCEDURE LOCALIZANDO:

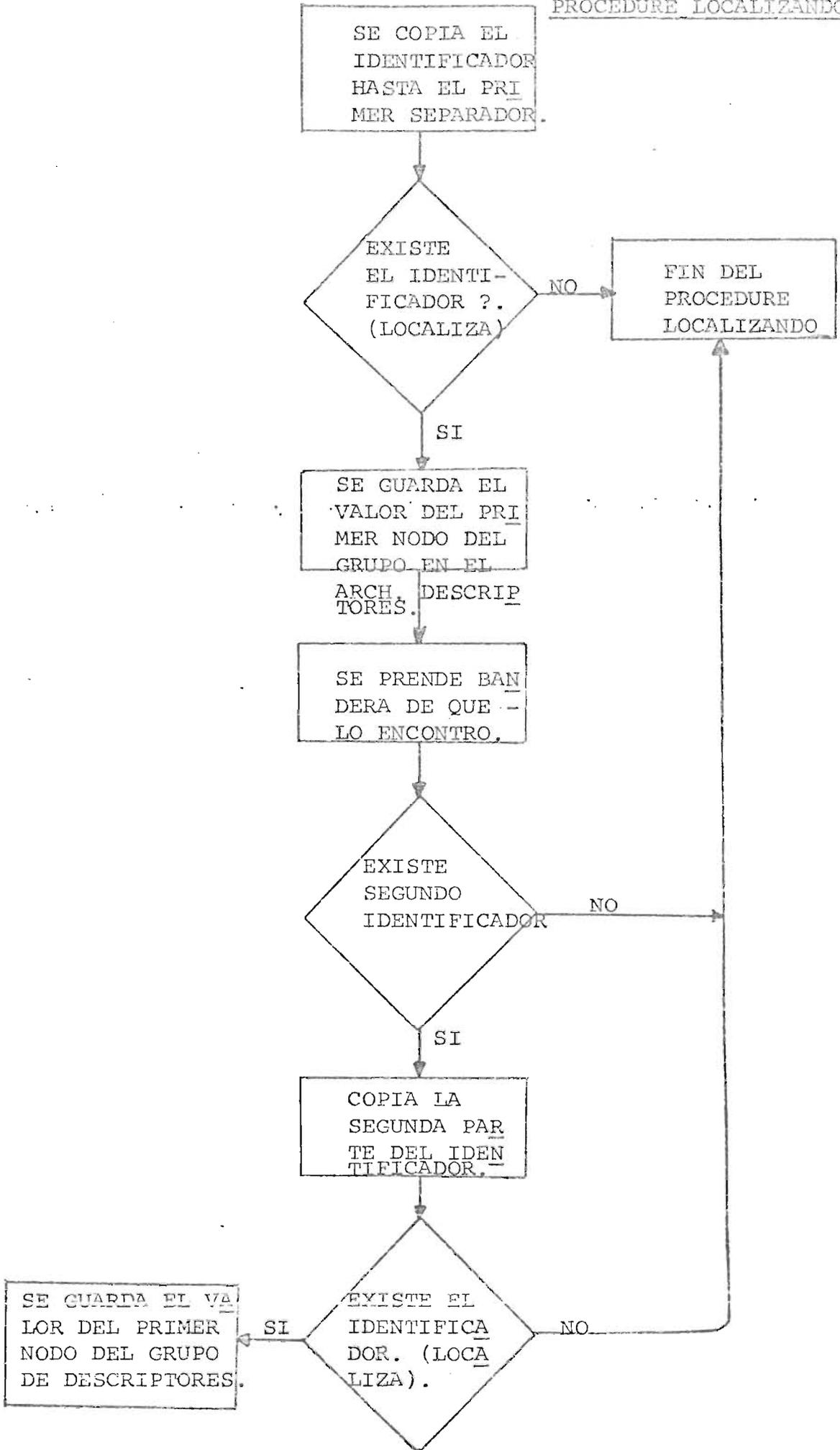
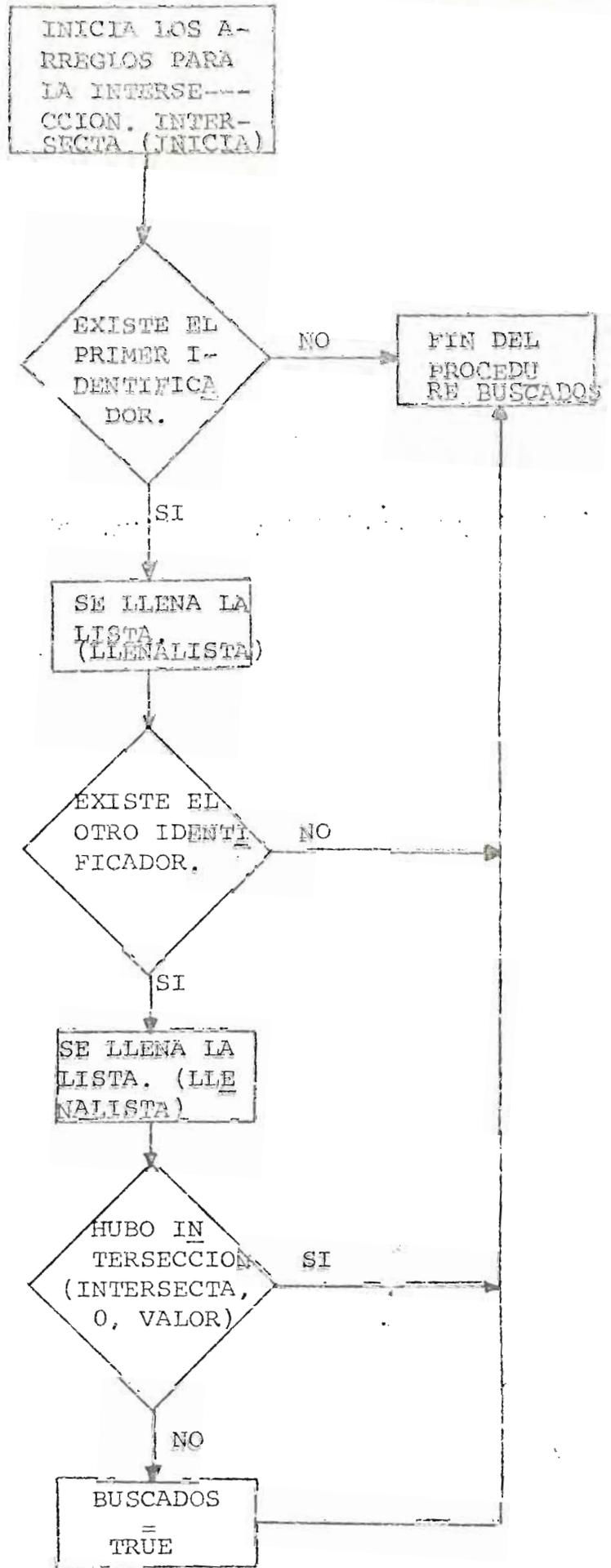


Diagrama de Flujo del
PROCEDURE BUSCADOS:



res solicitados sea mayor o igual al número de ejemplares entregados y las disponibilidades presupuestales de la dependencia sean suficientes. (PROCEDURE VALIDA)

Unicamente cuando no hay problemas con las tarjetas de factura y de pedidos que forman un paquete, se procede a almacenar toda la información y se generan los listados correspondientes para el Departamento de Egresos, el Departamento de Contabilidad, el Departamento de Adquisiciones y la Biblioteca Departamental. (PROCEDURE EMITELISTADOSRECEPCION)

PROCEDURE EMITELISTADOSRECEPCION

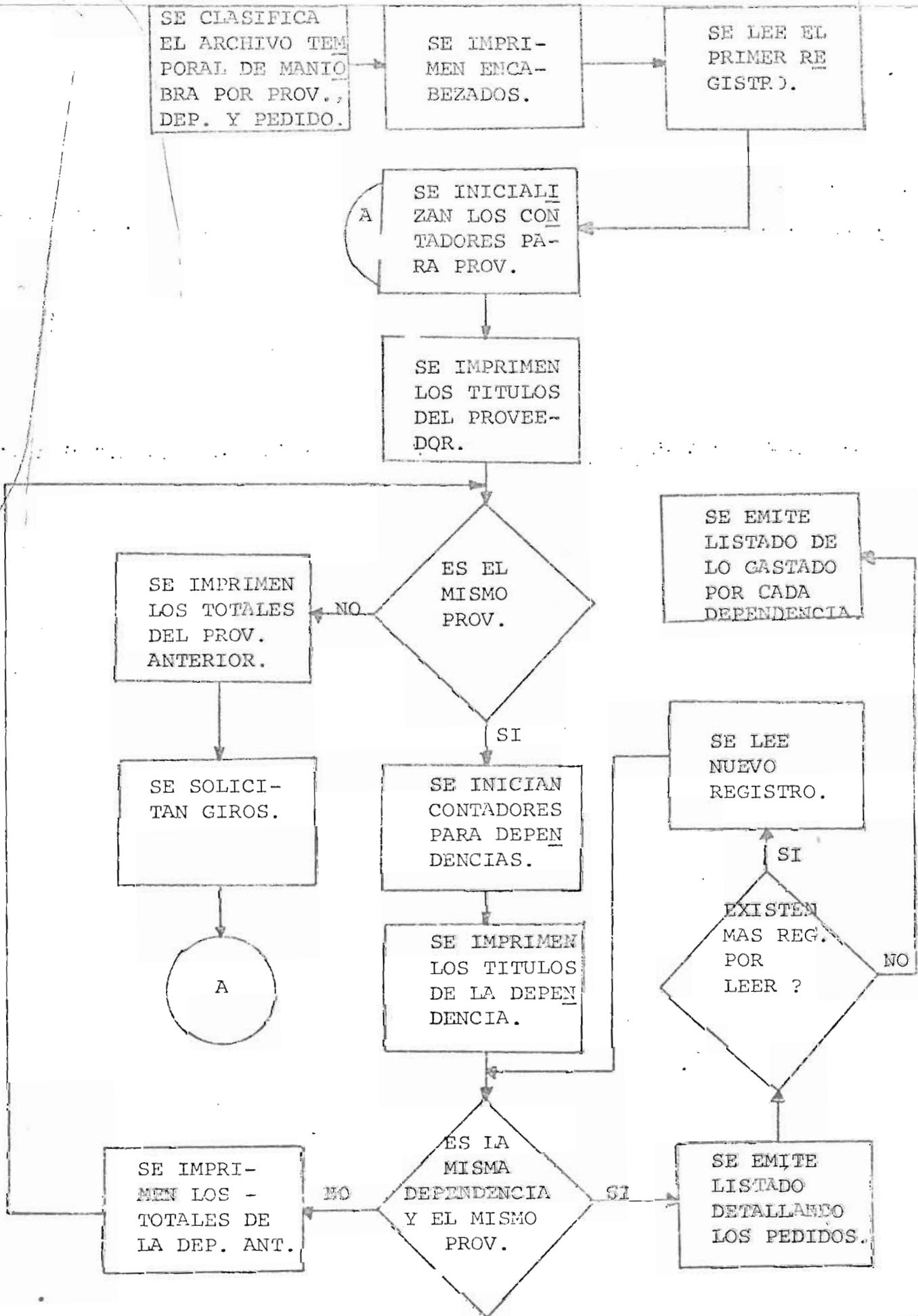
Este procedure genera simultáneamente 6 listados para los diferentes departamentos.

- 1) Solicitud de cheques para egresos
- 2) Desglose de cheques por factura para egresos
- 3) Desglose por factura para justificar Deuda en el Departamento de Contabilidad
- 4) Listado de cantidades erogadas por cada dependencia para egresos.
- 5) Listado de cantidades comprobadas por cada dependencia para el Departamento de Contabilidad
- 6) Detalle de pedidos por factura para el Departamento de egresos

Los datos de los pedidos recibidos dentro del proceso son clasificados por: PROVEEDOR, DEPENDENCIA, NUMERO DE FACTURA.

Mientras los pedidos son de la misma factura, se detallan para el Departamento de Adquisiciones. Al cambiar de factura, ésta se detalla para Contabilidad o Egresos, dependiendo del tipo de pago. Cuando cambia de proveedor, saca totales por proveedor en todos los listados y solicita el cheque del proveedor.

Cuando se terminan los pedidos y las facturas, se saca un resumen para Contabilidad y Egresos con las cantidades correspondientes para cada dependencia.



Después de solicitar todos los cheques, saca un listado con las cantidades erogadas por cada dependencia.

PROCEDURE IMPRESION

Las tarjetas en las cuales se detallan cada uno de los pedidos con sus copias correspondientes, son generadas por ésta rutina. Las tarjetas se imprimen de dos en dos; por la cual se aloja la imagen de tarjeta en un arreglo, cuando están las dos tarjetas listas, se manda a impresión.

Parámetros

ARCHIVO	Archivo de impresión
M	Número de tarjeta
N	Dirección del nodo de pedido. N=0 cuando los datos se sacan del archivo de facturas extra.
ARREGLO	Arreglo donde está alojada la información del pedido.
B	Variable lógica que le indica al procedimiento si emite toda la información del pedido o únicamente la referente a la solicitud del mismo.

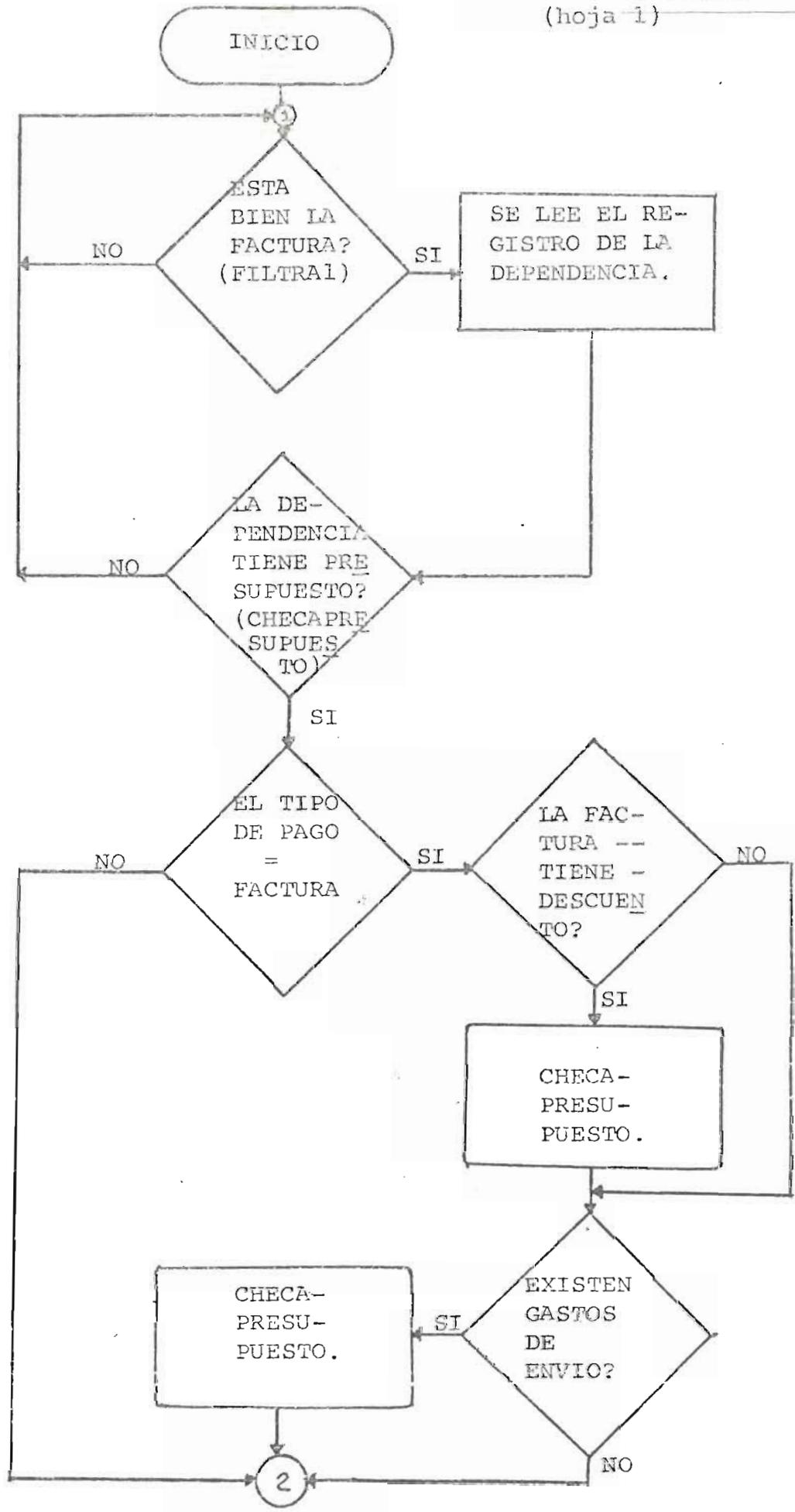
PROCEDURE RECEPCION

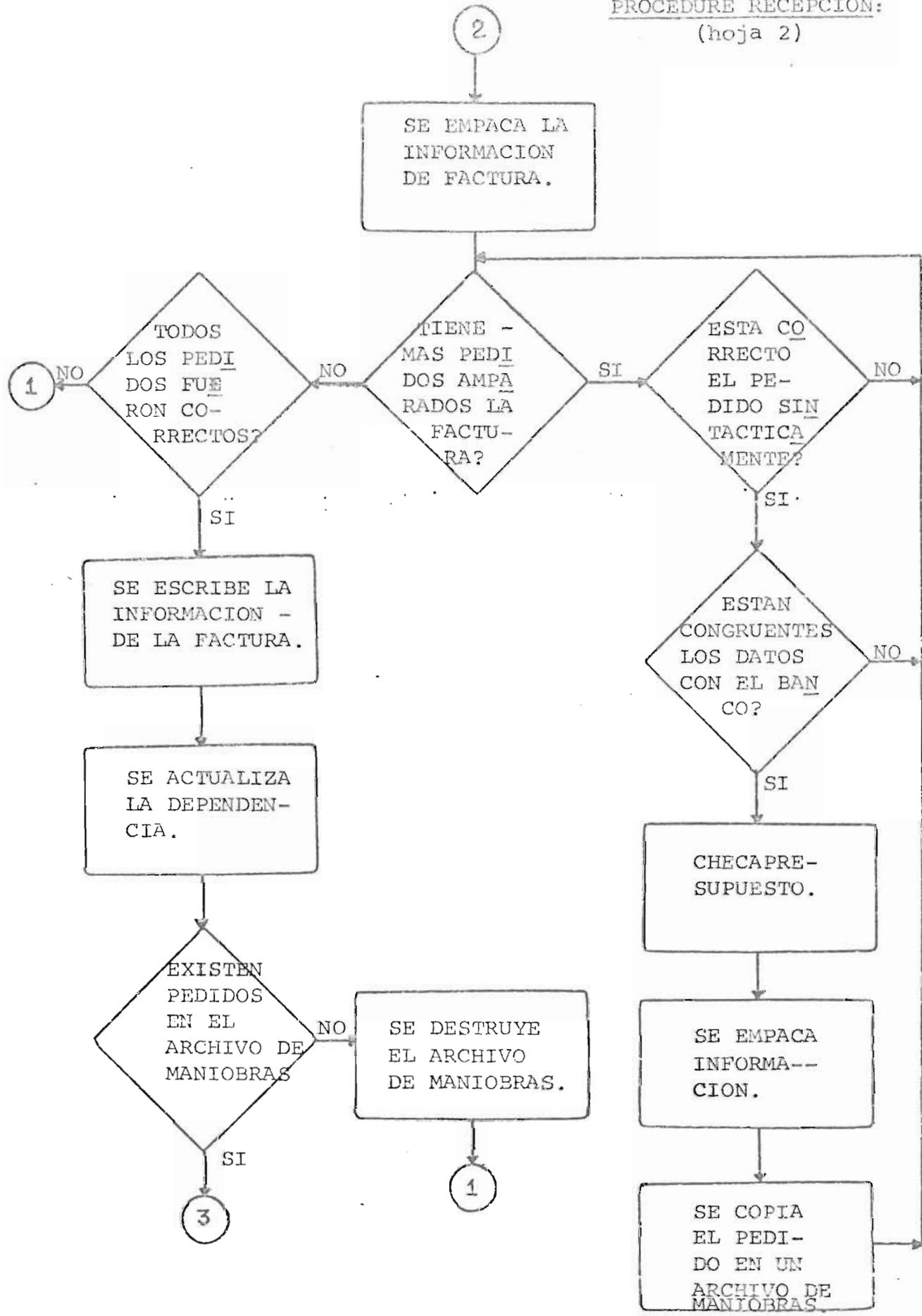
Cuando un pedido es surtido total o parcialmente, debe entrar al sistema amparado por una factura. Una factura puede contener uno o más pedidos, según el proveedor lo considere conveniente.

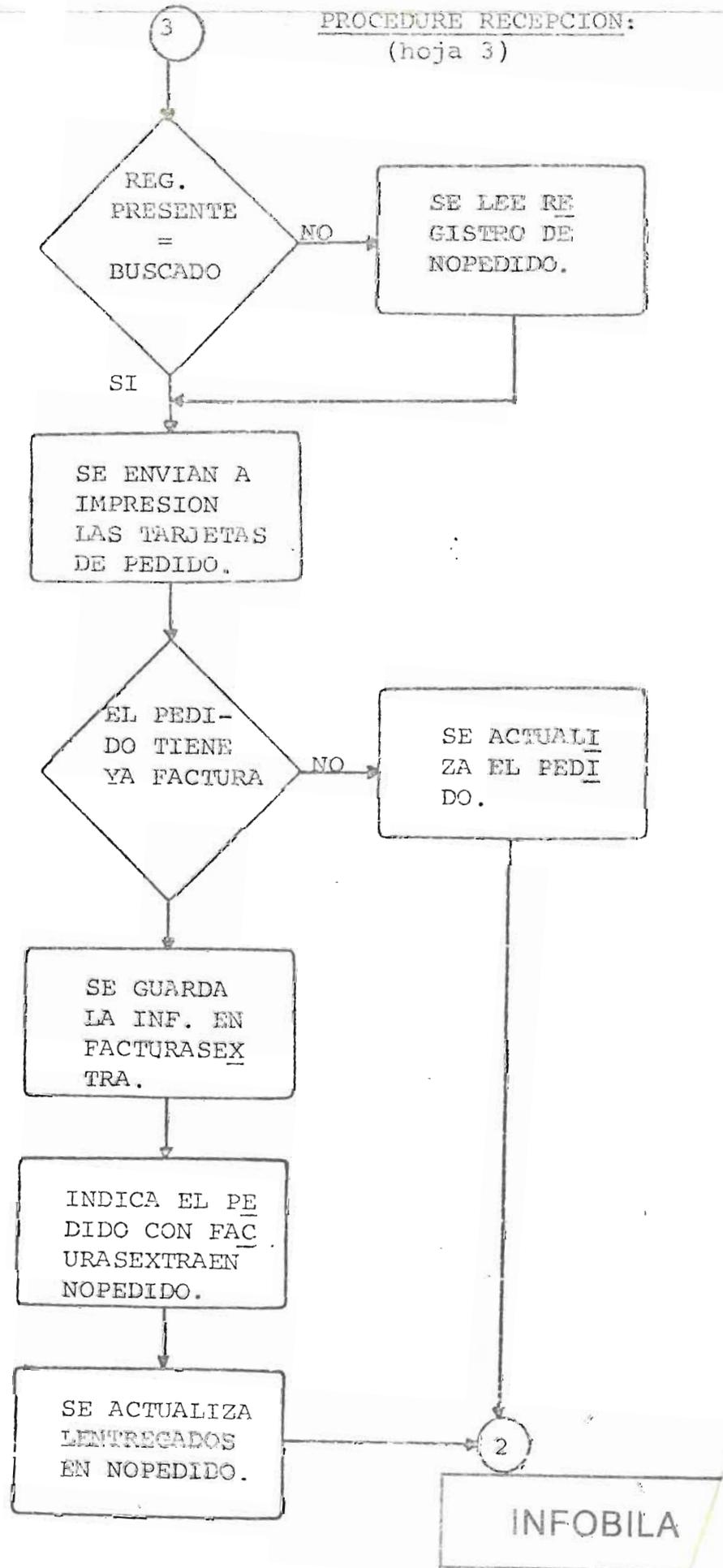
Inmediatamente después de verificar la validez de los archivos, revisa las facturas que entran al sistema, asienta los datos de los pedidos surtidos y emite la papelería correspondiente.

Los datos se captan y se validan dentro del Banco de Información por partes:

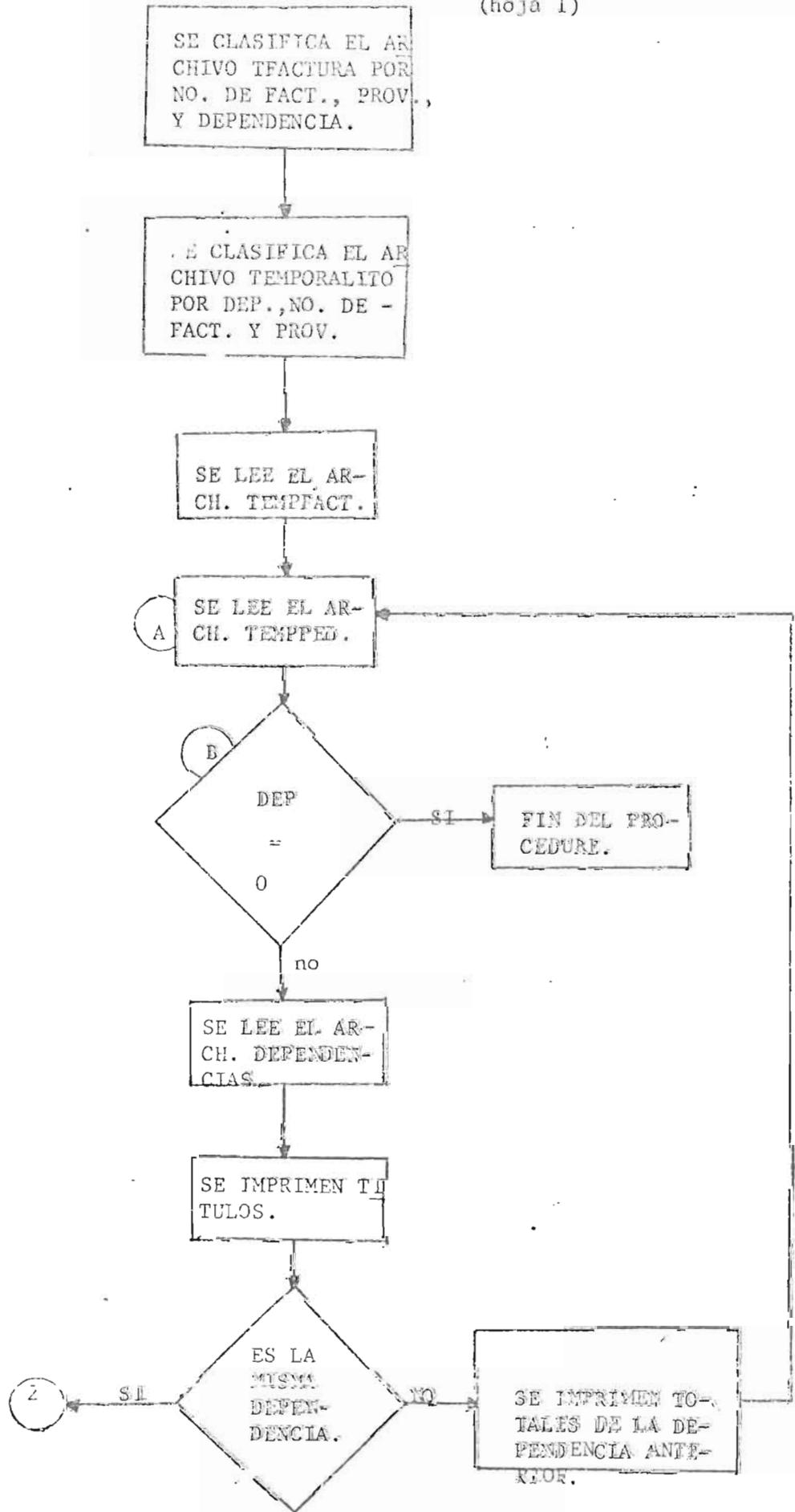
- 1) Se revisa sintácticamente la tarjeta que contiene los datos de la factura.
- 2) Revisan cada una de las tarjetas que contienen los libros amparados por la factura.
- 3) Esta revisión es tanto sintáctica como de cotejo dentro del Banco de Información. Si el pedido está en uso, verifica que la forma de pago sea la misma y el código de la dependencia adquiriente coincida con el número de ejemplo

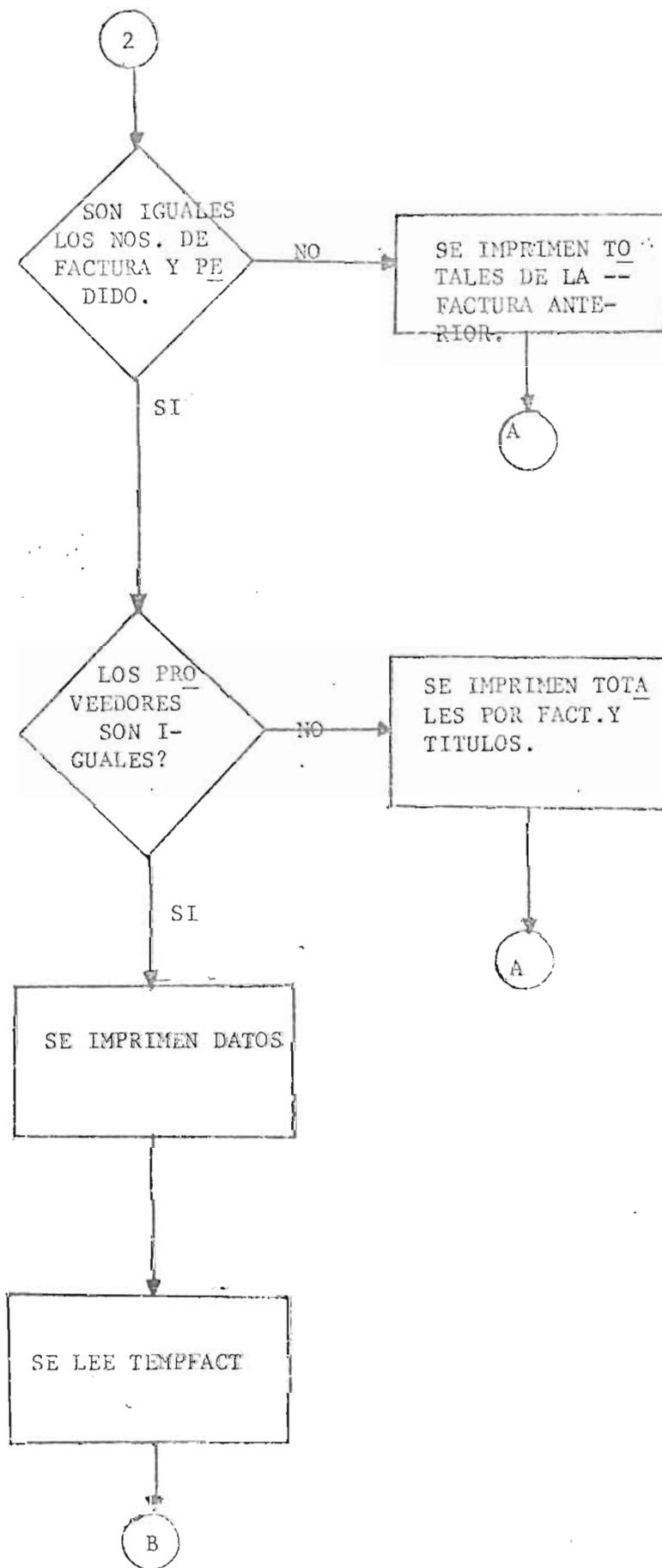






para proporcionar el desglose de los pedidos de cada
factura, dando los totales de la r'sma al cambio de
cada factura.





PROCEDURE INFORMA

Es un proceso que comunica al usuario del programa con el Banco de Datos. Está programado para efectuar las preguntas correspondientes al usuario para solicitarle todos los datos que el programa necesita para devolver la información. (PROCEDURE SEND)

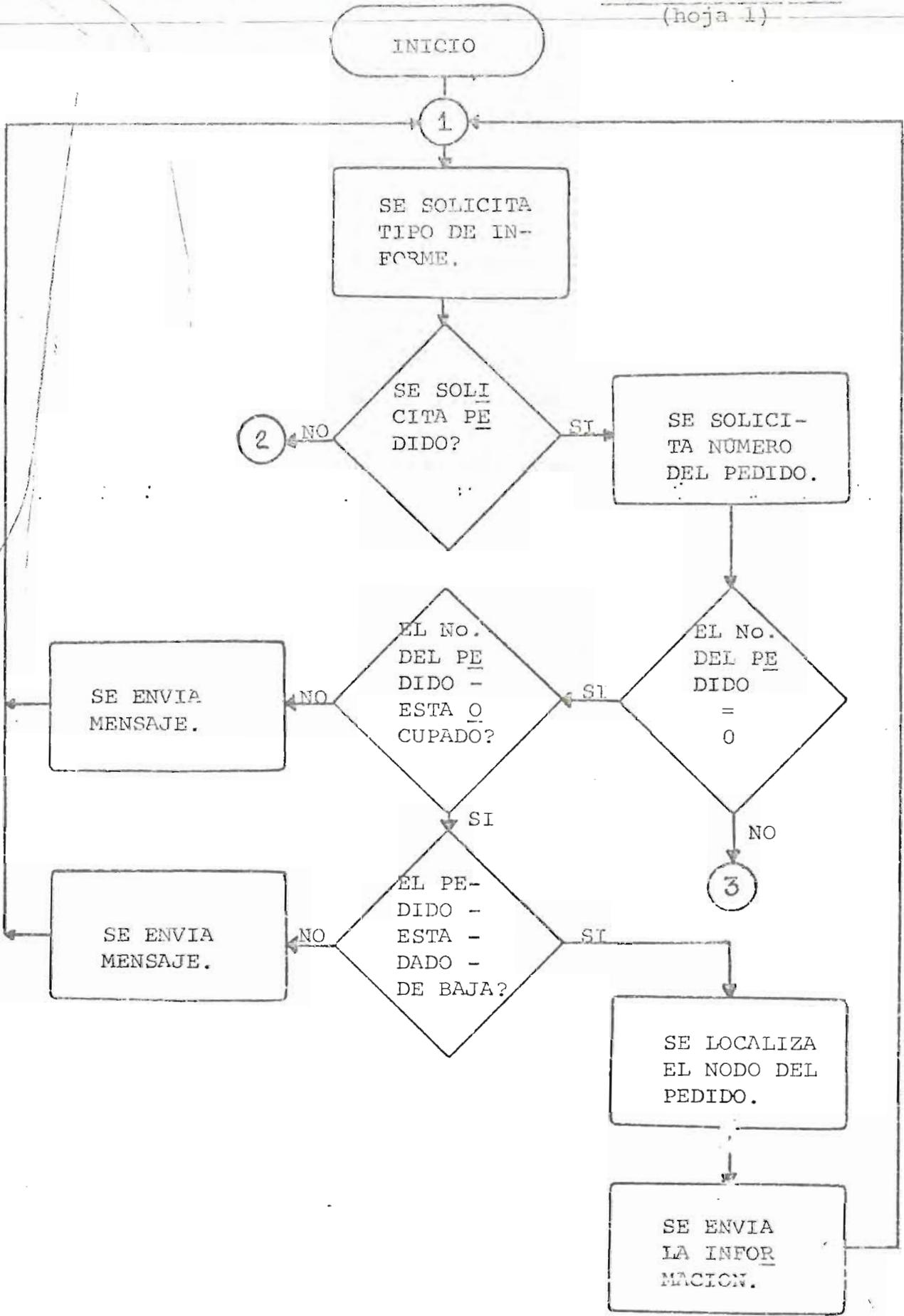
Esta rutina proporciona información al usuario sobre los pedidos, los proveedores y los estados de cuenta de cada dependencia. Cuando el informe solicitado es pedido, tiene dos opciones.

- 1) No. de pedido diferente de cero.

Encuentra en el directorio de dispersión numérica ARCHIVO P. EDISPNOPEDIDO la dirección del pedido, cuando éste existe, envía la tarjeta de pedido al usuario o avisa que el pedido no existe.

- 2) No. de pedido igual a cero.

En éste caso, sobreentiende que la búsqueda es por medio del autor y/o el título. Requiere del usuario los datos correspondientes e investiga dentro del Banco de Datos su existencia (PROCEDURE VALIDANDO).



Si se solicita informe acerca de los proveedores.

a) Número de proveedor diferente de cero.

Localiza la dirección del proveedor en el registro cero del archivo FProveedores y si éste existe, envía la información completa del proveedor (PROCEDURE SACAPROVEEDOR COMPLETO)

b) Número de proveedor igual a cero.

Emite un listado por orden alfabético de todos los proveedores registrados en el sistema. Cuando se quieren conocer los estados financieros de una dependencia, solicita el código programático de ésta lee el registro correspondiente en el archivo dependencias e imprime los datos solicitados.

PROCEDURE EJECUCION

Es el proceso que ejecuta el programa, es llamado después de la etiqueta de fin de archivo, por lo cual, después de que termina cada comando, vuelve a ejecutar el siguiente que le es solicitado.

El proceso es de interacción con el usuario cuando se trabaja por terminal y solicita al usuario, todos los datos necesarios para la ejecución del mismo.

Con los datos, prende los bits correspondientes dentro de un identificador (~~reservado~~) el cual es consultado, antes de empezar a efectuar cualquier comando, de ésta manera solo permite ejecutar el programa que el usuario tiene derecho a usar.

Cuando el usuario lo solicita, informa al pedido, sobre el uso del sistema o de algún comando en especial.

Ejecuta cada comando que le es solicitado y cierra los archivos usados.

Diagrama de Flujo de
PROCEDURE EJECUCION:

