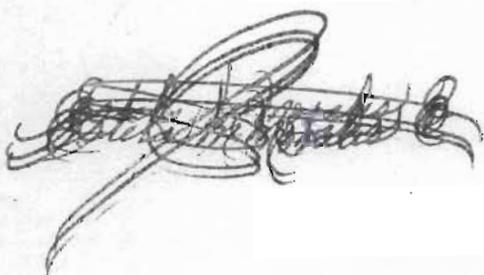


401

LIBRUNAM

SISTEMA AUTOMATIZADO PARA BIBLIOTECAS

MANUAL DE PROGRAMACION

A handwritten signature in dark ink, appearing to be 'Charlotte Bronsoiler', written in a cursive style.

CHARLOTTE BRONSOILER

ISRAEL STEINBERG

ALFREDO BRONSOILER

JUAN VOUTSSAS

ALEJANDRO RAMIREZ

INFOBILA

MEXICO: 1979

I N D I C E

	Pág.
1. Rutinas para lectura y validación de datos.	3
2. Procesos para la impresión	16
3. Rutinas para el manejo de información	28

MANUAL DE PROGRAMACION

La programación de LIBRUNAM se hizo en forma modular, de tal manera que es factible añadir nuevos procesos, sin alterar el funcionamiento de las rutinas implementadas.

Las rutinas que forman el programa de cómputo, se dividen en:

- I. Rutinas de Infraestructura
- II. Rutinas específicas para cada proceso.

I. RUTINAS DE INFRAESTRUCTURA

Son las rutinas generales usadas para crear la base del programa; se dividen básicamente en:

- i) Rutinas para lectura y validación de datos
 - ii) Rutinas para impresión
 - iii) Rutinas para manejo de información
- i) Rutinas de lectura de datos: Son las que permiten leer y validar la información de los archivos de lectura: terminales, discos, cintas o tarjetas.
 - ii) Las Rutinas de Impresión: Son las que se encargan de acomodar la información para las diferentes salidas. Se tienen las rutinas que justifican y silabeen los textos, que cambian de hoja, que imprimen encabezados, portadas, histogramas, tarjetas, etc.
 - iii) Las rutinas que manejan los diferentes tipos de archivos y procesan la información para alojarla en ellos, se dividen a su vez en:
 - a) Rutina de proceso de información que transforman

BIBLIOTECA



CENTRO UNIVERSITARIO
DE INVESTIGACIONES
BIBLIOTECOLÓGICAS

la misma aplicándole las funciones de ortografía, selección, dispersión, ordenamiento y empaquetan y alojan los nodos de cada archivo.

b) Rutinas para trabajar y controlar cada tipo de archivo.

II. Rutinas específicas de cada proceso. En general estos procesos llaman a las rutinas de infraestructura, para que internamente el computador realice una serie de funciones en forma simultánea y de esta forma se facilite el uso del sistema.

1. RUTINAS PARA LECTURA Y VALIDACION DE DATOS

PROCEDURE FILTRAR

Esta rutina lee todos los datos que se captan por tarjeta que verifica que

- Todas las tarjetas pertenecientes a un paquete, estén en orden y con el número de datos esperado.
- La sintaxis correcta de los datos, es decir: la construcción de las etiquetas, de los identificadores numéricos, de los códigos y emite mensajes de error en caso contrario.

Es un proceso de revisión dinámico y encuentra cada uno de los datos identificándolo con un separador; junta un dato dividido en varias tarjetas con un blanco como separador. Simultáneamente empaqueta la información.

Parámetros:

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

NOCOMAS	Número de separadores por paquete.
TEMP	Archivo de datos
EOF	Etiqueta de fin de archivo
CTR	Entero que indica la longitud de la información de cada registro
ETIQ	Caracter de separación.

En caso de que ocurra un error de programación se pre-
vee no descontinuar el proceso, inicializa apuntado
res y contadores; emite la ficha con problema y las
direcciones en el programa y continua revisando los
paquetes restantes.

RUTINAS PARA EL PROCESO "FILTRO".

Todas las rutinas amparadas bajo los nombres genéticos -
"filtra" o "checa" pertenecen a un proceso de revisión y vali-
dación de la información llamado "filtro".

Cada vez que ingresan fichas al sistema, pasan por el -
filtro para su depuración.

Entre las principales rutinas del filtro distinguimos:

CHECA	CHECA902
CHECATIPO	CHECA900
CHECAFECHA	CHECA650
CHECANUMERO	CHECA930
ERROR	CHECA300
CHECA035	CHECAISBN
CHECAILUS	CHECA020-
CHECAREPRODUCCION	CHECA100
CHECANUMERODIVISION	CHECA245
CHECACODIGODEFECHA	CHECA110
CHECA50	CHECA111
CHECA901	CHECA130
CHECATEXTO	

PROCEDURE CHECA

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

Parámetros:

P Apuntador al arreglo que aloja la cadena.
LC Largo de la cadena después de ser revisada.
MAX Número máximo de caracteres que puede contener la cadena.

PROCEDURE CHECATIPO.

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 Banco 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 las fechas y las 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.

PROCEDURE ERROR

Este proceso emite el mensaje de error correspondiente, dependiendo del valor de "ERRONEO", emite la imagen de tarjeta señalando con un asterisco el lugar donde está localizado el error.

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

Parámetros:

ERRONEO Número del error que se presentó.

PROCEDURE CHECA 035

Verifica que sólo exista una etiqueta 035 en el paquete y que no esté vacía. Checa que sea cadena numérica y verifica que no exista el número de tarjeta registrada en el banco y em paca la información.

PROCEDURE CHECA ILUS

Revisa un campo de 4 caracteres, verificándolo contra una tabla y traduce a un valor numérico cada carácter; si no se cumple, manda un mensaje de error.

PROCEDURE CHECA REPRODUCCION

Revisa un campo de un carácter contra una tabla y lo traduce a un valor numérico; manda un mensaje de error si no está en la tabla.

PROCEDURE CHECA NUMERO DIVISION

Normaliza la cadena y revisa que el identificador sea numérico y asigna el valor del identificador cuando esté dentro del rango permitido.

Parámetros:

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

PROCEDURE CHECACODIGO DE FECHA

Revisa el código determinador de fechas de publicación -- y las fechas, son los nueve primeros caracteres de la parte - fija.

El primer caracter se checa contra una tabla y verifica- que las fechas sean lógicas.

1.1.12 PROCEDURE CHECA 50

Checa que venga solo una etiqueta 050; prende una bande- ra indicando esto, y se comienza a filtrar la información de - la manera siguiente:

- 1o. Se quitan los blancos que preceden a la información-
- 2o. Se calcula la longitud de la clasificación contenida hasta antes del separador "/".
- 3o. Se revisa que la información comience con un caract- er alfabético, conteniendo no más de dos.
- 4o. Llama a "CHECATIPO" el cual nos indica si la cadena- se encuentra contenida en la tabla "ACLASES". Que - contiene todas las clases posibles válidas.
- 5o. Se verifica que la parte numérica no exceda de cua- tro dígitos.
- 6o. Si en la cadena nos encontramos con un separador "." se checa que traiga las estructuras N, LN, NLN, con- los siguientes rangos: El 1o. si es numérico tres - caracteres; el alfabético 1 y el 2o. numérico cua- tro caracteres.

PROCEDURE CHECA 901

Revisa la etiqueta 901 la cual, por cada ficha debe venir solo una vez; prende una bandera para indicarlo. Verifica que la información sea de la forma:

- a) Número
- b) Número - Número.

CASO a) Este caso es cuando la ficha ampara a un solo volumen.

Esta información es revisada por el "PROCEDURE CHECA NUMERO DIVISION", la guardan como cadena y en el directorio se indica que es un solo volumen.

CASO b) En este caso además de ser checado por el "PROCEDURE CHECANUMERO DIVISION", las cadenas, verifica que los rangos entre las cadenas sean lógicos y obtiene el número de volumen sacando la diferencia + 1.

Ejemplo:

256-57 Esta cadena está correcta y son dos volúmenes los que ampara.

289-88 Cadena errónea, la segunda cadena es menor que la primera.

Además se crea un directorio para estadísticas y la información se guarda como cadena en la parte variable y en el directorio como números de volúmenes.

PROCEDURE CHECATEXTO

Este proceso se encarga de revisar las cadenas alfanuméricas, llama al Procedure "Checa", crea el directorio de la cadena y empaca la información.

PROCEDURE CHECA 902

Este proceso revisa la etiqueta 902; por cada ficha debe venir sólo una de estas y prende una, bandera verifica que la información sea de la escritura.

CLAVE/CLAVE/CLAVE

Cada clave consta de tres caracteres, buscados en tablas por el "PROCEDURE CHECATIPO". Se crea además un directorio cuando la ficha es capturada por la D.G.B., para estadísticas.

PROCEDURE CHECA 900

Revisa la etiqueta 900; esta cadena alfanumérica se compara contra una tabla de claves de dependencias. Si no hay error en la información se empaca la información; además se crea un directorio para estadísticas.

PROCEDURE CHECA 650

Revisa que la cadena tenga información y localiza el separador "-" en caso de encontrarlo verifica que tenga "Ø" antes y después; revisa la cadena con el "PROCEDURE CHECA" y verifica que el número de separadores no exceda a "NSEP" y empaca la información si no hay error.

Parámetros:

"NSEP" Número de separadores permitidos por etiqueta.

PROCEDURE CHECA 440

Este proceso revisa información que sea de estructura:

Número

Número: cadena alfanumérica

La cadena numérica se revisa con el "PROCEDURE CHECANU--
MERODIVISION" y si encuentra el separador";" se revisa -
la segunda cadena con "Checa", si no hay error se empa-
ca la información.

PROCEDURE CHECA 430

Revisa información de la siguiente estructura:

cadena alfanumérica

cadena alfanumérica; cadena alfanumérica

Esta información se checa con el "PROCEDURE CHECA", si -
no hay error se empa ca la información verificando que la
longitud de la cadena no exceda de 1510 caracteres.

PROCEDURE CHECA 300

Revisa la información de las etiquetas "250, 260, 300, -
500, 501, 502, 505, 520"; verifica que de las "250, 260,
300" a lo más exista una por ficha, las etiquetas se che-
can por grupos de la siguiente forma:

- 1o. 250
- 2o. 300
- 3o. 500, 501, 502, 505, 520, 260

- 1o. (250).- La información puede venir de la siguiente-
forma:

N = número	NNNN	N
A = cadena alfanumérica	NNNN # AAAAA	N #
	# AAA	#

Se verifica la información Numérica con "CHECANUMERODIVISION" y en caso de que encuentre un "#" se valida la parte alfabética y la empaqa.

2o. (300) La información puede venir de la siguiente manera:

NNNN	separador		N	I
NNNN	separador	#	AAAAAA	I 'v'
			N	I #
			I	#
			#	

Se verifica la información Numérica con "CHECANUMERODIVISION". El caracter alfabético se valida que sea un identificador válido; si se encuentra separador se verifica la cadena con "CHECA".

3o. Solamente se revisa que los caracteres sean alfanuméricos con "CHECA".

Si no hubo error se procede a empacar.

BOOLEAN PROCEDURE CHECAISBN.

Este proceso recibe cadenas numéricas con 18 caracteres máximo y sólo el último caracter puede ser "x", la información debe estar justificada a la derecha.

Multiplica por 10 a partir del 10o. caracter de derecha a izquierda y acumula el producto, y la cadena restante la multiplica por el valor de su posición sumandolo con los anteriores: sólo el último caracter si es "x", lo -

toma como 10.

Si el módulo es cero nos retorna "TRUE" y "FALSO" si no lo es.

Parámetros:

Número Arreglo con la información.

PROCEDURE CHECA 020,

Este proceso corrobora que la información sólo tenga los siguientes caracteres "0123456789\0-x"; compacta la información, quita los separadores "-" y "\0" deja la información justificada a la derecha y se la pasa al "PROCEDURE CHECAISBN"; si la información esta correcta se procederá a empacarla y en caso contrario se manda mensaje de error.

Parámetros:

TIPO. Esta variable nos indica que etiqueta es.

B020 Si este parámetro es "TRUE" la información se empaca no importa que tenga error y si es "FALSE" se da mensaje de error.

PROCEDURE CAPTURADOR

El procedure "capturador" está asociado íntimamente con el comando captura. Su función es captar la información de la terminal, reorganizarla y grabarla en un archivo. Es un procedure de tipo real, sin parámetros de "pase". Trabaja con el arreglo global "fichas" y los "defines" "SAY" y "SEND" globales para envío de mensajes. Todo lo demás son parámetros locales.

INFOBILA

Envía y recibe mensajes para identificar al usuario y de finir si éste desea longitudes de grabación y formato normal o alguno en especial.

Lee la ficha completa en el arreglo "TEXTO"; si trae alguno de los caracteres de control de la rutina (< > que cierran la captura, @ que termina una ficha y ! que despliega el formato guía).

El procedimiento reescribe en el arreglo "escribe" la información de la ficha en registros de 119 caracteres; un "_" en el 120 si continúa la ficha ó un "Ø" si no, el número de ficha y de registro en las siguientes ocho columnas para un total de 128 caracteres (22 palabras) por registro (Longitud normal), ó a la longitud que especifique el usuario.

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 FILTRA 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

2 RUTINAS PARA LA IMPRESION

PROCEDURE ESCRIBE

Rutina que imprime en papel o terminal casi todos los resultados del programa. Un contador (LINEAS), se incrementa cada vez que escribe y cuando llega al valor máximo de líneas que se pueden imprimir. (NLINEAS), cambia de hoja y las numera (TIT) si los resultados son por impresora. Si los resultados son por terminal suspende la impresión y separa una lectura en blanco para seguir.

Parámetros:

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

PROCEDURE SEND

Inicia el contador de líneas en cero y llama al procedimiento 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 1975.

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 ficha compacta cuando es verdadera
- P Es el apuntador al arreglo donde se desea escribir la fecha.

PROCEDURE ENCABEZADO

Genera los encabezados de los procesos del sistema.

Parámetros

- A : Número del título que se lleva en el encabezado.
- B : Número de la dependencia en la cual se va a entregar el listado. Si B=0, se omite el nombre de la dependencia.

IMPRESOR: Nombre de archivo de impresión

LINEAS: Contador de líneas

TIT: Contador de hojas.

PROCEDURE CENTRA:

Proceso que ajusta los textos dentro de un espacio dado para dejar margen uniforme a los lados del mismo; silabea e introduce blancos adicionales cuando es necesario, ajusta un renglón y deja actualizados los apuntadores para la siguiente llamada 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 línea que se desea centrar.
LC : Largo del identificador que se va a centrar, este valor regresa alterado, devolviendo el número de caracteres que faltan por escribir.
D : Devuelve el número de caracteres que fueron ajustados.

RUTINA SACAINFORMACION

Se encarga de construir las fichas bases y escribirlas en un arreglo a partir de un nodo empacado. Prepara además dos archivos con los nodos que representan para otras rutinas, estos nodos incluyen los identificadores del registro de las fichas en caso de que se pidan juegos desarrollados.

PARAMETROS:

- AFILTRO. Arreglo con la información "empacada de la ficha.
REGLON Entero, número de renglones. En la recuperación de línea esta variable regresa el tamaño de la ficha.
ACONT 1 Arreglo que prepara la información para desplegar la línea.
CUAL Número de columnas para ampliar información.
ORIGINAL Variable real con los siguientes subparámetros.

BITS:

47-44	NUMERO DE FICHA ORIGINALES
43-41	NUMERO DE FICHAS DESARROLLADAS
40-37	NUMERO DE FICHAS ORIGINALES POR AUTOR
36-33	NUMERO DE FICHAS ORIGINALES POR CLASIFICACION
32-30	NUMERO DE FICHAS DESARROLLADAS POR DICCIONARIO
10	SE ORDENAN POR DEPENDENCIA Y # DE ADQUISICION LAS FICHAS INDICADAS EN 47-41
9	SE CONSTRUYE LA FICHA SIN NOTAS, ISEN, TEMAS NI COAUTORES.
8	SE IMPRIMEN REMISIONES
7	NO (OTRA RUTINA)
6	NO (OTRA RUTINA)
5	SE IMPRIMEN LAS FICHAS EN MINUSCULAS, PERO CIERTOS ENCABEZADOS EN MAYUSCULAS
4	NO (OTRA RUTINA)
3	SE HACEN JUEGOS DESARROLLADOS
2	NO
1	NO (OTRA RUTINA)
0	NO (OTRA RUTINA)

Las fichas base se encuentran en el archivo FSAL2. El de nodos para alfabetizar se llama FILEINTEM y el de nodos para ordenar por dependencia o simplemente imprimir sin orden se llama FICHA.

Usa las siguientes subrutinas:

SUBROUTINA SACA 50:

Desempaca: la clasificación, el número de tarjeta con dígito contro, la clave de la dependencia y el número de

adquisición, los escribe en la ficha base. Si la ficha es de continuación le escribe la abreviación "CONT...".

Manda hacer copias extras cuando es necesario; acumula en el archivo de remisiones cuando se requiere para más de una biblioteca y contabiliza el número de volúmenes y títulos por dependencia.

PARAMETROS:

Booleano que indica si es de continuación la ficha.

SUBROUTINA SACACIENES:

PARAMETRO:

QUETIO. Entero, indica las etiquetas 1XX, 6XX o 7XX.

1XX: Si alguna etiqueta de AUTOR, existe en ese caso la desempaca y la escribe en fichas base. (X es cualquier número)

6XX y 7XX: Todas las etiquetas que le corresponden a estas familias, las escriben en la ficha base, precedidas por un número arábigo secuencial para los 6XX y romano para los 7XX.

Si se hacen juegos desarrollados, los escribe en un arreglo extra.

SUBROUTINA SACACIENTO DIEZ:

Funciona igual que la anterior, con la diferencia del caso especial de las etiquetas:

110, 610, 710

PARAMETRO:

QUETIZ. Igual significado que el anterior.

SUBROUTINA SACA 245:

Desempaca el TITULO y los SUBTITULOS en el archivo de fichas base; de tener la opción "desarrollo", los escribe en un arreglo extra.

Asigna el número romano en forma secuencial como se hizo para los COAUTORES, a los SUBTITULOS y al TITULO.

PARAMETRO:

ETIQ. Entero, puede tener los valores 245 o 240.

SUBROUTINA SACACADENAS:

Desempaca las etiquetas únicas o múltiples:

Ø2Ø, 25Ø, 26Ø, 3ØØ, 43Ø, 44Ø, 5ØØ, 5Ø1, 5Ø5, 52Ø, 65Ø

Y las ilustraciones en los lugares asignados.

En caso de tener la opción de "desarrollo", escribe los TEMAS en un arreglo extra, para utilizarlos como encabezamiento de fichas.

PARAMETRO:

QUIETIQ. Entero, indica que etiqueta va a ser desempacada.

SUBROUTINA DFCENTRA:

En todas las subrutinas anteriores, la información que se desempaca, se escribe en el arreglo AYUDA.

Los caracteres que se encuentran en este arreglo, se escriben en el archivo de fichas base una vez que han sido silabeado y justificados con dimensiones solicitadas para construirlas.

En caso de que la información sobrepase el límite de renglones y recuperación en línea, se llama a la rutina de continuación de fichas.

PARAMETRO:

CARGO Entero, indica la longitud de la cadena centrada.

SUBROUTINA PRODOBLE:

Acondiciona la ficha presente que no tuvo espacio para -- ser finalizada en el arreglo normal, para que tenga las - características de una ficha de continuación, la escribe en el arreglo de fichas base.

SUBROUTINA POR CAMPOS:

Desempaca un nodo de ficha bibliográfica, deja la información con el formato tipo de captura..

PARAMETROS:

ÁFILTRO: Nodo con la información

B1: Entero: determina la construcción de la información

Ø : INFORMACION EN LINEA

1 : ESCRITURA EN UN ARCHIVO, CON LAS MISMAS CARACTERISTICAS QUE EL CAPTURADO ORIGINALMENTE

SACADOR: Entero:

NOT FASE: Desempaca todas las etiquetas, tal como -
la ficha original capturada.

OTRO VALOR: Desempaca las etiquetas que le corres--
ponde a los bits prendidos, de acuerdo a
su clave dentro del nodo.

Para información en línea, la información se desempaca -
en el arreglo ACONT1 de una dimensión; cada 8 caracte--
res representan un renglón en la terminal; el total de -
éstos se guarda en la variable ULTIMONUMFICHA.

El archivo con la información en el formato de captura -
se llama VALCAMPO.

Si sacador tiene el valor "cero".

Para desempacar la información BARRE el directorio glo--
bal del nodo, mientras existen etiquetas, desempaca su -
dirección y longitud correspondiente llama a la rutina -
que se encarga de dejarla como fue capturada originalmente.

En caso de que sacador tenga otro valor, investiga cada-
uno de los 33 bits menos significativos, si alguno está-
prendido y dicha clave de etiqueta existe, las desempacada
todas con la rutina correspondiente.

SUBROUTINA SACAOCHO:

Traduce los campos numéricos de la parte fija del campo-
del mismo nombre, los deja en forma numérica o en clave-
alfabética.

SUBROUTINA SACACIENES, SACACIENTO DIEZ, SACA 245:

Se encargan de reconstruir las etiquetas de la misma forma

ma que aparecían en la ficha capturada, desempaca las partes y añade los separadores particulares:

- + La subrutina sacacienes contempla las etiquetas: 100, 111, 130, 240, 600, 630, 700, 711, 730.
- + La subrutina sacaciento diez contempla las etiquetas: 110, 610, 710.
- + La subrutina saca 245 contempla las etiquetas: 245, 740.

SUBROUTINA SACACADENAS:

Desempaca con sus características particulares todas las etiquetas restantes por medio de un "CASE OF", agrupa por grupos las similares y desempaca de la misma manera.

RUTINA DESEMPACA

Localiza en el directorio global de la ficha en proceso una etiqueta. Regresa su longitud y dirección; en caso de no hallarla, estas variables son ceros.

Actualiza el apuntador donde encontró el identificador - en caso de que exista más de una etiqueta (por ejemplo TEMAS), la siguiente vez empezará a buscar la etiqueta a partir de él.

PARAMETROS:

ETIQUETA. Clave de la etiqueta.

LIM Apuntador desde donde empieza a buscar en el directorio del nodo. La primera vez debe ser cero y se actualiza por si se desean desempacar más etiquetas con la misma clave.

RUTINA GENERAL ISBN.

Calcula el dígito de control módulo 11 a una cadena de números que se encuentran en un arreglo.

PARAMETROS:

- NUMERO. Arreglo con longitud/hasta de 18 caracteres donde se encuentra la cadena dato.
- Q. Apuntador del arreglo donde se deja la cadena de números con su dígito de control.
- LOGNUM. Longitud de la cadena resultado.

RUTINA IMPREMISION.

Se utiliza para imprimir remisiones, desde el encabezado con el nombre de la dependencia, hasta el cuerpo de la remisión.

PARAMETROS:

- TITUL. Arreglo donde se encuentra el TITULO para el encabezado.
- NECABEZA. Posición del arreglo TITUL donde empieza.
- A900. Arreglo donde se localiza el nombre de la dependencia.
- IVOL. Número de volúmenes que ampara.
- FREMS. Archivo con los datos de los libros que ampara la remisión.

SUBRUTINA CARGOFICHA:

Lee un archivo de la forma:

NUMERO DE FICHA \$ DEPENDENCIA \$ NUMEROS DE ADQUISI-
CION

Lee la ficha del archivo y reemplaza la dependencia y el número de adquisición, en caso de que las tenga, sino simplemente los da de alta.

Se utiliza para hacer los cargos a dependencias. De este nodo modificado se pueden hacer fichas catalográficas y remisiones de cargo.

Checka que el número de ficha exista, que la clave de dependencia esté dada de alta y que los números de adquisición sean lógicos.

El caso del error, imprime a diagnóstico, ignora los datos equivocados y continúa con los siguientes.

PARAMETROS:

Bc1. Booleano; falso indica que leerá el archivo FSINFICHA y verdadero que leerá FCONFICHA.

Estos archivos son acumulables. Al terminar el proceso remueve el archivo.

PROCEDURE SILABEA.

El procedimiento se encarga de cortar las palabras en sílabas, con las reglas del Castellano. Estrictamente hablando, tiene un pequeño margen de error en las sílabas que se forman a partir de un diptongo acentuado.

En la práctica este error no es visible, pues el procedimiento corta hasta la siguiente sílaba.

El procedimiento es de tipo entero; recibe los parámetros CAMPOESC y PAY. CAMPOESC es una variable entera que indica los espacios disponibles en un renglón para escribir una palabra de un texto y PAY es un apuntador que trae el texto a silabear, posicionado en el primer carácter. El procedure al final toma el valor (entero) del número de caracteres que forman sílabas completas y que es el máximo que cabe en el espacio CAMPOESC disponible.

Verifica que la cadena sea alfabética; si no es así, el procedimiento regresa la longitud de la cadena como sílaba; la filosofía de la rutina es formar "árboles" con las vocales y las consonantes. Primero se checan las palabras o sílabas de 2, 3, y 4 caracteres, que son más fáciles de identificar. Después se recorre cada caracter formando un árbol y buscando combinaciones hasta que se obtiene una sílaba de "n" caracteres, cuyo valor N se guarda en LSIL. Después de formar una sílaba, se investiga si en la longitud disponible para escribir alcanza para otra sílaba y se reinicia el proceso; en el momento en que el número de caracteres acumulados en LSIL es el máximo que cabe en la longitud disponible y contiene sílabas completas regresa.

3. RUTINAS PARA EL MANEJO DE INFORMACION.

PROCEDIMIENTO ORTOGRAF:

Transforma la ortografía de un texto. Quita caracteres repetidos.

PARAMETROS:

A (v-Pointer) Descriptor que señala en donde se encuentra la cadena de caracteres a la que se le modifica la ortografía.

N (v-Entero) Longitud de la cadena.

PROCEDIMIENTO DAMEPALABRA: (Booleano)

Entrega una por una las palabras de un texto. (Usa los caracteres blanco para separar las palabras).

Variables globales utilizadas:

POINTERTRABAJO (Pointer) Descriptor que señala de que arreglo tomar las palabras. Avanza a una nueva posición cada vez que se llama al procedimiento.

NUMEROTRABAJO (Entero) Indica cuantos caracteres quedan del arreglo donde se toman las palabras. Se actualiza cada vez que se llama al procedimiento.

PARAMETROS:

(DAMEPALABRA) (Valor que regresa el procedimiento).

Un valor TRUE indica que ya no hubo más-

"tabla descriptora".

ELEMENTOSACA (V-Real) Un elemento de la "tabla descriptora".

QUITAR (V-Booleana) TRUE indica que se desean quitar palabras no significativas.

ORTOGRAFIA (V-Booleana) TRUE que se desea transformación fonética.

PROCEDIMIENTO ALTALISTAS:

Toma un archivo de listas-generadas, lo ordena formando un archivo de listas-ordenadas. Mezcla los apuntadores de este archivo con los de un archivo de listas o crea nuevas listas en este último y en el archivo de dispersión según sea necesario.

Al terminar el archivo de listas-ordenadas y de listas generadas se remueven automáticamente.

PROCEDIMIENTOS LLAMADOS:

ABRE, CIERRA, LEEARCHIVO, ESCRIBEARCHIVO, DIRECTORIODOBLE.

PARAMETROS:

PALDISP Archivo de dispersiones (Tipo Directorio Doble)

APALDISP Arreglo de trabajo de PALDISP

PALLIST Archivo de listas (Tipo Dinámico-Ligado)

APALLIS Arreglo de trabajo de PALLIST

LISTAIN Archivo de listas-generadas. Su formato es el siguiente:

- 1a. palabra: Dispersión
- 2a. palabra: Elemento de la lista
- 3a. palabra: Control: 1 = alta
0 = baja

PROCEDIMIENTO PREPARA ISBN:

Sirve para preparar un número de ISBN, obtenido de una "tabla-descriptora" (ver SACAFILTRO), para permitir hacerle una dispersión. La preparación consiste en seleccionar los caracteres numéricos o la letra "X" y unirlos todos formando una cadena continua.

VARIABLES GLOBALES UTILIZADAS:

POINTERTRABAJO (Pointer) Descriptor que señala donde poner el número preparado.

Al salir del procedimiento queda apuntando al final de la cadena.

NUMEROTRABAJO (Entero) Señala de cuántos caracteres consta la cadena.

PARAMETROS:

Ebase Arreglo de caracteres al que apunta la "tabla-descriptora".

ELEMENTOSACA (V-Real) Un elemento de la "Tabla-descriptora".

PROCEDIMIENTO REVISAS (BOOLEANO)

Revisa que el archivo sea tipo adecuado, que haya sido cerrado correctamente la última vez que se actualizó y que la versión sea correcta con el archivo. Cuando es dinámico-ligado (TIPO=1) comprueba que la dirección al próximo lugar disponible tenga marca de "fin de-archivo".

Si el programa encuentra una falla, el procedure regresa un valor TRUE y un número que indica el error. El valor que se conserva en la variable booleana global "ERRORENARCHIVOS" con el siguiente formato:

bits 0:1 El valor 1 indica error.

bits 4:4 Indica el error que se detectó.

- 1 = El tipo de archivo no corresponde al especificado.
- 2 = El archivo no se cerró correctamente la última vez que fué actualizado.
- *3 = No corresponde el último registro del archivo con el apuntador "próximo-disponible".
- *4 = No corresponde la marca de fin-de archivo con el apuntador "próximo disponible".

* Solo para tipo 1

PARAMETROS:

(REVISAS) (Valor que regresa el procedimiento)
ARCHIVO: Archivo a revisar
ARREGLO: Arreglo de trabajo
TIPO (V-Entero) Tipo de archivo:

- 1 = Dinámico-Ligado
- 2 = Directorio
- 3 = Directorio-Doble

PROCEDIMIENTO ABRE:

Lee del registro cero de los archivos las palabras de control y las coloca en el arreglo de trabajo. Revisa que la versión del archivo sea la correcta, usando el archivo de control que haya sido cerrado correctamente la última vez que se actualizó y que sea del tipo adecuado. Cuando el archivo es dinámico-ligado (TIPO=1) comprueba que el apuntador al próximo lugar disponible del archivo efectivamente apunte a la marca de fin-de-archivo del mismo.

Si el programa encuentra una falla, se detiene en una condición de "NO-FILE" (Indicando que el archivo no está disponible) para permitir al operador asignar un nuevo archivo o terminar el proceso.

Si todo está correcto, marca en las palabras de control que el archivo fué abierto.

Cuando se actualiza el atributo "MYUSE" tiene un valor 10, y el programa graba el registro 0, indicando que el archivo fué abierto. Cuando sólo se va a acceder el archivo, el atributo MYUSE tiene un valor IN.

PARAMETROS:

- ARCHIVO: Es el archivo que se desea abrir
- ARREGLO: Arreglo de trabajo de ARCHIVO
- TIPO (V-Entero): Señala el tipo de archivo de que se trata:

- 1 = Dinámico-Ligado
- 2 = Directorio
- 3 = Directorio-Doble

Condiciones de error:

- 1 = El tipo del archivo no corresponde al especificado.
- 2 = El archivo no se cerró correctamente la última vez - que fué actualizado.
- *3 = No corresponde el último registro del archivo con el apuntador "próximo-disponible".
- *4 = No corresponde la marca de fin-de-archivo con el - apuntador "proximo-disponible".

* Sólo para TIPO = 1

PROCEDIMIENTO CIERRA

Cierra un archivo abierto. Para actualización; graba en el registro cero las palabras de control, tomándolas del arreglo de trabajo, actualiza la versión del archivo en el archivo de control.

PARAMETROS:

ARCHIVO	Archivo
ARREGLO	Arreglo de Trabajo de ARCHIVO

PROCEDIMIENTO LEEDIRECTORIO (Booleano)

Lee un archivo tipo directorio. Antes de permitir el acceso revisa que el número de dato este entre los límites válidos (mayor o igual a cero y menor al último especificado del directorio).

PARAMETROS:

(LEEDIRECTORIO) (Valor que regresa el procedimiento. También se guarda en la variable global booleana ERRORENARCHIVOS).

Un valor TRUE indica que hubo error y los demás bits indican de que tipo de error se trata:

0:1 Un valor 1 indica error.

4:4 Tipo de error.

1 = Directorio no se abrió antes de actualizarlo.

2 = NUMERO fuera de rango.

3 = Error del sistema al tratar de leer un registro físico.

DIRECTORIO Directorio que se desea acceder.

ARREGLO Arreglo de trabajo de DIRECTORIO

NUMERO (V-Entero) Número de dato que se desea leer.

PDATOS (V-Pointer) Pointer donde se van a colocar los datos leídos.

PROCEDIMIENTO ESCRIBEDIRECTORIO (Booleano)

Escribe en un archivo tipo directorio. Antes revisa que el número de dato por escribir sea válido:

a) Mayor o igual a cero.

b) Menor o igual al límite especificado de datos en la creación del directorio ó (ver c).

- c) Que el directorio sea expandible (No se haya usado - la opción CRUNCH al crearlo) y el número de dato sea igual al número de control secuencial.

PARAMETROS

(ESCRIBEDIRECTORIO) (Valor que regresa el procedimiento También se guarda en la variable global-booleana ERRORENARCHIVOS).

Un valor TRUE indica que hubo error y - los demás bits indican que tipo de - error se trata:

0:1 Un valor 1 indica error.

4:4 Tipo de error.

1 = Directorio no se abrió antes de es-cribir o se abrió solo para lectura.

2 = Número fuera de rango.

3 = Error del sistema al tratar de leer un registro físico.

DIRECTORIO Directorio en el que se desea escribir.

ARREGLO Arreglo de trabajo de DIRECTORIO

NUMERO (V-Entero) Número de dato que se desea escribir

PDATOS (V-Pointer) Pointer donde viene la información - por escribir.

PROCEDIMIENTO LEEARCHIVO (Booleano)

Lee un archivo tipo dinámico-ligado. Antes de permitir el acceso revisa que el apuntador al nodo sea una dirección - aceptable;

- a) El valor sea mayor o igual a 256 (256 apunta al 1er. registro del archivo, palabra 0. El registro cero se usa como control) y menor al valor del fin-de-archivo.
- b) El número de la palabra del registro sea menor o igual a 238.
- c) Que el nodo sea válido.
 - i) Correspondan los caracteres de control.
 - ii) No esté dado de baja.
 - iii) No sea de continuación.

PARAMETROS:

(LEEARCHIVO) (Valor booleano: que regresa el procedimiento. También se guarda en la variable global booleana ERRORENARCHIVOS).

Un valor TRUE indica que hubo error y los demás bits indican de que tipo de error se trata:

0:1 Un valor 1 indica error.

4:4 Tipo de error.

1 = El archivo esta cerrado

2 = El apuntador es inválido

3 = Error del sistema al tratar de leer un registro físico.

4 = La liga a un nodo de continuación es inválida.

5 = El arreglo proporcionado por el programa para colocar la información no es de tamaño suficiente.

palabras.

PSAL (Pointer) Descriptor que indica en donde está la -
palabra seleccionada.

N (Entero) Longitud de la palabra seleccionada.

PROCEDIMIENTO PREPARACADENA:

Sirve para preparar un texto, obtenido de una "tabla-descriptora", (ver Sacafiltro), para permitir seleccionar palabras del mismo y hacerles una dispersión.

Las opciones que tiene son dos:

- I) Quitar palabras no significativas
- II) Hacer transformación fonética de las palabras.

El procedimiento se puede llamar varias veces seguidas - acumulando al final de la cadena el nuevo texto.

Variables globales utilizadas:

POINTERTRABAJO (Pointer) Descriptor que señala en donde poner el texto preparado. Al salir del procedimiento queda apuntando al final - de la cadena.

NUMEROTRABAJO (Entero) Señala de cuántos caracteres - consta el texto preparado.

PROCEDIMIENTOS QUE UTILIZA:

NORMALIZACION, DAMEPALABRA, ORTOGRAF.

PARAMETROS:

EBASE °(Arreglo de caracteres) Arreglo al que apunta la-

- ARCHIVO: Archivo que se desea accesar.
- ARREGLO: Arreglo de trabajo de ARCHIVO.
- APUNTADOR: (real) Al entrar: Apuntador que señala de -
que parte del archivo leer.
- Al salir: Liga del usuario. (0 si no -
hay liga)
- TEXTO (V-Pointer) Pointer que señala en donde colocar -
la información leída.
- LONGITUD (Entero) Longitud en caracteres de la informa-
ción leída.

PROCEDIMIENTO ESCRIBEARCHIVO (Booleano)

Escribe en un archivo tipo dinámico-ligado, previamente-revisa que el apuntador del nodo sea aceptable:

- a) El valor cero, indica se desea escribir al final de archivo; el valor NOT FALSE, indica que se escribe en registro nuevo.
- b) Otro valor debe ser mayor o igual a 256 (256 apunta al 1er. registro del archivo, palabra 0. El registro cero se utiliza como control), y menor o igual al valor del fin de archivo y el nodo debe ser válido (si no se trata del fin de archivo):
 - i) Corresponda el o los caracteres de control.
 - ii) El nodo no es de continuación.

PARAMETROS:

(ESCRIBEARCHIVO) Valor Booleano que regresa el procedimiento. También se guarda en la variable global ERRORENARCHIVOS).

Un valor TRUE indica error y los bits -
restantes indican de el tipo:

0:1 Un valor 1 indica error

4:4 Tipo de error

1 = El archivo esta cerrado o se abrió -
sólo para lectura.

2 = El apuntador es inválido.

3 = Error del sistema al tratar de leer-
o escribir un registro físico.

4 = La liga a un nodo de continuación es
inválida.

5 = El arreglo de los datos no es de ta-
maño suficiente.

ARCHIVO: Archivo

ARREGLO: Arreglo de trabajo de ARCHIVO.

APUNTADOR (Real) Al entrar: Apuntador que señala en -
que parte del archivo escribir:

0 = En el próximo disponible

NOT FALSE = En el próximo registro dispo
nible.

N = En la dirección N.

Al salir: Liga del usuario guardada (0
si no hay liga).

LIGA (Real) Al entrar: Contiene la liga del usuario:

0 = No se guarda liga

NOT FALSE = Se guarda liga a próximo dis
ponible.

N = Se guarda valor N.

Al salir: Contiene el valor del apuntador donde se escribió la información.

TEXTO (V-Pointer) Pointer que indica en donde se encuentra la información a escribir.

LONGITUD (V-Entero) Longitud en caracteres de la información.

PROCEDIMIENTO DIRECTORIODIBLE (Booleano)

Lee y escribe en un directorio doble de dispersión. A continuación se presenta una tabla que explica su funcionamiento:

PARAMETROS DE ENTRADA		VALOR INTERNO GUARDADO		PARAMETROS DE SALIDA	
Entrada	Asignar	No. Asociado Al entrar	Al salir	Dirección-Directorio Doble	
X	FALSE	0	0	0	FALSE
X	TRUE	0	X	X	FALSE
X	FALSE	Y	Y	Y	TRUE
X	TRUE	Y	Y	Y	TRUE

El parámetro TEXTO se utiliza para identificar cada elemento grabado en el archivo. El parámetro DISPERSION se usa para señalar en qué registro se guarda TEXTO junto con su número-asociado.

PARAMETROS:

(DIRECTORIODOBLE) (Valor que regresa el procedimiento.- También se guarda en la variable global booleana ERRORENARCHIVOS). FALSE-TEXTO-no existe en el directorio-doble.

TRUE-TEXTO ya existe en el directorio-doble.

ARCHIVO: Directorio-doble donde se va a leer o escribir.

ARREGLO: Arreglo de trabajo de ARCHIVO.

ENTRADA (Real) (Sólo se usa para escribir) Valor que se desea guardar en el directorio-doble asociado a TEXTO.

DISPERSION (V-Real) Número que se utiliza para indicar en que parte del directorio-doble se localiza TEXTO.

TEXTO (V-Real) Una palabra que se utiliza para identificar cada elemento guardado o por guardar en el directorio-doble.

ASIGNAR (V-Booleana) FALSE indica solo lectura de la información TRUE indica, en caso de no existir el TEXTO en el directorio-doble, se guarda junto con ENTRADA como número asociado.

DIRECCION (Real) Entrega el número guardado en el directorio-doble asociado a TEXTO.

PROCEDIMIENTO DISPERSION (Real)

Es el que realiza el encimamiento de bits de una cadena de caracteres.

PARAMETROS

(DISPERSION (Valor que regresa el procedimiento).

Contiene el resultado del encimamiento de bits.

ARREGLO (V-Pointer) Descriptor que señala en que parte está la cadena a la que se le va a hacer la dispersión.

NCARACT (V-Entero) Longitud de la cadena en caracteres.

NBITS (V-Entero) Número de bits que debe contener la dispersión.

PROCEDIMIENTO INTERSECCION (Booleano)

Se encarga de realizar la intersección entre las listas. Su funcionamiento está dividido en cuatro partes:

- I) Inicialización: Limpia las variables y arreglos necesarios para poder iniciar una nueva intersección.
- II) Carga de listas. Recibe los elementos de cada lista y los acomoda en los arreglos de trabajo.
- III) Intersecta. Realiza la intersección entre las listas cargadas en el punto II.
- IV) Entrega. Va entregando uno por uno los elementos de la lista resultado.

VARIABLES globales de trabajo:

NUMBUFINTERSECTAR (Definición) su valor debe ser mayor o igual a tres e indica con cuantos arreglos de trabajo debe funcionar la intersección.

CONTROLBUFFER Arreglo que se usa para administrar los arreglos de trabajo.

CONTROLES Arreglo que guarda los descriptores de las listas.

BUFFER (2 dimensiones) Son los arreglos de trabajo.

PARAMETROS

(INTERSECCION) (Valor que regresa el procedimiento).

Sólo se usa para TIPO = 0. Vale TRUE - cuando ya no hay más listas que entregar.

TIPO (V-Entero) Indica que paso de la intersección realizar:

- 2 = Inicializa

- 1 = Intersecta

0 = Entrega lista resultado

0>N = Recibe un elemento de la lista N.

VALOR (Real) Su uso depende del valor de TIPO:

Tipo = 2: No se usa

Tipo = 1: Regresa el número de elementos de la lista resultado.

Tipo = 0: SI INTERSECCION vale FALSE - regresa un elemento de la lista resultado.

Tipo = N: Contiene un elemento de la lista N.

PROCEDIMIENTO NORMALIZACION:

Normaliza una cadena de caracteres:

a) Quita puntuación convirtiéndola en blancos (solo de-

ja palabras y números).

b) Convierte minúsculas a mayúsculas.

c) Quita blancos múltiples.

PARAMETROS

PSAL (Pointer) Descriptor que indica donde se coloca la cadena normalizada.

PENT (V-Pointer) Descriptor que señala de donde tomar la cadena por normalizar.

NUM (V-Entero) Longitud de la cadena por normalizar.

PROCEDIMIENTO SORTEA

PARAMETROS:

CAMPOESC Entero que indica caracteres disponibles.

PAY Apuntador a la cadena.

EL "PROCEDURE" es entero y toma el valor del número de caracteres que se deben cortar.

Esta rutina esta asociada al comando "ORDENA" pero también la llaman otras rutinas del sistema. Su función es acomodar un número de fichas por orden alfabético de autor, dependencia ó clasificación.

Es un procedure de tipo REAL y recibe los parámetros BLLAM, BARCH, LIMINF, LIMSUP, ORDPARAM, ELARCH.

BLLAM: Es un booleano que si viene en "verdadero" indica que la llamada es desde otra rutina y que todos los parámetros los recibe de dicha rutina y si es "falso" indica que es comando directo y que los parámetros serán

proporcionados por el usuario vía terminal.

BARCH: booleano que indica, si es "verdadero" que los números de matriz vienen en el archivo elarch: si es falso se trabajará con el rango de números dado por LIMINF- Y LIMSUP.

LIMINF: entero que trae el límite inferior de los números de matriz a ordenar (sólo viene cuando se desea procesar un rango, por lo tanto BARCH es falso).

LIMSUP: entero con el límite superior de los números de matriz a ordenar.

ORDPARAM: entero con el tipo de ordenamiento deseado:

1: autor 2: dependencia 3: clasificación

ELARCH: Archivo con los números de matriz a ordenar; solo viene si BARCH es "verdadero".

El procedimiento revisa si los archivos del banco están presentes y correctos, así como el archivo que se pasa, si éste existe: En caso de no ser así no podrá ejecutar nada; manda un mensaje y termina.

Si todo está correcto; el procedimiento fija sus parámetros, ya sea los que recibió de la llamada ó pidiéndolos al usuario como clave del sorteo.

El usuario puede pedir total o parcial del banco ordenado por autor, dependencia o clasificación; si es parcial indica si desea números dados de matriz o rango. Si se desean números dados deberá teclearlos, número a número; si se desea un rango se da el límite superior e inferior.

Todas estas preguntas tienen filtros que validan su sin-

taxis, básica a fin de prevenir errores fatales a la rutina.

Una vez obtenidos todos los parámetros, la rutina comienza el sorteo: verifica si cada número de matriz proporcionado existe y si es así irá acomodándolo según el orden deseado. - Al final con el archivo de números sorteado procede a "desempacar" cada ficha y a generar su imagen en un back-up de impresión.