

ESCUELA UNIVERSITARIA POLITECNICA DE LAS PALMAS DE GRAN CANARIA

TITULO: VAX PASCAL: Ampliación sobre el Standard.

Aplicación: programa de archivo contable (ARCONT)
de la E.U.P.L.P.

Autor:

Tutor:

Carlos M. Ramírez Casañas

Sebastian Suarez Gil

PROLOGO

La finalidad de este proyecto ha sido la construcción de un programa de gestión de archivos aplicado a la Escuela Universitaria Politécnica de Las Palmas de Gran Canaria, para controlar la gestión contable de las distintas cuentas de dicha escuela.

Para conseguir el objetivo final de este proyecto se ha descrito, en una primera parte, las peculiaridades que tiene el lenguaje de alto nivel VAX PASCAL de Digital.

En la primera parte se hace una descripción de los tipos de datos, expresiones, sección de declaraciones, sentencias, procedimientos, funciones y procesos de entrada/salida propios de esta ampliación del Pascal Standard.

En una segunda parte se realiza un análisis del programa ARCONT, archivo contable de una Escuela Universitaria particularizado para la de Las Palmas, lo cual no es obstáculo para una posible adaptación a cualquier escuela de características similares. La descripción se ha dividido en módulos, que facilitan la comprensión de ARCONT, de modo que este programa sea de fácil manejo y asequible a cualquier usuario.

INDICE GENERAL

PARTE I. VAX PASCAL: Ampliación sobre el
Standard.

PARTE II. Aplicación; programa de archivo
contable (ARCONT) de la E.U.P.L.P.

P A R T E I

INDICE

PARTE I. VAX PASCAL: Ampliación sobre el Standard.

	Página
1. INTRODUCCION.....	1
2. TIPOS DE DATOS.....	2
2.1.1. Tipo 'Integer'.....	4
2.1.2. Tipo 'UNSIGNED'.....	6
2.1.3. Tipos 'Real'.....	6
2.1.4. Tipos 'estructurado'.....	7
3. EXPRESIONES.....	9
3.1. Tipo 'cast'.....	10
4. SECCION DE DECLARACIONES.....	11
4.1. Sección VALUE.....	12
5. SENTENCIAS.....	14
5.1. La sentencia CASE.....	14
6. PROCEDIMIENTOS Y FUNCIONES.....	15
6.1. Rutinas aritméticas del VAX PASCAL.....	18
6.2. Rutinas de cadenas de caracteres.....	21
6.3. Rutinas de asignación dinámica.....	30
6.4. Rutinas de Entrada/Salida.....	31
6.5. Rutinas de parámetros.....	33
6.6. Rutinas complementarias.....	38
7. PROCESOS DE ENTRADA Y SALIDA.....	41
7.1. Organización indexada.....	41
7.2. Acceso directo.....	42

	Página
7.3. Acceso 'Keyed'.....	43
7.4. OPEN.....	43
7.5. CLOSE.....	47
7.6. EXTEND.....	49
7.7. Procedimientos de acceso directo.....	51
7.8. Procedimientos de acceso 'Keyed'.....	54

1. INTRODUCCION.

El lenguaje de alto nivel VAX PASCAL es una versión más completa y extensa del lenguaje PASCAL. Esta versión utilizada por 'Digital' ha sido desarrollada para el uso del sistema operativo VAX/VMS. Incluye todos los elementos standard del lenguaje añadiendo las siguientes extensiones.

- ‡ Tipos de datos UNSIGNED.
- * Tipos de datos reales de doble y cuádruple precisión.
- * Estructura de datos VARYING OF CHAR que pueden aceptar cadenas de caracteres de longitud variable.
- * Operador de exponenciación.
- * Inicialización de variables en una sección VAR.
- * Claúsula OTHERWISE en la sentencia CASE.
- * Especificaciones más extensas y completas de los parámetros.
- * Mayor capacidad de entrada y salida, incluyendo el soporte para organización de ficheros relativo e indexados.
- * Compilación independiente.
- * Atributos que modifican los datos y los nombres de los procedimientos, funciones, programas y módulos.

- Unidades de compilación. VAX PASCAL utiliza el término "compilation unit" para denotar si un programa o un módulo puede ser compilado como una unidad separada. Un módulo VAX PASCAL consiste en una cabecera seguida de una sección de declaración; no puede contener sentencias ejecutables.
- Elementos léxicos. Un programa PASCAL se compone enteramente de elementos léxicos. Los símbolos especiales que se utilizan en VAX PASCAL son: Exponenciación: * * ; Tanto por ciento: % y el tipo de operador 'cast': :: .

Las palabras utilizadas en un programa VAX PASCAL son combinaciones de caracteres alfabéticos y numéricos. Ocasionalmente los signos: \$, dollar; 'underscore' ___; o tanto por ciento, %.

- Palabras reservadas 'nonstandard'.

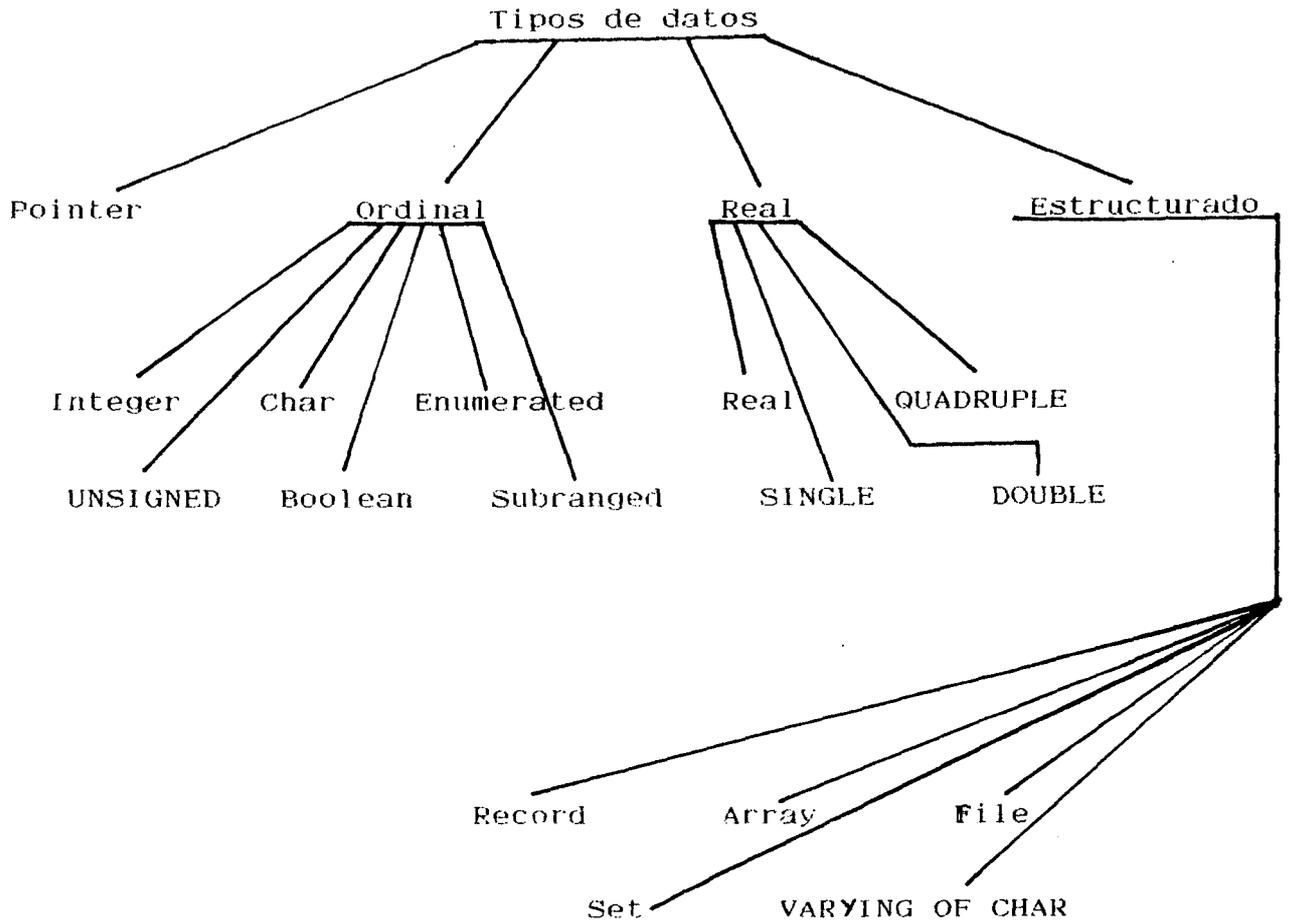
Estas palabras generalmente van precedidas del signo tanto por ciento si no son sentencias o declaraciones de módulos o variables.

% DESCR	MODULE
% DICTIONARY	OTHERWISE

* IMMED	REM
* INCLUDE	VALUE
* REF	VARYING
* STDECSR	

- Identificadores. Son utilizados para nombres de programas, módulos, constantes simbólicas, tipos de datos, variables, procedimientos, funciones y secciones de programas. Un identificador es una combinación de letras, dígitos, signos dollar y 'undercores' (___).

2. TIPOS DE DATOS.



2.1.1. Tipo Integer.

Los tipos de datos Integer son valores enteros positivos y negativos que van desde un rango de $-2^{31} + 1$ hasta $-2^{31} - 1$.

VAX PASCAL permite especificar valores enteros en las notaciones binaria, hexadecimal y octal. Se pueden utilizar estas notaciones en cual-

quier lugar en el que se permiten los decimales, excepto para las etiquetas (labels). Para especificar un valor binario, octal o hexadecimal es necesario colocar el signo de tanto por ciento, %, y una letra entre apóstrofes antes del número. Las letras aproximadas son: B (notación binaria), O (notación octal) y X (notación hexadecimal). Dentro de los apóstrofes se pueden incluir espacios y tabs para hacer más fácil la lectura del valor.

Ejemplos:

Notación binaria del número:

1000 0011 -----%b '1000 0011'

Notación octal del número:

7712 -----%O '7712'

Notación hexadecimal del número:

A1EF-----%X'A1EF'

Desde el valor entero especificado siempre crea una constante no-negativa, el signo del bit no puede ser colocado como el bit más significativo. Para especificar un entero con signo, va precedido por el signo % junto con el operador deseado: menos, -, o más, +. Esto permite señalar una expresión entera constante cuyo valor está entre el mínimo y el máximo que soporta VAX PASCAL. Un ejemplo de la notación del valor mínimo admisible en hexadecimal es:

`-%X '7FFF FFFF'.`

2.1.2. Tipo 'UNSIGNED'.

Los tipos de datos UNSIGNED denotan valores enteros no negativos desde 0 hasta $2^{31} - 1$. El mayor valor posible de los tipos de los UNSIGNED es 4294967265, el cual es dos veces mayor que el valor máximo en el tipo de datos enteros (positivos), MAXINT. Hay que aclarar que UNSIGNED es un tipo máquina-dependiente desarrollado para el uso en programación de sistemas, no para cada aplicación de valores enteros no negativos.

Cuando un programa VAX PASCAL contiene una constante entera mayor que MAXINT, la constante se trata como un dato UNSIGNED. Estos enteros pueden ser escritos en decimal, binario, octal y hexadecimal.

Para especificar que un valor entero ha de ser UNSIGNED, se utiliza la función predeclarada UINT. Un ejemplo de cómo se debe especificar la expresión es:

```
UINT ( %X '7FFF FFFF' );
```

2.1.3. Tipos 'Real'.

Los identificadores REAL, SINGLE, DOUBLE y QUADRUPLE son los tipos real. REAL y SINGLE son si

nónimos; ambos denotan valores reales de precisión simple. El tipo DOBLE es de doble precisión y CUADRUPLE son valores reales de precisión cuádruple.

DOUBLE existe en dos formatos , G-floating y D-floating, los cuales permiten escoger si los valores de doble - precisión expresarán un rango grande (G-floating) o un rango más limitado con mayor precisión (D-floating). No se pueden especificar ambos formatos de DOUBLE en la misma unidad de compilación.

Para indicar un número real de doble precisión, debe utilizar notación exponencial. Reemplace la letra E (de exponenciación) por la letra D (minúscula o mayúscula) para indicar el exponente.

Por ejemplo:

4.371528665 D-3

812d2

Análogamente , la letra Q (mayúscula o minúscula) en notación exponencial designa un valor de cuádruple precisión.

Por ejemplo:

0.11435q3

3362q2

2.1.4. Tipos 'estructurado'.

VAX PASCAL tiene cinco tipos de datos estructurados: record, array, VARYING OF CHAR, set y file.

El tipo de datos VARYING OF CHAR denota una cadena de caracteres. La máxima longitud de la cadena se establece por la definición de este tipo de datos. Al contrario que un paquete de longitud fija de caracteres, una cadena VARYING puede tener valores de diferentes longitudes, desde cero hasta el mínimo número especificado. Tiene la forma:

VARYING (valor entero máximo) OF CHAR

El valor entero máximo tiene un rango de 1 a 65535 caracteres que indica la mayor longitud posible de la cadena.

Este tipo de datos estructurado posee simultáneamente las propiedades del RECORD y el tipo ARRAY. Así, por ejemplo:

VARYING(upper-bound) OF CHAR

sustituye a:

RECORD

Length : (WORD) 0..upper-bound;

body : Packed Array (1..upper-bound) of char;

Ejemplos:

a) VARYING (25) OF CHAR

Para este tipo algunos posibles valores son:

'Wolfgang Amadeus Mozart'

'Bach'

b) ARRAY. (1..5) OF VARYING (20) OF CHAR.

Para este tipo de array tendrá cinco valores de cadenas, como ejemplo:

('Las Palmas, 'Valencia', 'Gandía', 'La Co
ruña', 'Bilbao');

c) RECORD

Título : VARYING (30) OF CHAR;

Autor VARYING (20) OF CHAR;

Categoría : (Ficción, Biografía, In
fantil, No-ficción);

END

Los datos serían:

('Pulgarcito', 'Perrault', 'Infantil')

3. EXPRESIONES.

VAX PASCAL reconoce dos formas de expresio
nes: en tiempo de compilación y en tiempo de ejecu
ción. Una expresión en tiempo de compilación con
siste en operandos de valores determinados cuando
el programa se compila. La expresión más simple es
una constante o un identificador de constante . O
tra expresiones en tiempo de compilación combinan

constantes e identificadores con operadores y las funciones predeclaradas siguientes:

- BITNEXT, BIT-OFFSET, BITSIZE, BYTE-OFFSET, NEXT, SIZE.
- Aritméticas: MAX, MIN, UAND, UNOT, UOR , UXOR, XOR.
- Cadena de caracteres: INDEX, LENGTH, PAD, SUBSTR.
- DBLE, INT, QUAD, SNGL, UINT, UROUND , UTRUNC.
- CARD, EXPO, ZERO.

3.1. Tipo 'Cast'.

Cada variable está asociada con uno y sólo un tipo de datos; el tipo con el que estaba declarado. Algunas veces, sin embargo, debe ser posible realizar una operación más eficientemente si las reglas del PASCAL fueran menos estrictas. VAX PASCAL provee el tipo de operador 'cast', que cambia el contexto en el que puede utilizar una variable o una expresión de un cierto tipo de datos. La representación actual del objeto siendo cast nunca se altera por el operador tipo cast. Tiene la forma:

Identificador-variable :: tipo-identificador

(expresión) :: tipo-identificador

El operador tipo cast (::) separa el nombre de una variable o una expresión entre paréntesis del tipo identificador.

Por ejemplo:

```

Type    F-Float=Packed Record
        Frac 1 : 0..127;
        Expo  : 0..255;
        Sign  : BOOLEAN;
        Frac 2 : 0..65535;
        END;

```

```

Var     A : Real
        :
        :
        :
        A :: F-Float. Expo:=A::
           F-Float. Expo+1;

```

4. SECCION DE DECLARACIONES.

Las dos primeras partes de un bloque PASCAL son la cabecera y la sección de declaraciones. La cabecera especifica el nombre del programa, módulo, procedimiento o función. Cada una de estas declaraciones se introducen con una palabra clave y por el siguiente orden: LABEL, CONST, TYPE, VAR, VALUE, PROCEDURE y FUNCTION. Por ejemplo:

```

Program      Nombre-programa (parámetro 1,
              parámetro 2, .....);

LABEL       declaraciones;

CONST       DECLARACIONES:

TYPE        declaraciones;

VAR         declaraciones;

VALUE       declaraciones;

PROCEDURE   declaraciones;

FUNCTION    declaraciones;

Begin       (*programa principal*)
            :      (*sentencias ejecutables*)
            :
            :
End.        (*fin del programa principal*)

```

4.1. Sección VALUE.

La sección VALUE se utiliza para inicializar variables ordinales, reales y estructuradas (excepto variables ficheros, file). Al contrario que las secciones de declaración, la sección VALUE puede aparecer solamente en la sección de declaraciones del programa principal, es decir, no puede utilizarse en la sección de declaraciones de procedimientos (procedimientos), functions, o módulos.

El formato exacto de una inicialización depende del tipo de variable a inicializar. Tiene la forma:

VALUE

{Identificador de la variable : = value};...

Un ejemplo de inicialización es el siguiente:

VAR

Escuela : RECORD

Clase : RECORD

Cursos : CHAR;

Orden : INTEGER;

END;

PASO : BOOLEAN;

END;

VALUE

Escuela : = (('B',14), TRUE)

En suma, para completar la inicialización, VAX PASCAL también permite hacer una inicialización parcial; es decir, puede asignar valores para especificar elementos de una estructura variable. Esta característica es particularmente útil cuando una estructura variable contiene un fichero variable, cuyas estructuras no pueden ser inicializadas de otra manera. Utilizando el anterior ejemplo, se puede inicializar el record Clase usando constantes:

VALUE

Escuela. Clase. Cursos:= 'A';

Escuela. Clase. Orden:= 1;

o, podría realizarse la misma inicialización usando:

```
VALUE
    Escuela. Clase := ('A', 1);
```

5. SENTENCIAS.

PASCAL está provisto de diversas sentencias que controlan las acciones realizadas por un programa. CASE, IF-THEN (-ELSE), FOR, REPEAT, WHILE, WITH, GOTO, llamada a procedimiento.

5.1. La sentencia CASE.

Esta sentencia ejecuta una de las varias instrucciones dependiendo del valor de la expresión ordinal denominada el 'case selector'. Tiene el formato:

```
CASE    case-selector OF
    {{case-label-list; .....: sentencia};.....
    (( ; ) OTHER WISE
        {sentencia} ;.....)
    ( ; )
END
```

Si el valor del 'case-selector' no aparece en la lista de las etiquetas del 'case', el sistema ejecuta la sentencia de la cláusula OTHERWISE.

Ejemplo:

```

Program OTHERWISE (Input, Output);
VAR I : Integer;
Begin.
    Writeln ('Escriba un número del 1 al 5');
    Read (I);
    CASE I. OF
        1: Writeln ('Este es el número uno');
        2: Writeln ('Este es el número dos');
        3: Writeln ('Este es el número tres');
        4: Writeln ('Este es el número cuatro');
        5: Writeln ('Este es el número cinco');
        OTHERWISE Writeln ('No está dentro del
            rango');
    END;
End.

```

Este es un ejemplo en el que se muestra de forma sencilla el empleo de la cláusula OTHERWISE, que ejecuta la instrucción no seleccionada en la lista o el rango definido anteriormente; cualquier otro valor distinto del especificado en la lista dará la última instrucción.

6. PROCEDURES y FUNCTIONS. (Procedimientos y Funciones.

Cuando se diseña un programa que resuelve un problema complejo, se deben encontrar convenien

temente los subproblemas más sencillos que equivalgan al primero de mayor complejidad. En PASCAL se segmentan los programas de esta manera mediante los procedimientos (procedures) y funciones (functions).

A los procedimientos y funciones, colectivamente los llamaremos rutinas. Un programa VAX PASCAL puede incluir: rutinas escritas y realizadas por el propio programador; rutinas 'external' del VAX/VMS system services, rutinas VAX Run-Time Library, rutinas escritas en otros lenguajes VAX y rutinas predeclaradas.

VAX PASCAL se suministra de procedimientos y funciones predeclaradas que realizan distintas operaciones comunmente útiles.

BITNEXT.- Esta función devuelve un valor entero que indica el número de bits que serían asignados por un componente de tipo X en un paquete de array. Tiene la forma:

BITNEXT (X)

En donde X puede ser de algún tipo.

BITSIZE.- Esta función devuelve un valor entero que indica el número de bits que se asignarían por un campo de tipo X en un paquete record. Tiene la forma:

BITSIZE (X)

En donde el parámetro X puede ser cual-

quier tipo.

BIT-OFFSET.- Esta función devuelve un valor entero que representa la posición del bit de un campo de tipo *f* en un record de tipo *t*. Tiene la forma:

BIT - OFFSET (*t*,*f*)

En donde el parámetro *t* puede ser de algún tipo record, y el parámetro *f* puede ser algún campo contenido en ese record.

BYTE-OFFSET.- Esta función devuelve un valor entero que representa la posición del byte de un campo de tipo *f* en un record de tipo *t*.

BYTE-OFFSET (*t*,*f*)

En donde el parámetro *t* puede ser cualquier tipo record, y el parámetro *f* cualquier campo contenido en ese record.

NEXT.- Esta función devuelve un valor entero que indica el número de bytes que se asignarían por un componente de tipo *X* en un 'unpocked' (no-paquete) array.

NEXT (*X*)

En donde el parámetro *X* puede ser de algún tipo. Si *X* representa un parámetro 'formal'; sin embargo, un 'warning' (aviso) sucede porque el alineamiento del correspondiente actual parámetro no puede ser determinado.

Tanto las funciones NEXT como SIZE devuelven los mismos tamaños de valores del byte para un tipo determinado , excepto cuando los componentes del tipo especificado, es un 'unpacked array'.

SIZE.- Esta función devuelve varios resultados, dependiendo del valor del parámetro X. Tiene la forma:

$$\text{SIZE} (.X, \{c1, c2, \dots, cn\})$$

En donde X puede ser tipo variable o identificador. Si X es un tipo identificador , entonces devuelve un valor que indica el número de bytes que serían asignados por una variable o campo del record de tipo X . Si X es una variable, entonces devuelve un valor entero que indica el número de bytes que se asignan para esa variable. En el caso de que X sea una variante de la variable record o una variante del tipo identificador, SIZE devuelve un valor que indica el número de bytes que serían asignados para ambas partes fijas del record.

Rutinas aritméticas del VAX PASCAL.

MAX.- Tiene la forma:

$$\text{MAX} (X1, X2, \dots, Xn)$$

MAX devuelve el valor máximo de una lista de parámetros X_1, \dots, X_n , en donde los parámetros pueden ser de algún tipo aritmético, pero todos deben ser del mismo tipo.

MIN.- Tiene la forma:

$$\text{MIN} (X_1, X_2, \dots, X_n)$$

MIN devuelve el valor mínimo de la lista de parámetros X_1, \dots, X_n , en donde los parámetros pueden ser de algún tipo aritmético, pero todos deben ser del mismo tipo.

UAND.- Tiene la forma:

$$\text{UAND} (u_1, u_2) \dots$$

UAND realiza una operación lógica binaria **AND** por cada par de bits de los parámetros u_1 y u_2 , los cuales deben ser del tipo **UNSIGNED**.

Por ejemplo:

```
Resultado : = UAND(%X'FF9', %X'703');
```

Este ejemplo devuelve el valor hexadecimal **UNSIGNED %X'701'**.

UNOT.- Tiene la forma:

$$\text{UNOT} (u)$$

UNOT realiza una operación lógica binaria **NOT** por cada bit del parámetro u , en donde u es una expresión del tipo **UNSIGNED**. Devuelve un valor **UNSIGNED**.

Por ejemplo:

Resultado : = UNOT (%X'FF9');

El valor de Resultado es del tipo hexadecimal

UNSIGNED %X'FFFFFF06'.

UOR.- Tiene la forma:

UOR (u1,u2)

Realiza una operación lógica binaria OR con los correspondientes pares de bits de dos parámetros u1 y u2, los cuales deben ser del tipo UNSIGNED. Devuelve un valor UNSIGNED.

Por ejemplo:

Resultado : = UOR(%X'FF9', %X'703');

Este ejemplo devuelve el valor hexadecimal %X'FFB'.

UXOR.- Tiene la forma:

UXOR (u1,u2)

Realiza una operación lógica binaria OR-exclusiva con los pares de bits de dos parámetros u1 y u2, que deben ser del tipo UNSIGNED, así como el valor de la operación.

Por ejemplo:

Resultado : = UXOR(%X'FF9', %X'703');

Este ejemplo devuelve el valor hexadecimal %X'8FA'.

XOR._ Es de la forma:

XOR (p1,p2)

Realiza una operación lógica exclusiva binaria OR-exclusiva de dos conjuntos 'set' (los cuales deben tener el mismo tipo base) o dos valores booleanos. Dependiendo de los parámetros escogidos, XOR devuelve bien si el conjunto de elementos que no aparece en ambos conjuntos, o bien un valor booleano.

Por ejemplo:

```
Resultado : = XOR (('A', 'B', 'C'), ('B', 'C', 'D'));
```

Este ejemplo devuelve el conjunto {'A', 'D'}.

Rutinas de cadena de caracteres del VAX PASCAL:

BIN.- Esta función convierte el valor del parámetro X a un equivalente binario. Tiene la siguiente forma:

```
BIN (X, length, digits)
```

El parámetro X es la expresión a convertir, este parámetro puede ser de cualquier tipo excepto VARYING OF CHAR.

Los parámetros 'length' y 'digits' son opcionales, y representan la longitud de la cadena y el mínimo número de dígitos significativos para ser devueltos.

Por ejemplo:

```
Type
```

```

Días-mes = set of 0..31;

Var

    Días-de-lluvia: Días-mes;
    :
    :
    :

Días-de-lluvia:={1,2,6,10,12,14,18,22,
                25,30};

Resultado : = BIN (Días-de-lluvia,32);

```

En este ejemplo la función BIN convierte el valor de Días-de-lluvia en un equivalente binario y devuelve este valor como una cadena de 32 caracteres. La cadena resultante tiene un 1 en cada posición donde un valor fue asignado a Días-de-lluvia, y un 0 en todas las otras posiciones. Así, el valor de la cadena es '0100001001000101010001000110'. (La representación binaria es de derecha a izquierda, con el bit más a la izquierda representando el elemento 31).

DEC... Esta función convierte el parámetro X a un equivalente decimal. Tiene la forma

```
DEC (X, length, digits)
```

El parámetro X es la expresión que va a ser tratada. Al igual que las funciones BIN, OCT y HEX, ésta puede tratar cualquier tipo de datos excepto VARYING; sin embargo,

al contrario que las anteriores funciones señaladas, requiere que el parámetro X sea de longitud menor o igual a 32 bits.

Por ejemplo:

```

Var
    Cuenta: Integer;

Begin
    Cuenta: = 5;
    Writeln (DEC(Cuenta, 8,7));
        :
        :
        :

End;
```

El valor de la variable entera Cuenta se convierte en un equivalente decimal y saldrá en 8 columnas: siete dígitos y un blanco.

HEX.- (X, length, digits)

Por HEX.- Esta función convierte el parámetro X en su equivalente hexadecimal. Tiene la forma:

```
HEX (X, length, digits)
```

Por ejemplo:

```

Var
    P : A;
    :
    :
    :

    Digits : = 8;

Resultado : = HEX (A, 10, Digits);
```

INDEX.- Esta función localiza el primer carácter de un string partiendo de otro string objeto. Tiene la forma:

INDEX (object, pattern)

Requiere dos expresiones de cadena de caracteres como parámetros: uno objeto o puente (object) y el que se busca en el anterior (pattern).

Esta función devuelve el valor entero que especifica la posición del componente más a la izquierda de la cadena de caracteres pattern, encontrada. La búsqueda finaliza tan pronto como se localice dicho string pattern. Si no se encuentra el valor de INDEX será cero.

Por ejemplo:

```
Object:= 'Erase una vez un lugar';
Pattern:= 'un lugar';
Posición:= INDEX (Object, Pattern);
```

La función INDEX busca el valor, en la cadena Object, de la cadena de caracteres Pattern. El valor entero que toma la variable Posición es 15, que indica la situación del primer carácter del Pattern en Object.

LENGTH.- Esta función determina la longitud de una expresión. Tiene la siguiente forma:

LENGTH (str)

Requiere una cadena de caracteres como parámetro, str. Devuelve el valor entero que especifica la longitud de la cadena.

Por ejemplo:

```
String : = 'Hoy es martes';
```

```
Longitud : = LENGTH (String);
```

En este caso el valor entero de la variable Longitud es 13, indicando el número de caracteres de la expresión.

OCT.- Esta función convierte el valor del parámetro X a un equivalente octal. Tiene la forma:

```
OCT (X, length, digits)
```

X puede ser de cualquier tipo, excepto VAR YING OF CHAR.

Por ejemplo:

```
Entero : = 427;
```

```
resultado : = OCT (Entero, 10,3);
```

El valor de Resultado es una cadena de diez caracteres y tres dígitos significativos, un valor, en este caso, es '653'.

PAD.- Esta función añade un carácter a una cadena de caracteres tantas veces como sea necesario para completar el tamaño especificado de la nueva cadena de caracteres.

Tiene la forma:

```
PAD (str, fill, size)
```

Debe tener tres parámetros: Str, una expresión de caracteres; fill, el carácter con el que se completa la cadena; y, size, el tamaño de la nueva expresión. Este último parámetro, size, debe ser mayor o igual que la longitud de str.

Por ejemplo:

```
Cadena : = 'Ejemplo';
Resultado : = PAD(Cadena,'*',20);
```

En este ejemplo el valor será: 'Ejemplo***
*****', es decir, que la longitud total es veinte.

READV.- Este procedimiento lee caracteres de una expresión, cadena de caracteres, y los asigna a las variables listadas como parámetros del procedimiento READV. Esta función es análoga a READLN. Tiene la forma:

```
READV (str,{identificador [ : especificador ]},.....{,Error : = error recovery})
```

El parámetro especificador es para BIN,OCT y HEX.

Por ejemplo:

```
Type
Color=(Amarillo,Rojo,Azul);
Flor=(Clavel,Tulipán,Rosa);
```

Var

```

Pinta: Color;
Bouquet: Flor;
Mes: VARYING (5) OF CHAR;
Variable: REAL;
Str: VARYING (17) OF CHAR;
:
:
:
:

```

```
Lectura-Str:='Rojo Julio 26.33805';
```

```
READV (Lectura-Str, Pinta, Mes, Variable);
```

Este procedimiento READV lee caracteres de la variable Lectura-Str y los asigna a las variables Pinta, Mes y Variable. Así pues las variables: Pinta contiene Rojo; Mes contiene 'Julio'; y la Variable contiene el valor real 26.33805.

STATUSV.- Esta función devuelve un valor entero que especifica el status del último READV o WRITEV completado. No tiene ningún parámetro.

Por ejemplo:

```

VAR
    Variable:VARYING(20)OF CHAR;
    Resultado: INTEGER;
Begin
    Variable:= '255';
    READV(Variable,Resultado,Error
        := Continue);
    IF STATUSV <>0

```

```

then
    writeln('Error in READV');
End;

```

SUBSTR.- Esta función escoge una 'sub-cadena' de otra cadena de caracteres. Tiene el formato:

```
SUBSTR (str, start, length)
```

Requiere tres parámetros: una cadena de caracteres, una expresión entera que indica la posición de comienzo de la 'sub-cadena'. Devuelve un string de la longitud especificada. Los valores de la posición de comienzo y la longitud deben ser mayor que cero. Por ejemplo:

```

Original: = 'Esta es la cadena original';
Comienzo: = 12;
Longitud: = 15;
Nueva-Cadena: = SUBSTR (Original,Comienzo, Longitud);

```

En este ejemplo la Nueva-Cadena tendrá el valor 'cadena original'.

UDEC.- Esta función convierte el parámetro X a un equivalente decimal 'unsigned'. Tiene el formato:

```
UDEC (X, length, digits)
```

Como las funciones BIN, OCT y HEX, UDEN puede tomar cualquier tipo exceptuando VAR-

YING. Requiere, el parámetro X, una longitud menor o igual a 32 bits.

Por ejemplo:

```

Var
    Cuenta: Integer;

Begin
    Cuenta: = 3;

    Writeln (UDEC (Cuenta));
    :
    :
    :

End;
```

WRITEV.- Este procedimiento escribe caracteres en una variable:cadena de caracteres del tipo VARYING OF CHAR. Es análogo a WRITELN. Tiene la forma:

```

WRITEV (str, parameter-hist, error:
        = error-recovery);
```

Por ejemplo:

```

Type
    Color=(Amarillo, Rojo, Azul);
    Flor=(Clavel,Rosa, Amapola,
          Tulipán);

Var
    Bouquet:Flor:=Rosa;
    Mes:VARYING (9)OF CHAR;
    Variable:REAL;
    Escritura:VARYING (30)OF CHAR;
    :
    :
    :
```

```
Variable:=232.705;
```

```
WRITEV(Escritura, Amarillo , Variable:8:3, PRED.(Bouquet));
```

La variable escritura quedará con el valor:

```
'Amarillo 232.705 Tulipán'
```

Rutinas de asignación dinámica del VAX PASCAL:

ADDRESS.- Esta función devuelve un valor puntero(pointer) que se relaciona con el parámetro X. Tiene la forma:

```
ADDRESS (X)
```

El parámetro X no puede ser del tipo 'packed structured' (paquete estructurado). El compilador VAX PASCAL asume que todos los punteros hacen referencia a variables dinámicas asignadas por el procedimiento NEW o para variables que tengan el atributo VOLATILE.

IADDRESS.- Esta función es similar a la función ADDRESS, excepto en que ésta devuelve un valor entero que hace referencia al parámetro X. Tiene la forma:

```
IADDRESS (X)
```

El parámetro X no puede ser del tipo 'packed structured'. El compilador VAX PASCAL asume automáticamente que todos los punteros se refieren a variables dinámicas asig

nadas al procedimiento NEW.

Por ejemplo:

```

Var
    Dirección: INTEGER;
    Variable: REAL;
Begin
    Dirección: = IADDRESS (Variable);
    WRITELN('La dirección de la variable es', Dirección);
End;
```

Este ejemplo produce que IADDRESS devuelva un valor entero que representa la dirección de la variable.

Rutinas de Entrada/Salida del VAX PASCAL:

ADD-INTERLOCKED.- Esta función añade el valor de una expresión al valor de una variable, utilizando la instrucción 'VAX Add Aliqued Word Interlocked' (ADAWI), y almacena el valor procesado en la variable. Tiene la forma:

```
ADD-INTERLOCKED (e, v)
```

El tipo de expresión, e, debe ser compatible con la variable v. La función devuelve el valor entero -1 si el nuevo valor de v es negativo, 0 si es cero, y +1 si es positivo.

CLEAR-INTERLOCKED.- Esta función asigna al valor FALSE a b y devuelve el valor original de b, usando la instrucción 'VAX Branch on Bit Clear and Clear Interlocked (BBCCI). Tiene la forma:

CLEAR-INTERLOCKED (b)

El parámetro b debe ser una variable de tipo BOOLEAN.

FIND-FIRST-BIT-CLEAR.- Esta función localiza el primer bit de un PACKED ARRAY OF BOOLEAN cuyo valor sea 0. Tiene la forma:

FIND-FIRST-BIT-CLEAR(vector, start-index)

El resultado de esta función es un indexamiento INTEGER del primer elemento que contiene el valor 1.

FIND-MEMBER.- Esta función localiza el primer caracter en un string que es miembro de un conjunto especificado. Tiene la forma:

FIND-MEMBER (string, charset)

El resultado es una indicación INTEGER de la posición en el string del primer caracter que es miembro de un conjunto de caracteres. El valor 1 indica que es el primer caracter en el string; el valor 0 indica que ninguno de los caracteres del string son miembros del conjunto.

FIND-NONMEMBER.- Esta función localiza el primer caracter en un string que no es miembro de un conjunto especificado. Tiene la forma:

FIND-NONMEMBER (string, charset)

El resultado es una indicación INTEGER de la posición en el string del primer caracter que no es miembro de un conjunto de caracteres. El valor 1 indica el primer caracter en el string; el caracter 0 indica que todos los caracteres del string son miembros del conjunto.

SET-INTERLOCKED.- Esta función asigna el valor TRUE a b y devuelve el valor original de b, utilizando la instrucción ' VAX Branch on Bit Set and Set Interlocked (BBSI). Tiene la forma:

SET-INTERLOCKED (b)

El parámetro b debe ser una variable de tipo BOOLEAN.

Rutinas de parámetros del VAX PASCAL:

ARGUMENT.- Esta función especifica un argumento que corresponde a un parámetro de una función o un procedimiento con el atributo LIST. Tiene la forma:

ARGUMENT (nombre-parámetro, n)

El argumento :nombre-parámetro, especifica el nombre de un parámetro declarado con el atributo LIST. El parámetro n especifica un valor entero positivo que identifica un argumento. El primer argumento de la lista siempre es 1.

ARGUMENT-LIST-LENGTH.- Esta función devuelve un valor entero representando el número de parámetros correspondientes a otro similar con el atributo LIST. Tiene la forma:

ARGUMENT-LIST-LENGTH(nombre-parámetro)

Nombre-parámetro especifica el nombre del parámetro declarado con el atributo LIST.

Por ejemplo:

```

Program Muestra (OUTPUT);
Procedure Mues (F1 : VARYING (LEN)
OF CHAR; Ax : (LIST) CHAR);
Var
    I : Integer;
Begin
    WRITE (F1, ', ');
    FOR I := 1 TO ARGUMENT - LIST -
    LENGTH (Ax) DO WRITE (ARGUMENT
    (Ax, I));
    WRITELN;
End;
Begin
Mues ('Ho', '!');
```

```
Mues ('Hola', 'c', 'h', 'a', 'c', 'A',
      'l');
```

```
End.
```

En este ejemplo, la primera llamada al procedimiento Mues asigna el string 'Ho' al parámetro F1, y asigna el string '!' al parámetro Ax. El segundo llamamiento a Mues asigna el string 'hello' a F1, y 'c', 'h', 'a', 'c', 'a', 'l' a Ax.

PRESENT.- Esta función indica si la muestra de argumentos de una rutina contiene uno de ellos que corresponda a un parámetro 'formal'. Tiene la forma:

```
PRESENT (nombre-parámetro)
```

Por ejemplo:

```
Program Trunc (OUTPUT);
```

```
Var
```

```
    A, B: CHAR;
```

```
    C : INTEGER;
```

```
Procedure Valores (A:CHAR;
```

```
                  B:(TRUNCATE) CHAR;
```

```
                  C:(TRUNCATE) INTE-
```

```
                  GER;);
```

```
Begin
```

```
    IF PRESENT(A) THEN WRITE('1-');
```

```
    IF PRESENT(B) THEN WRITE('2-');
```

```
    IF PRESENT(C) THEN WRITE('3!');
```

```
    WRITELN;
```

```

End;

Begin
Valores (A);
Valores (A, B);
Valores (A, B, C);

End.

```

En este ejemplo, la primera llamada a `Valores` causa la impresión: '1-' . La segunda llamada a `Valores` causa: '1-2-!' . La tercera causa: '1-2-3!' .

`DBLE.`- Esta función convierte el valor del parámetro `X` a su equivalente de doble precisión. Tiene la forma:

```
DBLE (X)
```

Devuelve un valor tipo `DOUBLE`. El parámetro `X` debe ser de tipo aritmético.

`INT.` Esta función convierte el valor de `X` a su equivalente entero. Tiene la forma:

```
INT (X)
```

El parámetro `X` debe ser de tipo ordinal.

`QUAD.`- Esta función convierte el valor de `X` a un equivalente de cuádruple precisión. Tiene la forma:

```
QUAD (X)
```

Devuelve un valor tipo `QUADRUPLE`. El parámetro `X` debe ser tipo aritmético.

SNGL.- Esta función redondea el valor de X a un equivalente de precisión simple. Tiene la forma:

SNGL (X)

Devuelve un valor tipo SINGLE. El parámetro X debe ser de tipo aritmético.

UINT.- Esta función convierte el valor de X a su equivalente como entero 'unsigned'. Tiene la forma:

UINT (X)

El parámetro X debe ser del tipo ordinal . Devuelve un valor de tipo UNSIGNED.

UROUND.- Esta función convierte el valor de X a su equivalente como un entero 'unsigned' para redondear la parte fraccionaria del valor. Tiene la forma:

UROUND (X)

El parámetro X debe ser tipo REAL, SINGLE, DOUBLE o QUADRUPLE. El valor devuelto es de tipo UNSIGNED.

UTRUNC.- Esta función convierte el valor de X a un equivalente como un entero 'unsigned' para truncar la parte fraccionaria de el valor. Tiene la forma:

UTRUNC (X)

El parámetro X debe ser del tipo REAL, SINGLE, DOUBLE o QUADRUPLE. El valor devuelto

es de tipo UNSIGNED.

Rutinas complementarias del VAX PASCAL:

CARD.- Esta función devuelve un valor entero indicando el número de componentes que son elementos del conjunto-expresión s.

CARD (s)

CLOCK.- Esta función devuelve un valor entero indicando la cantidad de tiempo del procesador central, en milisegundos, utilizados para el proceso en curso. Esta función no tiene parámetros.

CREATE-DIRECTORY.- Este procedimiento crea un nuevo directorio o subdirectorío VMS.

Tiene la forma:

CREATE-DIRECTORY (nombre-fichero, error-return)

El Nombre del fichero debe ser un nombre de dispositivo. El parámetro error-return es opcional, y devolverá un código de error recuperable VMS.

DATE and TIME.- Estos procedimientos asignan la fecha en curso y la hora a una variable string. Tiene la forma:

DATE (str)

TIME (str)

Cada parámetro requiere un string de tipo
PACKED ARRAY (1..11) OF CHAR.

Por ejemplo:

Var

```

    Fecha, Hora : PACKED ARRAY (1..
    11) OF CHAR; 11) OF CHAR;
    :
    :
    :
```

DATE (Fecha);

TIME (Hora);

Estas dos llamadas darán como resultado el
siguiente formato-ejemplo:

12-Feb-1.987

14:20:25,98

DELETE-FILE.- Este procedimiento borra uno
o más ficheros VMS. Tiene la forma:

```

DELETE-FILE (nombre-fichero, error-
return)
```

La especificación del primer fichero debe
tener un nombre explícito del directorio.

ESTABLISH.- Este procedimiento establece u
na condición titular VAX que procesa erro-
res y presenta el estado de excepciones y
condiciones. Tiene la forma:

```

ESTABLISH (identificador-función)
```

El parámetro a establecer debe ser el nom-

bre de una rutina que tiene el atributo ASYNCHRONOUS.

EXPO.- Esta función devuelve el exponente entero de la representación del parámetro X. Tiene la forma:

EXPO (X)

Cuando el parámetro X es del tipo REAL, SINGLE o D-floating DOUBLE, el exponente es un valor entero entre -128 y +127. Cuando es del tipo G-floating DOUBLE, oscila entre -1024 y +1023. Cuando es del tipo QUADRUPLE, el valor va desde -16384 hasta +16383.

HALT.- Este procedimiento llama al análogo LIB&STOF de la VAX Run-Time Library, con el valor-condición PAS&HALT. Sin una condición apropiada, HALT termina la ejecución del programa. Este procedimiento no tiene parámetros.

RENAME-FILE.- Este procedimiento cambia el nombre de un fichero VMS. Tiene la forma:

RENAME-File (nombre-antiguo; nombre
-nuevo, error-return)

REVERT.-Este procedimiento cancela una condición activada por el procedimiento ESTABLISH. No tiene parámetros.

UNDEFINED.- Esta función examina si X con-

tiene un operando reservado. Tiene la forma:

UNDEFINED (X)

El parámetro X debe ser una variable de tipo REAL, SINGLE, DOUBLE o QUADRUPLE. Devuelve el valor TRUE si X contiene un valor que ha sido reservado VAX/VMS. Será FALSE si X no contiene valor reservado.

ZERO.- Esta función se utiliza en asignamiento de sentencias y para inicializar una variable (excepto tipo fichero) al valor binario cero. No tiene parámetros.

7. PROCESOS DE ENTRADA Y SALIDA.

VAX PASCAL incluye un extenso conjunto de rutinas predeclaradas que gobiernan los procesos de entrada/salida (input/output, I/O). Estas rutinas establecen ficheros con organización secuencial, relativa o INDEXADA, y los procesa según acceso secuencial, DIRECTO o KEYED.

7.1. Organización indexada.

Los componentes de un fichero indexado se ordenan en ciertos campos de datos, llamados cla-

ves, que están contenidos en cada componente.

Cuando se diseña un fichero indexado, se deciden cuales son los campos que van a ser claves; los contenidos de estos campos serán usados para identificar sus componentes en operaciones subsiguientes. La longitud de un campo clave y su posición relativa es idéntica para todos los componentes de el fichero.

Cuando se crea un fichero indexado, debe definirse al menos una clave para el fichero, utilizando el atributo KEY; ésta se denomina clave primaria del fichero. También se pueden definir hasta 254 claves más, denominadas claves alternas. Este último tipo es un campo que tiene la misma longitud e igual posición en cada componente del fichero.

7.2. Acceso directo.

Significa que los componentes se procesan en un orden específico, mediante los procedimientos FIND y LOCATE.

FIND posiciona un fichero de acceso directo para aceptar la entrada; LOCATE posiciona el fichero para escribir la salida. Un fichero con organización secuencial debe tener records RMS de longitud fija para poder acceder mediante el método directo.

7.3. Acceso 'Keyed'.

Significa que los componentes se procesan en un orden determinado por el valor de un campo clave (Key). Utiliza el procedimiento FINDK para indicar el valor clave del componente a procesar ; posiciona el fichero en el componente que corresponde al valor clave para especificarse como un parámetro.

7.4. OPEN.

El procedimiento OPEN abre un fichero, define el método de acceso, y permite especificar los parámetros del fichero. El término "record", en los nombres de los parámetros de este procedimiento, indica un registro RMS. Tiene la forma:

```
OPEN (file-variable
      (, file-name)
      (, file-history)
      (, record-length)
      (, access-record)
      (, record-type)
      (, carriage-control)
      (, organization)
      (, disposition)
      (, file-sharing)
      (, user-action)
```

{, default-file-name)

{, ERROR : = error-recovery))

File-variable: Nombre de la variable asociada al fichero para ser abierto.

File-name: String de caracteres que representan el nombre del fichero representado por una variable fichero en un procedimiento OPEN. Si se omite toma el nombre de la variable-fichero.

History: Indica si el fichero especificado existe o debe ser creado. NEW indica que un fichero nuevo será creado en la apertura . OLD indica un fichero existente para que sea abierto. READONLY indica que un fichero existente será abierto sólo para lectura UNKNOWN indica que si no existe un fichero antiguo, será creado uno nuevo.

History : = OLD o NEW o READONLY o
UNKNOWN

Record-length: el valor del parámetro de la longitud del record es un entero positivo que especifica el máximo tamaño en bytes de una línea en un fichero texto o un fichero de tipo FILE OF VARYING (por defecto será 133 bytes). Para otros ficheros el parámetro se ignora.

Access-Method : = SEQUENTIAL o DIRECT o KE
YED. Son tres tipos: secuencial, directo o
'keyed' (por claves).

Record-type: especifica el tipo de record,
si es fijo, variable, stream , stream con
carry return o stream con line-feed.

Record-type : FIXED o VARIABLE o S-
TREAM O STREAM-CR o
STREAM-LF

El formato stream es un formato del record
por el cual los registros de un fichero se
delimitan por caracteres especiales o secu-
encias de caracteres denominados 'termina-
tors'.

Carriage-Control: especifica el formato de
el control del carro para el fichero. Pue-
de ser: LIST(para ficheros textos y del ti-
po FILE OF VARYING), CARRIAGE, FORTRAN (es-
tos dos últimos parámetros mediante carac-
teres de control de carro), NO CARRIAGE,NO
NE (no especifican los controles).

Organization: Especifica la organización
física del fichero; secuencial, relativa o
indexada. SEQUENTIAL, RELATIVE, INDEXED.

Disposition: Describe lo que se hace con

el fichero cuando se cierra. SAVE: guardar lo. DELETE el fichero se borra. PRINT se imprime y no se borra. PRINT-DELETE se borra el fichero después de imprimirse. SUBMIT el fichero se somete al lote de trabajo y no se borra. SUBMIT-DELETE igual que el anterior exceptuando que se borra el fichero . DELETE el fichero automáticamente se borra al cerrarse éste.

Sharing: Indica si sus programas pueden acceder al fichero mientras esté abierto. READONLY indica que el acceso puede ser sólo para lectura. READWRITE indica que otros programas pueden escribirse y leerse. NONE rechaza el acceso a otros programas.

Default-file-name: Permite a los usuarios el acceso a los parámetros 'RMS Default-file-name'.

Error-Recovery: Especifica la acción tomada por un programa si el procedimiento OPEN falla en la ejecución. Si se especifica CONTINUE el programa continúa a pesar de las condiciones erróneas. Si se especifica MESSAGE se genera un mensaje de error y la ejecución del programa cesa.

Ejemplos:

1. Programa Principal (Guía);

Var

```

Guía: TEXT;
:
:
:
OPEN (Guía);

```

2. OPEN (Albunes,

```

'SYS$DISK:[CARLOS INVENTO',
Acces-method:=DIRECT,
history:=OLD);

```

Este ejemplo abre el fichero VAX/VMS e xistente: [CARLOS INVENTO DAT para el acceso directo. Este fichero es reconocido en el programa PASCAL como la variable Albunes.

3. OPEN (Solar, 'Energía', NEW,
FIXED);

4. OPEN (Jounnal, UNKNOWN , KE-
YED, INDEXED);

7.5. CLOSE.

Este procedimiento cierra un fichero a bierito, y tiene la forma:

```

CLOSE (file-variable
      (, disposition)
      (, user-action)
      (, ERROR:= error-recovery))

```

File-variable: es el nombre de la variable asociada al fichero por cerrar.

Disposition: describe lo que se va a hacer con el fichero cuando se cierre. Los valores de los parámetros han sido descritos en el procedimiento OPEN.

User-action: este parámetro efectúa una llamada a la 'Run Time Library' para el cierre del fichero.

Error-recovery: parámetro que especifica la acción que tomaría el programa en el caso de fallo en el procedimiento CLOSE. Los parámetros han sido descritos en el procedimiento OPEN.

Ejemplos:

1. CLOSE (Albunes);
2. CLOSE (Productos,
DISPOSITION:= PRINT-DE
LETE);

Este ejemplo cierra el fichero Productos, lo vuelca en la impresora y más tarde lo borra.

3. CLOSE (Zapatos,
USER-ACTION:=Close-fi-
le);

Este procedimiento llama a función Close-File, que debe realizar el \$CLOSE para otras operaciones.

La función devuelve un valor para indicar si el fichero Zapatos ha sido suficientemente cerrado.

7.6. EXTEND.

El procedimiento EXTEND abre un fichero existente y lo prepara para su escritura. Se utiliza comunmente para añadir datos al fichero. Tiene la forma:

```
EXTEND (file-variable [,ERROR:=
      error-recovery])
```

file-variable: nombre de la variable del fichero asociado al fichero de salida.

error-recovery: es el valor del parámetro que indica la acción tomada en el caso de que ocurra un error.

Después de la ejecución de EXTEND, el fichero se posiciona a continuación del último componente, y EOF (end-of-file, fin del fichero) toma el valor FALSE. Si el fichero no existe se produce un error en el tiempo de ejecución.

Una llamada a EXTEND en un fichero relativo abierto por acceso directo lo posi-

ciona después de su último componente existente.

Una llamada a EXTEND en un fichero indexado abierto por acceso 'Keyed' lo posiciona después del último componente relativo a la clave primaria (primary Key).

Ejemplos:

1. EXTEND (Datos);

Esta sentencia abrirá un fichero existente denominado DATOS. DAT. El fichero será posicionado al final de este y lo prepara para escribir en él.

2. Var

```

      F: File of Integer;
      :
      :
      :

      OPEN (F, 'sample Dat', OLD, Relativa, Direct);

      EXTEND (F);

      F:= 20;

      PUT(F);
      :
      :
      :
```

Estas sentencias abren un fichero relativo existente llamado 'SAMPLE DAT'. El fichero será posicionado después del último record en el fichero. A continuación PUT añadirá nuevos registros al final del fichero.

7.7. Procedimientos de Acceso directo.

Los siguientes procedimientos son, generalmente, sólo para ficheros abiertos mediante el acceso directo. En algunos casos, como se indica , para acceso 'keyed' también.

- * DELETE (también para 'keyed access')
- * FIND
- * LOCATE
- * UP DATE (también para 'keyed access')

DELETE.- Este procedimiento borra el componente actual del fichero. Tiene la forma:

```
DELETE (file-variable(, ERROR:=error
                    -recovery))
```

file-variable: nombre de la variable-fichero asociada a éste, desde el que un componente será borrado.

error-recovery: indica la acción tomada en caso de error.

DELETE:solamente puede ser usado en ficheros con organización relativa o indexada que han sido abiertos por acceso directo o keyed. No puede ser utilizado en ficheros con organización secuencial.

Por ejemplo:

```
DELETE (Cuentas);
```

Borra el componente actual del fichero Cuentas. Antes de esta instrucción debe haber o

rrido un procedimiento FIND, FINDK, GET, RESET o RESETK.

FIND.- Este procedimiento posiciona un fichero en un componente específico. Tiene la forma:

```
FIND (file-variable, componente-member (, ERROR:= error-recovery))
```

file variable: nombre de una variable-fichero asociada a éste, que se abre por acceso directo.

componente number: expresión entera positiva que indica el componente al fichero que se va a posicionar. Si el componente es cero o negativo, causa un error en tiempo de ejecución.

error-recovery: indica la acción que se toma si ocurre algún error en ejecución.

El procedimiento FIND permite acceder directamente a los componentes del fichero. Puede usarse para moverse hacia adelante o hacia atrás en un fichero.

Después de la ejecución de este procedimiento, el fichero se posiciona en el componente especificado. La variable buffer asume el valor del componente, y el modo fichero se pone en Inspection.

Ejemplos:

FIND (Albunes, Actual+2);

Si el valor de Actual es 6, se produce que la posición se mueva al octavo componente ; la variable buffer Albunes asume el valor del componente.

FIND (Albunes, Actual-1)

Si el valor de Actual es 6, se moverá la posición al quinto componente. Albunes , variable buffer, toma el valor del quinto componente.

LOCATE.- Este procedimiento posiciona un fichero acceso-directo a un componente particular, de modo que el próximo procedimiento PUT modifica ese componente. Tiene la forma

LOCATE (file-variable, component-number [, Error-recovery])

component-number: expresión entera positiva que indica el número del componente por encontrar.

LOCATE posiciona el fichero de modo que el próximo PUT escribirá el contenido de la variable buffer dentro del componente seleccionado.

Ejemplo:

LOCATE (Cuentas, 63);

UPDATE.- Este procedimiento escribe los contenidos de la variable buffer dentro del

componente actual. Tiene la forma:

```
UPDATE (file-variable [, error-recovery])
```

El procedimiento es válido para organizaciones relativa e indexada abiertas mediante métodos directo o 'Keyed'.

Ejemplo:

```
UPDATE (Ventas);
```

7.8. Procedimientos de acceso 'Keyed'.

Son los siguientes procedimientos:

- * FINDK
- * RESETK

FINDK.- Este procedimiento busca el índice de un fichero indexado abierto según el acceso directo, y localiza un componente determinado. Tiene la forma:

```
FINDK (file-variable, Key-number, Key-value [,match-type][,error-recovery])
```

Key-number: expresión entera positiva que indica la posición de la clave (Key).

Key-value: indica la clave que va a ser encontrada, debe ser compatible el campo clave (Key) en la posición clave especificada.

match-type: identificador que indica la relación entre el valor clave en FINDK y el

valor de un componente.

Los match-type deben ser uno de los identificadores EQL, GTR o GEQ para indicar que la clave le encontró; tiene un valor igual o mayor que el valor del procedimiento FINDK.

Ejemplo:

```
FINDK (Indice, 1, 35, GEQ);
```

Este ejemplo busca el índice para la clave número 1 en el fichero Índice hasta que encuentra el primer componente cuyo valor de clave es mayor o igual que 35. Índice debe haber sido abierto mediante el método 'Keyed access'.

RESETK.- Este procedimiento es análogo al procedimiento RESET, prepara el fichero para su lectura. Tiene la siguiente forma:

```
RESETK (file-variable, Key-number (,
        error-recovery))
```

Key-number: es una expresión no - negativa que indica la posición de la clave.

RESETK sólo puede ser aplicado a ficheros indexados abiertos por 'keyed access'. Se asigna un número de clave de 0 a 254 para cada campo de un componente del fichero, con el atributo KEY. Busca el componente con más bajo valor de clave en el número 'Key' especificado. Así este componente se hace el actual, su valor se copia en el buffer del fi

chero.

Ejemplo;

RESETK (Libroíndice, Ø);

Busca el fichero Libroíndice por el componente con más bajo valor de clave primaria (primary Key).

P A R T E I I

INDICE

PARTE II. Aplicación: programa de archivo contable (ARCONT) de la E.U.P.L.P.

	Página
1. ANALISIS DEL PROGRAMA "ARCONT".....	1
1.1. Características generales.....	1
1.2. MODULO I. Programa de control.....	3
1.3. MODULO INICIAL. Procedimiento del Menú Principal.....	5
1.4. MODULO CONSULTA.....	6
1.5. MODULO AYUDA, Juego de instrucciones del programa.....	9
1.6. MODULO ACTIV. Menú de las cuentas de actividades docentes.....	9
1.7. MODULO CARRERAS. Menú de las Cuentas de las Carreras de la E.U.P.....	12
1.8. MODULO TELECO.....	14
1.9. MODULO E.U.P.....	16
1.10. MODULO CUENTAGEN.....	21
1.11. MODULO OPCUENTAS.....	22
1.12. MODULO INTRODATOS.....	23
1.13. MODULO ENCA.....	23
1.14. MODULO CENIN.....	24
1.15. MODULO OPCION.....	25

	Página
1.16. MODULO CENTRADO.....	25
1.17. MODULOS BORRADO,CURSOR e INVERSO...	26
2. Ficheros de datos	
correspondientes a las cuentas.....	29
2.1. Fichero de datos de	
instrucciones del programa.....	30
APENDICE A. CONTROL DE PANTALLA.....	34
A.1. Movimiento del cursor.....	34
A.2. Conjunto de caracteres.....	36
A.3. Características específicas	
de visualización.....	37
A.4. Borrado de pantalla.....	38
A.5. Creación de una	
región 'scrolling'.....	39
A.6. Características del terminal...	40
3. LISTADO DEL PROGRAMA "ARCONT".....	41

1. ANALISIS DEL PROGRAMA "ARCONT".

En esta sección se estudia el funcionamiento del programa ARCONT, cuya estructuración en procedimientos independientes permite un cómodo análisis de cada una de sus funciones.

1.1. Características generales.

El programa arcont está concebido para gestionar el archivo contable de una escuela universitaria, y aplicado concretamente a la Escuela Universitaria Politécnica de Las Palmas de Gran Canaria.

Se compone del módulo principal y de diferentes módulos o procedimientos que se seleccionan desde el primero.

Para entender la estructuración de esta tarea, se hace necesario explicar en qué se basa el trabajo de la gestión administrativo-económica de la Secretaría de la E.U.P. que a continuación se detalla.

Los presupuestos anuales asignados a la E.U.P. están divididos en los siguientes apartados: Actividades Docentes, Material inventariable, Cultura general, Educación física y deportes, Extensión universitaria, Transportes, Mantenimiento, Reuniones y conferencias, Mobiliario y equipo inventariable, Actos de carácter social, Dietas y Gastos de oficina. De todos estos apartados reseñados, el más importante, a la hora de su gestión, es el de

Actividades docentes, debido a la distribución del presupuesto asignado a éste, y al valor en cuanto a cantidad del mismo.

Actividades docentes se divide en siete secciones principales, cinco de las cuales corresponden a cada carrera (Telecomunicaciones , Industriales, Topografía, Obras Públicas y Navales) y dos divisiones más correspondientes, la primera, a "Asignaturas comunes" y, la segunda, a " Sin cargo a cátedra".

Las especificaciones del reparto de las cantidades asignadas a cada carrera son las divisiones que existen en cada una de ellas por: cátedras, especialidades o en su conjunto, según sean las carreras de Telecomunicaciones, Industriales , Navales, Topografía y Obras Públicas respectivamente.

1.2. MODULO I. Programa de control.

Este módulo primero selecciona el paso a la ejecución directa de lo que es propiamente AR-CONT, o antes seleccionar las instrucciones a seguir por el usuario para una utilización de éste sin impedimentos, y de la forma más adecuada y fácil.

La selección de la letra 'H' o 'h' siempre dará en cualquier momento paso al módulo de instrucciones del programa, que es el de 'AYUDA' al usuario.

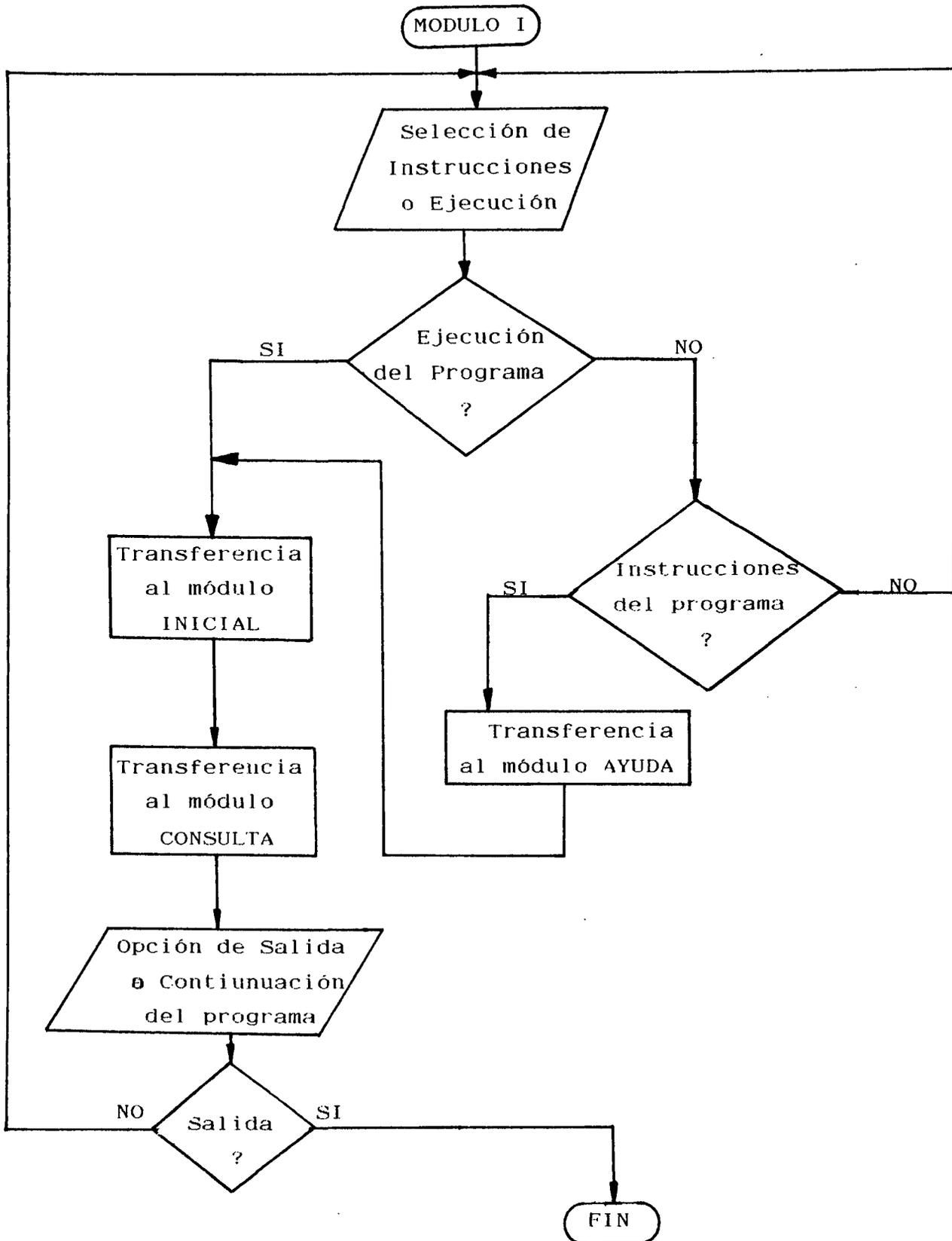
Una vez se haya seleccionado el módulo de instrucciones o, lo que es igual, el procedimiento 'AYUDA', dando el control al módulo inicial en donde se empiezan a escoger las opciones de las distintas cuentas o menús de cuentas para su posterior consulta o escritura.

En el caso de que se pretenda hacer una selección de otra clave que no sean las expuestas, 'S' o 'H', el proceso está a la espera de que una de estas entradas ocurra.

Una vez se haya hecho la elección de ejecución del módulo inicial y una vez acabado éste, transferirá el control al módulo de consulta que dará la opción de finalización del programa.

Llamadas a los procedimientos: INICIAL, BQRRADO, INVERSO, OPCION, CONSULTA, AYUDA.

-Organigrama del Programa de Control.



1.3. MODULO INICIAL. Procedimiento del Menú Principal.

Este módulo inicial tiene la misión de exponer las diferentes opciones que tiene ARCONT y seleccionar el módulo que corresponde con la opción tomada y transferir el control a este último. En el caso de que se tome la salida finaliza la ejecución del programa. Una vez realizada la función del procedimiento seleccionado pasaría el control al módulo INICIAL siempre que no se haya optado por tomar la 'SALIDA' del programa, es decir, el final de la ejecución.

En el bloque I, Selección de la Opción, hay que tener en cuenta que las opciones son : CUENTA GENERAL DE LA E.U.P., Material Inventariable, Cultura General, Educación Física y Deportes , Extensión Universitaria, Transportes, Mantenimiento, Reuniones y Conferencias, Mobiliario y equipo inventariable, Dietas, Gastos de Oficina y Actividades Docentes, que ramifica en cuatro módulos, de los cuales dos de ellos se subdividen en otros tantos módulos correspondientes a las carreras y asignaturas comunes.

Este módulo se encarga, en definitiva, únicamente de transferir el control a los otros procedimientos de cuentas o menús de cuentas del programa. Las variables que se utilizan aquí son de tipo char, caracter, y van del '1' al '9' y de la

'A' hasta la 'D', y también el caracter 'H' para acceder a las instrucciones del programa (MODULO IV.AYUDA).

Llamadas a procedimientos: EUP, MATER, CULTUR, EDUCA, EXTEN, TRANS, MANTE, REUNION, MOBILIA, ACTOS, DIETAS, GASTOS, AYUDA, BORRADO, CENIN , OPCION.

1.4. MODULO CONSULTA.

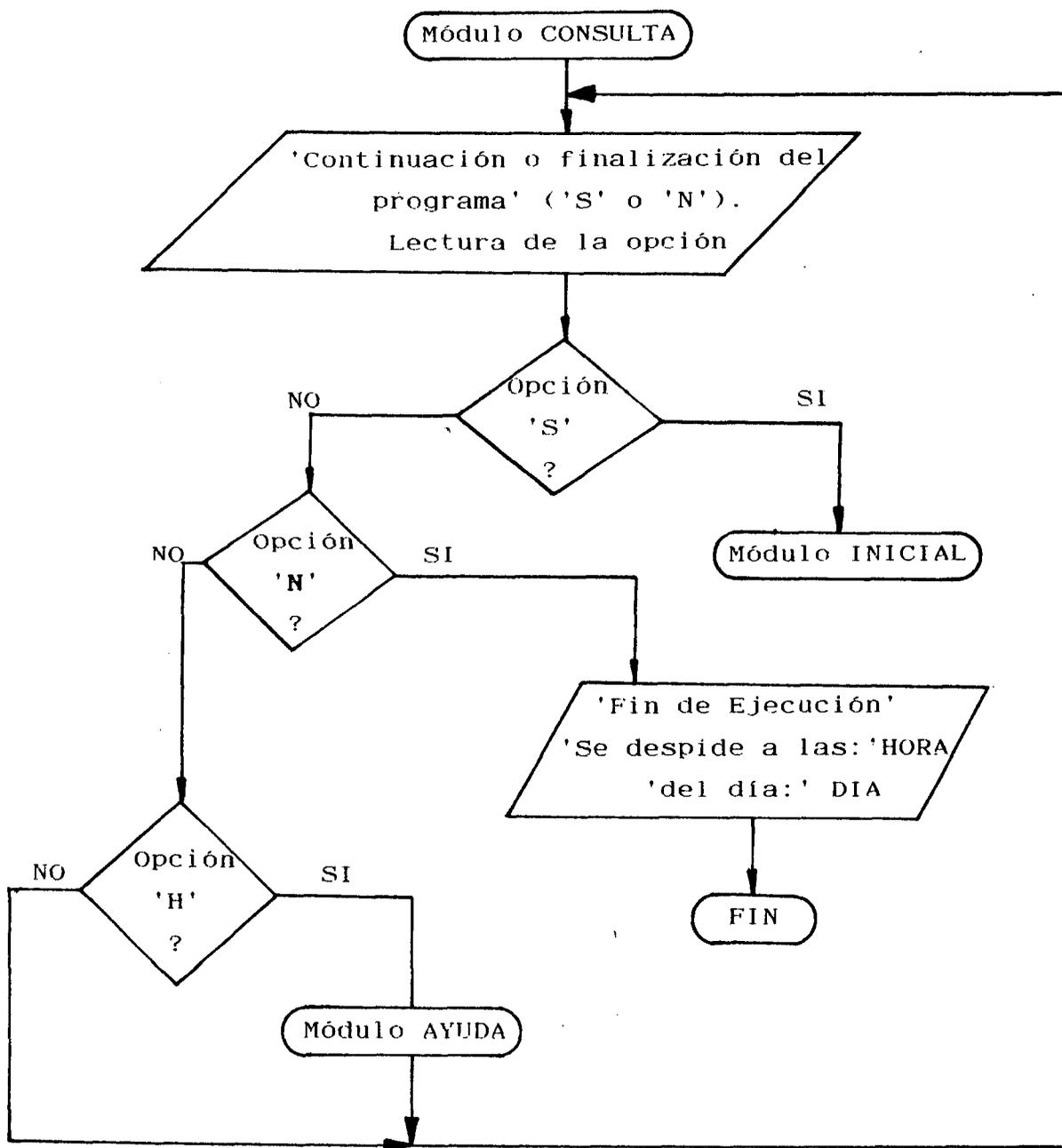
Este procedimiento se encarga de optar por la continuación de la ejecución del programa o por llegar a su fin. En el caso de que se adopte el último criterio, se señalará al usuario el fin de la ejecución, exponiendo además, para mayor información, la hora y fecha de la terminación de su tarea.

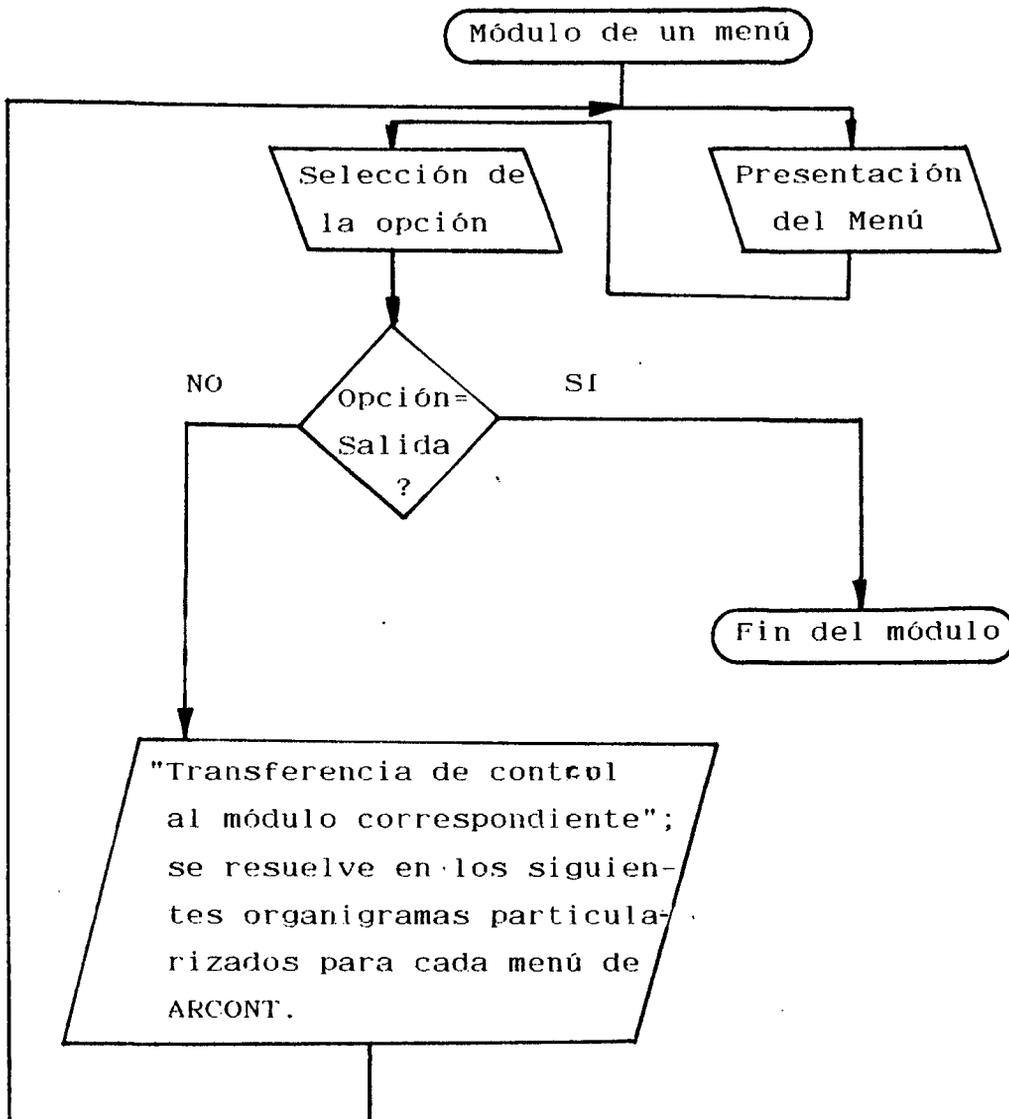
Si sucede que se tome la elección de continuación, el control se transfiere al módulo INICIAL nuevamente.

Siempre que no se opte por ninguna de estas tres situaciones (continuación, terminación o ayuda) el proceso permanecerá a la espera de una de estas entradas.

Llamadas a procedimientos INICIAL y AYUDA.

-Organigrama del módulo CONSULTA:



Organigrama general de los menús.

1.5. MODULO AYUDA. Juego de instrucciones del programa.

Este procedimiento hace una exposición de las instrucciones a seguir por el usuario para que tenga referencia de los pasos que debe seguir para un mejor manejo y utilización del programa . A la vez, para aquel que en un principio no haya hecho uso de este, sepa en cada momento la utilización de ARCONT:

El módulo realiza la lectura de un fichero texto formado por cadenas de caracteres. Este está dividido en diferentes páginas que una vez consultadas transfiere el control al módulo INICIAL.

Para pasar las diferentes páginas únicamente es necesario pulsar la tecla 'RETURN' (de retorno). El paso de una página a otra se ha realizado comparando la variable S,cadena de caracteres, con: '**' que sirve como indicación de la terminación de una página. Posteriormente se hace un borrado de pantalla, y así sucesivamente hasta el final de el fichero.

1.6. MODULO ACTIV. Menú de las cuentas de Actividades Docentes.

Este módulo corresponde al Menú de Actividades Docentes, y lo que realiza es la presenta-

ción de las cuatro opciones que constituyen las cuentas derivadas de este apartado. Entre ellas hay dos menús de cuentas, que son: el de las cuentas de las carreras y el de las cuentas de las asignaturas comunes.

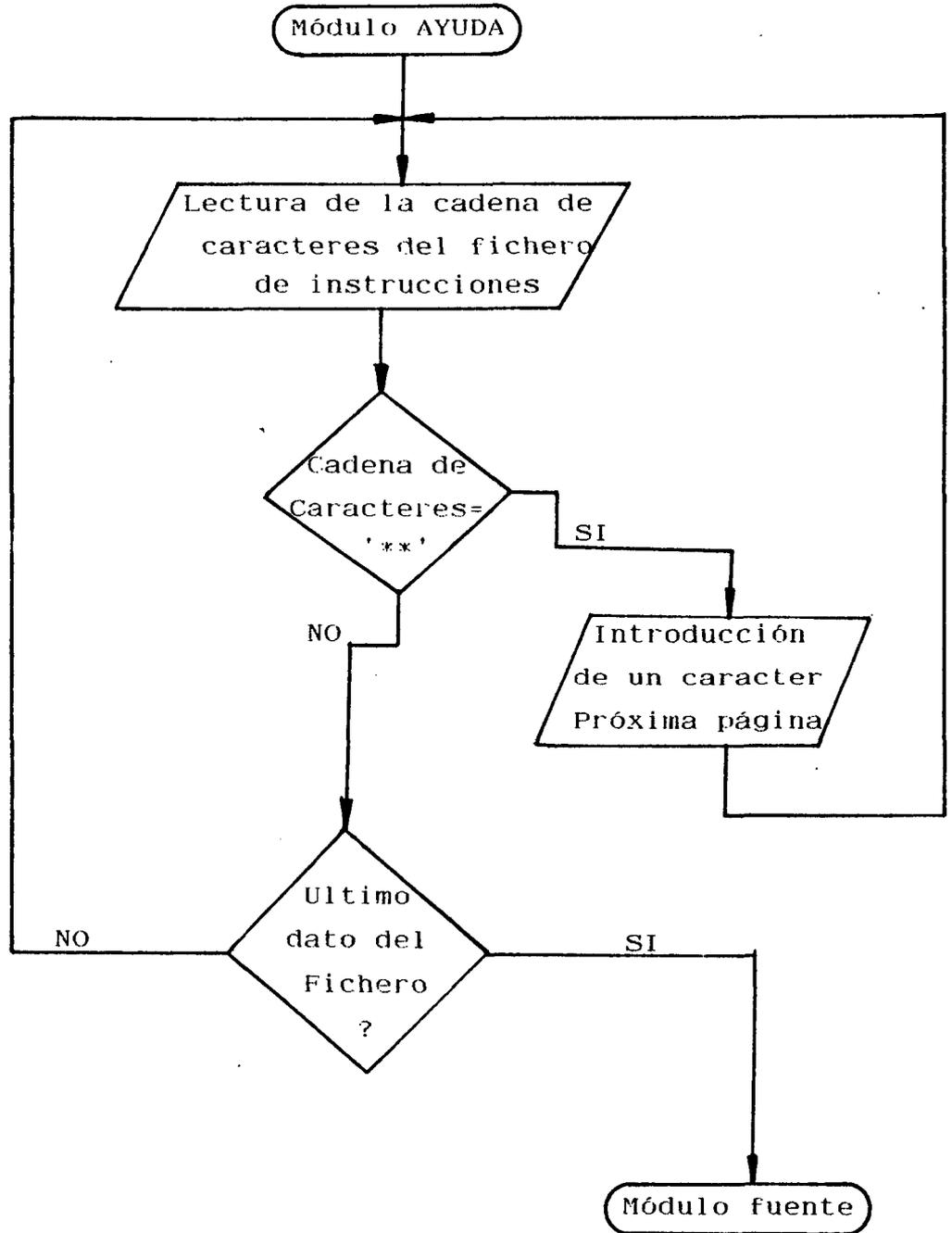
Como acceso directo de este apartado están las cuentas, una: la cuenta general de Actividades Docentes, y, la otra: la cuenta Sin Cargo a Cátedra.

Una vez se haya realizado la elección correspondiente a una cuenta o a un menú de cuentas, este módulo transfiere el control al procedimiento correspondiente. Para ello hace la lectura mediante la variable CH, tipo char(caracter), de la opción seleccionada. También como en los casos de los módulos anteriores siempre sea la entrada igual que el carácter 'H', el control pasa al módulo AYUDA. También en el caso de que la entrada sea distinta de las posibles preestablecidas, el ordenador permanecerá a la espera de una entrada adecuada.

El organigrama de este módulo viene a ser similar al del módulo INICIAL, lo cual significa que, exceptuando los procedimientos a los que hace llamadas, la estructura es idéntica.

Llamadas a los procedimientos: CUENTAGEN, CARRERAS, COMUNES, CUENTASIN, AYUDA, BORRADO, CENIN, OPCION.

-Organigrama del módulo AYUDA.



1.7. MODULO CARRERAS. Menú de las Cuentas de las Carreras de E.U.P.

Este procedimiento muestra las diferentes opciones que podemos tomar en cuanto a las cinco carreras que hay en la Escuela Universitaria Politécnica (Telecomunicaciones, Industriales, Topografía, Obras Públicas y Navales).

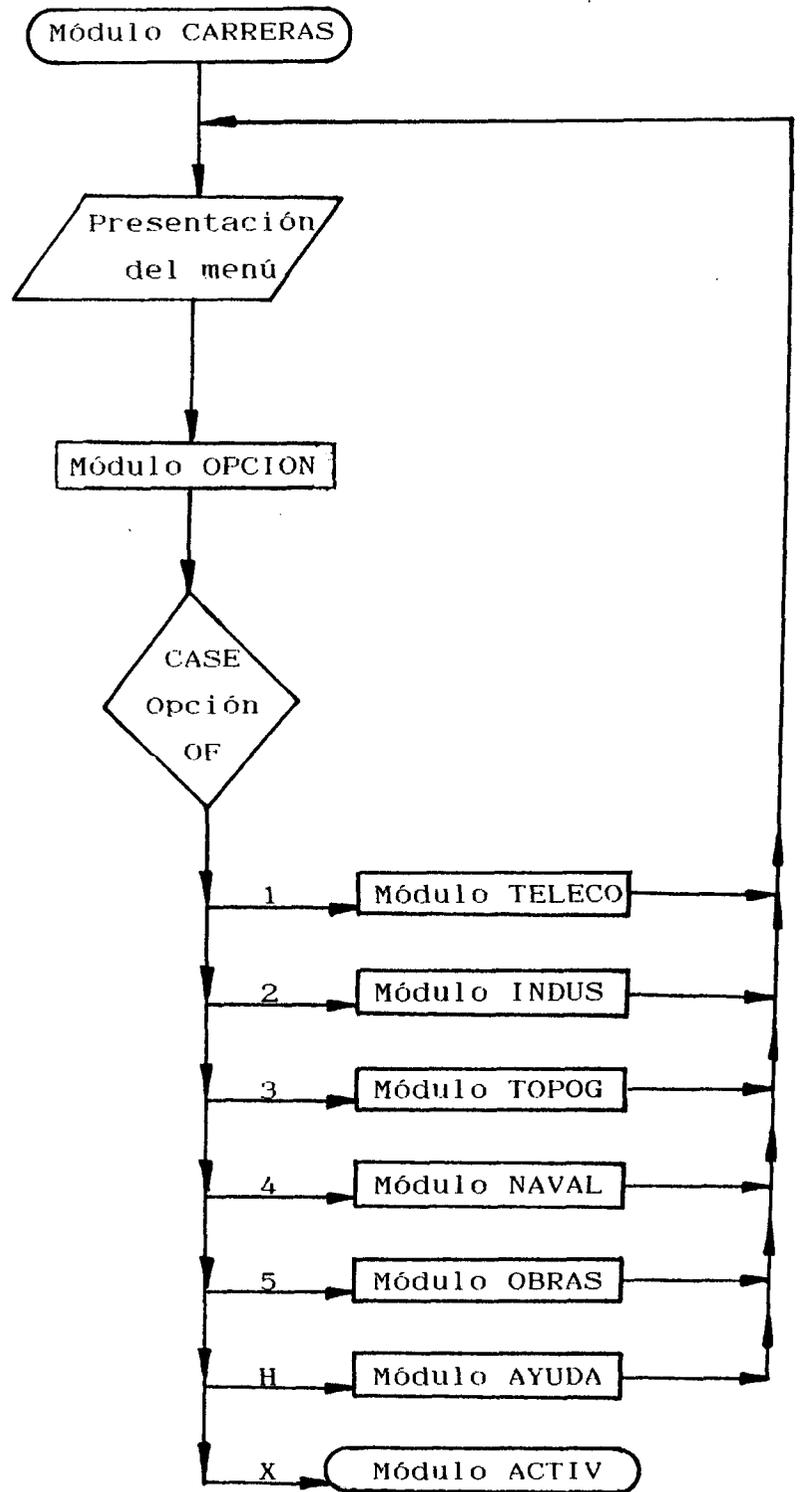
El control pasa al módulo o procedimiento OPCION, anteriormente reseñado, y posteriormente al módulo o procedimiento correspondiente.

Por tratarse de otro menú de cuentas la estructura y, por tanto, su organigrama, es análogo al de los módulos anteriores. La particularidad de este procedimiento es que hace referencia a otros dos menús de cuentas correspondientes a las carreras de Telecomunicaciones e Industriales (TELECO e INDUS respectivamente). Las otras tres carreras se controlan por una cuenta para cada una de ellas.

En lo sucesivo se ha de explicar que tanto los caracteres 'H', como cualquier otro que no corresponda a los establecidos en pantalla, se tratarán según lo explicado en el módulo anterior; es decir, se asegura la entrada del valor de la variable acotada.

Llamada a procedimientos: TELECO , INDUS, TOPOG, OBRAS, NAVAL, CENIN, BORRADO, OPCION.

-Organigrama del módulo CARRERAS.



1.8. MODULO TELECO. Menú de las Cuentas de la Carrera de Telecomunicaciones.

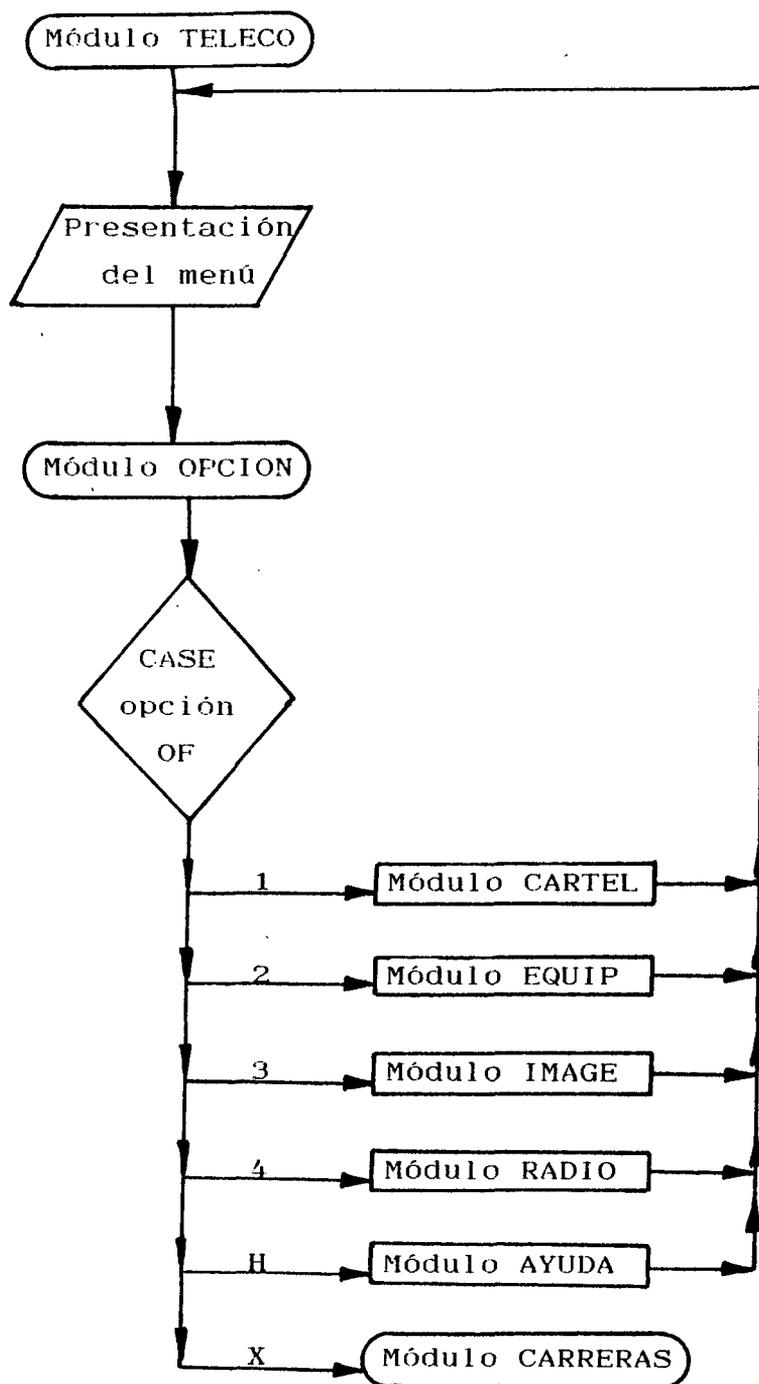
Este módulo se encarga de hacer la presentación del menú de las cuentas que corresponden a este apartado, y que son: Cuenta de la Carrera, Cuenta de Equipos Electrónicos, Cuenta de Imágen y Sonido y Cuenta de Radiocomunicación.

Deja al usuario la elección de una de las cuentas anteriormente citadas, y pasa al control del procedimiento concreto de cada cuenta.

En el caso de que haya cualquier entrada de datos que no corresponda con el debido comando, se produce un sonido y se coloca el puntero en el lugar anterior, para la introducción nuevamente de los datos requeridos.

Llamadas a procedimientos: CARTEL, EQUIP , IMAGE, RADIO, AYUDA, OPCION, CENIN, BORRADO.

-Organigrama del módulo TELECO.



1.9. MÓDULO E.U.P. Cuenta general de la Escuela Universitaria Politécnica.

Este procedimiento o módulo tiene por misión hacer posible tres tareas diferentes que afectan a la cuenta y, por tanto, a los ficheros que de ella dependen. Estas tres tareas son las de: Iniciación de la cuenta, Añadir operaciones a la cuenta y, por último, la consulta de dicha cuenta.

De una forma simplificada, se pueden tratar cada una de cada una de las posibilidades, operando con la cuenta por separado.

Antes hay que exponer un pequeño menú de las opciones por las que se puede acceder a hacer una rutina o no, en este último caso es en el que se acepte la entrada de salir de la cuenta y pasar el menú correspondiente para el acceso al resto de las cuentas que forman el programa.

Una vez se haya elegido una de las tres opciones se transfiere el control a cada uno de los apartados correspondientes. Esto se hace con la variable tipo caracter y con la sentencia en lenguaje Pascal CASE variable OF (en caso de que).

Para el caso de haber elegido la inicialización de la cuenta, lo primero es pedir la confirmación de que se va a realizar esta rutina. Se recuerda que al hacerlo se pierde toda la información de dicha cuenta. Por eso se hace este tipo de observación y, en el caso de que no haya seguridad

antes de hacer la selección, se recomienda que mediante la entrada 'H' se transfiera el proceso al módulo de AYUDA al usuario para más información. Una vez ratificada la inicialización de la cuenta se procede a la presentación del marco para la introducción de los nuevos datos, estos son guardados en los ficheros y se realiza nuevamente la transferencia al menú de opciones.

En el segundo caso, de haber elegido añadir nuevas operaciones a la cuenta, el proceso es inicial al anterior excepto en que se ha de tener en cuenta el número de operaciones en que este proceso se reitera. Una vez se introduce una operación se pide una confirmación para su posterior almacenamiento, y se vuelve a repetir la introducción de nuevos datos.

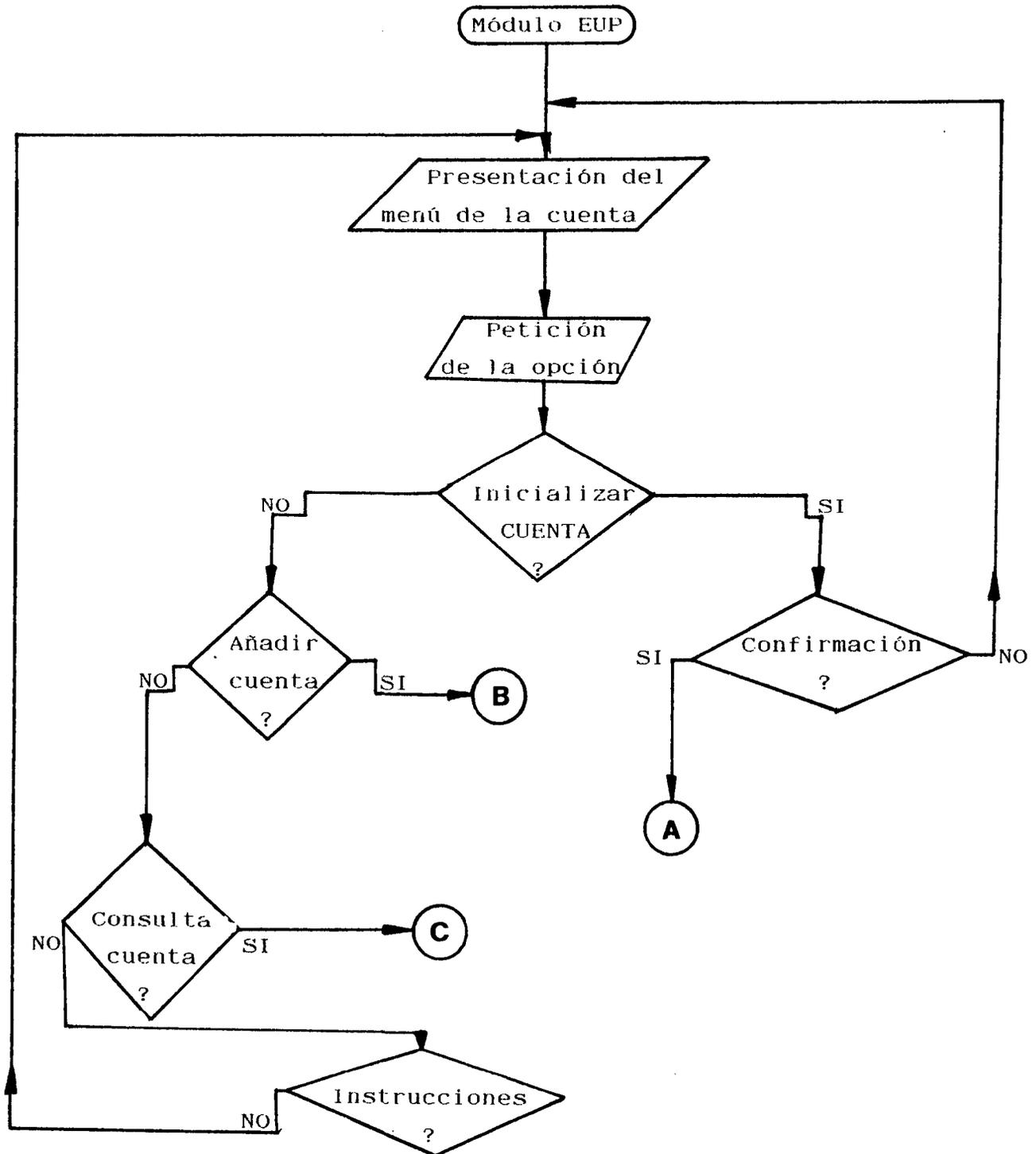
El tercer caso, consulta de la cuenta, está basado en la lectura de los datos almacenados en los ficheros para su posterior presentación. Se trata pues de una tarea de lectura y salida de los datos que estarán divididos en distintas páginas de la cuenta.

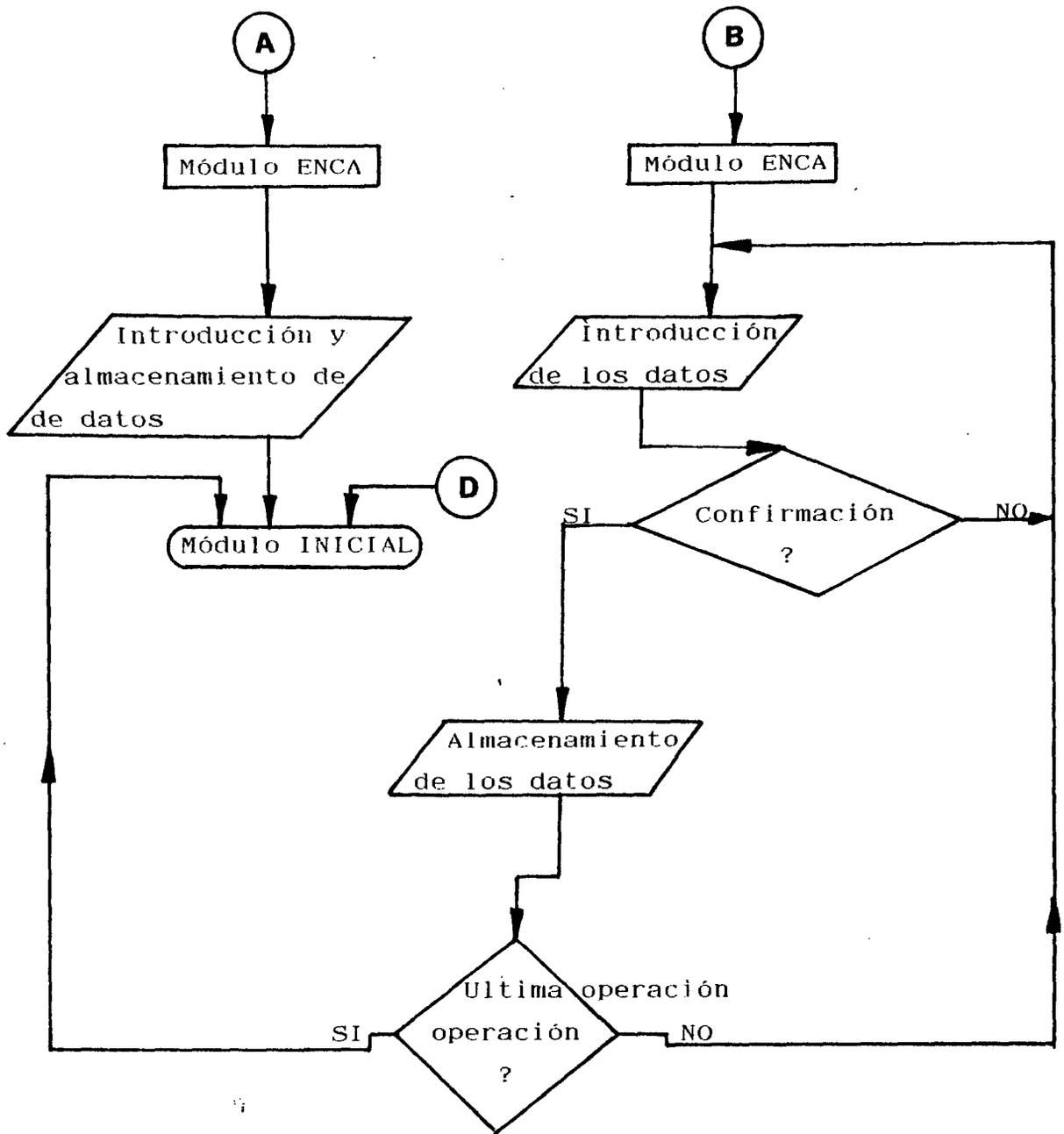
En los dos primeros casos el formato de introducción de datos es igual, y en el tercer caso es más amplio.

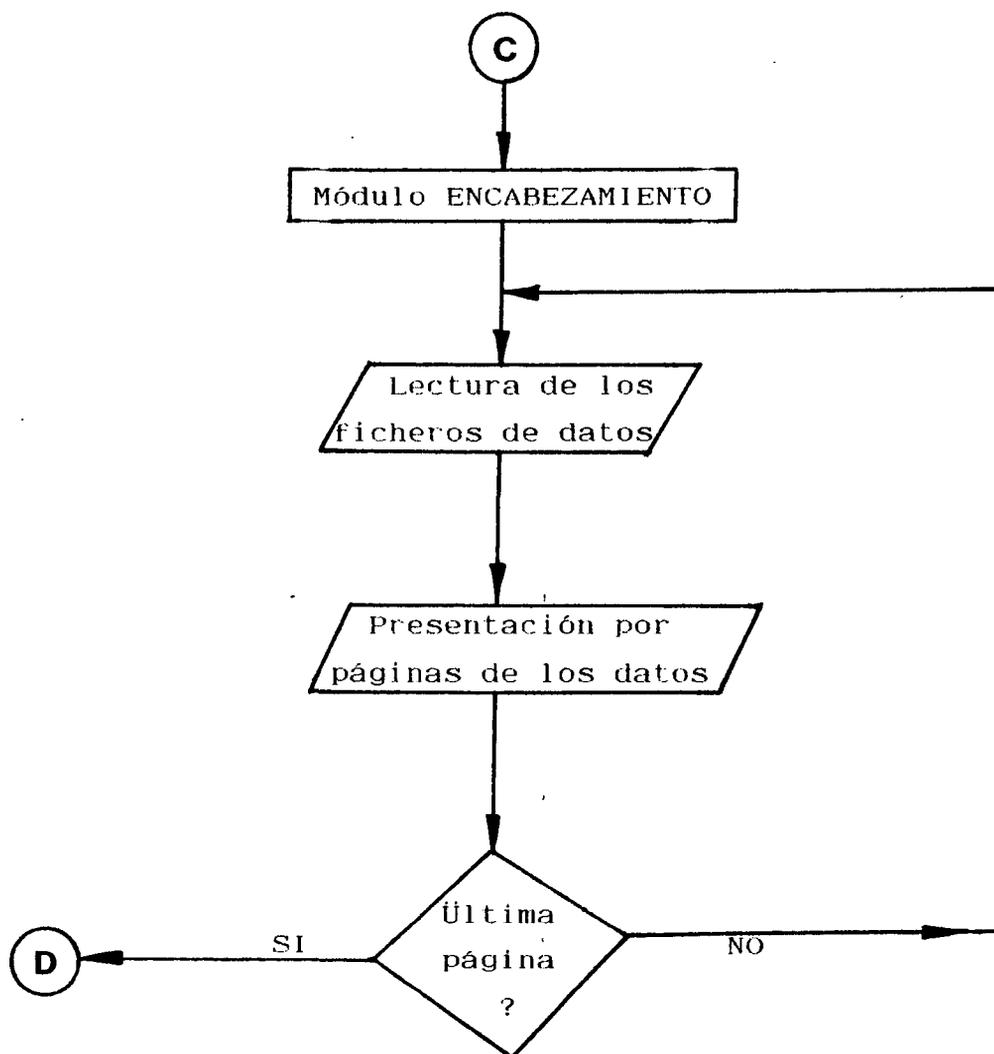
Existe una cuarta opción que es la de salida del módulo de la cuenta que transfiere el control al menú.

Este mismo tratamiento, o de forma similar,

-Organigrama de la Cuenta de la Escuela Universitaria Politécnica.







se han resuelto los módulos o procedimientos : MATER, CULTUR, EDUCA, EXTEN, TRANS, REUNION, MOBILIA, ACTOS, DIETAS y GASTOS. También tienen, así misma, parecido organigrama.

Llamadas a los procedimientos: BORRADO, OP CUENTAS, INTRODADOS, AYUDA, ENCA, CURSOR y ENCABEZAMIENTO.

1.10. MODULO CUENTAGEN. Cuenta General de Actividades Docentes.

Este módulo o procedimiento básicamente es idéntico al anteriormente descrito, salvo en los aspectos concernientes a la múltiple subdivisión de las cuentas dependientes de este apartado tan amplio. Es por el hecho de tener treinta cuentas con todos sus movimientos y menús.

Al igual que el módulo precedente, la cuenta general de actividades docentes, dispone de las cuatro opciones descritas, su acceso a ellas responde a diferentes tareas por realizar. El control del proceso acabará siempre en el menú correspondiente, después de haber aceptado la salida de dicha cuenta.

Las cuentas, y por consiguiente, los procedimientos que pertenecen a este apartado son los siguientes: CARTEL, EQUIP, IMAGE, RADIO, TOPOG, NAVAL, OBRAS, CARIN, ORGIN, CONSTR, CONOCIM, ELECTIR, MECAMAQ, ELECTRII, TERMOI, MATEMAT, ALG, CALC, AM-

PL, FISIC, QUIMIC, DIBU, INGL, CUENTASIN, QUIMICA-
ESP, QUIMVII, QUIMVIII, QUIMIX, ELECTRON y MECAIII.
Ello significa que el tratamiento de éstas es simi
lar y, por tanto, gozan de parecido estructuración

Llamada a los procedimientos: OPCUENTAS,BO
RRADO, ENCA, INTRODATOS, AYUDA, CURSOR y ENCABEZA-
MIENTO.

1.11. MODULO OPCUENTAS. Opciones de operaciones en las cuentas.

Este procedimiento sirve como apoyo al pro
grama principal y tiene por misión sacar por pant
lla las opciones de los cuatro caminos que puede
tomar el usuario.

Primero mediante los controles de pantalla
y otros procedimientos se realiza el borrado de
pantalla y el comentario de la cuenta a la que co-
rresponda el menú de las opciones anteriormente re
señado. Como ya se ha descrito en apartados prece-
dentes, las operaciones que se pueden realizar son:
Inicializar la cuenta, añadir operaciones a la cu-
enta, consulta de la cuenta y salida de la cuenta
al menú correspondiente.

Llamadas a los procedimientos: BORRADO,CE-
NIN.

1.12. MODULO INTRODATOS. Introducción de datos en la inicialización de una cuenta.

Este es un pequeño procedimiento que se repite su proceso a lo largo de todo el programa siempre que se opte por la inicialización de la cuenta. Posteriormente pasa el control al procedimiento de la cuenta en curso y al apartado de inicializar.

Únicamente tiene por misión colocar en un determinado lugar de la pantalla el número de factura, cuando se realiza la operación de inicialización su valor es '000000'. Para ello se han utilizado los controles propios de pantalla del ordenador. No tiene ninguna referencia a otro procedimiento.

1.13. MODULO ENCA. Presentación del marco de la cuenta en la inicialización y en la adquisición de nuevas operaciones.

Este procedimiento forma parte de los módulos dedicados a la presentación en la pantalla del formato en que se introducirán los datos.

Está encargado de realizar un marco en el que se colocarán los diferentes campos que forman la entrada de operaciones de la cuenta, ya sea para la inicialización o para la adquisición de nuevos movimientos. Para ello se coloca en la parte superior de la pantalla y, como referencia para el usuario, los tipos de datos que se han de introdu-

cir (número de factura, fecha en la que se realizó el gasto, el concepto por el que se ha producido tal hecho y el importe de la operación).

Básicamente se trata de un procedimiento dominado por los controles de pantalla, que realizan casi la totalidad de la tarea. Para ello ha sido necesario colocar bucles que ejecutan repetidamente instrucciones gráficas.

De forma similar se ha concebido el procedimiento ENCABEZAMIENTO, exceptuando que se utiliza cuando se desea una consulta de una cuenta determinada y, por tanto, es necesario aumentar el número de datos, que son los de: número de operación y saldo correspondiente para cada movimiento de la cuenta.

Llamada a los procedimientos: BORRADO, CENTRADO, CURSOR e INVERSO.

1.14. MODULO CENIN. Procedimiento de centrado y salida en "reverse" del comentario.

Este sencillo procedimiento está formado por otros dos que posteriormente se describirán. Se utiliza en el caso en que simultaneamente se desea obtener un comentario, como por ejemplo: "CUENTA DE LA E.U.P.", centrada en las ochenta columnas de que consta la pantalla y además aparezca en "reverse" (contrario a la exposición del resto de los carac-

teres representados en el "display").

Es otro de los módulos de salida del programa.

Llamadas a los procedimientos: CENTRADO, INVERSO.

1.15. MODULO OPCION. Petición de la opción deseada

Este procedimiento es uno más de los módulos dedicados a formar parte de las salidas por el periférico monitor.

Utiliza un único comentario que se repite a lo largo de todos los menús en los que salgan las cuentas a elegir. El comentario es : ' Escriba la opción (número o letra correspondiente) que de see: '.

Existe, después del comentario, la sentencia de lectura, del periférico teclado , tipo char (caracter). Posteriormente se transfiere el control al procedimiento del menú correspondiente (aquel que ha realizado la llamada a este módulo).

1.16. MODULO CENTRADO. Centrado del comentario en ochenta columnas.

Este procedimiento necesita que se haya declarado anteriormente el comentario que actúa como variable del tipo string de caracteres (en lengua-

je de alto nivel, Digital Vax Pascal : varying (c) of char;). A parte de esta declaración, existe otra que es la variable entera: J.

Empieza el procedimiento asignando a la variable J el valor de la resta entre: ochenta (número de columnas de la pantalla) y la longitud de la cadena de caracteres que representa el comentario. Una vez hecha la sustracción descrita, se divide por la mitad. Este resultado es el que toma la variable J que va a representar el límite del bucle. Este bucle significa que se van a colocar espacios en blanco hasta conseguir el valor J. Así quedará centrado cualquier valor que asuma la variable COMENTARIO.

Por último, la sentencia siguiente es la de salida por pantalla del valor de COMENTARIO.

Este procedimiento es muy adecuado para la representación de la información en pantalla de forma simétrica y estética.

Una vez se haya realizado el procedimiento el proceso continúa en el módulo desde el que se ha solicitado éste.

1.17. MODULOS BORRADO, CURSOR e INVERSO.

Estos módulos ilustran como borrar la pantalla (BORRADO), mover el cursor a un lugar específico de la pantalla (CURSOR) y enviar datos al terminal utilizando las rutinas de formato de pantalla.

lla.

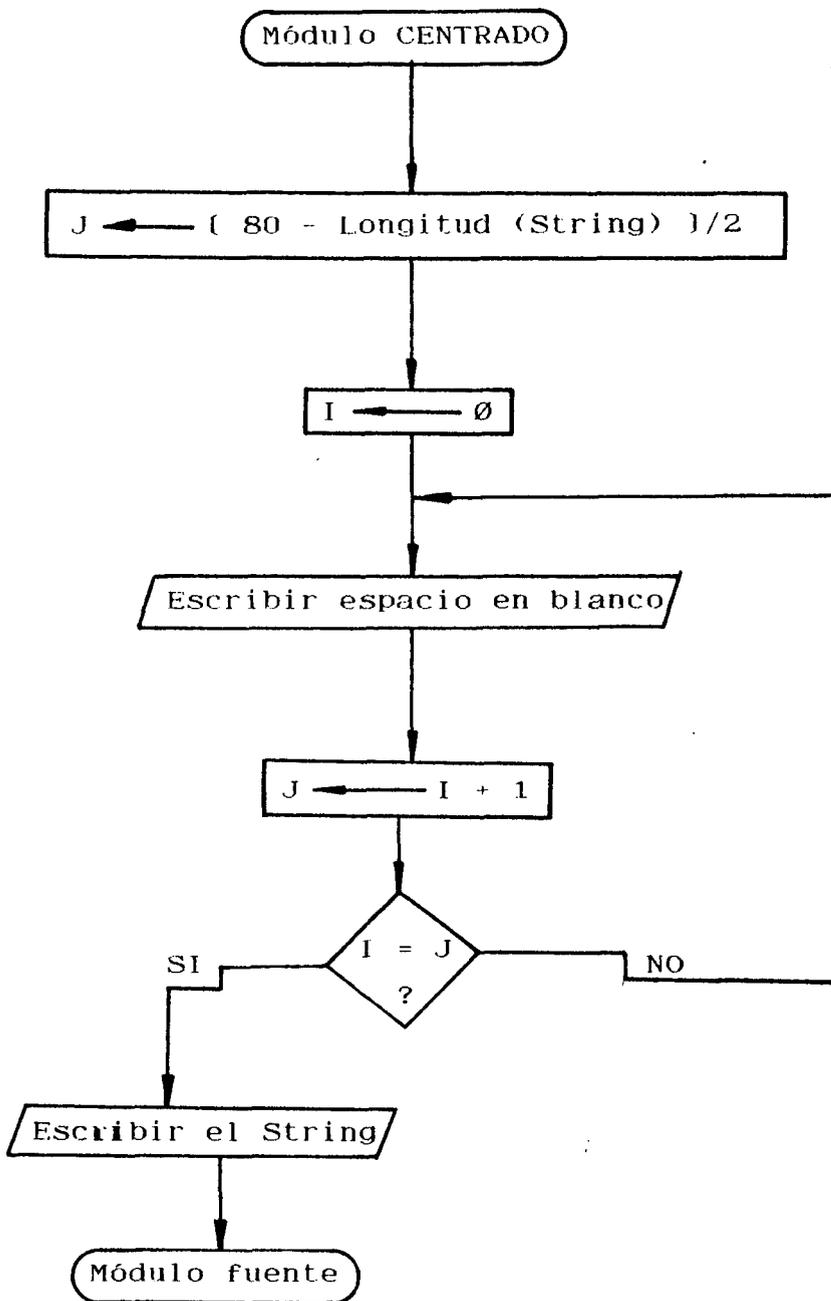
El procedimiento BORRADO utiliza la rutina 'ERASE-PAGE', que hace desaparecer todos los caracteres desde la línea y columna especificada (en este caso, desde el principio de la pantalla, línea 1 columna 1) hasta el final de la pantalla.

El procedimiento CURSOR debe posicionar en algún lugar de la pantalla el cursor, sin borrar ninguna información ya existente en el terminal. Así el cursor puede ser colocado en alguna dirección después de visualizar o recibir datos. Todo ello se realiza con la rutina SET_CURSOR.

El procedimiento INVERSO utiliza los atributos del "reverse-video" mediante la rutina PUT_SCREEN.

Todos estos procedimientos necesitan declarar variables de diferentes tipos para las tres rutinas anteriormente descritas. Ellos forman parte (módulos) de la librería de los 'RMS Procedure' de los servicios del sistema VAX/VMS.

-Organigrama del módulo CENTRADO.



2. Ficheros de datos correspondientes a las cuentas.

En el programa ARCONT se han declarado dos tipos fundamentales de ficheros, que son: el tipo texto (formado por cadena de caracteres) y el tipo entero (formado por los datos enteros).

Los datos asignados a las variables de tipo entero son: IMPORTE y SALDO. La fecha, el número de factura y el concepto son cadenas de caracteres almacenados en los ficheros TEXT (tipo texto). Este último tipo tiene como componente básico a un simple carácter, sin embargo en el record RMS (Record Management Service), de un fichero 'text', equivale a una línea completa de caracteres que termina con la marca de fin de línea (end-of-line , EOL).

En el fichero de tipo entero, cada record RMS contiene la representación de un valor entero. Así, para cada cuenta existen tipo 'Integer' (entero) y tipo 'text' (texto).

Las variables declaradas como ficheros textos tienen el siguiente formato: F + " Nombre del módulo de la cuenta ". Por ejemplo: FEUP, FMATER , FTRANS, etc. Las variables declaradas como ficheros de datos enteros se han nombrado como sigue : " Nombre de la cuenta " + IMPORTE o SALDO. Por ejemplo: EUPSALDO, EUPIMPORTE, MATERSALDO, etc.

2.1. Fichero de datos de instrucciones del programa.

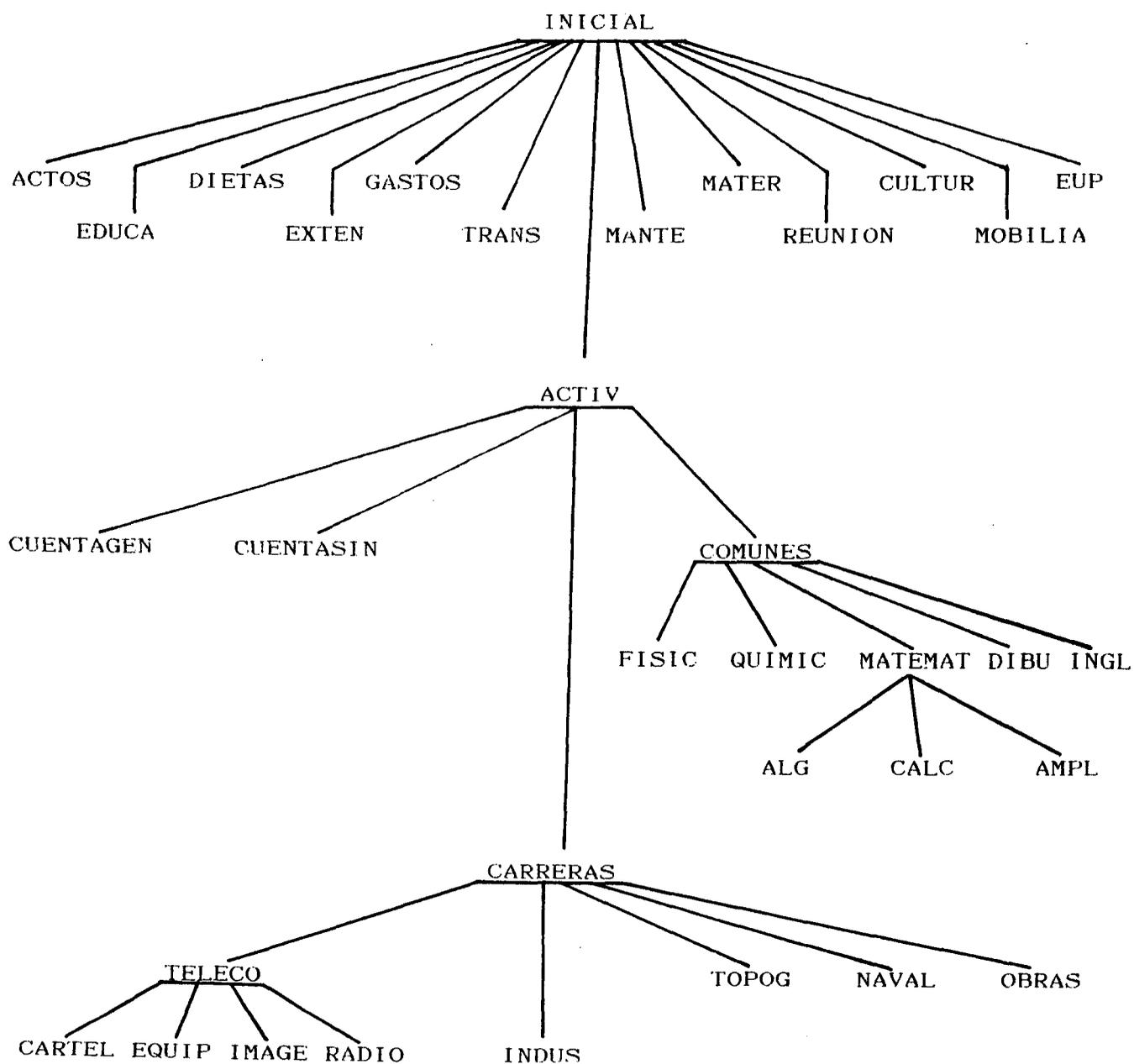
Este fichero de datos, tipo texto, es decir, constituido por caracteres, es el utilizado por el módulo AYUDA para que su lectura facilite la información del programa al usuario.

Está dividido por diferentes páginas, únicamente separada una de otra por los dos asteriscos, '**', que indican el final de una página. El resto de los datos que contiene dicho fichero son cadenas de caracteres (Varying (c) of char , o lo que es igual: string of char).

El modo en que está organizado este fichero de auxilio al usuario es secuencial , al igual que el método de acceso a los datos. Ello significa que la información es presentada línea a línea según se va efectuando la lectura de los datos contenidos en dicho fichero. La longitud de cada record es de 235 bytes, para poder aumentar algo más el valor en defecto de este parámetro construido por el VAX PASCAL tipo 'text', que es de 133 bytes. El parámetro 'Record Type' es variable, es decir , puede ir desde cero o dos caracteres hasta un número muy superior, así las líneas consiguen las longitudes deseadas en la construcción del fichero. Otro parámetro especificado para este fichero de instrucciones es el parámetro 'Sharing', que en es

te caso tiene el valor REASONLY, lo cual significa que mientras esté abierto (OPEN) sólo pueden leerse los datos y no se pueden escribir en él (para este último caso sería necesario especificar el parámetro 'Sharing' como READWRITE que no se especifica para la aplicación a la que nos referimos).

Estructura de Arbol del Programa ARCONT eg
 comenzando de un módulo de partida: módulo INICIAL y
 su ramificación. Es decir, aquí se marcan los cami
 nos que se han de seguir para acceder a las distin
 tas cuentas, unas dependientes de otras.



APENDICE A

APENDICE A. CONTROL DE PANTALLA (Terminales).

Este apartado describe básicamente las secuencias 'escape' para los terminales Digital, series VT-200 y VT-100.

Las divisiones en las que se van a exponer las secuencias de los controles de pantalla van a ser por este orden: Movimiento del cursor, Conjuntos de caracteres, Características de visualización por pantalla, Borrado de pantalla, Creación de pantallas movibles "scroll" y Características del terminal.

A.1. Movimiento del cursor.

Antes de describir las varias y diferentes posibilidades del manejo del cursor por la pantalla, hay que hacer constar que en la tabla de caracteres ASCII el símbolo <ESC> tiene el valor decimal veintisiete. Para conseguir esta secuencia, por tanto, se ha de especificar el carácter número 27 que se representa en lenguaje de alto nivel VAX PASCAL: CHR(27).

En la siguiente lista de caracteres que permiten mover el cursor, donde n representa un número entero que determina el número de posiciones o líneas que el cursor se ha de mover (si se omite el valor n, el número, por defecto, será el uno).

Los caracteres aquí representados, r y c , son enteros que indican el número de fila y columna respectivamente.

CHR(27){nA Mueve el cursor hacia arriba, sin mover la página (no scrolling).

CHR(27) M Mueve el cursor hacia arriba junto con la región 'scrolling'.

CHR(27){nB Mueve el cursor hacia abajo, sin 'scrolling'

CHR(27) D <LF> Mueve el cursor hacia abajo en conjunto con la región 'scrolling'.

CHR(27){nC Mueve el cursor a la derecha.

CHR(27){nD Mueve el cursor a la izquierda.

CHR(27)E Mueve el cursor al comienzo de la próxima línea.

CHR(27) {r; cH	Coloca el cursor en la posición especificada de r, fila y cā columna en la pantalla.
CHR(27) 7	Guarda la posición, en cuanto a columna, del cursor y los atributos de caracteres.
CHR(27) 8	Almacena la posición, columna, del cursor y sus atributos.

A.2. Conjunto de caracteres.

Estas secuencias permiten escoger un grupo de caracteres determinado. Las terminales de las series VT-100 y VT-200 de Digital consiguen diferenciar tres conjuntos:

1. US. Conjunto de caracteres ASCII. La secuencia 'escape' es CHR(27) B.

2. UK. Conjunto de caracteres de United Kingdom (Reino Unido) que tiene pequeñas diferencias con el anterior, como lo es el signo de la libra £. La secuencia 'escape' es CHR(27) CA.

3. Graphic. Conjunto de caracteres gráficos ,que se especifican con la secuencia 'escape' CD.

A.3. Características específicas de visualización.

La siguiente lista de secuencias 'escape' permiten especificar las características de visualización de los caracteres en pantalla.

CHR(27) {0m	Escribe caracteres normales.
CHR(27) {1m	Escribe caracteres en negrita.
CHR(27) {4m	Escribe caracteres subrayados.
CHR(27) {5m	Escribe caracteres parpadeando.
CHR(27) {7m	Escribe caracteres en 'reverse-video'.
CHR(27) #3	Escribe caracteres el doble de alto(en la m <i><u>i</u></i>

tad superior de la
pantalla.)

CHR(27) #4 Escribe caracteres
el doble de alto ,
en la mitad inferior
de la pantalla.

CHR(27) #5 Escribe caracteres
de ancho normal.

CHR(27) #6 Escribe caracteres
en doble ancho.

Las secuencias CHR(27) #3 y CHR(27) #4 no
pueden utilizarse con el conjunto de caracteres
gráficos.

A.4. Borrado de pantalla.

La siguiente lista de secuencias permite
borrar la zona de pantalla que desee.

CHR(27) {OK Borra desde el cursor has-
ta el final de la linea.

CHR(27) {ØK Borra desde el principio

de la línea hasta el cursor.

CHR(27) (2K Borra la línea entera.

CHR(27) (0J Borra desde el cursor hasta el final de la pantalla.

CHR(27) (1J Borra desde el principio de la pantalla hasta el cursor.

CHR(27) (2J Borra la pantalla entera.

A.5. Creación de una región 'scrolling'.

a región 'scrolling' es una sección de la pantalla dentro de la cual el texto se mueve.

En la primera secuencia 'escape' t y b representan valores enteros que indican la primera y última línea de la región 'scrolling'. Ambos valores deben estar comprendidos entre 1 y 24 ; y , es preciso que, t sea menor que b.

CHR(27) (t;b Sube la región 'scrolling'.

CHR(27) [?bh	Coloca el cursor en la posición 0, 0 igual a la esquina superior de la región 'scrolling'.
CHR(27) [?bl	Coloca el cursor en el principio de la pantalla.

A.6. Características del terminal.

Las siguientes secuencias permiten utilizar los atributos del terminal.

CHR(27) [?3l	Ancho de la pantalla: 80 columnas.
CHR(27) [?3h	Ancho de la pantalla:132 columnas.
CHR(27) H	Pone una marca de 'tab' en la columna en curso.
CHR(27) [g	Borra el 'tab' de la columna en curso.
CHR(27) [3g	Borra todos los 'tab'.

3. LISTADO DEL PROGRAMA "ARCONT"

-LINE-IDC-PL-SL-

```
00001      0 0 PROGRAM ARCONT (INPUT,OUTPUT);
00002      0 0 LABEL 100;
00003      0 0 TYPE WORD_INTEGER=iWORD& 0..65535;
00004      0 0 VAR CHA,CH: CHAR;
00005      0 0      U,IMPORTE,SALDO,J,I,FILA,COLUMNA,W: INTEGER;
00006      0 0      SCREEN_STAT: INTEGER;
00007      0 0      NOMBRE: CHAR;
00008      0 0      NFACTURA,FECHA,CONCEPTO,COMENTARIO: VARYING i300& OF CHAR;
00009      0 0      PUT_FLAG: WORD_INTEGER;
00010      0 0      FEUP: TEXT;
00011      0 0      EUPIMPORTE, EUPSALDO: FILE OF INTEGER;
00012  C 0 0 (*****
00013      1 0 PROCEDURE BORRADO;
00014      2 0 FUNCTION LIB$ERASE_PAGE (LINE_NO: WORD_INTEGER;
00015      1 0 COL_NO: WORD_INTEGER): INTEGER; EXTERN;
00016      0 0 BEGIN SCREEN_STAT:=LIB$ERASE_PAGE(1,1); END;
00017  C 0 0 (*****
00018      1 0 PROCEDURE CURSOR;
00019      2 0 FUNCTION LIB$SET_CURSOR (LINE_NO: WORD_INTEGER;
00020      1 0 COL_NO: WORD_INTEGER): INTEGER; EXTERN;
00021      0 0 BEGIN SCREEN_STAT:= LIB$SET_CURSOR (FILA,COLUMNA); END;
00022  C 0 0 (*****
00023      1 0 PROCEDURE ENTRADA;
00024      2 0 FUNCTION LIB$GET_SCREEN ( VAR INPUT_TEXT: CHAR;
00025      2 0 PROMPT_STR: VARYING iLIM2& OF CHAR;
00026      1 0 VAR OUT_LEN: WORD_INTEGER:= %IMMED 0): INTEGER; EXTERN;
00027      0 0 BEGIN SCREEN_STAT:= LIB$GET_SCREEN (NOMBRE,COMENTARIO); END;
00028  C 0 0 (*****
00029      1 0 PROCEDURE INVERSO;
00030      2 0 FUNCTION LIB$PUT_SCREEN ( OUT_TEXT: VARYING iC& OF CHAR;
00031      2 0 LINE_NO: WORD_INTEGER; COL_NO: WORD_INTEGER;
00032      1 0 FLAGS: WORD_INTEGER): INTEGER; EXTERN;
00033      1 1 BEGIN PUT_FLAG:= 2;
00034      0 0 SCREEN_STAT:= LIB$PUT_SCREEN (COMENTARIO,FILA,COLUMNA,PUT_FLAG);
00035  C 0 0 (*****
00036      1 0 PROCEDURE CENTRADO (COMENTARIO: VARYING iC& OF CHAR);
00037      1 0 VAR I,J: INTEGER;
00038      1 1 BEGIN
00039      1 1 J:= ( 80 - LENGTH (COMENTARIO)) DIV 2;
00040      1 1 FOR I:= 1 TO J DO WRITE (' ');
00041      1 1 WRITELN (COMENTARIO); FILA:=1; COLUMNA:= J;
00042      0 0 END;
00043  C 0 0 (*****
00044  C 0 0 (***** AYUDA *****)
00045  C 0 0 (*****
00046      1 0 PROCEDURE AYUDA;
00047      1 0 VAR K: CHAR; j: TEXT; S: VARYING i300& OF CHAR;
00048      1 1 BEGIN
00049      1 1 BORRADO;
00050      1 1 COMENTARIO:= ' AYUDA AL USUARIO ';
00051      1 1 CENTRADO (COMENTARIO); INVERSO;
00052      1 1 OPEN (j,'j.dat',HISTORY:= OLD,RECORD_LENGTH:= 235,SHARING:= READ
00053      1 1 RESET (j);
00054      1 1 WHILE NOT EOF (j) DO
00055      1 2 BEGIN
```

-LINE-IDC-FL-SL-

```

00056      1  2      READLN (j,S);  WRITELN (S);
00057      1  2      IF S = '**' THEN
00058      1  3          BEGIN
00059      1  3          COMENTARIO:='Pulse la tecla < RETURN > para contin
00060      1  3          ENTRADA;      FILA:=20;      COLUMNA:=1; INVERSO;
00061      C  1  3          (* READ (K); *)
00062      1  3          BORRADO;
00063      1  2          END;
00064      1  1          END;
00065      1  1  WRITE (CHR(27), 'i22;5H');
00066      1  1  WRITE (' Pulse <RETURN> para la ejecuci3rn del programa ...');
00067      1  1  READ (CH);
00068      0  0  END;
00069      C  0  0  (*****
00070      1  0  PROCEDURE OPCION;
00071      1  1  BEGIN
00072      1  1  WRITE (CHR(27), 'i22;5H');
00073      1  1  WRITE ('Escriba la opcion (numero o letra correspondiente) que d
00074      1  1  WRITE (CHR(27), 'i0J'); READLN (CH);
00075      0  0  END;
00076      C  0  0  (*****
00077      1  0  PROCEDURE CENIN;
00078      0  0  BEGIN CENTRADO (COMENTARIO); INVERSO; END;
00079      C  0  0  (*****
00080      1  0  PROCEDURE ENCABEZAMIENTO;
00081      1  1  BEGIN
00082      1  1  BORRADO; CENTRADO (COMENTARIO);
00083      1  1  WRITE (CHR(27), '(O'); WRITE ('1');
00084      1  1  FOR I:= 1 TO 78 DO
00085      1  1  WRITE ('q'); WRITELN ('k');
00086      1  1  FOR I:= 1 TO 18 DO
00087      1  2  BEGIN
00088      1  2  WRITE ('x'); WRITE (' ':78); WRITELN ('x');
00089      1  1  END;
00090      1  1  WRITE ('m');
00091      1  1  WRITE (' ':78);
00092      1  1  WRITELN ('j');
00093      1  1  WRITELN;
00094      1  1  WRITE ('t');
00095      1  1  FOR I:= 1 TO 78 DO WRITE ('q');
00096      1  1  WRITE ('u');
00097      1  1  FILA:=3; COLUMNA:=3; CURSOR;
00098      1  1  COMENTARIO:=
00099      1  1  ' N.OPER.      N.FACT.      FECHA      CONCEPTO      IMPOR
00100      1  1  INVERSO;
00101      1  1  FOR I:= 1 TO 17 DO WRITELN;
00102      1  1  WRITE ('m');
00103      1  1  FOR I:= 1 TO 78 DO WRITE ('q');
00104      1  1  WRITE ('j');
00105      0  0  WRITELN (CHR(27), '(B'); END;
00106      C  0  0  (*****
00107      1  0  PROCEDURE ENCA;
00108      1  1  BEGIN
00109      1  1  BORRADO; CENTRADO (COMENTARIO);
00110      1  1  WRITE (CHR(27), '(O');

```

-LINE-IDC-FL-SL-

```

00111      1 1 WRITE ('      l');
00112      1 1 FOR I:= 1 TO 63 DO WRITE ('q');
00113      1 1 WRITELN ('k');
00114      1 1 FOR I:= 1 TO 18 DO
00115      1 1     BEGIN WRITE ('      x'); WRITE (' ':63); WRITELN ('x'); EN
00116      1 1 WRITE ('      m');
00117      1 1 WRITE (' ':63);
00118      1 1 WRITELN ('j');
00119      1 1 WRITELN;
00120      1 1 WRITE ('      t');
00121      1 1 FOR I:= 1 TO 63 DO WRITE ('q');
00122      1 1 WRITE ('u');
00123      1 1 FILA:=3; COLUMNA:=9; CURSOR;
00124      1 1 COMENTARIO:=
00125      1 1 ' N.FACT.      FECHA      CONCEPTO      IMPORTE  ';
00126      1 1 INVERSO;
00127      1 1 FOR I:= 1 TO 17 DO WRITELN;
00128      1 1 WRITE ('      m');
00129      1 1 FOR I:= 1 TO 63 DO WRITE ('q');
00130      1 1 WRITE ('j');
00131      1 1 WRITELN (CHR(27),'(B)');
00132      0 0 END;
00133      C 0 0 (*****
00134      1 0 PROCEDURE INTRODATOS;
00135      1 1 BEGIN
00136      1 1 WRITE (CHR(27),'(O)'); WRITE (CHR(27),'i9;10H');
00137      1 1 WRITE ('000000'); (*      factura *)
00138      1 1 WRITE ('      x'); WRITE (CHR(27),'(B)');
00139      0 0 END;
00140      C 0 0 (*****
00141      1 0 PROCEDURE OPCUENTAS;
00142      1 1 BEGIN
00143      1 1 BORRADO; CENIN; WRITE (CHR(27),'i9;23H');
00144      1 1 WRITELN ('I: Inicializar la cuenta'); WRITELN;
00145      1 1 WRITELN (' ':22,'A: Agadir operaciones a la cuenta'); WRITELN;
00146      1 1 WRITELN (' ':22,'C: Consulta de la cuenta'); WRITELN;
00147      1 1 WRITELN (' ':22,'X: Salida de la cuenta al menz');
00148      0 0 END;
00149      C 0 0      (*****
00150      C 0 0      (***** EUP *****
00151      C 0 0      (*****
00152      1 0 PROCEDURE EUP;
00153      1 0 LABEL 10,11,AA,A,B,C,D,R;
00154      1 0 TYPE STRING= VARYING i300 OF CHAR;
00155      1 0 VAR FEUP: TEXT; EUPSALDO,EUPIMPORTE: FILE OF INTEGER;
00156      1 0 S,FECHA,NFACTURA,CONCEPTO: STRING; M,J,N,I,W,IMPORTE,SALDO:
00157      1 1 BEGIN
00158      1 1 COMENTARIO:= ' CUENTA GENERAL DE LA E.U.P. ';
00159      1 2 OPCUENTAS; 10: OPCION; CASE CH OF
00160      C 1 2 (*****,      Inicializaci3n de la cuenta *****
00161      1 3 'I','i';      BEGIN
00162      1 3 11: BORRADO;WRITE (CHR(27),'i11;3H'); WRITELN (
00163      1 3 'Pylse la "S" para confirmar la inicializaci3n, "N" en caso cont
00164      1 3 WRITE (' ':2,'(recuerde que todas las operaciones son borradas);
00165      1 4 WRITE (CHR(27),'i0J'); READLN (CH);      CASE CH OF

```

--LINE-IDC-PL-SL--

```
00166      1  5          'S','s': BEGIN ENCA; INTRODADOS;
00167      1  6      REPEAT AA: WRITE (CHR(27),'i5;20H'); READLN (FECHA);
00168      1  6          IF FECHA = '' THEN BEGIN WRITE (CHR(7));GOTO AA; END;
00169      1  5      UNTIL LENGTH (FECHA) < 10;
00170      1  5      WRITE (CHR(27),'(O)'); WRITE (CHR(27),'i5;28H'); WRITE (' x ');
00171      1  5      WRITE (CHR(27),'(B)');
00172      1  6      CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada ');
00173      1  6      WRITE (CHR(27),'i5;55H');WRITE (CHR(27),'(O)');WRITE (' x ');
00174      1  5      UNTIL IMPORTE < 100000000; WRITE (CHR(27),'(B)'); SALDO:= IMPOR
00175      1  5          OPEN (FEUP,'FEUP.DAT'); OPEN (EUPSALDO,'EUPSALDO.DAT');
00176      1  5          OPEN (EUPIMPORTE,'EUPIMPORTE.DAT');
00177      1  5      REWRITE (FEUP); REWRITE (EUPIMPORTE); REWRITE (EUPSALDO);
00178      1  5      WRITELN (FEUP,NFACTURA); WRITELN (FEUP,FECHA); WRITELN (FEUP,CO
00179      1  5      WRITE (EUPIMPORTE,IMPORTE); WRITE (EUPSALDO,IMPORTE);
00180      1  4          CLOSE (FEUP); CLOSE (EUPIMPORTE); CLOSE (EUPSALDO); END;
00181      1  4          'N','n': BEGIN WRITELN; END;
00182      1  4          'H','h': BEGIN AYUDA; GOTO 11; END;
00183      1  3          OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END; END;
00184      1  2      EUP; END; (* Desde Begin del 'i','I': *)
00185      C  1  2      (***** Agadir operaciones *****)
00186      1  3          'A','a': BEGIN ENCA; OPEN (FEUP,'FEUP.DAT',HISTORY:= OLD
00187      1  3      OPEN (EUPIMPORTE,'EUPIMPORTE.DAT',HISTORY:= OLD);
00188      1  3      OPEN (EUPSALDO,'EUPSALDO.DAT',HISTORY:= OLD);
00189      1  4      REPEAT WRITE (CHR(27),'i22;8H');
00190      1  4      WRITE ('Introduzca el nzmero de operaciones: ');
00191      1  3      WRITE (CHR (27),'i0J'); READLN (N); UNTIL N < 16;
00192      1  5      FOR I:= 1 TO N DO BEGIN REPEAT
00193      1  5      A: FILA:= I+4; COLUMNA:= 10; CURSOR; READLN (NFACTURA);
00194      1  5          IF NFACTURA = '' THEN BEGIN WRITE (CHR(7)); GOTO A; END;
00195      1  4      UNTIL LENGTH (NFACTURA) < 10;
00196      1  5      REPEAT B: FILA:= I+4; COLUMNA:= 18; CURSOR;
00197      1  5      WRITE (CHR(27),'(O)'); WRITE ('x '); WRITE (CHR(27),'(B)');
00198      1  5      READLN (FECHA); IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO
00199      1  4      UNTIL LENGTH (FECHA) < 10;
00200      1  5      REPEAT C: FILA:= I+4; COLUMNA:= 28; CURSOR;
00201      1  5      WRITE (CHR(27),'(O)'); WRITE (' x '); WRITE (CHR(27),'(B)');
00202      1  5      READLN (CONCEPTO); IF CONCEPTO = '' THEN BEGIN WRITE (CHR(7));
00203      1  4      UNTIL LENGTH (CONCEPTO) < 21;
00204      1  5      REPEAT D: FILA:= I+4; COLUMNA:= 55; CURSOR;
00205      1  5      WRITE (CHR(27),'(O)'); WRITE (' x '); WRITE (CHR(27),'(B)');
00206      1  5      READLN (IMPORTE); IF IMPORTE < 0 THEN BEGIN WRITE (CHR(7)); GOT
00207      1  4      UNTIL IMPORTE < 100000000; R: WRITE (CHR(27),'i22;8H');
00208      1  4      WRITE ('Confirme s no la operacisn ("S" s "N") : ');
00209      1  5      WRITE (CHR(27),'i0J'); READLN (CH); CASE CH OF
00210      1  5          'N','n': FILA:= FILA-1;
00211      1  6          'S','s': BEGIN EXTEND (FEUP); EXTEND (EUPIMPOR
00212      1  6      WRITELN (FEUP,NFACTURA); WRITELN (FEUP,FECHA);
00213      1  6      WRITELN (FEUP,CONCEPTO); WRITE (EUPIMPORTE,IMPORTE);
00214      1  6      RESET (EUPSALDO); WHILE NOT EOF (EUPSALDO) DO
00215      1  6          READ (EUPSALDO,SALDO); SALDO:= SALDO - IMPORTE;
00216      1  5      EXTEND (EUPSALDO); WRITE (EUPSALDO,SALDO); END;
00217      1  5          OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
00218      1  3          END; END;
00219      1  3      CLOSE (FEUP); CLOSE (EUPIMPORTE); CLOSE (EUPSALDO);
00220      1  2      EUP; END;
```

-LINE-IDC-FL-SL-

```

00221  C  1  2  (****          Consulta de la cuenta          *****)
00222  1  3          'C','c': BEGIN ENCABEZAMIENTO;
00223  1  3  OPEN (FEUP,'FEUP.DAT',HISTORY:=OLD);      RESET (FEUP);
00224  1  3  OPEN (EUPIMPORTE,'EUPIMPORTE.DAT',HISTORY:=OLD); RESET (EUPIMP
00225  1  3  OPEN (EUPSALDO,'EUPSALDO.DAT',HISTORY:=OLD); RESET (EUPSALDO);
00226  1  3  J:=5; M:=1; W:=1; WRITE (CHR(27),'i5;1H');
00227  1  4  WHILE NOT EOF (FEUP) DO BEGIN
00228  1  5      WHILE NOT EOF (EUPIMPORTE) DO BEGIN
00229  1  6          WHILE NOT EOF (EUPSALDO) DO BEGIN
00230  1  6              WRITE (CHR(27),'(O)'); WRITE ('x'); WRITE (CHR(27),'(B)');
00231  1  6                  FILA:=J; COLUMNAS:=0; CURSOR;
00232  1  6                      WRITE (M);
00233  1  6                          FILA:=J; COLUMNAS:=0; CURSOR;
00234  1  6                              WRITE (CHR(27),'i7D','nzm.',CHR(27),'i6C');
00235  1  6                                  READLN (FEUP,S);          WRITE (S,CHR(27),'i25D')
00236  1  6                                      FILA:=J; COLUMNAS:=0; CURSOR;
00237  1  6                                          READLN (FEUP,S);
00238  1  6                                              WRITE (CHR(27),'i25C',S);
00239  1  6                                                  WRITE (CHR(27),'i36D');
00240  1  6                                                      FILA:=J; COLUMNAS:=0; CURSOR;
00241  1  6                                                          READLN (FEUP,S);
00242  1  6                                                              WRITE (CHR(27),'i36C',S);
00243  1  6                                                                  WRITE (CHR(27),'i57D');
00244  1  6                                                                      FILA:=J; COLUMNAS:=0; CURSOR;
00245  1  6                                                                          WRITE (CHR(27),'i57C');
00246  1  6                                                                              READ (EUPIMPORTE,IMPORTE);
00247  1  6                                                                                  WRITE (IMPORTE,CHR(27),'i68D');
00248  1  6                                                                                          FILA:=J; COLUMNAS:=0; CURSOR;
00249  1  6                                                                                              READ (EUPSALDO,SALDO);
00250  1  6                                                                                                  WRITE (CHR(27),'i68C',SALDO);
00251  1  6                                                                                                      WRITE (CHR(27),'(O)'); WRITELN ('x'); WRITE (CHR(27),'(B)');
00252  1  6                                                                                                          J:=J+1; M:=M+1;
00253  1  7  IF M = 17*W THEN BEGIN
00254  1  7      WRITE (CHR(27),'(O','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
00255  1  7      WRITELN ('j'); WRITE (CHR(27),'(B)');
00256  1  7          WRITE ('Pzise <RETURN> para continuar
00257  1  7              WRITE (CHR(27),'i0J'); READ (CH); W:=W+1;
00258  1  7      WRITE (CHR(27),'i5;1H');          WRITE (CHR(27),'i0J');
00259  1  5  END;          END;          (* fin del WHILE ... IMPORTE *)
00260  1  4  END;          (* fin del WHILE ... SALDO *)
00261  1  3  END;
00262  1  3  WRITE (CHR(27),'(O','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
00263  1  3      WRITELN ('j'); WRITE (CHR(27),'(B)');
00264  1  3  WRITE ('Pzise <RETURN> para continuar ...');
00265  1  3  WRITE (CHR(27),'i0J'); READ (CH); BORRADO;
00266  1  3  CLOSE (FEUP); CLOSE (EUPIMPORTE);CLOSE (EUPSALDO);
00267  1  2  EUP;          END;          (* fin del BEGIN ... 'C' *)
00268  1  2      'H','h': BEGIN AYUDA; EUP; END;
00269  1  2      'X','x': BORRADO;
00270  1  2      OTHERWISE BEGIN WRITE (CHR(7)); GOTO 10; END;
00271  0  0  END; END;
00272  C  0  0          (*****
00273  C  0  0          (***** MATER *****
00274  C  0  0          (*****
00275  0  0

```

-LINE-IDC-PL-SL-

```
00276      1  0  PROCEDURE MATER;
00277      1  0  LABEL 10,11,AA,A,B,C,D,R;
00278      1  0  TYPE STRING= VARYING (300) OF CHAR;
00279      1  0  VAR FMATER: TEXT; MATERSALDO,MATERIMPORTE: FILE OF INTEGER;
00280      1  0      S,FECHA,NFACTURA,CONCEPTO: STRING;
00281      1  0      W,M,J,N,I,IMPORTE,SALDO: INTEGER;
00282      1  1  BEGIN
00283      1  1  COMENTARIO:= ' CUENTA DE MATERIAL INVENTARIABLE '; OPCUENTAS;
00284      1  2  10: OPCION; CASE CH OF
00285  C 1  2  (*****,      Inicializaci3rn de la cuenta      *****)
00286      1  3      'I','i': BEGIN
00287      1  3  11: BORRADO; WRITE (CHR(27),'i11:3H');
00288      1  3  WRITELN (
00289      1  3  'Pulse la "S" para confirmar la inicializaci3rn, "N" en caso cont
00290      1  3  WRITE (' ':2,'(recuerde que todas las operaciones son borradas):
00291      1  4  WRITE (CHR(27),'i0J'); READLN (CH); CASE CH OF
00292      1  5      'S','s': BEGIN ENCA;
00293  C 1  5  (*****          introduccion de datos          *****)
00294      1  5  INTRODATOS;
00295      1  6  REPEAT
00296      1  6  AA: WRITE (CHR(27),'i5:20H'); READLN (FECHA);
00297      1  6      IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO AA;
00298      1  5  UNTIL LENGTH (FECHA) < 10;
00299      1  5  WRITE (CHR(27),'(0)'); WRITE (CHR(27),'i5:28H'); WRITE (' x
00300      1  5  WRITE (CHR(27),'(B)');
00301      1  5  CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada
00302      1  6  REPEAT
00303      1  6  WRITE (CHR(27),'i5:55H'); WRITE (CHR(27),'(0)'); WRITE ('
00304      1  6  READ (IMPORTE);
00305      1  5  UNTIL IMPORTE < 100000000;
00306      1  5  WRITE (CHR(27),'(B)'); SALDO:= IMPORTE;
00307  C 1  5  (*****          fin de la introduccion de los datos          **
00308      1  5  OPEN (FMATER,'FMATER.DAT'); OPEN (MATERSALDO,'MATERSALDO.DAT'
00309      1  5  OPEN (MATERIMPORTE,'MATERIMPORTE.DAT');
00310      1  5  REWRITE (FMATER);REWRITE (MATERIMPORTE);REWRITE (MATERSALDO);
00311      1  5  WRITELN (FMATER,NFACTURA);WRITELN (FMATER,FECHA);
00312      1  5  WRITELN (FMATER,CONCEPTO); WRITE (MATERIMPORTE,IMPORTE);
00313      1  5  WRITE (MATERSALDO,IMPORTE);
00314      1  4  CLOSE (FMATER); CLOSE (MATERIMPORTE); CLOSE (MATERSALDO); END
00315      1  4      'N','n': WRITELN;
00316      1  4      'H','h': BEGIN AYUDA; GOTO 11; END;
00317      1  3      OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END; END;
00318      1  2  MATER; END; (* Desde Begin del 'i','I': *)
00319  C 1  2  (*****          Agadir operaciones          *****)
00320      1  3      'A','a': BEGIN ENCA;
00321      1  3  OPEN (FEUP,'FEUP.DAT',HISTORY:= OLD);
00322      1  3  OPEN (FMATER,'FMATER.DAT',HISTORY:= OLD);
00323      1  3  OPEN (EUPIMPORTE,'EUPIMPORTE.DAT',HISTORY:= OLD);
00324      1  3  OPEN (MATERIMPORTE,'MATERIMPORTE.DAT',HISTORY:= OLD);
00325      1  3  OPEN (EUPSALDO,'EUPSALDO.DAT',HISTORY:= OLD);
00326      1  3  OPEN (MATERSALDO,'MATERSALDO.DAT',HISTORY:= OLD);
00327      1  4  REPEAT WRITE (CHR(27),'i22:8H');
00328      1  4  WRITE (' Introduzca el n3mero de operaciones: ');
00329      1  3  WRITE (CHR(27),'i0J'); READLN (N); UNTIL N < 16;
00330      1  5  FOR I:= 1 TO N DO BEGIN REPEAT
```

-LINE-IDC-PL-SL-

```
00331 1 5 A: FILA:= I+4; COLUMNA:= 10; CURSOR; READLN (NFACTURA);
00332 1 5 IF NFACTURA = '' THEN BEGIN WRITE (CHR(7)); GOTO A; END;
00333 1 5 UNTIL LENGTH (NFACTURA) < 10; REPEAT
00334 1 5 B: FILA:= I+4; COLUMNA:= 18; CURSOR;
00335 1 5 WRITE (CHR(27),'(O)'); WRITE ('x '); WRITE (CHR(27),'(B)');
00336 1 5 READLN (FECHA); IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO
00337 1 4 UNTIL LENGTH (FECHA) < 10;
00338 1 5 REPEAT C: FILA:= I+4; COLUMNA:= 28; CURSOR;
00339 1 5 WRITE (CHR(27),'(O)'); WRITE (' x '); WRITE (CHR(27),'(B)');
00340 1 5 READLN (CONCEPTO); IF CONCEPTO = '' THEN BEGIN WRITE (CHR(7));
00341 1 4 UNTIL LENGTH (CONCEPTO) < 21;
00342 1 5 REPEAT D: FILA:= I+4; COLUMNA:= 55; CURSOR;
00343 1 5 WRITE (CHR(27),'(O)'); WRITE (' x '); WRITE (CHR(27),'(B)');
00344 1 5 READLN (IMPORTE); IF IMPORTE < 0 THEN BEGIN WRITE (CHR(7)); GO
00345 1 4 UNTIL IMPORTE < 100000000; R: WRITE (CHR(27),'i22;8H');
00346 1 4 WRITE ('Confirme s no la operacisn ("S" s "N") : ');
00347 1 4 WRITE (CHR(27),'iOJ'); READLN (CH);
00348 1 5 CASE CH OF 'N','n': FILA:= FILA-1;
00349 1 6 'S','s': BEGIN
00350 1 6 EXTEND (FEUP); EXTEND (EUIIMPORTE);
00351 1 6 EXTEND (FMATER); EXTEND (MATERIMPORTE);
00352 1 6 WRITELN (FEUP,NFACTURA); WRITELN (FEUP,FECHA);
00353 1 6 WRITELN (FMATER,NFACTURA); WRITELN (FMATER,FECHA);
00354 1 6 WRITELN (FEUP,'Material Inventariable'); WRITE (EUIIMPORTE,IM
00355 1 6 WRITELN (FMATER,CONCEPTO); WRITE (MATERIMPORTE,IMPORTE);
00356 1 6 RESET (EUPSALDO); WHILE NOT EOF (EUPSALDO) DO
00357 1 6 READ (EUPSALDO,SALDO); SALDO:= SALDO - IMPORTE;
00358 1 6 EXTEND (EUPSALDO); WRITE (EUPSALDO,SALDO);
00359 1 6 RESET (MATERIALSALDO); WHILE NOT EOF (MATERIALSALDO) DO
00360 1 6 READ (MATERIALSALDO,SALDO); SALDO:= SALDO - IMPORTE;
00361 1 5 EXTEND (MATERIALSALDO); WRITE (MATERIALSALDO,SALDO); END;
00362 1 5 OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
00363 1 3 END; END; (* fin del FOR *)
00364 1 3 CLOSE (FEUP); CLOSE (EUIIMPORTE); CLOSE (EUPSALDO);
00365 1 3 CLOSE (FMATER); CLOSE (MATERIMPORTE); CLOSE (MATERIALSALDO);
00366 1 2 MATER; END;
00367 C 1 2 (***** Consulta de la cuenta *****)
00368 1 3 'C','c': BEGIN ENCABEZAMIENTO;
00369 1 3 OPEN (FMATER,'FMATER.DAT',HISTORY:=OLD); RESET (FMATER);
00370 1 3 OPEN (MATERIMPORTE,'MATERIMPORTE.DAT',HISTORY:=OLD); RESET (MA
00371 1 3 OPEN (MATERIALSALDO,'MATERIALSALDO.DAT',HISTORY:=OLD); RESET (MATER
00372 1 3 J:=5; M:=1; W:=1; WRITE (CHR(27),'i5;1H');
00373 1 4 WHILE NOT EOF (FMATER) DO BEGIN
00374 1 5 WHILE NOT EOF (MATERIMPORTE) DO BEGIN
00375 1 6 WHILE NOT EOF (MATERIALSALDO) DO BEGIN
00376 1 6 WRITE (CHR(27),'(O)'); WRITE ('x'); WRITE (CHR(27),'(B)');
00377 1 6 FILA:=J; COLUMNA:=0; CURSOR;
00378 1 6 WRITE (M);
00379 1 6 FILA:=J; COLUMNA:=0; CURSOR;
00380 1 6 WRITE (CHR(27),'i7D','nzm.',CHR(27),'i6C');
00381 1 6 READLN (FMATER,S); WRITE (S,CHR(27),'i25D');
00382 1 6 FILA:=J; COLUMNA:=0; CURSOR;
00383 1 6 READLN (FMATER,S);
00384 1 6 WRITE (CHR(27),'i25C',S);
00385 1 6 WRITE (CHR(27),'i36D');
```

--LINE--IDC--PL--SL--

```

00386      1  6      FILA:=J; COLUMNA:=0;          (* CONCEPTO *)
00387      1  6      CURSOR;
00388      1  6      READLN (FMATER,S);
00389      1  6      WRITE (CHR(27),'i36C',S);
00390      1  6      WRITE (CHR(27),'i58D');
00391      1  6      FILA:=J; COLUMNA:=0;          (* IMPORTE *)
00392      1  6      CURSOR;
00393      1  6      WRITE (CHR(27),'i58C');
00394      1  6      READ (MATERIMPORTE,IMPORTE);
00395      1  6      WRITE (IMPORTE);
00396      1  6      WRITE (CHR(27),'i68D');
00397      1  6      FILA:=J; COLUMNA:=0;          (* SALDO *)
00398      1  6      CURSOR;
00399      1  6      WRITE (CHR(27),'i68C');
00400      1  6      READ (MATERMATER,SALDO);
00401      1  6      WRITE (SALDO);
00402      1  6      WRITE (CHR(27),'(0)'); WRITELN (' x'); WRITE (CHR(27),'(B)');
00403      1  6      J:=J+1; M:=M+1;
00404      1  7      IF M = 17*W THEN BEGIN
00405      1  7      WRITE (CHR(27),'(0)',m'); FOR I:= 1 TO 78 DO WRITE ('q');
00406      1  7      WRITELN ('j'); WRITE (CHR(27),'(B)');
00407      1  7      WRITE ('PzIse <RETURN> para continuar
00408      1  7      WRITE (CHR(27),'i0J'); READ (CH); W:=W+1;
00409      1  7      WRITE (CHR(27),'i5;1H'); WRITE (CHR(27),'i0J');
00410      1  6      END;
00411      1  5      END; (* fin del WHILE ... IMPORTE *)
00412      1  4      END; (* fin del WHILE ... SALDO *)
00413      1  3      END; (* fin del WHILE NOT (FMATER) *)
00414      1  3      WRITE (CHR(27),'(0)',m'); FOR I:= 1 TO 78 DO WRITE ('q');
00415      1  3      WRITELN ('j'); WRITE (CHR(27),'(B)');
00416      1  3      WRITE ('PzIse <RETURN> para continuar ...');
00417      1  3      WRITE (CHR(27),'i0J'); READ (CH);
00418      1  3      BORRADO;
00419      1  3      CLOSE (FMATER);CLOSE (MATERIMPORTE);CLOSE (MATERMATER);
00420      1  2      MATER; END;
00421      C  1  2      (*****
00422      1  2      'H','h': BEGIN AYUDA; MATER; END;
00423      1  2      'X','x': BORRADO;
00424      1  2      OTHERWISE BEGIN WRITE (CHR(7));.GOTO 10; END;
00425      0  0      END; (* fin del begin del case primero *) END;
00426      C  0  0      (*****
00427      C  0  0      (***** CULTUR *****
00428      C  0  0      (*****
00429      1  0      PROCEDURE CULTUR;
00430      1  0      LABEL 10,11,AA,A,B,C,D,R;
00431      1  0      TYPE STRING= VARYING i300 OF CHAR;
00432      1  0      VAR FCULTUR: TEXT; CULTURMATER,CULTURIMPORTE: FILE OF INTEGER;
00433      1  0      S,FECHA,NFACTURA,CONCEPTO: STRING;
00434      1  0      M,J,N,I,IMPORTE,SALDO: INTEGER;
00435      1  1      BEGIN
00436      1  1      COMENTARIO:= ' CUENTA DE CULTURA GENERAL ';
00437      1  1      OPCUENTAS; 10: OPCION;
00438      1  2      CASE CH OF
00439      C  1  2      (***** Inicializaci3n de la cuenta *****
00440      1  3      'I','i': BEGIN

```

-LINE-IDC-PL-SL-

```

00441      1 3 11: BORRADO; WRITE (CHR(27),'i11;3H'); WRITELN (
00442      1 3 'Fylse la "S" para confirmar la inicializaciorn, "N" en caso cont
00443      1 3 WRITE (' ':2,'(recuerde que todas las operaciones son borradas):
00444      1 3 WRITE (CHR(27),'i0J'); READLN (CH);
00445      1 5     CASE CH OF 'S','s': BEGIN ENCA; INTRODATOS;
00446      1 6     REPEAT
00447      1 6 AA: WRITE (CHR(27),'i5;20H'); READLN (FECHA);
00448      1 6     IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO AA; END;
00449      1 5     UNTIL LENGTH (FECHA) < 10;
00450      1 5     WRITE (CHR(27),'(0)'); WRITE (CHR(27),'i5;28H'); WRITE (' x
00451      1 5     WRITE (CHR(27),'(B)');
00452      1 5     CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada
00453      1 6     REPEAT
00454      1 6     WRITE (CHR(27),'i5;55H'); WRITE (CHR(27),'(0)'); WRITE (' x
00455      1 5     READ (IMPORTE); UNTIL IMPORTE < 100000000;
00456      1 5     WRITE (CHR(27),'(B)'); SALDO:= IMPORTE;
00457      1 5     OPEN (FCULTUR,'FCULTUR.DAT'); OPEN (CULTURSALDO,'CULTURSALDO.D
00458      1 5     OPEN (CULTURIMPORTE,'CULTURIMPORTE.DAT');
00459      1 5     REWRITE (FCULTUR); REWRITE (CULTURIMPORTE); REWRITE (CULTURSAL
00460      1 5     WRITELN (FCULTUR,NFACTURA); WRITELN (FCULTUR,FECHA);
00461      1 5     WRITELN (FCULTUR,CONCEPTO); WRITE (CULTURIMPORTE,IMPORTE);
00462      1 5     SALDO:= IMPORTE; WRITE (CULTURSALDO,IMPORTE);
00463      1 4     CLOSE (FCULTUR); CLOSE (CULTURIMPORTE); CLOSE (CULTURSALDO); E
00464      1 4     'N','n': BEGIN WRITELN; END;
00465      1 4     'H','h': BEGIN AYUDA; GOTO 11; END;
00466      1 3     OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END; END;
00467      1 2 CULTUR; END; (* Desde Begin del 'i','I': *)
00468      C 1 2 (***** Agadir operaciones *****
00469      1 3     'A','a': BEGIN ENCA;
00470      1 3     OPEN (FCULTUR,'FCULTUR.DAT',HISTORY:= OLD);
00471      1 3     OPEN (CULTURIMPORTE,'CULTURIMPORTE.DAT',HIST
00472      1 3     OPEN (CULTURSALDO,'CULTURSALDO.DAT',HISTORY:
00473      1 4 REPEAT WRITE (CHR(27),'i22;8H');
00474      1 4 WRITE ('Introduzca el nzmero de operaciones: ');
00475      1 4 WRITE (CHR(27),'i0J'); READLN (N);
00476      1 3 UNTIL N < 16;
00477      1 4 FOR I:= 1 TO N DO BEGIN
00478      1 5 REPEAT
00479      1 5 A: FILA:= I+4; COLUMNA:= 10; CURSOR; READLN (NFACTURA);
00480      1 5     IF NFACTURA = '' THEN BEGIN WRITE (CHR(7)); GOTO A; END;
00481      1 4 UNTIL LENGTH (NFACTURA) < 10;
00482      1 5 REPEAT B: FILA:= I+4; COLUMNA:= 18; CURSOR;
00483      1 5 WRITE (CHR(27),'(0)'); WRITE ('x '); WRITE (CHR(27),'(B)'); READL
00484      1 5     IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO B; END;
00485      1 4 UNTIL LENGTH (FECHA) < 10;
00486      1 5 REPEAT C: FILA:= I+4; COLUMNA:= 28; CURSOR;
00487      1 5 WRITE (CHR(27),'(0)'); WRITE (' x '); WRITE (CHR(27),'(B)');
00488      1 5     READLN (CONCEPTO);
00489      1 5     IF CONCEPTO = '' THEN BEGIN WRITE (CHR(7)); GOTO C; END;
00490      1 4 UNTIL LENGTH (CONCEPTO) < 21;
00491      1 5 REPEAT D: FILA:= I+4; COLUMNA:= 55; CURSOR;
00492      1 5 WRITE (CHR(27),'(0)'); WRITE (' x '); WRITE (CHR(27),'(B)');
00493      1 5     READLN (IMPORTE);
00494      1 5     IF IMPORTE < 0 THEN BEGIN WRITE (CHR(7)); GOTO D; END;
00495      1 4 UNTIL IMPORTE < 100000000;

```

ARCONT
01

Source Listing

11-Feb-1987 10:1
11-Feb-1987 10:1

-LINE-IDC-FL-SL-

```

00496      1  4 R: WRITE (CHR(27), 'i22;8H');
00497      1  4 WRITE ('Confirme s. no la operacion ("S" s "N") : ');
00498      1  4 WRITE (CHR(27), 'i0J?'); READLN (CH);
00499      1  5 CASE CH OF   'N', 'n': FILA:= FILA-1;
00500      1  6           'S', 's':           BEGIN
00501      1  6     EXTEND (FCULTUR);   EXTEND (CULTURIMPORTE);
00502      1  6     WRITELN (FCULTUR, NFACTURA);   WRITELN (FCULTUR, FECHA);
00503      1  6     WRITELN (FCULTUR, CONCEPTO);   WRITE (CULTURIMPORTE, IMPORTE);
00504      1  6     RESET (CULTURSALDO);   WHILE NOT EOF (CULTURSALDO) DO
00505      1  6       READ (CULTURSALDO, SALDO);   SALDO:= SALDO - IMPORTE;
00506      1  5     EXTEND (CULTURSALDO);   WRITE (CULTURSALDO, SALDO);   END;
00507      1  5     OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
00508      1  4     END;   (* fin del case ch of *)
00509      1  3     END;   (* fin del FOR *)
00510      1  3 CLOSE (FCULTUR); CLOSE (CULTURIMPORTE); CLOSE (CULTURSALDO);
00511      1  2 CULTUR;           END;
00512      C 1  2 (*****   Consulta de la cuenta   *****)
00513      1  3     'C', 'c':   BEGIN ENCABEZAMIENTO;
00514      1  3     OPEN (FCULTUR, 'FCULTUR.DAT', HISTORY:=OLD); RESET (FCULTUR);
00515      1  3     OPEN (CULTURIMPORTE, 'CULTURIMPORTE.DAT', HISTORY:=OLD); RESET (CULTURIMPORTE);
00516      1  3     OPEN (CULTURSALDO, 'CULTURSALDO.DAT', HISTORY:=OLD); RESET (CULTURSALDO);
00517      1  3     J:=5; M:=1; W:=1;
00518      1  3 WRITE (CHR(27), 'i5;1H');
00519      1  4 WHILE NOT EOF (FCULTUR) DO BEGIN
00520      1  5     WHILE NOT EOF (CULTURIMPORTE) DO BEGIN
00521      1  6     WHILE NOT EOF (CULTURSALDO) DO BEGIN
00522      1  6 WRITE (CHR(27), ' (O)'); WRITE (' x'); WRITE (CHR(27), ' (B)');
00523      1  6 FILA:=J; COLUMNA:=0; CURSOR;
00524      1  6     WRITE (M);
00525      1  6 FILA:=J; COLUMNA:=0; CURSOR;
00526      1  6     WRITE (CHR(27), 'i7D', 'nzm.', CHR(27), 'i6C');
00527      1  6 READLN (FCULTUR, S);
00528      1  6     WRITE (S, CHR(27), 'i25D');
00529      1  6 FILA:=J; COLUMNA:=0; CURSOR;
00530      1  6     READLN (FCULTUR, S);
00531      1  6     WRITE (CHR(27), 'i25C', S);
00532      1  6     WRITE (CHR(27), 'i36D');
00533      1  6     FILA:=J; COLUMNA:=0;           (* CONCEPTO *)
00534      1  6     CURSOR;
00535      1  6     READLN (FCULTUR, S);
00536      1  6     WRITE (CHR(27), 'i36C', S);
00537      1  6     WRITE (CHR(27), 'i58D');
00538      1  6     FILA:=J; COLUMNA:=0;           (* IMPORTE *)
00539      1  6     CURSOR;
00540      1  6     WRITE (CHR(27), 'i58C');
00541      1  6     READ (CULTURIMPORTE, IMPORTE);
00542      1  6     WRITE (IMPORTE, CHR(27), 'i68D');
00543      1  6     FILA:=J; COLUMNA:=0;           (* SALDO *)
00544      1  6     CURSOR;
00545      1  6     READ (CULTURSALDO, SALDO);
00546      1  6     WRITE (CHR(27), 'i68C', SALDO);
00547      1  6 WRITE (CHR(27), ' (O)'); WRITELN (' x'); WRITE (CHR(27), ' (B)');
00548      1  6     J:=J+1;           M:=M+1;
00549      1  7     IF M = 17*M THEN BEGIN
00550      1  7     WRITE (CHR(27), ' (O)', 'm'); FOR I:= 1 TO 78 DO WRITE ('q');

```

ARCONT
01

Source Listing

-LINE-IDC-PL-SL-

```

00551 1 7 WRITELN ('j'); WRITE (CHR(27),'(B)');
00552 1 7 WRITE ('Pzlse <RETURN> para continuar
00553 1 7 WRITE (CHR(27),'iOJ'); READ (CH); W:=W+1;
00554 1 7 WRITE (CHR(27),'i5;1H'); WRITE (CHR(27),'iOJ');
00555 1 5 END; END; (* fin del WHILE .. *)
00556 1 4 END; (* fin del WHILE . *)
00557 1 3 END; (* fin del WHILE NOT (FCULTUR) *)
00558 1 3 WRITE (CHR(27),'(O','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
00559 1 3 WRITELN ('j'); WRITE (CHR(27),'(B)');
00560 1 3 WRITE ('Pzlse <RETURN> para continuar ...');
00561 1 3 WRITE (CHR(27),'iOJ'); READ (CH); BORRADO;
00562 1 3 CLOSE (FCULTUR);CLOSE (CULTURIMPORTE);CLOSE (CULTURSALDO);
00563 1 2 CULTUR; END;
00564 C 1 2 (***** *****)
00565 1 2 'H','h': BEGIN AYUDA; CULTUR; END;
00566 1 2 'X','x': BORRADO;
00567 1 2 OTHERWISE BEGIN WRITE (CHR(7)); GOTO 10; END;
00568 0 0 END; (* fin del begin del case primero *) END;
00569 C 0 0 (*****
00570 C 0 0 (***** EDUCA *****
00571 C 0 0 (*****
00572 1 0 PROCEDURE EDUCA;
00573 1 0 LABEL 10,11,AA,A,B,C,D,R;
00574 1 0 TYPE STRING= VARYING (300) OF CHAR;
00575 1 0 VAR FEDUCA: TEXT; EDUCASALDO,EDUCAIMPORTE: FILE OF INTEGER;
00576 1 0 S,FECHA,NFACTURA,CONCEPTO: STRING; M,J,N,I,IMPORTE,SALDO: IN
00577 1 1 BEGIN
00578 1 1 COMENTARIO:= ' CUENTA DE EDUCACION FISICA Y DEPORTES ';
00579 1 2 OPCUENTAS; 10: OPCION; CASE CH OF
00580 C 1 2 (*****, Inicializacin de la cuenta *****
00581 1 3 'I','i': BEGIN 11: BORRADO;
00582 1 3 WRITE (CHR(27),'i11;3H'); WRITELN (
00583 1 3 'Pzlse la "S" para confirmar la inicializacin, "N" en caso cont
00584 1 3 WRITE (' ':2,'(recuerde que todas las operaciones son borradas):
00585 1 4 WRITE (CHR(27),'iOJ'); READLN (CH); CASE CH OF
00586 1 5 'S','s': BEGIN ENCA; INTRODATOS;
00587 1 6 REPEAT AA: WRITE (CHR(27),'i5;20H'); READLN (FECHA);
00588 1 6 IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO AA; END;
00589 1 5 UNTIL LENGTH (FECHA) < 10; WRITE (CHR(27),'(O)');
00590 1 5 WRITE (CHR(27),'i5;28H'); WRITE (' x '); WRITE (CHR(27),'(
00591 1 5 CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada
00592 1 6 REPEAT
00593 1 6 WRITE (CHR(27),'i5;55H'); WRITE (CHR(27),'(O)');
00594 1 6 WRITE (' x '); READ (IMPORTE);
00595 1 5 UNTIL IMPORTE < 10000000; WRITE (CHR(27),'(B)'); SALDO:= IMPOR
00596 1 5 OPEN (FEDUCA,'FEDUCA.DAT'); OPEN (EDUCASALDO,'EDUCASALDO.DAT'
00597 1 5 OPEN (EDUCAIMPORTE,'EDUCAIMPORTE.DAT');
00598 1 5 REWRITE (FEDUCA); REWRITE (EDUCAIMPORTE);
00599 1 5 REWRITE (EDUCASALDO);WRITELN (FEDUCA,NFACTURA);
00600 1 5 WRITELN (FEDUCA,FECHA); WRITELN (FEDUCA,CONCEPTO);
00601 1 5 WRITE (EDUCAIMPORTE,IMPORTE);WRITE (EDUCASALDO,IMPORTE);
00602 1 4 CLOSE (FEDUCA); CLOSE (EDUCAIMPORTE); CLOSE (EDUCASALDO); END;
00603 1 4 'N','n': BEGIN WRITELN; END;
00604 1 4 'H','h': BEGIN AYUDA; GOTO 11; END;
00605 1 3 OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END; END;

```

-LINE-IDC-PL-SL-

```

00606      1  2  EDUCA;          END; (* Desde      Begin del  'i','I':  *)
00607      C  1  2  (*****      Agadir operaciones      *****)
00608      1  3      'A','a': BEGIN ENCA;
00609      1  3                      OPEN (FEDUCA,'FEDUCA.DAT',HISTORY:= OLD);
00610      1  3                      OPEN (EDUCAIMPORTE,'EDUCAIMPORTE.DAT',HISTOR
00611      1  3                      OPEN (EDUCASALDO,'EDUCASALDO.DAT',HISTORY:=
00612      1  3
00613      1  4 REPEAT WRITE (CHR(27),'i22;8H');
00614      1  4 WRITE ('Introduzca el nzmero de operaciones: ');
00615      1  4 WRITE (CHR(27),'iOJ'); READLN (N);
00616      1  3 UNTIL N < 16;
00617      1  4 FOR I:= 1 TO N DO BEGIN
00618      1  5 REPEAT A: FILA:= I+4; COLUMNA:= 10; CURSOR; READLN (NFACTURA);
00619      1  5       IF NFACTURA = '' THEN BEGIN WRITE (CHR(7)); GOTO A; END;
00620      1  4 UNTIL LENGTH (NFACTURA) < 10;
00621      1  5 REPEAT B: FILA:= I+4; COLUMNA:= 18; CURSOR;
00622      1  5 WRITE (CHR(27),'O'); WRITE (' x '); WRITE (CHR(27),'(B)');
00623      1  5 READLN (FECHA);
00624      1  5       IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO B; END;
00625      1  4 UNTIL LENGTH (FECHA) < 10;
00626      1  5 REPEAT C: FILA:= I+4; COLUMNA:= 28; CURSOR;
00627      1  5 WRITE (CHR(27),'O');WRITE (' x ');WRITE (CHR(27),'(B)');
00628      1  5 READLN (CONCEPTO);
00629      1  5       IF CONCEPTO = '' THEN BEGIN WRITE (CHR(7)); GOTO C; END;
00630      1  4 UNTIL LENGTH (CONCEPTO) < 21;
00631      1  5 REPEAT D: FILA:= I+4; COLUMNA:= 55; CURSOR;
00632      1  5 WRITE (CHR(27),'O');WRITE (' x ');
00633      1  5 WRITE (CHR(27),'(B)'); READLN (IMPORTE);
00634      1  5       IF IMPORTE < 0 THEN BEGIN WRITE (CHR(7)); GOTO D; END;
00635      1  4 UNTIL IMPORTE < 100000000;
00636      1  4 R: WRITE (CHR(27),'i22;8H');
00637      1  4 WRITE ('Confirme s no la operacisn ("S" s "N") : ');
00638      1  4 WRITE (CHR(27),'iOJ');READLN (CH);
00639      1  5 CASE CH OF
00640      1  6           'N','n': FILA:= FILA-1;
00641      1  6           'S','s': BEGIN
00642      1  6 EXTEND (FEDUCA); EXTEND (EDUCAIMPORTE);
00643      1  6 WRITELN (FEDUCA,NFACTURA); WRITELN (FEDUCA,FECHA);
00644      1  6 WRITELN (FEDUCA,CONCEPTO); WRITE (EDUCAIMPORTE,IMPORTE);
00645      1  6 RESET (EDUCASALDO); WHILE NOT EOF (EDUCASALDO) DO
00646      1  6 READ (EDUCASALDO,SALDO); SALDO:= SALDO - IMPORTE;
00647      1  5 EXTEND (EDUCASALDO); WRITE (EDUCASALDO,SALDO); END;
00648      1  5 OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
00649      1  4 END; (* fin del case ch of *)
00650      1  3 END; (* fin del FOR *)
00651      1  3 CLOSE (FEDUCA); CLOSE (EDUCAIMPORTE); CLOSE (EDUCASALDO);
00652      C  1  2  (*****      Consulta de la cuenta      *****)
00653      1  2      'C','c':
00654      1  3      BEGIN
00655      1  3      ENCABEZAMIENTO;
00656      1  3      OPEN (FEDUCA,'FEDUCA.DAT',HISTORY:=OLD);
00657      1  3      RESET (FEDUCA);
00658      1  3      OPEN (EDUCAIMPORTE,'EDUCAIMPORTE.DAT',HISTOR
00659      1  3      RESET (EDUCAIMPORTE);
00660      1  3      OPEN (EDUCASALDO,'EDUCASALDO.DAT',HISTORY:=O

```

© Del documento, los autores. Digitalización realizada por U.L.P.C. Biblioteca Universitaria. 2008

-LINE-IDC-PL-SL-

```
00661      1  3          .RESET (EDUCASALDO);
00662      1  3          J:=5; M:=1; W:=1;
00663      1  3 WRITE (CHR(27),'i5;1H');
00664      1  4 WHILE NOT EOF (FEDUCA) DO BEGIN
00665      1  5          WHILE NOT EOF (EDUCAIMPORTE) DO BEGIN
00666      1  6              WHILE NOT EOF (EDUCASALDO) DO BEGIN
00667      1  6                  WRITE (CHR(27),'(O)'); WRITE ('x'); WRITE (CHR(27),'(B)');
00668      1  6                      FILA:=J; COLUMNA:=0; CURSOR;
00669      1  6                          WRITE (M);
00670      1  6                              FILA:=J; COLUMNA:=0;          (* NUM.FACTURA *)
00671      1  6                                  CURSOR;
00672      1  6                                      WRITE (CHR(27),'i7D','nzm.',CHR(27),'i6C');
00673      1  6                                          .
00674      1  6 READLN (FEDUCA,S);
00675      1  6 WRITE (S,CHR(27),'i25D');
00676      1  6 FILA:=J; COLUMNA:=0;
00677      1  6 CURSOR;          (* FECHA *)
00678      1  6 READLN (FEDUCA,S);
00679      1  6 WRITE (CHR(27),'i25C',S);
00680      1  6 WRITE (CHR(27),'i36D');
00681      1  6 FILA:=J; COLUMNA:=0;          (* CONCEPTO *)
00682      1  6 CURSOR;
00683      1  6 READLN (FEDUCA,S);
00684      1  6 WRITE (CHR(27),'i36C',S);
00685      1  6 WRITE (CHR(27),'i58D');
00686      1  6 FILA:=J; COLUMNA:=0;          (* IMPORTE *)
00687      1  6 CURSOR;
00688      1  6 WRITE (CHR(27),'i58C');
00689      1  6 READ (EDUCAIMPORTE,IMPORTE);
00690      1  6 WRITE (IMPORTE,CHR(27),'i68D');
00691      1  6 FILA:=J; COLUMNA:=0;          (* SALDO *)
00692      1  6 CURSOR;
00693      1  6 READ (EDUCASALDO,SALDO);
00694      1  6 WRITE (CHR(27),'i68C',SALDO);
00695      1  6 WRITE (CHR(27),'(O)'); WRITELN ('x'); WRITE (CHR(27),'(B)');
00696      1  6 J:=J+1; M:=M+1;
00697      1  7 IF M = 17*W THEN BEGIN
00698      1  7     WRITE (CHR(27),'(O','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
00699      1  7     WRITELN ('j'); WRITE (CHR(27),'(B)');
00700      1  7     WRITE ('Pzlse <RETURN> para continuar
00701      1  7     WRITE (CHR(27),'i0J'); READ (CH); W:=W+1;
00702      1  7     WRITE (CHR(27),'i5;1H'); WRITE (CHR(27),'i0J');
00703      1  6     END;
00704      1  5     END;          (* fin del WHILE .. *)
00705      1  4     END;          (* fin del WHILE ... *)
00706      1  3 END;          (* fin del WHILE NOT (FEDUCA) *)
00707      1  3 WRITE (CHR(27),'(O','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
00708      1  3 WRITELN ('j'); WRITE (CHR(27),'(B)');
00709      1  3 WRITE ('Pzlse <RETURN> para continuar ...');
00710      1  3 WRITE (CHR(27),'i0J'); READ (CH); BORRADO;
00711      1  3 CLOSE (FEDUCA); CLOSE (EDUCAIMPORTE); CLOSE (EDUCASALDO);
00712      1  2 EDUCA;
00713      C 1  2 (*****          *****)
00714      1  2 'H','h': BEGIN AYUDA; EDUCA; END;
00715      1  2 'X','x': BORRADO;
```

--LINE-IDC-PL-SL--

```

00716      1  2      OTHERWISE BEGIN WRITE (CHR(7)); GOTO 10; END;
00717      0  0 END; (*      fin del begin del case primero *) END;
00718      C  0  0      .(*****
00719      C  0  0      (***** EXTEN *****
00720      C  0  0      (*****
00721      1  0 PROCEDURE EXTEN;
00722      1  0      LABEL 10,11,AA,A,B,C,D,R;
00723      1  0 TYPE STRING= VARYING (300) OF CHAR;
00724      1  0 VAR FEXTEN: TEXT;      EXTENSALDO,EXTENIMPORTE: FILE OF INTEGER;
00725      1  0      S,FECHA,NFACTURA,CONCEPTO: STRING;
00726      1  0      M,J,N,I,IMPORTE,SALDO: INTEGER;
00727      1  1 BEGIN
00728      1  1 COMENTARIO:= ' CUENTA DE EXTENSION UNIVERSITARIA ';
00729      1  1 OPCUENTAS;      10: OPCION;
00730      1  2 CASE CH OF
00731      C  1  2 (*****,      Inicializaciorn de la cuenta *****
00732      1  3      'I','i': BEGIN 11: BORRADO;
00733      1  3 WRITE (CHR(27),'i11;3H');
00734      1  3 WRITELN (
00735      1  3 'Pulse la "S" para confirmar la inicializaciorn, "N" en caso cont
00736      1  3 WRITE (' ':2,'(recuerde que todas las operaciones son borradas):
00737      1  3 WRITE (CHR(27),'iOJ');
00738      1  4 READLN (CH);      CASE CH OF
00739      1  5      'S','s': BEGIN
00740      1  5      ENCA;      INTRODADOS;
00741      1  6 REPEAT AA; WRITE (CHR(27),'i5;20H');
00742      1  6 READLN (FECHA); IF FECHA = '' THEN BEGIN WRITE (CHR(7));GOTO A
00743      1  5 UNTIL LENGTH (FECHA) < 10;
00744      1  5 WRITE (CHR(27),'(O)'; WRITE (CHR(27),'i5;28H');
00745      1  5 WRITE (' x '); WRITE (CHR(27),'(B)';
00746      1  5 CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada '
00747      1  6 REPEAT WRITE (CHR(27),'i5;55H');
00748      1  6 WRITE (CHR(27),'(O)'; WRITE (' x '); READ (IMPORTE);
00749      1  5 UNTIL IMPORTE < 100000000;
00750      1  5 WRITE (CHR(27),'(B)'; SALDO:= IMPORTE;
00751      1  5 OPEN (FEXTEN,'FEXTEN.DAT'); OPEN (EXTENSALDO,'EXTENSALDO.DAT')
00752      1  5 OPEN (EXTENIMPORTE,'EXTENIMPORTE.DAT');
00753      1  5 REWRITE (FEXTEN); REWRITE (EXTENIMPORTE); REWRITE (EXTENSALDO)
00754      1  5 WRITELN (FEXTEN,NFACTURA); WRITELN (FEXTEN,FECHA);
00755      1  5 WRITELN (FEXTEN,CONCEPTO); WRITE (EXTENIMPORTE,IMPORTE);
00756      1  5 WRITE (EXTENSALDO,IMPORTE);
00757      1  4 CLOSE (FEXTEN); CLOSE (EXTENIMPORTE); CLOSE (EXTENSALDO); END;
00758      1  4      'N','n': BEGIN WRITELN; END;
00759      1  4      'H','h': BEGIN AYUDA; GOTO 11; END;
00760      1  4      OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END;
00761      1  3      END;      EXTEN;
00762      1  2      END; (* Desde Begin del 'i','I': *)
00763      C  1  2 (*****      Agadir operaciones *****
00764      1  3      'A','a': BEGIN ENCA;
00765      1  3      OPEN (FEXTEN,'FEXTEN.DAT',HISTORY:= OLD);
00766      1  3      OPEN (EXTENIMPORTE,'EXTENIMPORTE.DAT',HISTOR
00767      1  3      OPEN (EXTENSALDO,'EXTENSALDO.DAT',HISTORY:=
00768      1  4 REPEAT WRITE (CHR(27),'i22;8H');
00769      1  4 WRITE (' Introduzca el nzmero de operaciones: ');
00770      1  4 WRITE (CHR (27),'iOJ'); READLN (N);

```

© Del documento, los autores. Digitalización realizada por ULPGC. Biblioteca Universitaria, 2008

-LINE-IDC-PL-SL-

```
00771      1 3 UNTIL N < 16;
00772      1 4 FOR I:= 1 TO N DO          BEGIN
00773      1 5 REPEAT A: FILA:= I+4; COLUMNA:= 10;
00774      1 5 CURSOR; READLN (NFACTURA);
00775      1 5     IF NFACTURA = '?' THEN BEGIN WRITE (CHR(7)); GOTO A; END;
00776      1 4 UNTIL LENGTH (NFACTURA) < 10;
00777      1 5 REPEAT B: FILA:= I+4; COLUMNA:= 18;
00778      1 5 CURSOR; WRITE (CHR(27),'(O)');
00779      1 5 WRITE ('x '); WRITE (CHR(27),'(B)');
00780      1 5 READLN (FECHA); IF FECHA = '?' THEN BEGIN WRITE (CHR(7)); GOTO
00781      1 4 UNTIL LENGTH (FECHA) < 10;
00782      1 5 REPEAT C: FILA:= I+4; COLUMNA:= 28;
00783      1 5 CURSOR; WRITE (CHR(27),'(O)');
00784      1 5 WRITE (' x '); WRITE (CHR(27),'(B)');
00785      1 5 READLN (CONCEPTO);
00786      1 5     IF CONCEPTO = '?' THEN BEGIN WRITE (CHR(7)); GOTO C; END;
00787      1 4 UNTIL LENGTH (CONCEPTO) < 25;
00788      1 5 REPEAT D: FILA:= I+4; COLUMNA:= 55; CURSOR;
00789      1 5 WRITE (CHR(27),'(O)'); WRITE (' x '); WRITE (CHR(27),'(B)');
00790      1 5 READLN (IMPORTE);
00791      1 5     IF IMPORTE < 0 THEN BEGIN WRITE (CHR(7)); GOTO D; END;
00792      1 4 UNTIL IMPORTE < 100000000;
00793      1 4 R: WRITE (CHR(27),'i22;8H');
00794      1 4 WRITE ('Confirme s no la operacion ("S" s "N") : ');
00795      1 4 WRITE (CHR(27),'iOJ'); READLN (CH);
00796      1 5 CASE CH OF
00797      1 6           'S','s': BEGIN
00798      1 6     EXTEND (FEXTEN); EXTEND (EXTENIMPORTE);
00799      1 6     WRITELN (FEXTEN,NFACTURA); WRITELN (FEXTEN,FECHA);
00800      1 6     WRITELN (FEXTEN,CONCEPTO); WRITE (EXTENIMPORTE,IMPORTE);
00801      1 6     RESET (EXTENSALDO); WHILE NOT EOF (EXTENSALDO) DO
00802      1 6     READ (EXTENSALDO,SALDO); SALDO:= SALDO - IMPORTE;
00803      1 5     EXTEND (EXTENSALDO); WRITE (EXTENSALDO,SALDO); END;
00804      1 5     OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
00805      1 4     END; (* fin del case ch of *)
00806      1 3     END; (* fin del FOR *)
00807      1 3 CLOSE (FEXTEN); CLOSE (EXTENIMPORTE); CLOSE (EXTENSALDO);
00808      1 2 EXTEN; END;
00809      C 1 2 (***** Consulta de la cuenta *****)
00810      1 3 'C','c': BEGIN ENCABEZAMIENTO;
00811      1 3 OPEN (FEXTEN,'FEXTEN.DAT',HISTORY:=OLD); RESET (FEXTEN);
00812      1 3 OPEN (EXTENIMPORTE,'EXTENIMPORTE.DAT',HISTORY:=OLD); RESET (EXT
00813      1 3 OPEN (EXTENSALDO,'EXTENSALDO.DAT',HISTORY:=OLD); RESET (EXTENSA
00814      1 3 J:=5; M:=1; W:=1;
00815      1 3 WRITE (CHR(27),'i5;1H');
00816      1 4 WHILE NOT EOF (FEXTEN) DO BEGIN
00817      1 5     WHILE NOT EOF (EXTENIMPORTE) DO BEGIN
00818      1 6     WHILE NOT EOF (EXTENSALDO) DO BEGIN
00819      1 6     WRITE (CHR(27),'(O)'); WRITE ('x'); WRITE (CHR(27),'(B)');
00820      1 6     FILA:=J; COLUMNA:=0; CURSOR;
00821      1 6     WRITE (M);
00822      1 6     FILA:=J; COLUMNA:=0; (* NUM.FACTURA *)
00823      1 6     CURSOR;
00824      1 6     WRITE (CHR(27),'i7D','nzm.',CHR(27),'i6C');
00825      1 6     READLN (FEXTEN,S);
```

-LINE-IDC-PL-SL-

```

00826      1  6      WRITE (S,CHR(27),'i25D');
00827      1  6      FILA:=J; COLUMNA:=0;
00828      1  6      CURSOR; (* FECHA *)
00829      1  6      READLN (FEXTEN,S);
00830      1  6      WRITE (CHR(27),'i25C',S);
00831      1  6      WRITE (CHR(27),'i36D');
00832      1  6      FILA:=J; COLUMNA:=0; (* CONCEPTO *)
00833      1  6      CURSOR;
00834      1  6      READLN (FEXTEN,S);
00835      1  6      WRITE (CHR(27),'i36C',S);
00836      1  6      WRITE (CHR(27),'i58D');
00837      1  6      FILA:=J; COLUMNA:=0; (* IMPORTE *)
00838      1  6      CURSOR;
00839      1  6      WRITE (CHR(27),'i58C');
00840      1  6      READ (EXTENIMPORTE,IMPORTE);
00841      1  6      WRITE (IMPORTE,CHR(27),'i68D');
00842      1  6      FILA:=J; COLUMNA:=0; (* SALDO *)
00843      1  6      CURSOR;
00844      1  6      READ (EXTENSALDO,SALDO);
00845      1  6      WRITE (CHR(27),'i68C',SALDO);
00846      1  6      WRITE (CHR(27),'(O)'); WRITELN (' x'); WRITE (CHR(27),'(B)');
00847      1  6      J:=J+1; M:=M+1;
00848      1  7      IF M = 17*W THEN BEGIN
00849      1  7      WRITE (CHR(27),'(O','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
00850      1  7      WRITELN ('j'); WRITE (CHR(27),'(B)');
00851      1  7      WRITE ('Pzlse <RETURN> para continuar
00852      1  7      WRITE (CHR(27),'iOJ'); READ (CH); W:=W+1;
00853      1  7      WRITE (CHR(27),'i5;1H'); WRITE (CHR(27),'iOJ')
00854      1  6      END;
00855      1  5      END; (* fin del WHILE ... *)
00856      1  4      END; (* fin del WHILE ... *)
00857      1  3      END; (* fin del WHILE NOT (FEXTEN) *)
00858      1  3      WRITE (CHR(27),'(O','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
00859      1  3      WRITELN ('j'); WRITE (CHR(27),'(B)');
00860      1  3      WRITE ('Pzlse <RETURN> para continuar ...');
00861      1  3      WRITE (CHR(27),'iOJ'); READ (CH); BORRADO;
00862      1  3      CLOSE (FEXTEN); CLOSE (EXTENIMPORTE); CLOSE (EXTENSALDO);
00863      1  2      EXTEN; END;
00864      C  1  2      (***** *)
00865      1  2      'H','h': BEGIN AYUDA; EXTEN; END;
00866      1  2      'X','x': BORRADO;
00867      1  2      OTHERWISE BEGIN WRITE (CHR(7)); GOTO 10; END;
00868      0  0      END; (* fin del begin del case primero *) END;
00869      C  0  0      (******)
00870      C  0  0      (***** TRANS *****)
00871      C  0  0      (******)
00872      1  0      PROCEDURE TRANS;
00873      1  0      LABEL 10,11,AA,A,B,C,D,R;
00874      1  0      TYPE STRING= VARYING i300 OF CHAR;
00875      1  0      VAR FTRANS: TEXT; TRANSSALDO,TRANSIMPORTE: FILE OF INTEGER;
00876      1  0      S,FECHA,NFACTURA,CONCEPTO: STRING; M,J,N,I,IMPORTE,SALDO: I
00877      1  1      BEGIN
00878      1  1      COMENTARIO:= ' CUENTA DE TRANSPORTES ';
00879      1  1      OPCUENTAS; 10: OPCION;
00880      1  2      CASE CH OF

```

-LINE-IDC-FL-SL-

```
00881 C 1 2 (*****, Inicializaciorn de la cuenta *****)
00882 1 3 'I','i': BEGIN
00883 1 3 11: BORRADO; WRITE (CHR(27),'i11;3H');
00884 1 3 WRITELN (
00885 1 3 'Pulse la "S" para confirmar la inicializaciorn, "N" en caso con
00886 1 3 WRITE (' ':2,'(recuerde que todas las operaciones son borradas)
00887 1 3 WRITE (CHR(27),'i0J'); READLN (CH);
00888 1 5 CASE CH OF 'S','s': BEGIN ENCA; INTRODADOS;
00889 1 6 REPEAT AA: WRITE (CHR(27),'i5;20H'); READLN (FECHA);
00890 1 6 IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO AA; END;
00891 1 5 UNTIL LENGTH (FECHA) < 10;
00892 1 5 WRITE (CHR(27),'( '); WRITE (CHR(27),'i5;28H');
00893 1 5 WRITE (' x '); WRITE (CHR(27),'(B');
00894 1 5 CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada
00895 1 6 REPEAT WRITE (CHR(27),'i5;55H');
00896 1 6 WRITE (CHR(27),'( '); WRITE (' x ');
00897 1 5 READ (IMPORTE); UNTIL IMPORTE < 100000000;
00898 1 5 WRITE (CHR(27),'(B'); SALDO:= IMPORTE;
00899 1 5 OPEN (FTRANS,'FTRANS.DAT'); OPEN (TRANSSALDO,'TRANSSALDO.DAT'
00900 1 5 OPEN (TRANSIMPORTE,'TRANSIMPORTE.DAT');
00901 1 5 REWRITE (FTRANS);REWRITE (TRANSIMPORTE);REWRITE (TRANSSALDO);
00902 1 5 WRITELN (FTRANS,NFACTURA); WRITELN (FTRANS,FECHA);
00903 1 5 WRITELN (FTRANS,CONCEPTO);WRITE (TRANSIMPORTE,IMPORTE);
00904 1 5 WRITE (TRANSSALDO,IMPORTE);
00905 1 5 CLOSE (FTRANS); CLOSE (TRANSIMPORTE); CLOSE (TRANSSALDO);
00906 1 4 END;
00907 1 4 'N','n': BEGIN WRITELN; END;
00908 1 4 'H','h': BEGIN AYUDA; GOTO 11; END;
00909 1 4 OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END;
00910 1 3 END; TRANS;
00911 1 2 END; (* Desde Begin del 'i','I': *)
00912 C 1 2 (***** Agadir operaciones *****)
00913 1 3 'A','a': BEGIN ENCA;
00914 1 3 OPEN (FTRANS,'FTRANS.DAT',HISTORY:= OLD);
00915 1 3 OPEN (TRANSIMPORTE,'TRANSIMPORTE.DAT',HISTO
00916 1 3 OPEN (TRANSSALDO,'TRANSSALDO.DAT',HISTORY:=
00917 1 4 REPEAT WRITE (CHR(27),'i22;8H');
00918 1 4 WRITE (' Introduzca el nzmero de operaciones: ');
00919 1 4 WRITE (CHR(27),'i0J'); READLN (N);
00920 1 3 UNTIL N < 16;
00921 1 4 FOR I:= 1 TO N DO BEGIN
00922 1 5 REPEAT A: FILA:= I+4; COLUMNA:= 10; CURSOR; READLN (NFACTURA);
00923 1 5 IF NFACTURA = '' THEN BEGIN WRITE (CHR(7)); GOTO A; END;
00924 1 4 UNTIL LENGTH (NFACTURA) < 10;
00925 1 5 REPEAT B: FILA:= I+4; COLUMNA:= 18; CURSOR;
00926 1 5 WRITE (CHR(27),'( '); WRITE (' x ');
00927 1 5 WRITE (CHR(27),'(B'); READLN (FECHA);
00928 1 5 IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO B; END;
00929 1 4 UNTIL LENGTH (FECHA) < 10;
00930 1 5 REPEAT C: FILA:= I+4; COLUMNA:= 28; CURSOR;
00931 1 5 WRITE (CHR(27),'( '); WRITE (' x ');
00932 1 5 WRITE (CHR(27),'(B'); READLN (CONCEPTO);
00933 1 5 IF CONCEPTO = '' THEN BEGIN WRITE (CHR(7)); GOTO C; END;
00934 1 4 UNTIL LENGTH (CONCEPTO) < 21;
00935 1 5 REPEAT D: FILA:= I+4; COLUMNA:= 55; CURSOR;
```

ARCUNT
01

Source Listing

11-Feb-1987 10:2
11-Feb-1987 10:2

-LINE-IDC-FL-SL-

```

00936      1 5 WRITE (CHR(27),'(O)'); WRITE (' x ');
00937      1 5 WRITE (CHR(27),'(B)'); READLN (IMPORTE);
00938      1 5      IF IMPORTE < 0 THEN BEGIN WRITE (CHR(7)); GOTO D; END;
00939      1 4 UNTIL IMPORTE < 100000000;
00940      1 4 R: WRITE (CHR(27),'i22;8H');
00941      1 4 WRITE ('Confirme s no la operacion ("S" s "N") : ');
00942      1 4 WRITE (CHR(27),'i0J'); READLN (CH);
00943      1 5 CASE CH OF      'N','n': FILA:= FILA-1;
00944      1 6      'S','s': BEGIN
00945      1 6      EXTEND (FTRANS); EXTEND (TRANSIMPORTE);
00946      1 6      WRITELN (FTRANS,FACTURA); WRITELN (FTRANS,FECHA);
00947      1 6      WRITELN (FTRANS,CONCEPTO); WRITE (TRANSIMPORTE,IMPORTE);
00948      1 6      RESET (TRANSSALDO); WHILE NOT EOF (TRANSSALDO) DO
00949      1 6      READ (TRANSSALDO,SALDO); SALDO:= SALDO - IMPORTE;
00950      1 6      EXTEND (TRANSSALDO); WRITE (TRANSSALDO,SALDO);
00951      1 5      END;
00952      1 5      OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
00953      1 4      END; (* fin del case ch of *)
00954      1 3      END; (* fin del FOR *)
00955      1 3 CLOSE (FTRANS); CLOSE (TRANSIMPORTE); CLOSE (TRANSSALDO);
00956      1 2 TRANS; END;
00957      C 1 2 (***** Consulta de la cuenta *****)
00958      1 3      'C','c': BEGIN ENCABEZAMIENTO;
00959      1 3 OPEN (FTRANS,'FTRANS.DAT',HISTORY:=OLD); RESET (FTRANS);
00960      1 3 OPEN (TRANSIMPORTE,'TRANSIMPORTE.DAT',HISTORY:=OLD); RESET (TRA
00961      1 3 OPEN (TRANSSALDO,'TRANSSALDO.DAT',HISTORY:=OLD); RESET (TRANSSA
00962      1 3      J:=5; M:=1; W:=1;
00963      1 3 WRITE (CHR(27),'i5;1H');
00964      1 4 WHILE NOT EOF (FTRANS) DO BEGIN
00965      1 5      WHILE NOT EOF (TRANSIMPORTE) DO BEGIN
00966      1 6      WHILE NOT EOF (TRANSSALDO) DO BEGIN
00967      1 6      WRITE (CHR(27),'(O)'); WRITE ('x'); WRITE (CHR(27),'(B)');
00968      1 6      FILA:=J; COLUMNA:=0; CURSOR;
00969      1 6      WRITE (M);
00970      1 6      FILA:=J; COLUMNA:=0; (* NUM.FACTURA *)
00971      1 6      CURSOR;
00972      1 6      WRITE (CHR(27),'i7D','nzm.',CHR(27),'i6C');
00973      1 6      READLN (FTRANS,S);
00974      1 6      WRITE (S,CHR(27),'i25D');
00975      1 6      FILA:=J; COLUMNA:=0;
00976      1 6      CURSOR; (* FECHA *)
00977      1 6      READLN (FTRANS,S);
00978      1 6      WRITE (CHR(27),'i25C',S);
00979      1 6      WRITE (CHR(27),'i36D');
00980      1 6      FILA:=J; COLUMNA:=0; (* CONCEPTO *)
00981      1 6      CURSOR;
00982      1 6      READLN (FTRANS,S);
00983      1 6      WRITE (CHR(27),'i36C',S);
00984      1 6      WRITE (CHR(27),'i58D');
00985      1 6      FILA:=J; COLUMNA:=0; (* IMPORTE *)
00986      1 6      CURSOR;
00987      1 6      WRITE (CHR(27),'i58C');
00988      1 6      READ (TRANSIMPORTE,IMPORTE);
00989      1 6      WRITE (IMPORTE,CHR(27),'i68D');
00990      1 6      FILA:=J; COLUMNA:=0; (* SALDO *)

```

-LINE-IDC-FL-SL-

```

00991      1  6          CURSOR;
00992      1  6          READ (TRANSSALDO,SALDO);
00993      1  6          WRITE (CHR(27),'i68C',SALDO);
00994      1  6          WRITE (CHR(27),'(O)'; WRITELN (' x'); WRITE (CHR(27),'(B)';
00995      1  6          J:=J+1; M:=M+1;
00996      1  7          IF M = 17*W THEN BEGIN
00997      1  7          WRITE (CHR(27),'(O','m)'; FOR I:= 1 TO 78 DO WRITE ('q');
00998      1  7          WRITELN ('j'); WRITE (CHR(27),'(B)';
00999      1  7          WRITE ('Pzlse <RETURN> para continuar
01000      1  7          WRITE (CHR(27),'iOJ'); READ (CH); W:=W+1;
01001      1  7          WRITE (CHR(27),'i5;1H'); WRITE (CHR(27),'iOJ');
01002      1  6          END;
01003      1  5          END; (* fin del WHILE ... *)
01004      1  4          END; (* fin del WHILE ... *)
01005      1  3          END; (* fin del WHILE NOT (FTRANS) *)
01006      1  3          WRITE (CHR(27),'(O','m)'; FOR I:= 1 TO 78 DO WRITE ('q');
01007      1  3          WRITELN ('j'); WRITE (CHR(27),'(B)';
01008      1  3          WRITE ('Pzlse <RETURN> para continuar ...');
01009      1  3          WRITE (CHR(27),'iOJ'); READ (CH); BORRADO;
01010      1  3          CLOSE (FTRANS); CLOSE (TRANSIMPORTE); CLOSE (TRANSSALDO);
01011      1  2          TRANS; END;
01012      C  1  2          (***** *)
01013      1  2          'H','h': BEGIN AYUDA; TRANS; END;
01014      1  2          'X','x': BORRADO;
01015      1  2          OTHERWISE BEGIN WRITE (CHR(7)); GOTO 10; END;
01016      0  0          END; (* fin del begin del case primero *) END;
01017      C  0  0          (*****)
01018      C  0  0          (***** MANTE *****)
01019      C  0  0          (*****)
01020      1  0          PROCEDURE MANTE;
01021      1  0          LABEL 10,11,AA,A,B,C,D,R;
01022      1  0          TYPE STRING= VARYING i3002 OF CHAR;
01023      1  0          VAR FMANTE: TEXT; MANTESALDO,MANTEIMPORTE: FILE OF INTEGER;
01024      1  0          S,FECHA,NFACTURA,CONCEPTO: STRING; M,J,N,I,W,IMPORTE,SALDO:
01025      1  1          BEGIN
01026      1  1          COMENTARIO:= ' CUENTA DE MANTENIMIENTO ';
01027      1  1          OPCUENTAS; 10: OPCION;
01028      1  2          CASE CH OF
01029      C  1  2          (*****, Inicializaci3rn de la cuenta *****)
01030      1  3          'I','i': BEGIN
01031      1  3          11; BORRADO; WRITE (CHR(27),'i11;3H');
01032      1  3          WRITELN (
01033      1  3          'Pylse la "S" para confirmar la inicializaci3rn, "N" en caso cont
01034      1  3          WRITE (' ':2,'(recuerde que todas las operaciones son borradas):
01035      1  4          WRITE (CHR(27),'iOJ'); READLN (CH); CASE CH OF
01036      1  5          'S','s': BEGIN ENCA;
01037      C  1  5          (***** introduccion de datos *****)
01038      1  6          INTRODATOS; REPEAT
01039      1  6          AA: WRITE (CHR(27),'i5;20H'); READLN (FECHA);
01040      1  6          IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO AA; EN
01041      1  5          UNTIL LENGTH (FECHA) < 10; WRITE (CHR(27),'(O)';
01042      1  5          WRITE (CHR(27),'i5;28H'); WRITE (' x '); WRITE (CHR(27),'(
01043      1  5          CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada
01044      1  6          REPEAT WRITE (CHR(27),'i5;55H'); WRITE (CHR(27),'(O)';
01045      1  6          WRITE (' x '); READ (IMPORTE);

```

-LINE-IDC-PL-SL-

```
01046      1 5  UNTIL IMPORTE < 100000000; WRITE (CHR(27),'(B)'); SALDO:= IMP
01047  C 1 5  (***** fin de la introduccion de los datos *****)
01048      1 5  OPEN (FMANTE,'FMANTE.DAT'); OPEN (MANTESALDO,'MANTESALDO.DAT
01049      1 5  OPEN (MANTEIMPORTE,'MANTEIMPORTE.DAT');
01050      1 5  REWRITE (FMANTE); REWRITE (MANTEIMPORTE); REWRITE (MANTESALDO)
01051      1 5  WRITELN (FMANTE,NFACTURA);
01052      1 5  WRITELN (FMANTE,FECHA); WRITELN (FMANTE,CONCEPTO);
01053      1 5  WRITE (MANTEIMPORTE,IMPORTE); WRITE (MANTESALDO,IMPORTE);
01054      1 4  CLOSE (FMANTE); CLOSE (MANTEIMPORTE); CLOSE (MANTESALDO); END
01055      1 4  'N','n': BEGIN WRITELN; END;
01056      1 4  'H','h': BEGIN AYUDA; GOTO 11; END;
01057      1 4  OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END;
01058      1 3  END; MANTE;
01059      1 2  END; (* Desde Begin del 'i','I': *)
01060  C 1 2  (***** Agadir operaciones *****)
01061      1 3  'A','a': BEGIN ENCA;
01062      1 3  OPEN (FMANTE,'FMANTE.DAT',HISTORY:= OLD);
01063      1 3  OPEN (MANTEIMPORTE,'MANTEIMPORTE.DAT',HISTOR
01064      1 3  OPEN (MANTESALDO,'MANTESALDO.DAT',HISTORY:=
01065      1 4  REPEAT WRITE (CHR(27),'i22;8H');
01066      1 4  WRITE ('Introduzca el nzmero de operaciones: ');
01067      1 4  WRITE (CHR(27),'iOJ'); READLN (N);
01068      1 4  UNTIL N < 16; FOR I:= 1 TO N DO BEGIN
01069      1 5  REPEAT A: FILA:= I+4; COLUMNA:= 10;
01070      1 5  CURSOR; READLN (NFACTURA);
01071      1 5  IF NFACTURA = '' THEN BEGIN WRITE (CHR(7)); GOTO A; END;
01072      1 5  UNTIL LENGTH (NFACTURA) < 10; REPEAT
01073      1 5  B: FILA:= I+4; COLUMNA:= 18; CURSOR;
01074      1 5  WRITE (CHR(27),'(O)'); WRITE (' x ');
01075      1 5  WRITE (CHR(27),'(B)'); READLN (FECHA);
01076      1 5  IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO B; END;
01077      1 5  UNTIL LENGTH (FECHA) < 10; REPEAT
01078      1 5  C: FILA:= I+4; COLUMNA:= 28; CURSOR;
01079      1 5  WRITE (CHR(27),'(O)'); WRITE (' x ');
01080      1 5  WRITE (CHR(27),'(B)'); READLN (CONCEPTO);
01081      1 5  IF CONCEPTO = '' THEN BEGIN WRITE (CHR(7)); GOTO C; END;
01082      1 5  UNTIL LENGTH (CONCEPTO) < 21; REPEAT
01083      1 5  D: FILA:= I+4; COLUMNA:= 55; CURSOR;
01084      1 5  WRITE (CHR(27),'(O)'); WRITE (' x ');
01085      1 5  WRITE (CHR(27),'(B)'); READLN (IMPORTE);
01086      1 5  IF IMPORTE < 0 THEN BEGIN WRITE (CHR(7)); GOTO D; END;
01087      1 4  UNTIL IMPORTE < 100000000; R: WRITE (CHR(27),'i22;8H');
01088      1 4  WRITE ('Confirme s no la operacisn ("S" s "N") : ');
01089      1 4  WRITE (CHR(27),'iOJ'); READLN (CH);
01090      1 5  CASE CH OF 'N','n': FILA:= FILA-1;
01091      1 6  'S','s': BEGIN
01092      1 6  EXTEND (FMANTE); EXTEND (MANTEIMPORTE);
01093      1 6  WRITELN (FMANTE,NFACTURA); WRITELN (FMANTE,FECHA);
01094      1 6  WRITELN (FMANTE,CONCEPTO); WRITE (MANTEIMPORTE,IMPORTE);
01095      1 6  RESET (MANTESALDO); WHILE NOT EOF (MANTESALDO) DO
01096      1 6  READ (MANTESALDO,SALDO); SALDO:= SALDO - IMPORTE;
01097      1 5  EXTEND (MANTESALDO); WRITE (MANTESALDO,SALDO); END;
01098      1 5  OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
01099      1 4  END; (* fin del case ch of .*)
01100      1 3  END; (* fin del FOR *)
```

-LINE-IDC-FL-SL-

```

01101      1 3  CLOSE (FMANTE); CLOSE (MANTEIMPORTE);  CLOSE (MANTESALDO);
01102      1 2  MANTE;          END;
01103      C 1 2  (*****      Consulta de la cuenta      *****)
01104      1 3      'C','c': BEGIN ENCABEZAMIENTO;
01105      1 3  OPEN (FMANTE,'FMANTE.DAT',HISTORY:=OLD); RESET (FMANTE);
01106      1 3  OPEN (MANTEIMPORTE,'MANTEIMPORTE.DAT',HISTORY:=OLD); RESET (MAN
01107      1 3  OPEN (MANTESALDO,'MANTESALDO.DAT',HISTORY:=OLD); RESET (MANTESA
01108      1 3      J:=5; M:=1; W:=1;      WRITE (CHR(27),'i5;1H');
01109      1 4  WHILE NOT EOF (FMANTE) DO BEGIN
01110      1 5      WHILE NOT EOF (MANTEIMPORTE) DO BEGIN
01111      1 6      WHILE NOT EOF (MANTESALDO) DO BEGIN
01112      1 6  WRITE (CHR(27),'(O)'); WRITE ('x'); WRITE (CHR(27),'(B)');
01113      1 6      FILA:=J; COLUMN:=0; CURSOR;
01114      1 6      WRITE (M);
01115      1 6      FILA:=J; COLUMN:=0;          (* NUM.FACTURA *)
01116      1 6      CURSOR;
01117      1 6      WRITE (CHR(27),'i7D','nzm.',CHR(27),'i6C');
01118      1 6      READLN (FMANTE,S);
01119      1 6      WRITE (S,CHR(27),'i25D');
01120      1 6      FILA:=J; COLUMN:=0;
01121      1 6      CURSOR;          (* FECHA *)
01122      1 6      READLN (FMANTE,S);
01123      1 6      WRITE (CHR(27),'i25C',S);
01124      1 6      WRITE (CHR(27),'i36D');
01125      1 6      FILA:=J; COLUMN:=0;          (* CONCEPTO *)
01126      1 6      CURSOR;
01127      1 6      READLN (FMANTE,S);
01128      1 6      WRITE (CHR(27),'i36C',S);
01129      1 6      WRITE (CHR(27),'i58D');
01130      1 6      FILA:=J; COLUMN:=0;          (* IMPORTE *)
01131      1 6      CURSOR;
01132      1 6      WRITE (CHR(27),'i58C');
01133      1 6      READ (MANTEIMPORTE,IMPORTE);
01134      1 6      WRITE (IMPORTE,CHR(27),'i68D');
01135      1 6      FILA:=J; COLUMN:=0;          (* SALDO *)
01136      1 6      CURSOR;
01137      1 6      READ (MANTESALDO,SALDO);
01138      1 6      WRITE (CHR(27),'i68C',SALDO);
01139      1 6  WRITE (CHR(27),'(O)'); WRITELN ('x'); WRITE (CHR(27),'(B)');
01140      1 6      J:=J+1; M:=M+1;
01141      1 7      IF M = 17*W THEN BEGIN
01142      1 7      WRITE (CHR(27),'(O','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
01143      1 7      WRITELN ('j'); WRITE (CHR(27),'(B)');
01144      1 7      WRITE ('Pzlse <RETURN> para continuar
01145      1 7      WRITE (CHR(27),'iOJ'); READ (CH); W:=W+1;
01146      1 7      WRITE (CHR(27),'i5;1H'); WRITE (CHR(27),'iOJ');
01147      1 6      END;
01148      1 5      END;          (* fin del WHILE .. *)
01149      1 4      END;          (* fin del WHILE ... *)
01150      1 3  END;          (* fin del WHILE NOT (FMANTE) *)
01151      1 3  WRITE (CHR(27),'(O','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
01152      1 3  WRITELN ('j'); WRITE (CHR(27),'(B)');
01153      1 3  WRITE ('Pzlse <RETURN> para continuar ...');
01154      1 3  WRITE (CHR(27),'iOJ'); READ (CH); BORRADO;
01155      1 3  CLOSE (FMANTE);CLOSE (MANTEIMPORTE);CLOSE (MANTESALDO);

```

-LINE-IDC-PL-SL-

```

01156      1  2 MANTE;                END;                (* fin del BEGIN ... 'C' *)
01157  C  1  2 (*****
01158      1  2      'H','h': BEGIN AYUDA; MANTE; END;
01159      1  2      'X','x': BORRADO;
01160      1  2      OTHERWISE BEGIN WRITE (CHR(7)); GOTO 10; END;
01161      1  1 END; (* fin del begin del case primero *)
01162      0  0 END;
01163  C  0  0      (*****
01164  C  0  0      (***** REUNION *****
01165  C  0  0      (*****
01166      1  0 PROCEDURE REUNION;
01167      1  0 LABEL 10,11,AA,A,B,C,D,R;
01168      1  0 TYPE STRING= VARYING i300& OF CHAR;
01169      1  0 VAR FREU: TEXT; REUSALDO,REUIMPORTE: FILE OF INTEGER;
01170      1  0      S,FECHA,NFACTURA,CONCEPTO: STRING; M,J,N,I,W,IMPORTE,SALDO:
01171      1  1 BEGIN
01172      1  1 COMENTARIO:= ' CUENTA DE REUNIONES Y CONFERENCIAS ';
01173      1  2 OPCUENTAS; 10: OPCION; CASE CH OF
01174  C  1  2 (*****, Inicializaciom de la cuenta *****
01175      1  3      'I','i': BEGIN
01176      1  3 11: BORRADO; WRITE (CHR(27),'i11;3H');
01177      1  3 WRITELN (
01178      1  3 'Pylse la "S" para confirmar la inicializaciom, "N" en caso cont
01179      1  3 WRITE (' ':2,'(recuerde que todas las operaciones son borradas):
01180      1  3 WRITE (CHR(27),'i0J'); READLN (CH);
01181      1  5 CASE CH OF 'S','s': BEGIN ENCA;
01182  C  1  5 (***** introduccion de datos *****
01183      1  6 INTRODADOS; REPEAT
01184      1  6 AA: WRITE (CHR(27),'i5;20H'); READLN (FECHA);
01185      1  6      IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO AA
01186      1  5 UNTIL LENGTH (FECHA) < 10; WRITE (CHR(27),'(0');
01187      1  5 WRITE (CHR(27),'i5;28H'); WRITE (' x '); WRITE (CHR(27),'(E
01188      1  5 CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada '
01189      1  6 REPEAT WRITE (CHR(27),'i5;55H');
01190      1  6 WRITE (CHR(27),'(0'); WRITE (' x ');
01191      1  5 READ (IMPORTE); UNTIL IMPORTE < 100000000;
01192      1  5 WRITE (CHR(27),'(B'); SALDO:= IMPORTE;
01193  C  1  5 (***** fin de la introduccion de los datos ****
01194      1  5 OPEN (FREU,'FREU.DAT'); OPEN (REUSALDO,'REUSALDO.DAT');
01195      1  5 OPEN (REUIMPORTE,'REUIMPORTE.DAT');
01196      1  5 REWRITE (FREU); REWRITE (REUIMPORTE); REWRITE (REUSALDO);
01197      1  5 WRITELN (FREU,NFACTURA); WRITELN (FREU,FECHA);
01198      1  5 WRITELN (FREU,CONCEPTO); WRITE (REUIMPORTE,IMPORTE);
01199      1  5 WRITE (REUSALDO,IMPORTE);
01200      1  4 CLOSE (FREU); CLOSE (REUIMPORTE); CLOSE (REUSALDO); END;
01201      1  4      'N','n': BEGIN WRITELN; END;
01202      1  4      'H','h': BEGIN AYUDA; GOTO 11; END;
01203      1  4      OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END;
01204      1  3      END; REUNION;
01205      1  2      END; (* Desde Begin del 'i','I': *)
01206  C  1  2 (***** Agadir operaciones *****
01207      1  3      'A','a': BEGIN ENCA;
01208      1  3 OPEN (FREU,'FREU.DAT',HISTORY:= OLD);
01209      1  3 OPEN (REUIMPORTE,'REUIMPORTE.DAT',HISTORY:=
01210      1  3 OPEN (REUSALDO,'REUSALDO.DAT',HISTORY:= OLD)

```

© Del documento, los autores. Digitalización realizada por UPEC-Biblioteca Universitaria. 2018

-LINE-IDC-PL-SL-

```
01211 1 4 REPEAT WRITE (CHR(27),'i22;8H');
01212 1 4 WRITE ('Introduzca el nmero de operaciones: ');
01213 1 4 WRITE (CHR(27),'i0J'); READLN (N);
01214 1 3 UNTIL N < 16;
01215 1 4 FOR I:= 1 TO N DO BEGIN
01216 1 5 REPEAT A: FILA:= I+4; COLUMNA:= 10;
01217 1 5 CURSOR; READLN (NFACTURA);
01218 1 5 IF NFACTURA = '' THEN BEGIN WRITE (CHR(7)); GOTO A; END;
01219 1 5 UNTIL LENGTH (NFACTURA) < 10; REPEAT
01220 1 5 B: FILA:= I+4; COLUMNA:= 18; CURSOR;
01221 1 5 WRITE (CHR(27),'(0)'); WRITE ('x ');
01222 1 5 WRITE (CHR(27),'(B)'); READLN (FECHA);
01223 1 5 IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO B; END;
01224 1 5 UNTIL LENGTH (FECHA) < 10; REPEAT
01225 1 5 C: FILA:= I+4; COLUMNA:= 28; CURSOR;
01226 1 5 WRITE (CHR(27),'(0)'); WRITE ('x ');
01227 1 5 WRITE (CHR(27),'(B)'); READLN (CONCEPTO);
01228 1 5 IF CONCEPTO = '' THEN BEGIN WRITE (CHR(7)); GOTO C; END;
01229 1 4 UNTIL LENGTH (CONCEPTO) < 21;
01230 1 5 REPEAT D: FILA:= I+4; COLUMNA:= 55;
01231 1 5 CURSOR; WRITE (CHR(27),'(0)');
01232 1 5 WRITE ('x '); WRITE (CHR(27),'(B)'); READLN (IMPORTE);
01233 1 5 IF IMPORTE < 0 THEN BEGIN WRITE (CHR(7)); GOTO D; END;
01234 1 4 UNTIL IMPORTE < 100000000; R: WRITE (CHR(27),'i22;8H');
01235 1 4 WRITE ('Confirme s no la operacion ("S" s "N") : ');
01236 1 4 WRITE (CHR(27),'i0J'); READLN (CH);
01237 1 5 CASE CH OF 'N','n': FILA:= FILA-1;
01238 1 6 'S','s': BEGIN
01239 1 6 EXTEND (FREU); EXTEND (REUIMPORTE);
01240 1 6 WRITELN (FREU,NFACTURA); WRITELN (FREU,FECHA);
01241 1 6 WRITELN (FREU,CONCEPTO); WRITE (REUIMPORTE,IMPORTE);
01242 1 6 RESET (REUSALDO); WHILE NOT EOF (REUSALDO) DO
01243 1 6 READ (REUSALDO,SALDO); SALDO:= SALDO - IMPORTE;
01244 1 5 EXTEND (REUSALDO); WRITE (REUSALDO,SALDO); END;
01245 1 5 OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
01246 1 4 END; (* fin del case ch of *)
01247 1 3 END; (* fin del FOR *)
01248 1 3 CLOSE (FREU); CLOSE (REUIMPORTE); CLOSE (REUSALDO);
01249 1 2 REUNION; END;
01250 C 1 2 (***** Consulta de la cuenta *****)
01251 1 3 'C','c': BEGIN ENCABEZAMIENTO;
01252 1 3 OPEN (FREU,'FREU.DAT',HISTORY:=OLD); RESET (FREU);
01253 1 3 OPEN (REUIMPORTE,'REUIMPORTE.DAT',HISTORY:=OLD); RESET (REUIMPO
01254 1 3 OPEN (REUSALDO,'REUSALDO.DAT',HISTORY:=OLD); RESET (REUSALDO);
01255 1 3 J:=5; M:=1; W:=1;
01256 1 3 WRITE (CHR(27),'i5;1H');
01257 1 4 WHILE NOT EOF (FREU) DO BEGIN
01258 1 5 WHILE NOT EOF (REUIMPORTE) DO BEGIN
01259 1 6 WHILE NOT EOF (REUSALDO) DO BEGIN
01260 1 6 WRITE (CHR(27),'(0)'); WRITE ('x'); WRITE (CHR(27),'(B)');
01261 1 6 FILA:=J; COLUMNA:=0; CURSOR;
01262 1 6 WRITE (M);
01263 1 6 FILA:=J; COLUMNA:=0; (* NUM.FACTURA *)
01264 1 6 CURSOR;
01265 1 6 WRITE (CHR(27),'i7D','nzm.',CHR(27),'i6C');
```

-LINE-IDC-FL-SL-

```
01266      1 6      READLN (FREU,S);
01267      1 6      WRITE (S,CHR(27),'i25D');
01268      1 6      FILA:=J; COLUMNA:=0;
01269      1 6      CURSOR; (* FECHA *)
01270      1 6      READLN (FREU,S);
01271      1 6      WRITE (CHR(27),'i25C',S);
01272      1 6      WRITE (CHR(27),'i36D');
01273      1 6      FILA:=J; COLUMNA:=0; (* CONCEPTO *)
01274      1 6      CURSOR;
01275      1 6      READLN (FREU,S);
01276      1 6      WRITE (CHR(27),'i36C',S);
01277      1 6      WRITE (CHR(27),'i58D');
01278      1 6      FILA:=J; COLUMNA:=0; (* IMPORTE *)
01279      1 6      CURSOR;
01280      1 6      WRITE (CHR(27),'i58C');
01281      1 6      READ (REUIMPORTE,IMPORTE);
01282      1 6      WRITE (IMPORTE,CHR(27),'i68D');
01283      1 6      FILA:=J; COLUMNA:=0; (* SALDO *)
01284      1 6      CURSOR;
01285      1 6      READ (REUSALDO,SALDO);
01286      1 6      WRITE (CHR(27),'i68C',SALDO);
01287      1 6      WRITE (CHR(27),'(O)'; WRITELN ('x'); WRITE (CHR(27),'(B)');
01288      1 6      J:=J+1; M:=M+1;
01289      1 7      IF M = 17*W THEN BEGIN
01290      1 7      WRITE (CHR(27),'(O','m)'; FOR I:= 1 TO 78 DO WRITE ('q');
01291      1 7      WRITELN ('j'); WRITE (CHR(27),'(B)');
01292      1 7      WRITE ('Pzlse <RETURN> para continuar...');
01293      1 7      WRITE (CHR(27),'i0J'); READ (CH); W:=W+1;
01294      1 7      WRITE (CHR(27),'i5;1H'); WRITE (CHR(27),'i0J');
01295      1 6      END;
01296      1 6
01297      1 5      END; (* fin del WHILE ... EUPIMPORTE *)
01298      1 5
01299      1 4      END; (* fin del WHILE ... EUPSALDO *)
01300      1 4
01301      1 3 END; (* fin del WHILE NOT (FEUP) *)
01302      1 3 WRITE (CHR(27),'(O','m)'; FOR I:= 1 TO 78 DO WRITE ('q');
01303      1 3 WRITELN ('j'); WRITE (CHR(27),'(B)');
01304      1 3 WRITE ('Pzlse <RETURN> para continuar ...');
01305      1 3 WRITE (CHR(27),'i0J'); READ (CH); BORRADO;
01306      1 3 CLOSE (FREU); CLOSE (REUIMPORTE); CLOSE (REUSALDO);
01307      1 2 REUNION; END; (* fin del BEGIN ... 'C'
01308      C 1 2 (*****
01309      1 2 'H','h': BEGIN AYUDA; REUNION; END;
01310      1 2 'X','x': BORRADO;
01311      1 2 OTHERWISE BEGIN WRITE (CHR(7)); GOTO 10; END;
01312      0 0 END; (* fin del begin del case primero *) END;
01313      C 0 0 (*****
01314      C 0 0 (***** MOBILIA *****
01315      C 0 0 (*****
01316      1 0 PROCEDURE MOBILIA;
01317      1 0 LABEL 10,11,AA,A,B,C,D,R;
01318      1 0 TYPE STRING= VARYING i300 OF CHAR;
01319      1 0 VAR FMOB: TEXT; MOBSALDO,MOBIMPORTE: FILE OF INTEGER;
01320      1 0 S,FECHA,NFACTURA,CONCEPTO: STRING; M,J,N,I,W,IMPORTE,SALDO;
```

-LINE-IDC-PL-SL-

```
01321      1 1 BEGIN
01322      1 1 COMENTARIO:= ' CUENTA DE MOBILIARIO Y EQUIPO INVENTARIAL ';
01323      1 2 OPCUENTAS; 10: OPCION; CASE CH OF
01324      C 1 2 (*****,      Inicializaci3rn de la cuenta      *****)
01325      1 3      'I','i': BEGIN
01326      1 3 11: BORRADO; WRITE (CHR(27),'i11;3H');
01327      1 3 WRITELN (
01328      1 3 'Pulse la "S" para confirmar la inicializaci3rn, "N" en caso cont
01329      1 3 WRITE (' ':2,'(recuerde que todas las operaciones son borradas):
01330      1 3 WRITE (CHR(27),'i0J'); READLN (CH);
01331      1 5 CASE CH OF      'S','s': BEGIN ENCA;
01332      C 1 5 (*****      introduccion de datos      *****)
01333      1 5 INTRODATOS;
01334      1 6 REPEAT AA: WRITE (CHR(27),'i5;20H'); READLN (FECHA);
01335      1 6 IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO AA;
01336      1 5 UNTIL LENGTH (FECHA) < 10;
01337      1 5 WRITE (CHR(27),'(0'); WRITE (CHR(27),'i5;28H');
01338      1 5 WRITE (' x '); WRITE (CHR(27),'(B');
01339      1 5 CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada
01340      1 6 REPEAT WRITE (CHR(27),'i5;55H');
01341      1 6 WRITE (CHR(27),'(0'); WRITE (' x ');
01342      1 5 READ (IMPORTE); UNTIL IMPORTE < 100000000;
01343      1 5 WRITE (CHR(27),'(B'); SALDO:= IMPORTE;
01344      C 1 5 (*****      fin de la introduccion de los datos      *****)
01345      1 5 OPEN (FMOB,'FMOB.DAT'); OPEN (MOBSALDO,'MOBSALDO.DAT');
01346      1 5 OPEN (MOBIMPORTE,'MOBIMPORTE.DAT');
01347      1 5 REWRITE (FMOB); REWRITE (MOBIMPORTE);REWRITE (MOBSALDO);
01348      1 5 WRITELN (FMOB,NFACTURA);WRITELN (FMOB,FECHA);
01349      1 5 WRITELN (FMOB,CONCEPTO);WRITE (MOBIMPORTE,IMPORTE);WRITE (MOBSAL
01350      1 4 CLOSE (FMOB); CLOSE (MOBIMPORTE); CLOSE (MOBSALDO); END;
01351      1 4      'N','n': BEGIN WRITELN; END;
01352      1 4      'H','h': BEGIN AYUDA; GOTO 11; END;
01353      1 4 OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END;
01354      1 3      . END; MOBILIA;
01355      1 2      END; (* Desde Begin del 'i','I': *)
01356      C 1 2 (*****      Agadir operaciones      *****)
01357      1 3      'A','a': BEGIN ENCA;
01358      1 3 OPEN (FMOB,'FMOB.DAT',HISTORY:= OLD);
01359      1 3 OPEN (MOBIMPORTE,'MOBIMPORTE.DAT',HISTORY:=
01360      1 3 OPEN (MOBSALDO,'MOBSALDO.DAT',HISTORY:= OLD)
01361      1 4 REPEAT WRITE (CHR(27),'i22;8H');
01362      1 4 WRITE (' Introduzca el nzmero de operaciones: ');
01363      1 3 WRITE (CHR(27),'i0J'); READLN (N); UNTIL N < 16;
01364      1 4 FOR I:= 1 TO N DO BEGIN
01365      1 5 REPEAT A: FILA:= I+4; COLUMNA:= 10;
01366      1 5 CURSOR; READLN (NFACTURA);
01367      1 5 IF NFACTURA = '' THEN BEGIN WRITE (CHR(7)); GOTO A; END;
01368      1 4 UNTIL LENGTH (NFACTURA) < 10;
01369      1 5 REPEAT B: FILA:= I+4; COLUMNA:= 18;
01370      1 5 CURSOR; WRITE (CHR(27),'(0');
01371      1 5 WRITE ('x '); WRITE (CHR(27),'(B'); READLN (FECHA);
01372      1 5 IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO B; END;
01373      1 4 UNTIL LENGTH (FECHA) < 10;
01374      1 5 REPEAT C: FILA:= I+4; COLUMNA:= 28;
01375      1 5 CURSOR;WRITE (CHR(27),'(0');
```

-LINE-IDC-FL-SL-

```
01376 1 5 WRITE (' x ');WRITE (CHR(27),'(B)'); READLN (CONCEPTO);
01377 1 5 IF CONCEPTO = ' ' THEN BEGIN WRITE (CHR(7)); GOTO C; END;
01378 1 4 UNTIL LENGTH (CONCEPTO) < 21;
01379 1 5 REPEAT D: FILA:= I+4; COLUMNA:= 55;
01380 1 5 CURSOR;WRITE (CHR(27),'(O)');
01381 1 5 WRITE (' x '); WRITE (CHR(27),'(B)'); READLN (IMPORTE);
01382 1 5 IF IMPORTE < 0 THEN BEGIN WRITE (CHR(7)); GOTO D; END;
01383 1 4 UNTIL IMPORTE < 100000000;
01384 1 4 R: WRITE (CHR(27),'i22;8H');
01385 1 4 WRITE ('Confirme s no la operacion ("S" s "N") : ');
01386 1 4 WRITE (CHR(27),'iOJ');READLN (CH);
01387 1 5 CASE CH OF 'N','n': FILA:= FILA-1;
01388 1 6 'S','s': BEGIN
01389 1 6 EXTEND (FMOB); EXTEND (MOBIMPORTE);
01390 1 6 WRITELN (FMOB,NFACTURA); WRITELN (FMOB,FECHA);
01391 1 6 WRITELN (FMOB,CONCEPTO); WRITE (MOBIMPORTE,IMPORTE);
01392 1 6 RESET (MOBSALDO); WHILE NOT EOF (MOBSALDO) DO
01393 1 6 READ (MOBSALDO,SALDO); SALDO:= SALDO - IMPORTE;
01394 1 6 EXTEND (MOBSALDO); WRITE (MOBSALDO,SALDO);
01395 1 5 END;
01396 1 5 OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
01397 1 4 END; (* fin del case ch of *)
01398 1 3 END; (* fin del FOR *)
01399 1 3 CLOSE (FMOB); CLOSE (MOBIMPORTE);CLOSE (MOBSALDO);
01400 1 2 MOBILIA; END;
01401 C 1 2 (***** Consulta de la cuenta *****)
01402 1 3 'C','c': BEGIN ENCABEZAMIENTO;
01403 1 3 OPEN (FMOB,'FMOB.DAT',HISTORY:=OLD);RESET (FMOB);
01404 1 3 OPEN (MOBIMPORTE,'MOBIMPORTE.DAT',HISTORY:=OLD);RESET (MOBIMPORTE);
01405 1 3 OPEN (MOBSALDO,'MOBSALDO.DAT',HISTORY:=OLD);RESET (MOBSALDO);
01406 1 3 J:=5; M:=1; W:=1;
01407 1 3 WRITE (CHR(27),'i5;1H');
01408 1 4 WHILE NOT EOF (FMOB) DO BEGIN
01409 1 5 WHILE NOT EOF (MOBIMPORTE) DO BEGIN
01410 1 6 WHILE NOT EOF (MOBSALDO) DO BEGIN
01411 1 6 WRITE (CHR(27),'(O)'); WRITE ('x'); WRITE (CHR(27),'(B)');
01412 1 6 FILA:=J; COLUMNA:=0; CURSOR;
01413 1 6 WRITE (M);
01414 1 6 FILA:=J; COLUMNA:=0; (* NUM.FACTURA *)
01415 1 6 CURSOR;
01416 1 6 WRITE (CHR(27),'i7D','nzm.',CHR(27),'i6C');
01417 1 6 READLN (FMOB,S);
01418 1 6 WRITE (S,CHR(27),'i25D');
01419 1 6 FILA:=J; COLUMNA:=0; CURSOR; (* FECHA *)
01420 1 6 READLN (FMOB,S);
01421 1 6 WRITE (CHR(27),'i25C',S);
01422 1 6 WRITE (CHR(27),'i36D');
01423 1 6 FILA:=J; COLUMNA:=0; (* CONCEPTO *)
01424 1 6 CURSOR;
01425 1 6 READLN (FMOB,S);
01426 1 6 WRITE (CHR(27),'i36C',S);
01427 1 6 WRITE (CHR(27),'i58D');
01428 1 6 FILA:=J; COLUMNA:=0; (* IMPORTE *)
01429 1 6 CURSOR;
01430 1 6 WRITE (CHR(27),'i58C');
```

--LINE-IDC-PL-SL--

```

01431      1  6          READ (MOBIMPORTE, IMPORTE);
01432      1  6          WRITE (IMPORTE, CHR(27), 'i68D');
01433      1  6          FILA:=J; COLUMNNA:=0;          (* SALDO *)
01434      1  6          CURSOR;
01435      1  6          READ (MOBSALDO, SALDO);
01436      1  6          WRITE (CHR(27), 'i68C', SALDO);
01437      1  6          J:=J+1; M:=M+1;
01438      1  6          WRITE (CHR(27), 'O'); WRITELN (' x'); WRITE (CHR(27), 'B');
01439      1  7          IF M = 17*W THEN BEGIN
01440      1  7          WRITE (CHR(27), 'O', 'm'); FOR I:= 1 TO 78 DO WRITE ('q');
01441      1  7          WRITELN ('j'); WRITE (CHR(27), 'B');
01442      1  7          WRITE ('Pzlse <RETURN> para continuar
01443      1  7          WRITE (CHR(27), 'iOJ'); READ (CH); W:=W+1;
01444      1  7          WRITE (CHR(27), 'i5;1H'); WRITE (CHR(27), 'iOJ');
01445      1  6          END;
01446      1  6
01447      1  5          END;          (* fin del WHILE ... MOBIMPORTE *)
01448      1  5
01449      1  4          END;          (* fin del WHILE ... MOBSALDO *)
01450      1  4
01451      1  3          END;          (* fin del WHILE NOT (FMOB) *)
01452      1  3
01453      1  3          WRITE (CHR(27), 'O', 'm'); FOR I:= 1 TO 78 DO WRITE ('q');
01454      1  3          WRITELN ('j'); WRITE (CHR(27), 'B');
01455      1  3          WRITE ('Pzlse <RETURN> para continuar ...');
01456      1  3          WRITE (CHR(27), 'iOJ'); READ (CH); BORRADO;
01457      1  3          CLOSE (FMOB); CLOSE (MOBIMPORTE); CLOSE (MOBSALDO);
01458      1  2          MOBILIA;          END;          (* fin del BEGIN ... 'C'
01459      C  1  2          (*****
01460      1  2          'H', 'h': BEGIN AYUDA; MOBILIA; END;
01461      1  2          'X', 'x': BORRADO;
01462      1  3          OTHERWISE BEGIN WRITE (CHR(7)); GOTO 10;
01463      1  1          END;          END;          (* fin del begin del case primero *)
01464      0  0          END;
01465      C  0  0          (*****
01466      C  0  0          (***** ACTOS *****
01467      C  0  0          (*****
01468      1  0          PROCEDURE ACTOS;
01469      1  0          LABEL 10,11,AA,A,B,C,D,R;
01470      1  0          TYPE STRING= VARYING i300 OF CHAR;
01471      1  0          VAR FAC: TEXT; ACSALDO, ACIMPORTE: FILE OF INTEGER;
01472      1  0          S, FECHA, NFACTURA, CONCEPTO: STRING; M, J, N, I, W, IMPORTE, SALDO;
01473      1  1          BEGIN
01474      1  1          COMENTARIO:= ' CUENTA DE ACTOS DE CARACTER SOCIAL ';
01475      1  1          OPCUENTAS; 10: OPCION;
01476      1  2          CASE CH OF
01477      C  1  2          (*****, Inicializaci3rn de la cuenta *****
01478      1  3          'I', 'i': BEGIN
01479      1  3          11: BORRADO; WRITE (CHR(27), 'i11;3H');
01480      1  3          WRITELN (
01481      1  3          'Pzlse la "S" para confirmar la inicializaci3rn, "N" en caso cont
01482      1  3          WRITE (' ':2, '(recuerde que todas las operaciones son borradas);
01483      1  3          WRITE (CHR(27), 'iOJ'); READLN (CH);
01484      1  5          CASE CH OF 'S', 's': BEGIN ENCA;
01485      C  1  5          (***** introduccion de datos *****

```

ARCONT

01

Source Listing

-LINE-IDC-PL-SL-

```

01486      1 6  INTRODADOS; REPEAT AA: WRITE (CHR(27),'i5;20H'); READLN (F
01487      1 6  IF FECHA = '?' THEN BEGIN WRITE (CHR(7)); GOTO AA; END;
01488      1 5  UNTIL LENGTH (FECHA) < 10;
01489      1 5  WRITE (CHR(27),'(O)'); WRITE (CHR(27),'i5;28H');
01490      1 5  WRITE (' x '); WRITE (CHR(27),'(B)');
01491      1 5  CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada
01492      1 6  REPEAT WRITE (CHR(27),'i5;55H');
01493      1 6  WRITE (CHR(27),'(O)'); WRITE (' x '); READ (IMPORTE);
01494      1 5  UNTIL IMPORTE < 100000000; WRITE (CHR(27),'(B)'); SALDO:= IMP
01495      C 1 5  (***** fin de la introduccion de los datos *****)
01496      1 5  OPEN (FAC,'FAC.DAT'); OPEN (ACSALDO,'ACSALDO.DAT');
01497      1 5  OPEN (ACIMPORTE,'ACIMPORTE.DAT');
01498      1 5  REWRITE (FAC); REWRITE (ACIMPORTE); REWRITE (ACSALDO);
01499      1 5  WRITELN (FAC,NFACTURA); WRITELN (FAC,FECHA);
01500      1 5  WRITELN (FAC,CONCEPTO); WRITE (ACIMPORTE,IMPORTE); WRITE (ACSALDO,
01501      1 4  CLOSE (FAC); CLOSE (ACIMPORTE); CLOSE (ACSALDO); END;
01502      1 4  'N','n': BEGIN WRITELN; END;
01503      1 4  'H','h': BEGIN AYUDA; GOTO 11; END;
01504      1 4  OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END;
01505      1 3  END; ACTOS;
01506      1 2  END; (* Desde Begin del 'i','I': *)
01507      C 1 2  (***** Agadir operaciones *****)
01508      1 3  'A','a': BEGIN ENCA;
01509      1 3  OPEN (FAC,'FAC.DAT',HISTORY:= OLD);
01510      1 3  OPEN (ACIMPORTE,'ACIMPORTE.DAT',HISTORY:= DL
01511      1 3  OPEN (ACSALDO,'ACSALDO.DAT',HISTORY:= OLD);
01512      1 4  REPEAT WRITE (CHR(27),'i22;8H');
01513      1 4  WRITE (' Introduzca el nmero de operaciones: ');
01514      1 4  WRITE (CHR(27),'iOJ'); READLN (N);
01515      1 4  UNTIL N < 16; FOR I:= 1 TO N DO BEGIN
01516      1 5  REPEAT A: FILA:= I+4; COLUMNA:= 10;
01517      1 5  CURSOR; READLN (NFACTURA);
01518      1 5  IF NFACTURA = '?' THEN BEGIN WRITE (CHR(7)); GOTO A; END;
01519      1 4  UNTIL LENGTH (NFACTURA) < 10;
01520      1 5  REPEAT B: FILA:= I+4; COLUMNA:= 18; CURSOR;
01521      1 5  WRITE (CHR(27),'(O)'); WRITE (' x ');
01522      1 5  WRITE (CHR(27),'(B)'); READLN (FECHA);
01523      1 5  IF FECHA = '?' THEN BEGIN WRITE (CHR(7)); GOTO B; END;
01524      1 4  UNTIL LENGTH (FECHA) < 10;
01525      1 5  REPEAT C: FILA:= I+4; COLUMNA:= 28; CURSOR;
01526      1 5  WRITE (CHR(27),'(O)'); WRITE (' x ');
01527      1 5  WRITE (CHR(27),'(B)'); READLN (CONCEPTO);
01528      1 5  IF CONCEPTO = '?' THEN BEGIN WRITE (CHR(7)); GOTO C; END;
01529      1 4  UNTIL LENGTH (CONCEPTO) < 21;
01530      1 5  REPEAT D: FILA:= I+4; COLUMNA:= 55; CURSOR;
01531      1 5  WRITE (CHR(27),'(O)'); WRITE (' x ');
01532      1 5  WRITE (CHR(27),'(B)'); READLN (IMPORTE);
01533      1 5  IF IMPORTE < 0 THEN BEGIN WRITE (CHR(7)); GOTO D; END;
01534      1 4  UNTIL IMPORTE < 100000000;
01535      1 4  R: WRITE (CHR(27),'i22;8H');
01536      1 4  WRITE (' Confirme s no la operacion ("S" s. "N") : ');
01537      1 4  WRITE (CHR(27),'iOJ'); READLN (CH);
01538      1 5  CASE CH OF 'N','n': FILA:= FILA-1;
01539      1 6  'S','s': BEGIN
01540      1 6  EXTEND (FAC); EXTEND (ACIMPORTE);

```

-LINE-IDC-PL-SL-

```
01541      1  6      WRITELN (FAC,NFACTURA);      WRITELN (FAC,FECHA);
01542      1  6      WRITELN (FAC,CONCEPTO);      WRITE (ACIMPORTE,IMPORTE);
01543      1  6      RESET (ACXSALDO);      WHILE NOT EOF (ACXSALDO) DO
01544      1  6          READ (ACXSALDO,SALDO);      SALDO:= SALDO - IMPORTE;
01545      1  5      EXTEND (ACXSALDO);      WRITE (ACXSALDO,SALDO);      END;
01546      1  5          OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
01547      1  4          END;      (* fin del case ch of *)
01548      1  3          END;      (* fin del FOR *)
01549      1  3      CLOSE (FAC); CLOSE (ACIMPORTE); CLOSE (ACXSALDO);
01550      1  2      ACTOS;      END;
01551      C  1  2      (****      Consulta de la cuenta      ****)
01552      1  3          'C', 'c': BEGIN ENCABEZAMIENTO;
01553      1  3      OPEN (FAC, 'FAC.DAT', HISTORY:=OLD); RESET (FAC);
01554      1  3      OPEN (ACIMPORTE, 'ACIMPORTE.DAT', HISTORY:=OLD); RESET (ACIMPORTE);
01555      1  3      OPEN (ACXSALDO, 'ACXSALDO.DAT', HISTORY:=OLD); RESET (ACXSALDO);
01556      1  3          J:=5; M:=1; W:=1;      WRITE (CHR(27), 'i5;1H');
01557      1  4      WHILE NOT EOF (FAC) DO BEGIN
01558      1  5          WHILE NOT EOF (ACIMPORTE) DO BEGIN
01559      1  6              WHILE NOT EOF (ACXSALDO) DO BEGIN
01560      1  6                  WRITE (CHR(27), 'O'); WRITE ('x'); WRITE (CHR(27), 'B');
01561      1  6                  FILA:=J; COLUMNA:=0; CURSOR;
01562      1  6                      WRITE (M);
01563      1  6                      FILA:=J; COLUMNA:=0;      (* NUM.FACTURA *)
01564      1  6                      CURSOR;
01565      1  6                      WRITE (CHR(27), 'i7D', 'nzm.', CHR(27), 'i6C');
01566      1  6                      READLN (FAC,S);
01567      1  6                      WRITE (S, CHR(27), 'i25D');
01568      1  6                      FILA:=J; COLUMNA:=0;
01569      1  6                      CURSOR;      (* FECHA *)
01570      1  6                      READLN (FAC,S);
01571      1  6                      WRITE (CHR(27), 'i25C', S);
01572      1  6                      WRITE (CHR(27), 'i36D');
01573      1  6                      FILA:=J; COLUMNA:=0;      (* CONCEPTO *)
01574      1  6                      CURSOR;
01575      1  6                      READLN (FAC,S);
01576      1  6                      WRITE (CHR(27), 'i36C', S);
01577      1  6                      WRITE (CHR(27), 'i58D');
01578      1  6                      FILA:=J; COLUMNA:=0;      (* IMPORTE *)
01579      1  6                      CURSOR;
01580      1  6                      WRITE (ACIMPORTE, IMPORTE);
01581      1  6                      WRITE (CHR(27), 'i58C');
01582      1  6                      WRITE (IMPORTE);
01583      1  6                      WRITE (CHR(27), 'i68D');
01584      1  6                      FILA:=J; COLUMNA:=0;      (* SALDO *)
01585      1  6                      CURSOR;
01586      1  6                      READ (ACXSALDO, SALDO);
01587      1  6                      WRITE (CHR(27), 'i68C', SALDO);
01588      1  6      WRITE (CHR(27), 'O'); WRITELN ('x'); WRITE (CHR(27), 'B');
01589      1  6          J:=J+1;      M:=M+1;
01590      1  7          IF M = 17*M THEN BEGIN
01591      1  7              WRITE (CHR(27), 'O', 'm'); FOR I:= 1 TO 78 DO WRITE ('q');
01592      1  7              WRITELN ('j'); WRITE (CHR(27), 'B');
01593      1  7                  WRITE ('Pulse <RETURN> para continuar
01594      1  7              WRITE (CHR(27), 'iOJ'); READ (CH); W:=W+1;
01595      1  7              WRITE (CHR(27), 'i5;1H');      WRITE (CHR(27), 'iOJ');
```

--LINE-IDC-PL-SL--

```
01596      1 6                                END;
01597      1 5      END;      (* fin del WHILE ... *)
01598      1 4      END;      (* fin del WHILE ... *)
01599      1 3 END;      (* fin del WHILE NOT (FAC) *)
01600      1 3      WRITE (CHR(27),'(O','m'); FOR I:= 1 TO 78 DO WRITE ('a');
01601      1 3      WRITELN ('j'); WRITE (CHR(27),'(B');
01602      1 3      WRITE ('Pulse <RETURN> para continuar ...');
01603      1 3      WRITE (CHR(27),'(OJ'); READ (CH); BORRADO;
01604      1 3      CLOSE (FAC);CLOSE (ACIMPORTE);CLOSE (ACSALDO);
01605      1 2 ACTOS;                                END;      (* fin del BEGIN ... 'C' *)
01606      C 1 2 (*****
                                *****
01607      1 2      'H','h': BEGIN AYUDA; ACTOS; END;
01608      1 2      'X','x': BORRADO;
01609      1 2      OTHERWISE BEGIN WRITE (CHR(7)); GOTO 10; END;
01610      1 1 END;      (* fin del begin del case primero *)
01611      0 0 END;
01612      C 0 0      (*****
01613      C 0 0      (***** DIETAS *****
01614      C 0 0      (*****
01615      1 0 PROCEDURE DIETAS;
01616      1 0 LABEL 10,11,AA,A,B,C,D,R;
01617      1 0 TYPE STRING= VARYING (300) OF CHAR;
01618      1 0 VAR FDIE: TEXT; DIESALDO,DIEIMPORTE: FILE OF INTEGER;
01619      1 0      S,FECHA,NFACTURA,CONCEPTO: STRING; M,J,N,I,W,IMPORTE,SALDO:
01620      1 1 BEGIN
01621      1 1 COMENTARIO:= ' CUENTA DE DIETAS '; OPCUENTAS;
01622      1 2 JO: OPCION; CASE CH OF
01623      C 1 2 (*****, Inicializaci3n de la cuenta *****
01624      1 3      'I','i': BEGIN 11: BORRADO; WRITE (CHR(27),'i11;3H');
01625      1 3 WRITELN (
01626      1 3 'Pulse la "S" para confirmar la inicializaci3n, "N" en caso cont
01627      1 3 WRITE (' ':2,'(recuerde que todas las operaciones son borradas);
01628      1 4 WRITE (CHR(27),'(OJ'); READLN (CH); CASE CH OF
01629      1 5      'S','s': BEGIN ENCA;
01630      C 1 5 (***** introducci3n de datos *****
01631      1 5 INTRODATOS;
01632      1 6 REPEAT AA: WRITE (CHR(27),'i5;20H'); READLN (FECHA);
01633      1 6 IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO AA;
01634      1 5 UNTIL LENGTH (FECHA) < 10;
01635      1 5 WRITE (CHR(27),'(O'); WRITE (CHR(27),'i5;28H');
01636      1 5 WRITE (' x '); WRITE (CHR(27),'(B');
01637      1 5 CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada
01638      1 6 REPEAT WRITE (CHR(27),'i5;55H');
01639      1 6 WRITE (CHR(27),'(O'); WRITE (' x ');
01640      1 5 READ (IMPORTE); UNTIL IMPORTE < 100000000;
01641      1 5 WRITE (CHR(27),'(B'); SALDO:= IMPORTE;
01642      C 1 5 (***** fin de la introducci3n de los datos ***
01643      1 5 OPEN (FDIE,'FDIE.DAT'); OPEN (DIESALDO,'DIESALDO.DAT');
01644      1 5 OPEN (DIEIMPORTE,'DIEIMPORTE.DAT');
01645      1 5 REWRITE (FDIE); REWRITE (DIEIMPORTE); REWRITE (DIESALDO);
01646      1 5 WRITELN (FDIE,NFACTURA); WRITELN (FDIE,FECHA);WRITELN (FDIE,CONI
01647      1 5 WRITE (DIEIMPORTE,IMPORTE); WRITE (DIESALDO,IMPORTE);
01648      1 4 CLOSE (FDIE); CLOSE (DIEIMPORTE); CLOSE (DIESALDO); END;
01649      1 4      'N','n': BEGIN WRITELN; END;
01650      1 4      'H','h': BEGIN AYUDA; GOTO 11; END;
```

-LINE-IDC-PL-SL-

```
01651      1 4      OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END;
01652      1 3      END;
01653      1 2 DIETAS;      END; (* Desde Begin del 'i','I': *)
01654      C 1 2 (***** Agadir operaciones *****
01655      1 3      'A','a': BEGIN ENCA;
01656      1 3      OPEN (FDIE,'FDIE.DAT',HISTORY:= OLD);
01657      1 3      OPEN (DIEIMPORTE,'DIEIMPORTE.DAT',HISTORY:= OLD);
01658      1 3      OPEN (DIESALDO,'DIESALDO.DAT',HISTORY:= OLD);
01659      1 4 REPEAT WRITE (CHR(27),'i22;8H');
01660      1 4 WRITE ('Introduzca el nmero de operaciones: ');
01661      1 3 WRITE (CHR(27),'i0J'); READLN (N); UNTIL N < 16;
01662      1 5 FOR I:= 1 TO N DO BEGIN REPEAT
01663      1 5 A: FILA:= I+4; COLUMNA:= 10; CURSOR; READLN (NFACTURA);
01664      1 5      IF NFACTURA = '' THEN BEGIN WRITE (CHR(7)); GOTO A; END;
01665      1 4 UNTIL LENGTH (NFACTURA) < 10;
01666      1 5 REPEAT B: FILA:= I+4; COLUMNA:= 18; CURSOR;
01667      1 5 WRITE (CHR(27),'(O)'); WRITE ('x ');
01668      1 5 WRITE (CHR(27),'(B)'); READLN (FECHA);
01669      1 5      IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO B; END;
01670      1 4 UNTIL LENGTH (FECHA) < 10;
01671      1 5 REPEAT C: FILA:= I+4; COLUMNA:= 28; CURSOR;
01672      1 5 WRITE (CHR(27),'(O)'); WRITE (' x ');
01673      1 5 WRITE (CHR(27),'(B)'); READLN (CONCEPTO);
01674      1 5      IF CONCEPTO = '' THEN BEGIN WRITE (CHR(7)); GOTO C; END;
01675      1 4 UNTIL LENGTH (CONCEPTO) < 21;
01676      1 5 REPEAT D: FILA:= I+4; COLUMNA:= 55; CURSOR;
01677      1 5 WRITE (CHR(27),'(O)'); WRITE (' x ');
01678      1 5 WRITE (CHR(27),'(B)'); READLN (IMPORTE);
01679      1 5      IF IMPORTE < 0 THEN BEGIN WRITE (CHR(7)); GOTO D; END;
01680      1 4 UNTIL IMPORTE < 100000000;
01681      1 4 R: WRITE (CHR(27),'i22;8H');
01682      1 4 WRITE ('Confirme s no la operacion ("S" s "N") : ');
01683      1 4 WRITE (CHR(27),'i0J'); READLN (CH);
01684      1 5 CASE CH OF      'N','n': FILA:= FILA-1;
01685      1 6      'S','s': BEGIN
01686      1 6      EXTEND (FDIE); EXTEND (DIEIMPORTE);
01687      1 6      WRITELN (FDIE,NFACTURA); WRITELN (FDIE,FECHA);
01688      1 6      WRITELN (FDIE,CONCEPTO); WRITE (DIEIMPORTE,IMPORTE);
01689      1 6      RESET (DIESALDO); WHILE NOT EOF (DIESALDO) DO
01690      1 6      READ (DIESALDO,SALDO); SALDO:= SALDO - IMPORTE;
01691      1 5 EXTEND (DIESALDO); WRITE (DIESALDO,SALDO); END;
01692      1 5      OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
01693      1 4      END; (* fin del case ch of *)
01694      1 3      END; (* fin del FOR *)
01695      1 3 CLOSE (FDIE); CLOSE (DIEIMPORTE); CLOSE (DIESALDO);
01696      1 2 DIETAS;      END;
01697      C 1 2 (***** Consulta de la cuenta *****
01698      1 3      'C','c': BEGIN ENCABEZAMIENTO;
01699      1 3 OPEN (FDIE,'FDIE.DAT',HISTORY:=OLD); RESET (FDIE);
01700      1 3 OPEN (DIEIMPORTE,'DIEIMPORTE.DAT',HISTORY:=OLD); RESET (DIEIMPOR
01701      1 3 OPEN (DIESALDO,'DIESALDO.DAT',HISTORY:=OLD); RESET (DIESALDO);
01702      1 3 J:=5; M:=1; W:=1; WRITE (CHR(27),'i5;1H');
01703      1 4 WHILE NOT EOF (FDIE) DO BEGIN
01704      1 5      WHILE NOT EOF (DIEIMPORTE) DO BEGIN
01705      1 6      WHILE NOT EOF (DIESALDO) DO BEGIN
```

-LINE-IDC-PL-SL-

```
01706      1 6 WRITE (CHR(27),'(O)'); WRITE ('x'); WRITE (CHR(27),'(B)');
01707      1 6      FILA:=J; COLUMNA:=0; CURSOR;
01708      1 6      WRITE (M);
01709      1 6      FILA:=J; COLUMNA:=0; (* NUM.FACTURA *)
01710      1 6      CURSOR;
01711      1 6 WRITE (CHR(27),'i7D', 'nzm.',CHR(27),'i6C');
01712      1 6      READLN (FDIE,S);
01713      1 6      WRITE (S,CHR(27),'i25D');
01714      1 6      FILA:=J; COLUMNA:=0;
01715      1 6      CURSOR; (* FECHA *)
01716      1 6      READLN (FDIE,S);
01717      1 6      WRITE (CHR(27),'i25C',S);
01718      1 6      WRITE (CHR(27),'i36D');
01719      1 6      FILA:=J; COLUMNA:=0; (* CONCEPTO *)
01720      1 6      CURSOR;
01721      1 6      READLN (FDIE,S);
01722      1 6      WRITE (CHR(27),'i36C',S);
01723      1 6      WRITE (CHR(27),'i58D');
01724      1 6      FILA:=J; COLUMNA:=0; (* IMPORTE *)
01725      1 6      CURSOR;
01726      1 6      WRITE (CHR(27),'i58C');
01727      1 6      READ (DIEIMPORTE,IMPORTE);
01728      1 6      WRITE (IMPORTE,CHR(27),'i68D');
01729      1 6      FILA:=J; COLUMNA:=0; (* SALDO *)
01730      1 6      CURSOR;
01731      1 6      READ (DIESALDO,SALDO);
01732      1 6      WRITE (CHR(27),'i68C',SALDO);
01733      1 6 WRITE (CHR(27),'(O)'); WRITELN ('x'); WRITE (CHR(27),'(B)');
01734      1 6      J:=J+1; M:=M+1;
01735      1 7      IF M = 17*W THEN BEGIN
01736      1 7      WRITE (CHR(27),'(O','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
01737      1 7      WRITELN ('j'); WRITE (CHR(27),'(B)');
01738      1 7      WRITE ('Pzise <RETURN> para continuar
01739      1 7      WRITE (CHR(27),'i0J'); READ (CH); W:=W+1;
01740      1 7      WRITE (CHR(27),'i5;1H'); WRITE (CHR(27),'i0J')
01741      1 6      END;
01742      1 5      END; (* fin del WHILE ... *)
01743      1 4      END; (* fin del WHILE ... *)
01744      1 3 END; (* fin del WHILE NOT (FDIE) *)
01745      1 3      WRITE (CHR(27),'(O','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
01746      1 3      WRITELN ('j'); WRITE (CHR(27),'(B)');
01747      1 3 WRITE ('Pzise <RETURN> para continuar ...');
01748      1 3      WRITE (CHR(27),'i0J'); READ (CH); BORRADO;
01749      1 3 CLOSE (FDIE); CLOSE (DIEIMPORTE); CLOSE (DIESALDO);
01750      1 2 DIETAS; END; (* fin del BEGIN ... 'C' *)
01751      C 1 2 (*****
01752      1 2      'H','h': BEGIN AYUDA; DIETAS; END;
01753      1 2      'X','x': BORRADO;
01754      1 2      OTHERWISE BEGIN WRITE (CHR(7)); GOTO 10; END;
01755      1 1 END; (* fin del begin del case primero *)
01756      0 0 END;
01757      C 0 0      (*****
01758      C 0 0      (***** GASTOS *****
01759      C 0 0      (*****
01760      1 0 PROCEDURE GASTOS;
```

--LINE--IDC--PL--SL--

```
01761      1  0 LABEL 10,11,AA,A,B,C,D,R;
01762      1  0 TYPE STRING= VARYING (300) OF CHAR;
01763      1  0 VAR FGAS: TEXT;      GASSALDO,GASIMPORTE: FILE OF INTEGER;
01764      1  0      S,FECHA,NFACTURA,CONCEPTO: STRING;      M,J,N,I,W,IMPORTE,SALD(
01765      1  1 BEGIN
01766      1  1 COMENTARIO:= ' CUENTA DE GASTOS DE OFICINA ';
01767      1  2 OPCUENTAS; 10: OPCION; CASE CH OF
01768  C 1  2 (*****,      Inicializacion de la cuenta      *****)
01769      1  3      'I','i':      BEGIN
01770      1  3 11: BORRADO; WRITE (CHR(27),'i11;3H');
01771      1  3 WRITELN (
01772      1  3 'Pulse la "S" para confirmar la inicializacion, "N" en caso con
01773      1  3 WRITE (' ':2,'(recuerde que todas las operaciones son borradas)
01774      1  3 WRITE (CHR(27),'i0J'); READLN (CH);
01775      1  5 CASE CH OF      'S','s': BEGIN ENCA;
01776  C 1  5 (*****      introduccion de datos      *****)
01777      1  6 INTRODATOS; REPEAT
01778      1  6 AA: .WRITE (CHR(27),'i5;20H'); READLN (FECHA);
01779      1  6      IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO AA; END;
01780      1  5 UNTIL LENGTH (FECHA) < 10; WRITE (CHR(27),'(O');
01781      1  5 WRITE (CHR(27),'i5;28H'); WRITE (' x '); WRITE (CHR(27),'
01782      1  5 CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada
01783      1  6 REPEAT WRITE (CHR(27),'i5;55H');
01784      1  6 WRITE (CHR(27),'(O'); WRITE (' x '); READ (IMPORTE);
01785      1  5 UNTIL IMPORTE < 100000000; WRITE (CHR(27),'(B'); SALDO:= IM
01786  C 1  5 (*****      fin de la introduccion de los datos      *
01787      1  5 OPEN (FGAS,'FGAS.DAT'); OPEN (GASSALDO,'GASSALDO.DAT');
01788      1  5 OPEN (GASIMPORTE,'GASIMPORTE.DAT');
01789      1  5 REWRITE (FGAS); REWRITE (GASIMPORTE); REWRITE (GASSALDO);
01790      1  5 WRITELN (FGAS,NFACTURA); WRITELN (FGAS,FECHA);
01791      1  5 WRITELN (FGAS,CONCEPTO); WRITE (GASIMPORTE,IMPORTE);
01792      1  5 WRITE (GASSALDO,IMPORTE);
01793      1  4 CLOSE (FGAS); CLOSE (GASIMPORTE); CLOSE (GASSALDO); .END;
01794      1  4      'N','n': BEGIN WRITELN; END;
01795      1  4      'H','h': BEGIN AYUDA; GOTO 11; END;
01796      1  4      OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END;
01797      1  3      END;      GASTOS;
01798      1  2      END; (* Desde Begin del 'i','I': *)
01799  C 1  2 (*****      Agadir operaciones      *****)
01800      1  3      'A','a': BEGIN ENCA;
01801      1  3 OPEN (FGAS,'FGAS.DAT',HISTORY:= OLD);
01802      1  3 OPEN (GASIMPORTE,'GASIMPORTE.DAT',HISTORY:= OLD);
01803      1  3 OPEN (GASSALDO,'GASSALDO.DAT',HISTORY:= OLD);
01804      1  4 REPEAT WRITE (CHR(27),'i22;8H');
01805      1  4 WRITE ('Introduzca el nmero de operaciones: ');
01806      1  3 WRITE (CHR(27),'i0J'); READLN (N); UNTIL N < 16;
01807      1  5 FOR I:= 1 TO N DO BEGIN REPEAT
01808      1  5 A: FILA:= I+4; COLUMNA:= 10; CURSOR; READLN (NFACTURA);
01809      1  5 IF NFACTURA = '' THEN BEGIN WRITE (CHR(7)); GOTO A; END;
01810      1  5 UNTIL LENGTH (NFACTURA) < 10; REPEAT
01811      1  5 B: FILA:= I+4; COLUMNA:= 18; CURSOR;
01812      1  5 WRITE (CHR(27),'(O'); WRITE ('x ');
01813      1  5 WRITE (CHR(27),'(B'); READLN (FECHA);
01814      1  5 IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO B; END;
01815      1  4 UNTIL LENGTH (FECHA) < 10;
```

-LINE-IDC-PL-SL-

```

01816      1 5 REPEAT C: FILA:= I+4; COLUMNA:= 28; CURSOR;
01817      1 5 WRITE (CHR(27),'(O)'); WRITE (' x '); WRITE (CHR(27),'(B)');
01818      1 5 READLN (CONCEPTO);
01819      1 5 IF CONCEPTO = '?' THEN BEGIN WRITE (CHR(7)); GOTO C; END;
01820      1 4 UNTIL LENGTH (CONCEPTO) < 21;
01821      1 5 REPEAT D: FILA:= I+4; COLUMNA:= 55; CURSOR;
01822      1 5 WRITE (CHR(27),'(O)'); WRITE (' x ');
01823      1 5 WRITE (CHR(27),'(B)'); READLN (IMPORTE);
01824      1 5 IF IMPORTE < 0 THEN BEGIN WRITE (CHR(7)); GOTO D; END;
01825      1 4 UNTIL IMPORTE < 100000000;
01826      1 4 R: WRITE (CHR(27),'i22;8H');
01827      1 4 WRITE ('Confirme s no la operaci3n ("S" s "N") : ');
01828      1 4 WRITE (CHR(27),'i0J'); READLN (CH);
01829      1 5 CASE CH OF 'N','n': FILA:= FILA-1;
01830              'S','s': BEGIN
01831          1 6 EXTEND (FGAS); EXTEND (GASIMPORTE);
01832          1 6 WRITELN (FGAS,NFACTURA); WRITELN (FGAS,FECHA);
01833          1 6 WRITELN (FGAS,CONCEPTO); WRITE (GASIMPORTE,IMPORTE);
01834          1 6 RESET (GASSALDO); WHILE NOT EOF (GASSALDO) DO
01835          1 6 READ (GASSALDO,SALDO); SALDO:= SALDO - IMPORTE;
01836          1 6 EXTEND (GASSALDO); WRITE (GASSALDO,SALDO);
01837          1 5 END;
01838          1 5 OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
01839      1 4 END; (* fin del case ch of *)
01840      1 3 END; (* fin del FOR *)
01841      1 3 CLOSE (FGAS); CLOSE (GASIMPORTE); CLOSE (GASSALDO);
01842      1 2 GASTOS; END;
01843      C 1 2 (***** Consulta de la cuenta *****
01844      1 3 'C','c': BEGIN ENDABEZAMIENTO;
01845      1 3 OPEN (FGAS,'FGAS.DAT',HISTORY:=OLD); RESET (FGAS);
01846      1 3 OPEN (GASIMPORTE,'GASIMPORTE.DAT',HISTORY:=OLD); RESET (GASIMPORTE);
01847      1 3 OPEN (GASSALDO,'GASSALDO.DAT',HISTORY:=OLD); RESET (GASSALDO);
01848      1 3 J:=5; M:=1; W:=1; WRITE (CHR(27),'i5;1H');
01849      1 4 WHILE NOT EOF (FGAS) DO BEGIN
01850          1 5 WHILE NOT EOF (GASIMPORTE) DO BEGIN
01851              1 6 WHILE NOT EOF (GASSALDO) DO BEGIN
01852                  1 6 WRITE (CHR(27),'(O)'); WRITE ('x'); WRITE (CHR(27),'(B)');
01853                  1 6 FILA:=J; COLUMNA:=0; CURSOR;
01854                  1 6 WRITE (M);
01855                  1 6 FILA:=J; COLUMNA:=0; (* NUM.FACTURA *)
01856                  1 6 CURSOR;
01857                  1 6 WRITE (CHR(27),'i7D','nzm.',CHR(27),'i6C');
01858                  1 6 READLN (FGAS,S);
01859                  1 6 WRITE (S,CHR(27),'i25D');
01860                  1 6 FILA:=J; COLUMNA:=0;
01861                  1 6 CURSOR; (* FECHA *)
01862                  1 6 READLN (FGAS,S);
01863                  1 6 WRITE (CHR(27),'i25C',S);
01864                  1 6 WRITE (CHR(27),'i36D');
01865                  1 6 FILA:=J; COLUMNA:=0; (* CONCEPTO *)
01866                  1 6 CURSOR;
01867                  1 6 READLN (FGAS,S);
01868                  1 6 WRITE (CHR(27),'i36C',S);
01869                  1 6 WRITE (CHR(27),'i58D');
01870                  1 6 FILA:=J; COLUMNA:=0; (* IMPORTE *)

```

-LINE-IDC-PL-SL-

```

01871      1   6           CURSOR;
01872      1   6           WRITE (CHR(27),'i58C');
01873      1   6           READ (GASIMPORTE,IMPORTE);
01874      1   6           WRITE (IMPORTE,CHR(27),'i68D');
01875      1   6           FILA:=J; COLUMNA:=0;           (* SALDO *)
01876      1   6           CURSOR;
01877      1   6           READ (GASSALDO,SALDO);
01878      1   6           WRITE (CHR(27),'i68C',SALDO);
01879      1   6           WRITE (CHR(27),'(0)'; WRITELN (' x'); WRITE (CHR(27),'(B)';
01880      1   6           J:=J+1;           M:=M+1;
01881      1   7           IF M = 17*W THEN BEGIN
01882      1   7           WRITE (CHR(27),'(0','m)'; FOR I:= 1 TO 78 DO WRITE ('q');
01883      1   7           WRITELN ('j'); WRITE (CHR(27),'(B)';
01884      1   7           WRITE ('Pulse <RETURN> para continuar
01885      1   7           WRITE (CHR(27),'i0J'); READ (CH); W:=W+1;
01886      1   7           WRITE (CHR(27),'i5;1H'); WRITE (CHR(27),'i0J')
01887      1   6           END;
01888      1   5           END;           (* fin del WHILE ... *)
01889      1   4           END;           (* fin del WHILE ... *)
01890      1   3           END;           (* fin del WHILE NOT (FGAS) *)
01891      1   3           WRITE (CHR(27),'(0','m)'; FOR I:= 1 TO 78 DO WRITE ('q');
01892      1   3           WRITELN ('j'); WRITE (CHR(27),'(B)';
01893      1   3           WRITE ('Pulse <RETURN> para continuar ...');
01894      1   3           WRITE (CHR(27),'i0J'); READ (CH); BORRADO;
01895      1   3           CLOSE (FGAS); CLOSE (GASIMPORTE); CLOSE (GASSALDO);
01896      1   2           GASTOS;           END;           (* fin del BEGIN ... 'C'
01897      C   1   2           (*****
01898      1   2           'H','h': BEGIN AYUDA; GASTOS; END;
01899      1   2           'X','x': BORRADO;
01900      1   2           OTHERWISE BEGIN WRITE (CHR(7));GOTO 10; END;
01901      0   0           END; END;
01902      C   0   0           (*****
01903      C   0   0           (***** CARTEL *****
01904      C   0   0           (*****
01905      1   0           PROCEDURE CARTEL;
01906      1   0           LABEL 10,11,AA,A,B,C,D,R;
01907      1   0           TYPE STRING= VARYING i3000 OF CHAR;
01908      1   0           VAR FTCL: TEXT; TELSALDO,TELIMPORTE: FILE OF INTEGER;
01909      1   0           S,FECHA,NFACTURA,CONCEPTO: STRING; M,J,N,I,W,IMPORTE,SALDO;
01910      1   1           BEGIN
01911      1   1           COMENTARIO:= ' CUENTA DE LA CARRERA DE TELECOMUNICACIONES ';
01912      1   1           OPCUENTAS; 10: OPCION;
01913      1   2           CASE CH OF
01914      C   1   2           (*****, Inicializaciorn de la cuenta *****
01915      1   3           'I','i': BEGIN
01916      1   3           11: BORRADO; WRITE (CHR(27),'i11;3H');
01917      1   3           WRITELN (
01918      1   3           'Pulse la "S" para confirmar la inicializaciorn, "N" en caso con
01919      1   3           WRITE (' ':2,'(recuerde que todas las operaciones son borradas)
01920      1   3           WRITE (CHR(27),'i0J'); READLN (CH);
01921      1   4           CASE CH OF
01922      1   5           'S','s': BEGIN ENCA;
01923      C   1   5           (***** introduccion de datos *****
01924      1   5           INTRODATOS;
01925      1   6           REPEAT AA: WRITE (CHR(27),'i5;20H'); READLN (FECHA);

```

-LINE-IDC-PL-SL-

```
01926      1  6          IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO AA;
01927      1  5 UNTIL LENGTH (FECHA) < 10;
01928      1  5 WRITE (CHR(27),'(O)'); WRITE (CHR(27),'i5;28H');
01929      1  5 WRITE (' x '); WRITE (CHR(27),'(B)');
01930      1  5 CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada
01931      1  6 REPEAT WRITE (CHR(27),'i5;55H');
01932      1  6 WRITE (CHR(27),'(O)'); WRITE (' x ');
01933      1  5 READ (IMPORTE); UNTIL IMPORTE < 100000000;
01934      1  5 WRITE (CHR(27),'(B)'); SALDO:= IMPORTE;
01935      C 1  5 (***** fin de la introduccion de los datos ***
01936      1  5 OPEN (FTEL,'FTEL.DAT');OPEN (TELSALDO,'TELSALDO.DAT');
01937      1  5 OPEN (TELIMPORTE,'TELIMPORTE.DAT');
01938      1  5 REWRITE (FTEL); REWRITE (TELIMPORTE); REWRITE (TELSALDO);
01939      1  5 WRITELN (FTEL,NFACTURA); WRITELN (FTEL,FECHA);
01940      1  5 WRITELN (FTEL,CONCEPTO); WRITE (TELIMPORTE,IMPORTE);
01941      1  5 WRITE (TELSALDO,IMPORTE);
01942      1  4 CLOSE (FTEL); CLOSE (TELIMPORTE); CLOSE (TELSALDO); END;
01943      1  4 'N','n': BEGIN WRITELN; END;
01944      1  4 'H','h': BEGIN AYUDA; GOTO 11; END;
01945      1  4 OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END;
01946      1  3 END;
01947      1  2 CARTEL; END; (* Desde Begin del 'i','I': *)
01948      C 1  2 (***** Agadir operaciones *****
01949      1  3 'A','a': BEGIN ENCA;
01950      1  3 OPEN (FTEL,'FTEL.DAT',HISTORY:= OLD);
01951      1  3 OPEN (TELIMPORTE,'TELIMPORTE.DAT',HISTORY:= OLD);
01952      1  3 OPEN (TELSALDO,'TELSALDO.DAT',HISTORY:= OLD);
01953      1  4 REPEAT WRITE (CHR(27),'i22;8H');
01954      1  4 WRITE ('Introduzca el nzmero de operaciones: ');
01955      1  3 WRITE (CHR(27),'iOJ'); READLN (N); UNTIL N < 16;
01956      1  4 FOR I:= 1 TO N DO BEGIN
01957      1  5 REPEAT A: FILA:= I+4; COLUMNA:= 10;
01958      1  5 CURSOR; READLN (NFACTURA);
01959      1  5 IF NFACTURA = '' THEN BEGIN WRITE (CHR(7)); GOTO A; END;
01960      1  4 UNTIL LENGTH (NFACTURA) < 10;
01961      1  5 REPEAT B: FILA:= I+4; COLUMNA:= 18;
01962      1  5 CURSOR; WRITE (CHR(27),'(O)');
01963      1  5 WRITE ('x '); WRITE (CHR(27),'(B)'); READLN (FECHA);
01964      1  5 IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO B; END;
01965      1  4 UNTIL LENGTH (FECHA) < 10;
01966      1  5 REPEAT C: FILA:= I+4; COLUMNA:= 28;
01967      1  5 CURSOR; WRITE (CHR(27),'(O)');
01968      1  5 WRITE (' x '); WRITE (CHR(27),'(B)'); READLN (CONCEPTO);
01969      1  5 IF CONCEPTO = '' THEN BEGIN WRITE (CHR(7)); GOTO C; END
01970      1  4 UNTIL LENGTH (CONCEPTO) < 21;
01971      1  5 REPEAT D: FILA:= I+4; COLUMNA:= 55;
01972      1  5 CURSOR; WRITE (CHR(27),'(O)'); WRITE (' x ');
01973      1  5 WRITE (CHR(27),'(B)'); READLN (IMPORTE);
01974      1  5 IF IMPORTE < 0 THEN BEGIN WRITE (CHR(7)); GOTO D; END
01975      1  4 UNTIL IMPORTE < 100000000;
01976      1  4 R: WRITE (CHR(27),'i22;8H');
01977      1  4 WRITE ('Confirme s no la operacisn ("S" s "N") : ');
01978      1  4 WRITE (CHR(27),'iOJ'); READLN (CH);
01979      1  5 CASE CH OF 'N','n': FILA:= FILA-1;
01980      1  6 'S','s': BEGIN
```

--LINE--IDC--PL--SL--

```
01981      1  6  EXTEND (FTEL); EXTEND (TELIMPORTE);
01982      1  6  WRITELN (FTEL,NFACTURA); WRITELN (FTEL,FECHA);
01983      1  6  WRITELN (FTEL,CONCEPTO); WRITE (TELIMPORTE,IMPORTE);
01984      1  6  RESET (TELSALDO); WHILE NOT EOF (TELSALDO) DO
01985      1  6  READ (TELSALDO,SALDO); SALDO:= SALDO - IMPORTE;
01986      1  5  EXTEND (TELSALDO); WRITE (TELSALDO,SALDO); END;
01987      1  5  OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
01988      1  4  END; (* fin del case ch of *)
01989      1  3  END; (* fin del FOR *)
01990      1  3  CLOSE (FTEL); CLOSE (TELIMPORTE); CLOSE (TELSALDO);
01991      1  2  CARTEL; END;
01992      C 1  2  (***** Consulta de la cuenta *****)
01993      1  3  'C','c': BEGIN ENCABEZAMIENTO;
01994      1  3  OPEN (FTEL,'FTEL.DAT',HISTORY:=OLD); RESET (FTEL);
01995      1  3  OPEN (TELIMPORTE,'TELIMPORTE.DAT',HISTORY:=OLD); RESET (TELIMPORTE);
01996      1  3  OPEN (TELSALDO,'TELSALDO.DAT',HISTORY:=OLD); RESET (TELSALDO);
01997      1  3  J:=5; M:=1; W:=1; WRITE (CHR(27),'i5;1H');
01998      1  4  WHILE NOT EOF (FTEL) DO BEGIN
01999      1  5  WHILE NOT EOF (TELIMPORTE) DO BEGIN
02000      1  6  WHILE NOT EOF (TELSALDO) DO BEGIN
02001      1  6  WRITE (CHR(27),'(0)'); WRITE ('x'); WRITE (CHR(27),'(B)');
02002      1  6  FILA:=J; COLUMNA:=0; CURSOR;
02003      1  6  WRITE (M);
02004      1  6  FILA:=J; COLUMNA:=0; (* NUM.FACTURA *)
02005      1  6  CURSOR;
02006      1  6  WRITE (CHR(27),'i7D','nzm.',CHR(27),'i6C');
02007      1  6  READLN (FTEL,S);
02008      1  6  WRITE (S,CHR(27),'i25D');
02009      1  6  FILA:=J; COLUMNA:=0;
02010      1  6  CURSOR; (* FECHA *)
02011      1  6  READLN (FTEL,S);
02012      1  6  WRITE (CHR(27),'i25C',S);
02013      1  6  WRITE (CHR(27),'i36D');
02014      1  6  FILA:=J; COLUMNA:=0; (* CONCEPTO *)
02015      1  6  CURSOR;
02016      1  6  READLN (FTEL,S);
02017      1  6  WRITE (CHR(27),'i36C',S);
02018      1  6  WRITE (CHR(27),'i58D');
02019      1  6  FILA:=J; COLUMNA:=0; (* IMPORTE *)
02020      1  6  CURSOR;
02021      1  6  WRITE (CHR(27),'i58C');
02022      1  6  READ (TELIMPORTE,IMPORTE);
02023      1  6  WRITE (IMPORTE,CHR(27),'i68D');
02024      1  6  FILA:=J; COLUMNA:=0; (* SALDO *)
02025      1  6  CURSOR;
02026      1  6  READ (TELSALDO,SALDO);
02027      1  6  WRITE (CHR(27),'i68C',SALDO);
02028      1  6  J:=J+1; M:=M+1;
02029      1  6  WRITE (CHR(27),'(0)'); WRITELN ('x'); WRITE (CHR(27),'(B)');
02030      1  7  IF M = 17*W THEN BEGIN
02031      1  7  WRITE (CHR(27),'(0','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
02032      1  7  WRITELN ('j'); WRITE (CHR(27),'(B)');
02033      1  7  WRITE ('Pulse <RETURN> para continuar');
02034      1  7  WRITE (CHR(27),'i0J'); READ (CH); W:=W+1;
02035      1  7  WRITE (CHR(27),'i5;1H'); WRITE (CHR(27),'i0J');
```

-LINE-IDC-PL-SL-

```
02036      1 6                                END;
02037      1 5      END;      (* fin del WHILE ... *)
02038      1 4      END;      (* fin del WHILE ... *)
02039      1 3 END;      (* fin del WHILE NOT (FTEL) *)
02040      1 3      WRITE (CHR(27),'(O','m'); FOR I:= 1 TO 78 DO WRITE ('q');
02041      1 3      WRITELN ('j'); WRITE (CHR(27),'(B');
02042      1 3      WRITE ('Pzise <RETURN> para continuar ...');
02043      1 3      WRITE (CHR(27),'(iOJ'); READ (CH); BORRADO;
02044      1 3      CLOSE (FTEL); CLOSE (TELIMPORTE); CLOSE (TELSALDO);
02045      1 2      CARTEL;                                END;      (* fin del BEGIN ... 'C'
02046      C 1 2      (*****
                                *****
02047      1 2      'H','h': BEGIN AYUDA; CARTEL; END;
02048      1 2      'X','x': BORRADO;
02049      1 2      OTHERWISE BEGIN WRITE (CHR(7)); GOTO 10; END;
02050      1 1 END;      (* fin del begin del case primero *)
02051      0 0 END;
02052      C 0 0      (*****
02053      C 0 0      (*****      EQUIP      *****
02054      C 0 0      (*****
02055      1 0 PROCEDURE EQUIP;
02056      1 0 LABEL 10,11,AA,A,B,C,D,R;
02057      1 0 TYPE STRING= VARYING i3000 OF CHAR;
02058      1 0 VAR FEQ: TEXT; EQSALDO,EQIMPORTE: FILE OF INTEGER;
02059      1 0      S,FECHA,FACTURA,CONCEPTO: STRING;
02060      1 0      M,J,N,I,W,IMPORTE,SALDO: INTEGER;
02061      1 1      BEGIN
02062      1 1      COMENTARIO:= ' CUENTA DE EQUIPOS ELECTRONICOS ';
02063      1 1      OPCUENTAS;10: OPCION;
02064      1 2      CASE CH OF
02065      C 1 2      (*****,      Inicializaci0n de la cuenta      *****
02066      1 3      'I','i': BEGIN
02067      1 3      11: BORRADO;WRITE (CHR(27),'(i11;3H');
02068      1 3      WRITELN (
02069      1 3      'Pzise la "S" para confirmar la inicializaci0n, "N" en caso con
02070      1 3      WRITE (' ':2,'(recuerde que todas las operaciones son borradas)
02071      1 3      WRITE (CHR(27),'(iOJ');READLN (CH);
02072      1 5      CASE CH OF      'S','s': BEGIN ENCA;
02073      C 1 5      (*****      introducci0n de datos      *****
02074      1 5      INTRODATOS;
02075      1 6      REPEAT AA: WRITE (CHR(27),'(i5;20H');
02076      1 6      READLN (FECHA);
02077      1 7      IF FECHA = '' THEN BEGIN WRITE (CHR(7));
02078      1 6      GOTO AA;                                END;
02079      1 5      UNTIL LENGTH (FECHA) < 10;
02080      1 5      WRITE (CHR(27),'(O'); WRITE (CHR(27),'(i5;28H');
02081      1 5      WRITE (' x '); WRITE (CHR(27),'(B');
02082      1 5      CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada
02083      1 6      REPEAT WRITE (CHR(27),'(i5;55H');
02084      1 6      WRITE (CHR(27),'(O'); WRITE (' x ');
02085      1 5      READ (IMPORTE); UNTIL IMPORTE < 100000000;
02086      1 5      WRITE (CHR(27),'(B'); SALDO:= IMPORTE;
02087      C 1 5      (*****      fin de la introducci0n de los datos      ***
02088      1 5      OPEN (FEQ,'FEQ.DAT'); OPEN (EQSALDO,'EQSALDO.DAT');
02089      1 5      OPEN (EQIMPORTE,'EQIMPORTE.DAT');
02090      1 5      REWRITE (FEQ); REWRITE (EQIMPORTE); REWRITE (EQSALDO);
```

-LINE-IDC-PL-SL-

```
02091      1 5  WRITELN (FEQ,NFACTURA); WRITELN (FEQ,FECHA);
02092      1 5  WRITELN (FEQ,CONCEPTO); WRITE (EQIMPORTE,IMPORTE);
02093      1 5      WRITE (EQSALDO,IMPORTE);
02094      1 4  CLOSE (FEQ); CLOSE (EQIMPORTE); CLOSE (EQSALDO); END;
02095      1 4      'N','n': BEGIN WRITELN; END;
02096      1 4      'H','h': BEGIN AYUDA; GOTO 11; END;
02097      1 4      OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END;
02098      1 3      END;
02099      1 2  EQUIP; END; (* Desde Begin del 'i','I': *)
02100  C 1 2  (***** Agadir operaciones *****)
02101      1 2      'A','a':
02102      1 3      BEGIN ENCA;
02103      1 3  OPEN (FEQ,'FEQ.DAT',HISTORY:= OLD);
02104      1 3  OPEN (EQIMPORTE,'EQIMPORTE.DAT',HISTORY:= OLD);
02105      1 3  OPEN (EQSALDO,'EQSALDO.DAT',HISTORY:= OLD);
02106      1 4  REPEAT WRITE (CHR(27),'i22;BH');
02107      1 4  WRITE ('Introduzca el numero de operaciones: ');
02108      1 4  WRITE (CHR(27),'i0J'); READLN (N);
02109      1 4  UNTIL N < 16; FOR I:= 1 TO N DO BEGIN
02110      1 5  REPEAT A: FILA:= I+4; COLUMNA:= 10;
02111      1 5  CURSOR; READLN (NFACTURA);
02112      1 5      IF NFACTURA = '' THEN BEGIN WRITE (CHR(7)); GOTO A;
02113      1 4  UNTIL LENGTH (NFACTURA) < 10;
02114      1 5  REPEAT B: FILA:= I+4; COLUMNA:= 18;
02115      1 5  CURSOR; WRITE (CHR(27),'(0)'); WRITE ('x ');
02116      1 5  WRITE (CHR(27),'(B)'); READLN (FECHA);
02117      1 5      IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO B; END;
02118      1 4  UNTIL LENGTH (FECHA) < 10;
02119      1 5  REPEAT C: FILA:= I+4; COLUMNA:= 28;
02120      1 5  CURSOR; WRITE (CHR(27),'(0)'); WRITE (' x ');
02121      1 5  WRITE (CHR(27),'(B)'); READLN (CONCEPTO);
02122      1 5      IF CONCEPTO = '' THEN BEGIN WRITE (CHR(7)); GOTO C; END
02123      1 4  UNTIL LENGTH (CONCEPTO) < 21;
02124      1 5  REPEAT D: FILA:= I+4; COLUMNA:= 55;
02125      1 5  CURSOR; WRITE (CHR(27),'(0)'); WRITE (' x ');
02126      1 5  WRITE (CHR(27),'(B)'); READLN (IMPORTE);
02127      1 5      IF IMPORTE < 0 THEN BEGIN WRITE (CHR(7)); GOTO D; END;
02128      1 4  UNTIL IMPORTE < 100000000;
02129      1 4  R: WRITE (CHR(27),'i22;BH');
02130      1 4  WRITE ('Confirme s no la operacion ("S" s "N") : ');
02131      1 4  WRITE (CHR(27),'i0J'); READLN (CH);
02132      1 5  CASE CH OF 'N','n': FILA:= FILA-1;
02133      1 6      'S','s': BEGIN
02134      1 6      EXTEND (FEQ); EXTEND (EQIMPORTE);
02135      1 6      WRITELN (FEQ,NFACTURA); WRITELN (FEQ,FECHA);
02136      1 6      WRITELN (FEQ,CONCEPTO); WRITE (EQIMPORTE,IMPORTE);
02137      1 6      RESET (EQSALDO); WHILE NOT EOF (EQSALDO) DO
02138      1 6      READ (EQSALDO,SALDO); SALDO:= SALDO - IMPORTE;
02139      1 5      EXTEND (EQSALDO); WRITE (EQSALDO,SALDO); END;
02140      1 5      OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
02141      1 4      END; (* fin del case ch of *)
02142      1 3      END; (* fin del FOR *)
02143      1 3  CLOSE (FEQ); CLOSE (EQIMPORTE); CLOSE (EQSALDO);
02144      1 2  EQUIP; END;
02145  C 1 2  (***** Consulta de la cuenta *****)
```

-LINE-IDC-PL-SL-

```
02146      1 3      'C','c': BEGIN      ENCABEZAMIENTO;
02147      1 3 OPEN (FEG,'FEG.DAT',HISTORY:=OLD);RESET (FEG);
02148      1 3 OPEN (EQIMPORTE,'EQIMPORTE.DAT',HISTORY:=OLD);RESET (EQIMPORTE);
02149      1 3 OPEN (EGSALDO,'EGSALDO.DAT',HISTORY:=OLD);RESET (EGSALDO);
02150      1 3      J:=5; M:=1; W:=1;      WRITE (CHR(27),'i5;1H');
02151      1 4 WHILE NOT EOF (FEG) DO BEGIN
02152      1 5          WHILE NOT EOF (EQIMPORTE) DO BEGIN
02153      1 6              WHILE NOT EOF (EGSALDO) DO BEGIN
02154      1 6 WRITE (CHR(27),'(0)'); WRITE ('x'); WRITE (CHR(27),'(B)');
02155      1 6          FILA:=J; COLUMNA:=0;      CURSOR;
02156      1 6              WRITE (M);
02157      1 6              FILA:=J; COLUMNA:=0;      (* NUM.FACTURA *)
02158      1 6              CURSOR;
02159      1 6 WRITE (CHR(27),'i7D','nzm.',CHR(27),'i6C');
02160      1 6 READLN (FEG,S);
02161      1 6 WRITE (S,CHR(27),'i25D');
02162      1 6 FILA:=J; COLUMNA:=0;
02163      1 6 CURSOR;      (* FECHA *)
02164      1 6 READLN (FEG,S);
02165      1 6 WRITE (CHR(27),'i25C',S);
02166      1 6 WRITE (CHR(27),'i36D');
02167      1 6 FILA:=J; COLUMNA:=0;      (* CONCEPTO *)
02168      1 6 CURSOR;
02169      1 6 READLN (FEG,S);
02170      1 6 WRITE (CHR(27),'i25C',S);
02171      1 6 WRITE (CHR(27),'i58D');
02172      1 6 FILA:=J; COLUMNA:=0;      (* IMPORTE *)
02173      1 6 CURSOR;
02174      1 6 WRITE (CHR(27),'i58C');
02175      1 6 READ (EQIMPORTE,IMPORTE);
02176      1 6 WRITE (IMPORTE,CHR(27),'i68D');
02177      1 6 FILA:=J; COLUMNA:=0;      (* SALDO *)
02178      1 6 CURSOR;
02179      1 6 READ (EGSALDO,SALDO);
02180      1 6 WRITE (CHR(27),'i68C',SALDO);
02181      1 6      J:=J+1;      M:=M+1;
02182      1 6 WRITE (CHR(27),'(0)'); WRITELN ('x'); WRITE (CHR(27),'(B)');
02183      1 7          IF M = 17*W THEN BEGIN
02184      1 7 WRITE (CHR(27),'(0','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
02185      1 7 WRITELN ('j'); WRITE (CHR(27),'(B)');
02186      1 7          WRITE ('Pzlse <RETURN> para continuar
02187      1 7          WRITE (CHR(27),'i0J');      READ (CH); W:=W+1;
02188      1 7 WRITE (CHR(27),'i5;1H');      WRITE (CHR(27),'i0J')
02189      1 6          END;
02190      1 3          END;      END;      END;
02191      1 3 WRITE (CHR(27),'(0','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
02192      1 3 WRITELN ('j'); WRITE (CHR(27),'(B)');
02193      1 3 WRITE ('Pzlse <RETURN> para continuar ...');
02194      1 3 WRITE (CHR(27),'i0J');      READ (CH); BORRADO;
02195      1 3 CLOSE (FEG); CLOSE (EQIMPORTE); CLOSE (EGSALDO);
02196      1 2 EQUIP;      END;      (* fin del BEGIN ... 'C' *)
02197      C 1 2 (*****      *****
02198      1 2      'H','h': BEGIN AYUDA;      EQUIP;      END;
02199      1 2      'X','x': BORRADO;
02200      1 2      OTHERWISE BEGIN WRITE (CHR(7)); GOTO 10; END;
```

--LINE--IDC--PL--SL--

```
02201      1  1  END;      (*      fin del begin del case primero *)
02202      0  0  END;
02203      C  0  0      (*****
02204      C  0  0      (*****      IMAGE      *****)
02205      C  0  0      (*****
02206      1  0  PROCEDURE IMAGE;
02207      1  0  LABEL 10,11,AA,A,B,C,D,R;
02208      1  0  TYPE STRING= VARYING (300) OF CHAR;
02209      1  0  VAR FIMA: TEXT;      IMASALDO,IMAIMPORTE: FILE OF INTEGER;
02210      1  0      S,FECHA,NFACTURA,CONCEPTO: STRING;
02211      1  0      M,J,N,I,W,IMPORTE,SALDO: INTEGER;
02212      1  1  BEGIN
02213      1  1  COMENTARIO:= ' CUENTA DE IMAGEN Y SONIDO ';
02214      1  2  OPCUENTAS;10: OPCION; CASE CH OF
02215      C  1  2  (*****,      Inicializaci3n de la cuenta      *****)
02216      1  3      'I','i':      BEGIN
02217      1  3  11: BORRADO; WRITE (CHR(27),'i11;3H');
02218      1  3  WRITELN (
02219      1  3  'Pyase la "S" para confirmar la inicializaci3n, "N" en caso cont
02220      1  3  WRITE (' ':2,'(recuerde que todas las operaciones son borradas);
02221      1  3  WRITE (CHR(27),'i0J'); READLN (CH);
02222      1  5      CASE CH OF      'S','s': BEGIN      ENCA;
02223      C  1  5  (*****      introducci3n de datos      *****)
02224      1  6  INTRODADOS;      REPEAT
02225      1  6  AA: WRITE (CHR(27),'i5;20H');      READLN (FECHA);
02226      1  6      IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO AA;      END;
02227      1  5      UNTIL LENGTH (FECHA) < 10;
02228      1  5      WRITE (CHR(27),'i0');      WRITE (CHR(27),'i5;28H');
02229      1  5      WRITE (' x ');      WRITE (CHR(27),'(B');
02230      1  5      CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada
02231      1  6      REPEAT      WRITE (CHR(27),'i5;55H');
02232      1  6      WRITE (CHR(27),'(0');      WRITE (' x ');
02233      1  5      READ (IMPORTE); UNTIL IMPORTE < 100000000;
02234      1  5      WRITE (CHR(27),'(B');      SALDO:= IMPORTE;
02235      C  1  5  (*****      fin de la introducci3n de los datos      ***
02236      1  5  OPEN (FIMA,'FIMA.DAT');OPEN (IMASALDO,'IMASALDO.DAT');
02237      1  5      OPEN (IMAIMPORTE,'IMAIMPORTE.DAT');
02238      1  5  REWRITE (FIMA); REWRITE (IMAIMPORTE);REWRITE (IMASALDO);
02239      1  5  WRITELN (FIMA,NFACTURA); WRITELN (FIMA,FECHA);
02240      1  5  WRITELN (FIMA,CONCEPTO); WRITE (IMAIMPORTE,IMPORTE);
02241      1  5      WRITE (IMASALDO,IMPORTE);
02242      1  4  CLOSE (FIMA); CLOSE (IMAIMPORTE); CLOSE (IMASALDO);      END;
02243      1  4      'N','n': BEGIN WRITELN; END;
02244      1  4      'H','h': BEGIN AYUDA; GOTO 11;      END;
02245      1  4      OTHERWISE BEGIN WRITE (CHR(7));      GOTO 11;      END;
02246      1  3      END;
02247      1  2  IMAGE;      END; (* Desde      Begin del      'i','I':      *)
02248      C  1  2  (*****      Agadir operaciones      *****)
02249      1  3      'A','a':      BEGIN
02250      1  3      ENCA;
02251      1  3      OPEN (FIMA,'FIMA.DAT',HISTORY:= OLD);
02252      1  3      OPEN (IMAIMPORTE,'IMAIMPORTE.DAT',HISTORY:= OLD);
02253      1  3      OPEN (IMASALDO,'IMASALDO.DAT',HISTORY:= OLD);
02254      1  4  REPEAT WRITE (CHR(27),'i22;8H');
02255      1  4  WRITE ('Introduzca el nzmero de operaciones: ');
```

-LINE-IDC-PL-SL-

```
02256      1  4 WRITE (CHR (27),';iOJ'); READLN (N);
02257      1  3 UNTIL N < 16; FOR I:= 1 TO N DO
02258      1  4          BEGIN
02259      1  5 REPEAT  A: FILA:= I+4; COLUMNA:= 10;
02260      1  5 CURSOR; READLN (NFACTURA);
02261      1  5 IF NFACTURA = '' THEN BEGIN WRITE (CHR(7)); GOTO A; END;
02262      1  4 UNTIL LENGTH (NFACTURA) < 10;
02263      1  5 REPEAT  B: FILA:= I+4; COLUMNA:= 18;
02264      1  5 CURSOR; WRITE (CHR(27),'(O)'); WRITE (' x ');
02265      1  5 WRITE (CHR(27),'(B)'); READLN (FECHA);
02266      1  6 IF FECHA = '' THEN BEGIN
02267      1  6         WRITE (CHR(7));
02268      1  6         GOTO B;
02269      1  5         END;
02270      1  4 UNTIL LENGTH (FECHA) < 10;
02271      1  5 REPEAT  C: FILA:= I+4; COLUMNA:= 28;
02272      1  5 CURSOR; WRITE (CHR(27),'(O)'); WRITE (' x ');
02273      1  5 WRITE (CHR(27),'(B)'); READLN (CONCEPTO);
02274      1  6 IF CONCEPTO = '' THEN BEGIN
02275      1  6         WRITE (CHR(7));
02276      1  6         GOTO C;
02277      1  5         END;
02278      1  4 UNTIL LENGTH (CONCEPTO) < 21;
02279      1  5 REPEAT  D: FILA:= I+4; COLUMNA:= 55;
02280      1  5 CURSOR; WRITE (CHR(27),'(O)');
02281      1  5 WRITE (' x ');
02282      1  5 WRITE (CHR(27),'(B)'); READLN (IMPORTE);
02283      1  6 IF IMPORTE < 0 THEN BEGIN
02284      1  6         WRITE (CHR(7));
02285      1  6         GOTO D;
02286      1  5         END;
02287      1  4 UNTIL IMPORTE < 100000000;
02288      1  4 R: WRITE (CHR(27),';i22;OH');
02289      1  4 WRITE ('Confirme s no la operacion ("S" s "N") : ');
02290      1  4 WRITE (CHR(27),';iOJ'); READLN (CH);
02291      1  5 CASE CH OF
02292      1  5         'S', 's':
02293      1  6         BEGIN
02294      1  6         EXTEND (FIMA); EXTEND (IMAIMPORTE);
02295      1  6         WRITELN (FIMA,NFACTURA); WRITELN (FIMA,FECHA);
02296      1  6         WRITELN (FIMA,CONCEPTO); WRITE (IMAIMPORTE,IMPORTE);
02297      1  6         RESET (IMASALDO); WHILE NOT EOF (IMASALDO) DO
02298      1  6         READ (IMASALDO,SALDO); SALDO:= SALDO - IMPORTE;
02299      1  6         EXTEND (IMASALDO); WRITE (IMASALDO,SALDO);
02300      1  5         END;
02301      1  6         OTHERWISE BEGIN WRITE (CHR(7));
02302      1  5         GOTO R; END;
02303      1  4         END; (* fin del case ch of *)
02304      1  3         END; (* fin del FOR *)
02305      1  3 CLOSE (FIMA); CLOSE (IMAIMPORTE); CLOSE (IMASALDO);
02306      1  2 IMAGE; END;
02307      C 1  2 (***** Consulta de la cuenta *****
02308      1  3 'C', 'c': BEGIN ENCABEZAMIENTO;
02309      1  3 OPEN (FIMA,'FIMA.DAT',HISTORY:=OLD); RESET (FIMA);
02310      1  3 OPEN (IMAIMPORTE,'IMAIMPORTE.DAT',HISTORY:=OLD); RESET (IMAIMPOR
```

--LINE--IDC--PL--SL--

```
02311      1 3 OPEN (IMASALDO,'IMASALDO.DAT',HISTORY:=OLD);RESET (IMASALDO);
02312      1 3           J:=5; M:=1; W:=1;           WRITE (CHR(27),'i5;1H');
02313      1 4 WHILE NOT EOF (FIMA) DO BEGIN
02314      1 5           WHILE NOT EOF (IMAIMPORTE) DO BEGIN
02315      1 6                   WHILE NOT EOF (IMASALDO) DO BEGIN
02316      1 6 WRITE (CHR(27),'(O)'); WRITE ('X'); WRITE (CHR(27),'(B)');
02317      1 6           FILA:=J; COLUMNNA:=0; CURSOR;
02318      1 6                   WRITE (M);
02319      1 6                   FILA:=J; COLUMNNA:=0;           (* NUM.FACTURA *)
02320      1 6                   CURSOR;
02321      1 6 WRITE (CHR(27),'i7D', 'nzm.',CHR(27),'i6C');
02322      1 6           READLN (FIMA,S);
02323      1 6           WRITE (S,CHR(27),'i25D');
02324      1 6           FILA:=J; COLUMNNA:=0;
02325      1 6           CURSOR;           (* FECHA *)
02326      1 6           READLN (FIMA,S);
02327      1 6           WRITE (CHR(27),'i25C',S);
02328      1 6           WRITE (CHR(27),'i36D');
02329      1 6           FILA:=J; COLUMNNA:=0;           (* CONCEPTO *)
02330      1 6           CURSOR;
02331      1 6           READLN (FIMA,S);
02332      1 6           WRITE (CHR(27),'i36C',S);
02333      1 6           WRITE (CHR(27),'i58D');
02334      1 6           FILA:=J; COLUMNNA:=0;           (* IMPORTE *)
02335      1 6           CURSOR;
02336      1 6           WRITE (CHR(27),'i58C');
02337      1 6           READ (IMAIMPORTE,IMPORTE);
02338      1 6           WRITE (IMPORTE,CHR(27),'i68D');
02339      1 6           FILA:=J; COLUMNNA:=0;           (* SALDO *)
02340      1 6           CURSOR;
02341      1 6           READ (IMASALDO,SALDO);
02342      1 6           WRITE (CHR(27),'i68C',SALDO);
02343      1 6           J:=J+1;           M:=M+1;
02344      1 6 WRITE (CHR(27),'(O)'); WRITELN (' X'); WRITE (CHR(27),'(B)');
02345      1 7           IF M = 17*W THEN BEGIN
02346      1 7 WRITE (CHR(27),'(O','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
02347      1 7 WRITELN ('j'); WRITE (CHR(27),'(B)');
02348      1 7           WRITE ('Pzlse <RETURN> para continuar
02349      1 7           WRITE (CHR(27),'i0J'); READ (CH); W:=W+1;
02350      1 7 WRITE (CHR(27),'i5;1H');           WRITE (CHR(27),'i0J')
02351      1 6           END;
02352      1 5           END;
02353      1 4           END;
02354      1 3 END;
02355      1 3 WRITE (CHR(27),'(O','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
02356      1 3 WRITELN ('j'); WRITE (CHR(27),'(B)');
02357      1 3 WRITE ('Pzlse <RETURN> para continuar ...');
02358      1 3           WRITE (CHR(27),'i0J'); READ (CH); BORRADO;
02359      1 3 CLOSE (FIMA); CLOSE (IMAIMPORTE); CLOSE (IMASALDO);
02360      1 2 IMAGE;           END;           (* fin del BEGIN ... 'C' *)
02361      C 1 2 (*****
02362      1 2           'H','h': BEGIN AYUDA; IMAGE; END;
02363      1 2           'X','x': BORRADO;
02364      1 2           OTHERWISE BEGIN WRITE (CHR(7)); GOTO 10; END;
02365      1 1 END; (* fin del begin del case primero *)
```

-LINE-IDC-PL-SL-

```
02366      0  0  END;
02367      C  0  0      (*****
02368      C  0  0      (*****      RADIO      *****)
02369      C  0  0      (*****
02370      1  0  PROCEDURE RADIO;
02371      1  0  LABEL 10,11,AA,A,B,C,D,R;
02372      1  0  TYPE STRING= VARYING (300) OF CHAR;
02373      1  0  VAR FRA: TEXT;      RASALDO,RAIMPORTE: FILE OF INTEGER;
02374      1  0      S,FECHA,NFACTURA,CONCEPTO: STRING;
02375      1  0      M,J,N,I,W,IMPORTE,SALDO: INTEGER;
02376      1  1  BEGIN COMENTARIO:= ' CUENTA DE RADIOCOMUNICACIONES ';
02377      1  1  OPCUENTAS; 10: OPCION;
02378      1  2  CASE CH OF
02379      C  1  2  (*****,      Inicializaci3n de la cuenta      *****)
02380      1  3      'I','i':      BEGIN
02381      1  3  11: BORRADO; WRITE (CHR(27),'i11;3H');
02382      1  3  WRITELN (
02383      1  3  'Pyise la "S" para confirmar la inicializaci3n, "N" en caso con
02384      1  3  WRITE (' ':2,'(recuerde que todas las operaciones son borradas)
02385      1  3  WRITE (CHR(27),'i0J'); READLN (CH);
02386      1  4      CASE CH OF
02387      1  5      'S','s': BEGIN ENCA;
02388      C  1  5  (*****      introduccion de datos      *****)
02389      1  5  INTRODATOS;
02390      1  6  REPEAT AA; WRITE (CHR(27),'i5;20H'); READLN (FECHA);
02391      1  7  IF FECHA = '' THEN BEGIN WRITE (CHR(7));
02392      1  6  GOTO AA; END;
02393      1  5  UNTIL LENGTH (FECHA) < 10;
02394      1  5  WRITE (CHR(27),'(0)'); WRITE (CHR(27),'i5;28H');
02395      1  5  WRITE (' x '); WRITE (CHR(27),'(B)');
02396      1  5  CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada
02397      1  6  REPEAT WRITE (CHR(27),'i5;55H');
02398      1  6  WRITE (CHR(27),'(0)'); WRITE (' x ');
02399      1  5  READ (IMPORTE); UNTIL IMPORTE < 100000000;
02400      1  5  WRITE (CHR(27),'(B)'); SALDO:= IMPORTE;
02401      C  1  5  (*****      fin de la introduccion de los datos      ***
02402      1  5  OPEN (FRA,'FRA.DAT'); OPEN (RASALDO,'RASALDO.DAT');
02403      1  5  OPEN (RAIMPORTE,'RAIMPORTE.DAT');
02404      1  5  REWRITE (FRA);REWRITE (RAIMPORTE);REWRITE (RASALDO);
02405      1  5  WRITELN (FRA,NFACTURA); WRITELN (FRA,FECHA);
02406      1  5  WRITELN (FRA,CONCEPTO); WRITE (RAIMPORTE,IMPORTE);
02407      1  5  WRITE (RASALDO,IMPORTE);
02408      1  4  CLOSE (FRA); CLOSE (RAIMPORTE); CLOSE (RASALDO); END;
02409      1  4      'N','n': BEGIN WRITELN; END;
02410      1  4      'H','h': BEGIN AYUDA; GOTO 11; END;
02411      1  4      OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END;
02412      1  3      END;
02413      1  2  RADIO;      END; (* Desde      Begin del 'i','I': *)
02414      C  1  2  (*****      Agadir operaciones      *****)
02415      1  3      'A','a':      BEGIN
02416      1  3      ENCA;
02417      1  3      OPEN (FRA,'FRA.DAT',HISTORY:= OLD);
02418      1  3      OPEN (RAIMPORTE,'RAIMPORTE.DAT',HISTORY:= OLD);
02419      1  3      OPEN (RASALDO,'RASALDO.DAT',HISTORY:= OLD);
02420      1  4  REPEAT WRITE (CHR(27),'i22;8H');
```

-LINE-IDC-PL-SL-

```
02421      1  4 WRITE ('Introduzca el nmero de operaciones: ');
02422      1  4 WRITE (CHR(27),'iOJ'); READLN (N);
02423      1  3 UNTIL N < 16; FOR I:= 1 TO N DO
02424      1  5          BEGIN REPEAT
02425      1  5 A: FILA:= I+4; COLUMNA:= 10;
02426      1  5 CURSOR; READLN (NFACTURA);
02427      1  5     IF NFACTURA = '' THEN BEGIN WRITE (CHR(7)); GOTO A; END;
02428      1  4 UNTIL LENGTH (NFACTURA) < 10;
02429      1  5 REPEAT B: FILA:= I+4; COLUMNA:= 18;
02430      1  5 CURSOR; WRITE (CHR(27),'(O)'); WRITE ('x ');
02431      1  5 WRITE (CHR(27),'(B)'); READLN (FECHA);
02432      1  5     IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO B; END;
02433      1  4 UNTIL LENGTH (FECHA) < 10;
02434      1  5 REPEAT C: FILA:= I+4; COLUMNA:= 28;
02435      1  5 CURSOR; WRITE (CHR(27),'(O)'); WRITE (' x ');
02436      1  5 WRITE (CHR(27),'(B)'); READLN (CONCEPTO);
02437      1  5     IF CONCEPTO = '' THEN BEGIN WRITE (CHR(7)); GOTO C; END;
02438      1  4 UNTIL LENGTH (CONCEPTO) < 21;
02439      1  5 REPEAT D: FILA:= I+4; COLUMNA:= 55;
02440      1  5 CURSOR; WRITE (CHR(27),'(O)'); WRITE (' x ');
02441      1  5 WRITE (CHR(27),'(B)'); READLN (IMPORTE);
02442      1  6     IF IMPORTE < 0 THEN BEGIN WRITE (CHR(7));
02443      1  5         GOTO D; END;
02444      1  4 UNTIL IMPORTE < 100000000;
02445      1  4 R: WRITE (CHR(27),'i22;BH');
02446      1  4 WRITE ('Confirme s no la operacin ("S" s "N") : ');
02447      1  4 WRITE (CHR(27),'iOJ'); READLN (CH);
02448      1  5 CASE CH OF 'N','n': FILA:= FILA-1;
02449      1  6     'S','s': BEGIN
02450      1  6     EXTEND (FRA); EXTEND (RAIMPORTE);
02451      1  6     WRITELN (FRA,NFACTURA); WRITELN (FRA,FECHA);
02452      1  6     WRITELN (FRA,CONCEPTO); WRITE (RAIMPORTE,IMPORTE);
02453      1  6     RESET (RASALDO); WHILE NOT EOF (RASALDO) DO
02454      1  6     READ (RASALDO,SALDO); SALDO:= SALDO - IMPORTE;
02455      1  5     EXTEND (RASALDO); WRITE (RASALDO,SALDO); END;
02456      1  5     OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
02457      1  4     END; (* fin del case ch of *)
02458      1  3     END; (* fin del FOR *)
02459      1  3 CLOSE (FRA); CLOSE (RAIMPORTE); CLOSE (RASALDO);
02460      1  2 RADIO; END;
02461      C 1  2 (***** Consulta de la cuenta *****)
02462      1  3     'C','c': BEGIN ENCABEZAMIENTO;
02463      1  3 OPEN (FRA,'FRA.DAT',HISTORY:=OLD);RESET (FRA);
02464      1  3 OPEN (RAIMPORTE,'RAIMPORTE.DAT',HISTORY:=OLD);RESET (RAIMPORTE);
02465      1  3 OPEN (RASALDO,'RASALDO.DAT',HISTORY:=OLD);RESET (RASALDO);
02466      1  3     J:=5; M:=1; W:=1; WRITE (CHR(27),'i5;1H');
02467      1  4 WHILE NOT EOF (FRA) DO BEGIN
02468      1  5     WHILE NOT EOF (RAIMPORTE) DO BEGIN
02469      1  6     WHILE NOT EOF (RASALDO) DO BEGIN
02470      1  6 WRITE (CHR(27),'(O)'); WRITE ('x'); WRITE (CHR(27),'(B)');
02471      1  6     FILA:=J; COLUMNA:=0; CURSOR;
02472      1  6     WRITE (M);
02473      1  6     FILA:=J; COLUMNA:=0; (* NUM.FACTURA *)
02474      1  6     CURSOR;
02475      1  6     WRITE (CHR(27),'i7D','nzm.',CHR(27),'i6C');
```

-LINE-IDC-PL-SL-

```
02476      1 6      READLN (FRA,S);
02477      1 6      WRITE (S,CHR(27),';25D');
02478      1 6      FILA:=J; COLUMNA:=0;
02479      1 6      CURSOR; (* FECHA *)
02480      1 6      READLN (FRA,S);
02481      1 6      WRITE (CHR(27),';25C',S);
02482      1 6      WRITE (CHR(27),';36D');
02483      1 6      FILA:=J; COLUMNA:=0; (* CONCEPTO *)
02484      1 6      CURSOR;
02485      1 6      READLN (FRA,S);
02486      1 6      WRITE (CHR(27),';36C',S);
02487      1 6      WRITE (CHR(27),';58D');
02488      1 6      FILA:=J; COLUMNA:=0; (* IMPORTE *)
02489      1 6      CURSOR;
02490      1 6      WRITE (CHR(27),';58C');
02491      1 6      READ (RAIMPORTE,IMPORTE);
02492      1 6      WRITE (IMPORTE,CHR(27),';68D');
02493      1 6      FILA:=J; COLUMNA:=0; (* SALDO *)
02494      1 6      CURSOR;
02495      1 6      READ (RASALDO,SALDO);
02496      1 6      WRITE (CHR(27),';68C',SALDO);
02497      1 6      J:=J+1; M:=M+1;
02498      1 6      WRITE (CHR(27),'(O)'); WRITELN (' x'); WRITE (CHR(27),'(B)');
02499      1 7      IF M = 17*W THEN BEGIN
02500      1 7      WRITE (CHR(27),'(O','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
02501      1 7      WRITELN ('j'); WRITE (CHR(27),'(B)');
02502      1 7      WRITE ('Pzlse <RETURN> para continuar
02503      1 7      WRITE (CHR(27),';0J'); READ (CH); W:=W+1;
02504      1 7      WRITE (CHR(27),';5;1H'); WRITE (CHR(27),';0J')
02505      1 6      END;
02506      1 6
02507      1 5      END;
02508      1 4      END;
02509      1 3      END;
02510      1 3      WRITE (CHR(27),'(O','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
02511      1 3      WRITELN ('j'); WRITE (CHR(27),'(B)');
02512      1 3      WRITE ('Pzlse <RETURN> para continuar ...');
02513      1 3      WRITE (CHR(27),';0J'); READ (CH); BORRADO;
02514      1 3      CLOSE (FRA); CLOSE (RAIMPORTE); CLOSE (RASALDO);
02515      1 2      RADIO; END; (* fin del BEGIN ... 'C' *)
02516      C 1 2      (*****
02517      1 2      'H','h': BEGIN AYUDA; RADIO; END;
02518      1 2      'X','x': BORRADO;
02519      1 2      OTHERWISE BEGIN WRITE (CHR(7)); GOTO 10; END;
02520      1 1      END; (* fin del begin del case primero *)
02521      0 0      END;
02522      C 0 0      (*****
02523      C 0 0      (***** TELECO *****
02524      C 0 0      (*****
02525      1 0      PROCEDURE TELECO;
02526      1 0      LABEL 1;
02527      1 1      BEGIN
02528      1 1      BORRADO;
02529      1 1      COMENTARIO:= ' Cuentas de la carrera de Telecomunicaciones ';
02530      1 1      CENIN; WRITELN;
```

--LINE--IDC--PL--SL--

```
02531      1 1 WRITE (' ':5);
02532      1 1 WRITE (CHR(27),'(0)');
02533      1 1 FOR I:= 1 TO 70 DO WRITE ('q');
02534      1 1 WRITELN (CHR(27),'(B)');
02535      1 1 WRITELN ;WRITELN;
02536      1 1 WRITELN (' ':35,'1. CUENTA DE LA CARRERA');
02537      1 1 WRITELN;WRITELN;
02538      1 1 WRITELN (' ':8,'Especialidades:');
02539      1 1 WRITELN (' ':35,'2. EQUIPOS ELECTRONICOS');
02540      1 1 WRITELN;
02541      1 1 WRITELN (' ':35,'3. IMAGEN Y SONIDO');
02542      1 1 WRITELN;
02543      1 1 WRITELN (' ':35,'4. RADIOCOMUNICACION');
02544      1 1 WRITELN;
02545      1 1 WRITELN (' ':35,'X. Salida del Menz ');
02546      1 1 WRITELN;
02547      1 1 WRITE (CHR(27),'(0)');
02548      1 1 WRITE (' ':5);
02549      1 1 FOR I:= 1 TO 70 DO WRITE ('q');
02550      1 1 WRITELN (CHR(27),'(B)');
02551      1 1 1: OPCION;
02552      1 2 CASE CH OF
02553      1 2 '1': BEGIN CARTEL; TELECO; END;
02554      1 2 '2': BEGIN EQUIP; TELECO; END;
02555      1 2 '3': BEGIN IMAGE; TELECO; END;
02556      1 2 '4': BEGIN RADIO; TELECO; END;
02557      1 2 'X','x': BORRADO;
02558      1 2 'H','h': BEGIN AYUDA; TELECO; END;
02559      1 2 OTHERWISE BEGIN WRITE (CHR(7)); GOTO 1; END;
02560      0 0 END; END;
02561      C 0 0 (*****
02562      C 0 0 (***** INDUS *****
02563      C 0 0 (*****
02564      1 0 PROCEDURE INDUS;
02565      1 0 LABEL 10,11,AA,A,B,C,D,R;
02566      1 0 TYPE STRING= VARYING i3002 OF CHAR;
02567      1 0 VAR FCARIN: TEXT; CARINSALDO,CARINIMPORTE: FILE OF INTEGER;
02568      1 0 S,FECHA,NFACTURA,CONCEPTO: STRING; M,J,N,I,W,IMPORTE,SALDO:
02569      1 1 BEGIN
02570      1 1 COMENTARIO:= ' CUENTA DE LA CARRERA DE INDUSTRIALES ';
02571      1 2 OPCUENTAS; 10: OPCION; CASE CH OF
02572      C 1 2 (*****, Inicializaciorn de la cuenta *****)
02573      1 3 'I','i': BEGIN
02574      1 3 11: BORRADO;WRITE (CHR(27),'i11;3H'); WRITELN (
02575      1 3 'Pylyse la "S" para confirmar la inicializaciorn, "N" en caso con
02576      1 3 WRITE (' ':2,'(recuerde que todas las operaciones son borradas)
02577      1 4 WRITE (CHR(27),'i0J'); READLN (CH); CASE CH OF
02578      1 5 'S','s': BEGIN ENCA; INTRODADOS;
02579      1 6 REPEAT AA: WRITE (CHR(27),'i5;20H'); READLN (FECHA);
02580      1 6 IF FECHA = '' THEN BEGIN WRITE (CHR(7));GOTO AA; END;
02581      1 5 UNTIL LENGTH (FECHA) < 10;
02582      1 5 WRITE (CHR(27),'(0)'; WRITE (CHR(27),'i5;28H'); WRITE (' x
02583      1 5 WRITE (CHR(27),'(B)');
02584      1 6 CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada
02585      1 6 WRITE (CHR(27),'i5;55H');WRITE (CHR(27),'(0)';WRITE (' x ');
```

-LINE-IDC-PL-SL-

```
02586 1 5 UNTIL IMPORTE < 100000000; WRITE (CHR(27),'(B)'); SALDO:= IMPOR
02587 1 5 OPEN (FCARIN,'FCARIN.DAT'); OPEN (CARINSALDO,'CARINSALDO.DAT
02588 1 5 OPEN (CARINIMPORTE,'CARINIMPORTE.DAT');
02589 1 5 REWRITE (FCARIN); REWRITE (CARINIMPORTE); REWRITE (CARINSALDO)
02590 1 5 WRITELN (FCARIN,NFACTURA); WRITELN (FCARIN,FECHA); WRITELN (FC
02591 1 5 WRITE (CARINIMPORTE,IMPORTE); WRITE (CARINSALDO,IMPORTE);
02592 1 4 CLOSE (FCARIN); CLOSE (CARINIMPORTE); CLOSE (CARINSALDO);
02593 1 4 'N','n': BEGIN WRITELN; END;
02594 1 4 'H','h': BEGIN AYUDA; GOTO 11; END;
02595 1 3 OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END; END;
02596 1 2 INDUS; END; (* Desde Begin del 'i','I': *)
02597 C 1 2 (***** Aqadir operaciones *****)
02598 1 3 'A','a': BEGIN ENCA; OPEN (FCARIN,'FCARIN.DAT',HISTORY:=
02599 1 3 OPEN (CARINIMPORTE,'CARINIMPORTE.DAT',HISTORY:= OLD);
02600 1 3 OPEN (CARINSALDO,'CARINSALDO.DAT',HISTORY:= OLD);
02601 1 4 REPEAT WRITE (CHR(27),'i22;8H');
02602 1 4 WRITE ('Introduzca el nzmero de operaciones: ');
02603 1 3 WRITE (CHR(27),'i0J'); READLN (N); UNTIL N < 16;
02604 1 5 FOR I:= 1 TO N DO BEGIN REPEAT
02605 1 5 A: FILA:= I+4; COLUMNA:= 10; CURSOR; READLN (NFACTURA);
02606 1 5 IF NFACTURA = '' THEN BEGIN WRITE (CHR(7)); GOTO A; END;
02607 1 4 UNTIL LENGTH (NFACTURA) < 10;
02608 1 5 REPEAT B: FILA:= I+4; COLUMNA:= 18; CURSOR;
02609 1 5 WRITE (CHR(27),'(O)'); WRITE (' x '); WRITE (CHR(27),'(B)');
02610 1 5 READLN (FECHA); IF FECHA = '' THEN BEGIN WRITE (CHR(7)); GOTO
02611 1 4 UNTIL LENGTH (FECHA) < 10;
02612 1 5 REPEAT C: FILA:= I+4; COLUMNA:= 28; CURSOR;
02613 1 5 WRITE (CHR(27),'(O)'); WRITE (' x '); WRITE (CHR(27),'(B)');
02614 1 5 READLN (CONCEPTO); IF CONCEPTO = '' THEN BEGIN WRITE (CHR(7));
02615 1 4 UNTIL LENGTH (CONCEPTO) < 21;
02616 1 5 REPEAT D: FILA:= I+4; COLUMNA:= 55; CURSOR;
02617 1 5 WRITE (CHR(27),'(O)'); WRITE (' x '); WRITE (CHR(27),'(B)');
02618 1 5 READLN (IMPORTE); IF IMPORTE < 0 THEN BEGIN WRITE (CHR(7)); GO
02619 1 4 UNTIL IMPORTE < 100000000; R: WRITE (CHR(27),'i22;8H');
02620 1 4 WRITE ('Confirme s no la operaci3n ("S" s "N") : ');
02621 1 5 WRITE (CHR(27),'i0J'); READLN (CH); CASE CH OF
02622 1 5 'N','n': FILA:= FILA-1;
02623 1 6 'S','s': BEGIN EXTEND (FCARIN); EXTEND (CARINIMPORTE);
02624 1 6 WRITELN (FCARIN,NFACTURA); WRITELN (FCARIN,FECHA);
02625 1 6 WRITELN (FCARIN,CONCEPTO); WRITE (CARINIMPORTE,IMPORTE);
02626 1 6 RESET (CARINSALDO); WHILE NOT EOF (CARINSALDO) DO
02627 1 6 READ (CARINSALDO,SALDO); SALDO:= SALDO - IMPORTE;
02628 1 5 EXTEND (CARINSALDO); WRITE (CARINSALDO,SALDO); END;
02629 1 5 OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
02630 1 3 END; END;
02631 1 3 CLOSE (FCARIN); CLOSE (CARINIMPORTE); CLOSE (CARINSALDO);
02632 1 2 INDUS; END;
02633 C 1 2 (***** Consulta de la cuenta *****)
02634 1 3 'C','c': BEGIN ENCABEZAMIENTO;
02635 1 3 OPEN (FCARIN,'FCARIN.DAT',HISTORY:=OLD); RESET (FCARIN);
02636 1 3 OPEN (CARINIMPORTE,'CARINIMPORTE.DAT',HISTORY:=OLD); RESET (C
02637 1 3 OPEN (CARINSALDO,'CARINSALDO.DAT',HISTORY:=OLD); RESET (CARIN
02638 1 3 J:=5; M:=1; W:=1; WRITE (CHR(27),'i5;1H');
02639 1 4 WHILE NOT EOF (FCARIN) DO BEGIN
02640 1 5 WHILE NOT EOF (CARINIMPORTE) DO BEGIN
```

-LINE-IDC-PL-SL-

```
02641      1 6          WHILE NOT EDF (CARINSALDO) DO BEGIN
02642      1 6      WRITE (CHR(27),'(O)'); WRITE ('x'); WRITE (CHR(27),'(B)');
02643      1 6          FILA:=J; COLUMNA:=0; CURSOR;
02644      1 6          WRITE (M);
02645      1 6          FILA:=J; COLUMNA:=0; CURSOR;
02646      1 6          WRITE (CHR(27),'i7D','nzm.',CHR(27),'i6C');
02647      1 6          READLN (FCARIN,S);
02648      1 6          WRITE (S,CHR(27),'i25D');
02649      1 6          FILA:=J; COLUMNA:=0; CURSOR;
02650      1 6          READLN (FCARIN,S);
02651      1 6          WRITE (CHR(27),'i25C',S);
02652      1 6          WRITE (CHR(27),'i36D');
02653      1 6          FILA:=J; COLUMNA:=0; CURSOR;
02654      1 6          READLN (FCARIN,S);
02655      1 6          WRITE (CHR(27),'i36C',S);
02656      1 6          WRITE (CHR(27),'i58D');
02657      1 6          FILA:=J; COLUMNA:=0; CURSOR;
02658      1 6          WRITE (CHR(27),'i58C');
02659      1 6          READ (CARINIMPORTE,IMPORTE);
02660      1 6          WRITE (IMPORTE,CHR(27),'i68D');
02661      1 6          FILA:=J; COLUMNA:=0; CURSOR;
02662      1 6          READ (CARINSALDO,SALDO);
02663      1 6          WRITE (CHR(27),'i68C',SALDO);
02664      1 6          J:=J+1; M:=M+1;
02665      1 7          IF M = 17*W THEN BEGIN
02666      1 7      WRITE (CHR(27),'(O','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
02667      1 7      WRITELN ('j'); WRITE (CHR(27),'(B)');
02668      1 7          WRITE ('Pzlse <RETURN> para continuar
02669      1 7          WRITE (CHR(27),'i0J'); READ (CH); W:=W+1;
02670      1 7      WRITE (CHR(27),'i5;1H'); WRITE (CHR(27),'i0J')
02671      1 5 END;          END;          (* fin del WHILE ... IMPORTE *)
02672      1 4      END;          (* fin del WHILE ... SALDO *)
02673      1 3 END;
02674      1 3      WRITE (CHR(27),'(O','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
02675      1 3      WRITELN ('j'); WRITE (CHR(27),'(B)');
02676      1 3      WRITE ('Pzlse <RETURN> para continuar ...');
02677      1 3      WRITE (CHR(27),'i0J'); READ (CH); BORRADO;
02678      1 3      CLOSE (FCARIN); CLOSE (CARINIMPORTE);CLOSE (CARINSALDO);
02679      1 2      INDUS;          END;          (* fin del BEGIN ... 'C' *)
02680      1 2          'H','h': BEGIN AYUDA; INDUS; END;
02681      1 2          'X','x': BORRADO;
02682      1 2          OTHERWISE BEGIN WRITE (CHR(7)); GOTO 10; END;
02683      0 0      END; END;
02684      C 0 0          (*****
02685      C 0 0          (***** TOPOG *****
02686      C 0 0          (*****
02687      1 0      PROCEDURE TOPOG;
02688      1 0      LABEL 10,11,AA,A,B,C,D,R;
02689      1 0      TYPE STRING= VARYING i3002 OF CHAR;
02690      1 0      VAR FTO: TEXT;
02691      1 0          TOSALDO,TOIMPORTE: FILE OF INTEGER;
02692      1 0          S,FEDCHA,NFACTURA,CONCEPTO: STRING;
02693      1 0          M,J,N,I,W,IMPORTE,SALDO: INTEGER;
02694      1 1      BEGIN
02695      1 1      COMENTARIO:= ' CUENTA DE TOPOGRAFIA ';
```

-LINE-IDC-PL-SL-

```
02696      1 1 OPCUENTAS;
02697      1 1 10: OPCION;
02698      1 2 CASE CH OF
02699      C 1 2 (*****,          Inicializaci3rn de la cuenta          *****)
02700      1 3 'I','i':          BEGIN          11: BORRADO;
02701      1 3 WRITE (CHR(27),'i11;3H');
02702      1 3 WRITELN (
02703      1 3 'Pulse la "S" para confirmar la inicializaci3rn, "N" en caso con
02704      1 3 WRITE (' ':2,'(recuerde que todas las operaciones son borradas)
02705      1 3 WRITE (CHR(27),'i0J'); READLN (CH);
02706      1 5 CASE CH OF 'S','s': BEGIN ENCA;
02707      1 5 INTRODATOS;
02708      1 6 REPEAT
02709      1 6 AA: WRITE (CHR(27),'i5;20H'); READLN (FECHA);
02710      1 6 IF FECHA = '' THEN BEGIN WRITE (CHR(7));GOTO AA; END;
02711      1 5 UNTIL LENGTH (FECHA) < 10;
02712      1 5 WRITE (CHR(27),'(0'); WRITE (CHR(27),'i5;28H'); WRITE (' x
02713      1 5 WRITE (CHR(27),'(B');
02714      1 5 CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada
02715      1 6 REPEAT
02716      1 6 WRITE (CHR(27),'i5;55H'); WRITE (CHR(27),'(0'); WRITE ('
02717      1 6 READ (IMPORTE);
02718      1 5 UNTIL IMPORTE < 100000000;
02719      1 5 WRITE (CHR(27),'(B'); SALDO:= IMPORTE;
02720      1 5 OPEN (FTO,'FTO.DAT');
02721      1 5 OPEN (TOSALDO,'TOSALDO.DAT');
02722      1 5 OPEN (TOIMPORTE,'TOIMPORTE.DAT');
02723      1 5 REWRITE (FTO);
02724      1 5 REWRITE (TOIMPORTE);
02725      1 5 REWRITE (TOSALDO);
02726      1 5 WRITELN (FTO,NFACTURA);
02727      1 5 WRITELN (FTO,FECHA);
02728      1 5 WRITELN (FTO,CONCEPTO);
02729      1 5 WRITE (TOIMPORTE,IMPORTE);
02730      1 5 WRITE (TOSALDO,IMPORTE);
02731      1 5 CLOSE (FTO); CLOSE (TOIMPORTE); CLOSE (TOSALDO);
02732      1 4 END;
02733      1 4 'N','n': BEGIN WRITELN; END;
02734      1 4 'H','h': BEGIN AYUDA; GOTO 11; END;
02735      1 4 OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END;
02736      1 3 END;
02737      1 2 TOPOG;          END; (* Desde Begin del 'i','I': *)
02738      C 1 2 (*****          Agadir operaciones          *****)
02739      1 2 'A','a':
02740      1 3 BEGIN
02741      1 3 ENCA;
02742      1 3 OPEN (FTO,'FTO.DAT',HISTORY:= OLD);
02743      1 3 OPEN (TOIMPORTE,'TOIMPORTE.DAT',HISTORY:= 0
02744      1 3 OPEN (TOSALDO,'TOSALDO.DAT',HISTORY:= OLD);
02745      1 4 REPEAT
02746      1 4 WRITE (CHR(27),'i22;8H');
02747      1 4 WRITE (' Introduzca el nzmero de operaciones: ');
02748      1 4 WRITE (CHR (27),'i0J');
02749      1 4 READLN (N);
02750      1 3 UNTIL N < 16;
```

-LINE-IDC-PL-SL-

```
02751      1  3 FOR I:= 1 TO N DO
02752      1  4          BEGIN
02753      1  5 REPEAT
02754      1  5 A: FILA:= I+4; COLUMNA:= 10;
02755      1  5 CURSOR;
02756      1  5   READLN (NFACTURA);
02757      1  6       IF NFACTURA = '?' THEN BEGIN
02758      1  6           WRITE (CHR(7));
02759      1  6           GOTO A;
02760      1  5       END;
02761      1  4 UNTIL LENGTH (NFACTURA) < 10;
02762      1  5 REPEAT
02763      1  5 B: FILA:= I+4; COLUMNA:= 18;
02764      1  5 CURSOR;
02765      1  5 WRITE (CHR(27), '(O)');
02766      1  5 WRITE (' x ');
02767      1  5 WRITE (CHR(27), '(B)');
02768      1  5   READLN (FECHA);
02769      1  6       IF FECHA = '?' THEN BEGIN
02770      1  6           WRITE (CHR(7));
02771      1  6           GOTO B;
02772      1  5       END;
02773      1  4 UNTIL LENGTH (FECHA) < 10;
02774      1  5 REPEAT
02775      1  5 C: FILA:= I+4; COLUMNA:= 28;
02776      1  5 CURSOR;
02777      1  5 WRITE (CHR(27), '(D)');
02778      1  5 WRITE (' x ');
02779      1  5 WRITE (CHR(27), '(B)');
02780      1  5   READLN (CONCEPTO);
02781      1  6       IF CONCEPTO = '?' THEN BEGIN
02782      1  6           WRITE (CHR(7));
02783      1  6           GOTO C;
02784      1  5       END;
02785      1  4 UNTIL LENGTH (CONCEPTO) < 25;
02786      1  5 REPEAT
02787      1  5 D: FILA:= I+4; COLUMNA:= 55;
02788      1  5 CURSOR;
02789      1  5 WRITE (CHR(27), '(O)');
02790      1  5 WRITE (' x ');
02791      1  5 WRITE (CHR(27), '(B)');
02792      1  5   READLN (IMPORTE);
02793      1  6       IF IMPORTE < 0 THEN BEGIN
02794      1  6           WRITE (CHR(7));
02795      1  6           GOTO D;
02796      1  5       END;
02797      1  4 UNTIL IMPORTE < 100000000;
02798      1  4 R: WRITE (CHR(27), '(22;BH)');
02799      1  4 WRITE ('Confirme s no la operacisn ("S" s "N") : ');
02800      1  4 WRITE (CHR(27), '(OJ)');
02801      1  4 READLN (CH);
02802      1  5 CASE CH OF
02803      1  5     'N', 'n': FILA:= FILA-1;
02804      1  5     'S', 's':
02805      1  6         BEGIN
```

--LINE--IDC--PL--SL--

```
02806      1  6      EXTEND (FTO);
02807      1  6      EXTEND (TOIMPORTE);
02808      1  6      WRITELN (FTO,NFACTURA);
02809      1  6      WRITELN (FTO,FECHA);
02810      1  6      WRITELN (FTO,CONCEPTO);
02811      1  6      WRITE (TOIMPORTE,IMPORTE);
02812      1  6      RESET (TOSALDO);
02813      1  6          WHILE NOT EOF (TOSALDO) DO
02814      1  6              READ (TOSALDO,SALDO);
02815      1  6          SALDO:= SALDO - IMPORTE;
02816      1  6          EXTEND (TOSALDO);
02817      1  6          WRITE (TOSALDO,SALDO);
02818      1  5              END;
02819      1  5          OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
02820      1  3              END;
02821      1  3                  CLOSE (FTO);
02822      1  3                  CLOSE (TOIMPORTE);
02823      1  3                  CLOSE (TOSALDO);
02824      1  2      TUPOG;                                END;
02825      C  1  2      (*****      Consulta de la cuenta      *****)
02826      1  2          'C','c':
02827      1  3              BEGIN
02828      1  3                  ENCABEZAMIENTO;
02829      1  3                  OPEN (FTO,'FTO.DAT',HISTORY:=OLD);
02830      1  3                  RESET (FTO);
02831      1  3                  OPEN (TOIMPORTE,'TOIMPORTE.DAT',HISTORY:=OLD);
02832      1  3                  RESET (TOIMPORTE);
02833      1  3                  OPEN (TOSALDO,'TOSALDO.DAT',HISTORY:=OLD);
02834      1  3                  RESET (TOSALDO);
02835      1  3          J:=5; M:=1; W:=1;
02836      1  3          WRITE (CHR(27),' ;5;1H');
02837      1  3      WHILE NOT EOF (FTO) DO
02838      1  4          BEGIN
02839      1  4              WHILE NOT EOF (TOIMPORTE) DO
02840      1  5                  BEGIN
02841      1  5                      WHILE NOT EOF (TOSALDO) DO
02842      1  6                          BEGIN
02843      1  6                              WRITE (CHR(27),' (0)'); WRITE ('x'); WRITE (CHR(27),' (B)');
02844      1  6                              FILA:=J; COLUMNA:=0; CURSOR;
02845      1  6                              WRITE (M);
02846      1  6                              FILA:=J; COLUMNA:=0; CURSOR;
02847      1  6                              WRITE (CHR(27),' ;1C');
02848      1  6                              WRITE (CHR(27),' (0)'); WRITE ('x'); WRITE (CHR(27),' (B)');
02849      1  6                              READLN (FTO,S); WRITE (S);
02850      1  6                              WRITE (CHR(27),' ;12D');
02851      1  6                              FILA:=J; COLUMNA:=0; CURSOR;
02852      1  6                              WRITE (CHR(27),' ;12C');
02853      1  6                              WRITE (CHR(27),' (0)'); WRITE ('x'); WRITE (CHR(27),' (B)');
02854      1  6                              READLN (FTO,S); WRITE (S);
02855      1  6                              WRITE (CHR(27),' ;25D');
02856      1  6                              FILA:=J; COLUMNA:=0; CURSOR;
02857      1  6                              WRITE (CHR(27),' ;25C');
02858      1  6                              WRITE (CHR(27),' (0)'); WRITE ('x'); WRITE (CHR(27),' (B)');
02859      1  6                              READLN (FTO,S); WRITE (S);
02860      1  6                              WRITE (CHR(27),' ;40D');
```

-LINE-IDC-PL-SL-

```
02861      1  6          FILA:=J; COLUMNA:=0; CURSOR;
02862      1  6          WRITE (CHR(27),'i40C');
02863      1  6          WRITE (CHR(27),'(0)'); WRITE ('x');
02864      1  6          READ (TOIMPORTE,IMPORTE);
02865      1  6          WRITE (IMPORTE);
02866      1  6          WRITE (CHR(27),'i50D');
02867      1  6          FILA:=J; COLUMNA:=0; CURSOR;
02868      1  6          WRITE (CHR(27),'i50C'); WRITE ('x'); WRITE (CHR(27),'
02869      1  6          READ (TOSALDO,SALDO); WRITELN (SALDO);
02870      1  6          J:=J+1; M:=M+1;
02871      1  7          IF M = 17*W THEN BEGIN
02872      1  7          WRITE (CHR(27),'(0','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
02873      1  7          WRITELN ('j'); WRITE (CHR(27),'(B)');
02874      1  7          WRITE ('Pzlse <RETURN> para continuar
02875      1  7          WRITE (CHR(27),'i0J'); READ (CH); W:=W+1;
02876      1  7          WRITE (CHR(27),'i5;1H'); WRITE (CHR(27),'i0J')
02877      1  5 END;          END; (* fin del WHILE ... TOIMPORTE *)
02878      1  4          END; (* fin del WHILE ... TOSALDO *)
02879      1  3 END;
02880      1  3 FILA:=22; COLUMNA:=3;
02881      1  3 CURSOR;
02882      1  3 WRITE ('Pzlse <RETURN> para continuar ...');
02883      1  3 BORRADO;
02884      1  3 CLOSE (FTD); CLOSE (TOIMPORTE);CLOSE (TOSALDO);
02885      1  3 TOPOG;
02886      1  2          END; (* fin del BEGIN ... 'C' *)
02887      1  2          'H','h': BEGIN AYUDA; TOPOG; END;
02888      1  2          'X','x': BORRADO;
02889      1  2          OTHERWISE BEGIN WRITE (CHR(7)); GOTO 10; END;
02890      0  0 END; END;
02891      C  0  0          (*****
02892      C  0  0          (***** NAVAL *****
02893      L  0  0          (*****
02894      1  0 PROCEDURE NAVAL;
02895      1  0
02896      1  0 LABEL 10,11,AA,A,B,C,D,R;
02897      1  0 TYPE STRING= VARYING i3002 OF CHAR;
02898      1  0 VAR FNAV: TEXT;
02899      1  0 NAVSALDO,NAVIMPORTE: FILE OF INTEGER;
02900      1  0 S,FECHA,NFACTURA,CONCEPTO: STRING;
02901      1  0 M,J,N,I,W,IMPORTE,SALDO: INTEGER;
02902      1  1 BEGIN
02903      1  1 COMENTARIO:= ' CUENTA DE NAVALES ';
02904      1  1 OPCUENTAS;
02905      1  1 10: OPCION;
02906      1  2 CASE CH OF
02907      C  1  2 (*****, Inicializaciorn de la cuenta *****)
02908      1  2 'I','i':
02909      1  3 BEGIN
02910      1  3 11: BORRADO;
02911      1  3 WRITE (CHR(27),'i11;3H');
02912      1  3 WRITELN (
02913      1  3 'Pylse la "S" para confirmar la inicializaciorn, "N" en caso con
02914      1  3 WRITE (' ':2,'(recuerde que todas las operaciones son borradas)
02915      1  3 WRITE (CHR(27),'i0J');
```

-LINE-IDC-PL-SL-

```
02916      1   3 READLN (CH);
02917      1   4                               CASE CH OF
02918      1   5                 'S','s': BEGIN
02919      1   5                               ENCA;
02920      1   5   INTRODADOS;
02921      1   6   REPEAT
02922      1   6 AA:  WRITE (CHR(27),'i5;20H');   READLN (FECHA);
02923      1   6       IF FECHA = '' THEN BEGIN WRITE (CHR(7));GOTO AA; END;
02924      1   5   UNTIL LENGTH (FECHA) < 10;
02925      1   5   WRITE (CHR(27),'(O)'); WRITE (CHR(27),'i5;28H'); WRITE (' x
02926      1   5   WRITE (CHR(27),'(B)');
02927      1   5   CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada
02928      1   6   REPEAT
02929      1   6   WRITE (CHR(27),'i5;55H'); WRITE (CHR(27),'(O)'); WRITE ('
02930      1   6   READ (IMPORTE);
02931      1   5   UNTIL IMPORTE < 100000000;
02932      1   5   WRITE (CHR(27),'(B)'); SALDO:= IMPORTE;
02933      1   5                               OPEN (FNAV,'FNAV.DAT');
02934      1   5                               OPEN (NAVSALDO,'NAVSALDO.DAT');
02935      1   5                               OPEN (NAVIMPORTE,'NAVIMPORTE.DAT');
02936      1   5                               REWRITE (FNAV);
02937      1   5                               REWRITE (NAVIMPORTE);
02938      1   5                               REWRITE (NAVSALDO);
02939      1   5                               WRITELN (FNAV,NFACTURA);
02940      1   5                               WRITELN (FNAV,FECHA);
02941      1   5                               WRITELN (FNAV,CONCEPTO);
02942      1   5                               WRITE (NAVIMPORTE,IMPORTE);
02943      1   5                               WRITE (NAVSALDO,IMPORTE);
02944      1   5   CLOSE (FNAV); CLOSE (NAVIMPORTE); CLOSE (NAVSALDO);
02945      1   4   END;
02946      1   4   'N','n': BEGIN WRITELN; END;
02947      1   4   'H','h': BEGIN AYUDA; GOTO 11; END;
02948      1   4   OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END;
02949      1   3   END;
02950      1   2   NAVAL;   END; (* Desde   Begin del   'i','I':   *)
02951      C 1   2   (*****   Agadir operaciones   *****)
02952      1   2   'A','a':
02953      1   3   BEGIN
02954      1   3   ENCA;
02955      1   3   OPEN (FNAV,'FNAV.DAT',HISTORY:= OLD);
02956      1   3   OPEN (NAVIMPORTE,'NAVIMPORTE.DAT',HISTORY:=
02957      1   3   OPEN (NAVSALDO,'NAVSALDO.DAT',HISTORY:= OLD
02958      1   4   REPEAT
02959      1   4   WRITE (CHR(27),'i22;8H');
02960      1   4   WRITE ('Introduzca el nzmbero de operaciones: ');
02961      1   4   WRITE (CHR (27),'i0J');
02962      1   4   READLN (N);
02963      1   3   UNTIL N < 16;
02964      1   3   FOR I:= 1 TO N DO
02965      1   4   BEGIN
02966      1   5   REPEAT
02967      1   5   A:  FILA:= I+4; COLUMNA:= 10;
02968      1   5   CURSOR;
02969      1   5   READLN (NFACTURA);
02970      1   6   IF NFACTURA = '' THEN BEGIN
```

-LINE-IDC-PL-SL-

```
02971      1  6                WRITE (CHR(7));
02972      1  6                GOTO A;
02973      1  5                END;
02974      1  4 UNTIL LENGTH (NFACTURA) < 10;
02975      1  5 REPEAT
02976      1  5 B: FILA:= I+4; COLUMNA:= 18;
02977      1  5 CURSOR;
02978      1  5 WRITE (CHR(27), 'O');
02979      1  5 WRITE (' x ');
02980      1  5 WRITE (CHR(27), 'B');
02981      1  5   READLN (FECHA);
02982      1  6       IF FECHA = '' THEN BEGIN
02983      1  6                   WRITE (CHR(7));
02984      1  6                   GOTO B;
02985      1  5                   END;
02986      1  4 UNTIL LENGTH (FECHA) < 10;
02987      1  5 REPEAT
02988      1  5 C: FILA:= I+4; COLUMNA:= 28;
02989      1  5 CURSOR;
02990      1  5 WRITE (CHR(27), 'O');
02991      1  5 WRITE (' x ');
02992      1  5 WRITE (CHR(27), 'B');
02993      1  5   READLN (CONCEPTO);
02994      1  6       IF CONCEPTO = '' THEN BEGIN
02995      1  6                   WRITE (CHR(7));
02996      1  6                   GOTO C;
02997      1  5                   END;
02998      1  4 UNTIL LENGTH (CONCEPTO) < 25;
02999      1  5 REPEAT
03000      1  5 D: FILA:= I+4; COLUMNA:= 55;
03001      1  5 CURSOR;
03002      1  5 WRITE (CHR(27), 'O');
03003      1  5 WRITE (' x ');
03004      1  5 WRITE (CHR(27), 'B');
03005      1  5   READLN (IMPORTE);
03006      1  6       IF IMPORTE < 0 THEN BEGIN
03007      1  6                   WRITE (CHR(7));
03008      1  6                   GOTO D;
03009      1  5                   END;
03010      1  4 UNTIL IMPORTE < 100000000;
03011      1  4 R: WRITE (CHR(27), 'i22;8H');
03012      1  4 WRITE ('Confirme s no la operacisn ("S" s "N") : ');
03013      1  4 WRITE (CHR(27), 'iOJ');
03014      1  4 READLN (CH);
03015      1  5 CASE CH OF
03016      1  5             'N', 'n': FILA:= FILA-1;
03017      1  5             'S', 's':
03018      1  6                 BEGIN
03019      1  6             EXTEND (FNAV);
03020      1  6             EXTEND (NAVIMPORTE);
03021      1  6             WRITELN (FNAV, NFACTURA);
03022      1  6             WRITELN (FNAV, FECHA);
03023      1  6             WRITELN (FNAV, CONCEPTO);
03024      1  6             WRITE (NAVIMPORTE, IMPORTE);
03025      1  6             RESET (NAVSALDO);
```

--LINE--IDC--PL--SL--

```
03026      1  6      WHILE NOT EOF (NAVSALDO) DO
03027      1  6          READ (NAVSALDO,SALDO);
03028      1  6      SALDO:= SALDO - IMPORTE;
03029      1  6      EXTEND (NAVSALDO);
03030      1  6      WRITE (NAVSALDO,SALDO);
03031      1  5          END;
03032      1  5      OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
03033      1  3      END;          END;
03034      1  3          CLOSE (FNAV);
03035      1  3          CLOSE (NAVIMPORTE);
03036      1  3          CLOSE (NAVSALDO);
03037      1  2  NAVAL;          END;
03038      C  1  2  (*****)      Consulta de la cuenta      (*****)
03039      1  2          'C','c':
03040      1  3          BEGIN
03041      1  3              ENCABEZAMIENTO;
03042      1  3              OPEN (FNAV,'FNAV.DAT',HISTORY:=OLD);
03043      1  3              RESET (FNAV);
03044      1  3              OPEN (NAVIMPORTE,'NAVIMPORTE.DAT',HISTORY:=(
03045      1  3              RESET (NAVIMPORTE);
03046      1  3              OPEN (NAVSALDO,'NAVSALDO.DAT',HISTORY:=OLD)
03047      1  3              RESET (NAVSALDO);
03048      1  3          J:=5; M:=1; W:=1;
03049      1  3          WRITE (CHR(27),'i5;1H');
03050      1  3  WHILE NOT EOF (FNAV) DO
03051      1  4      BEGIN
03052      1  4          WHILE NOT EOF (NAVIMPORTE) DO
03053      1  5      BEGIN
03054      1  5          WHILE NOT EOF (NAVSALDO) DO
03055      1  6      BEGIN
03056      1  6          WRITE (CHR(27),'(0)'); WRITE ('x'); WRITE (CHR(27),'(B)');
03057      1  6          FILA:=J; COLUMNA:=0; CURSOR;
03058      1  6          WRITE (M);
03059      1  6          FILA:=J; COLUMNA:=0; CURSOR;
03060      1  6          WRITE (CHR(27),'i10');
03061      1  6          WRITE (CHR(27),'(0)'); WRITE ('x'); WRITE (CHR(27),'(B)');
03062      1  6          READLN (FNAV,S); WRITE (S);
03063      1  6          WRITE (CHR(27),'i12D');
03064      1  6          FILA:=J; COLUMNA:=0; CURSOR;
03065      1  6          WRITE (CHR(27),'i12C');
03066      1  6          WRITE (CHR(27),'(0)'); WRITE ('x'); WRITE (CHR(27),'(B)');
03067      1  6          READLN (FNAV,S); WRITE (S);
03068      1  6          WRITE (CHR(27),'i25D');
03069      1  6          FILA:=J; COLUMNA:=0; CURSOR;
03070      1  6          WRITE (CHR(27),'i25C');
03071      1  6          WRITE (CHR(27),'(0)'); WRITE ('x'); WRITE (CHR(27),'(B)');
03072      1  6          READLN (FNAV,S); WRITE (S);
03073      1  6          WRITE (CHR(27),'i40D');
03074      1  6          FILA:=J; COLUMNA:=0; CURSOR;
03075      1  6          WRITE (CHR(27),'i40C');
03076      1  6          WRITE (CHR(27),'(0)'); WRITE ('x');
03077      1  6          READ (NAVIMPORTE,IMPORTE);
03078      1  6          WRITE (IMPORTE);
03079      1  6          WRITE (CHR(27),'i50D');
03080      1  6          FILA:=J; COLUMNA:=0; CURSOR;
```

--LINE-IDC-FL-SL--

```

03081      1 6      WRITE (CHR(27),'i50C');      WRITE ('x');      WRITE (CHR(27),'
03082      1 6      READ (NAVSALDO,SALDO);      WRITELN (SALDO);
03083      1 6              J:=J+1; M:=M+1;
03084      1 7              IF M = 15*W THEN BEGIN
03085      1 7                      WRITE (CHR(27),'22;3H');
03086      1 7                      WRITE ('Pzlse <RETURN> para continuar
03087      1 7                      READ (CH); W:=W+1;
03088      1 6                      WRITE (CHR(27),'i5;iH');      END;
03089      1 5      END;      (* fin del WHILE ... NAVIMPORTE *)
03090      1 4      END;      (* fin del WHILE ... NAVSALDO *)
03091      1 3 END;
03092      1 3 FILA:=22; COLUMNNA:=3;
03093      1 3 CURSOR;
03094      1 3 WRITE ('Pzlse <RETURN> para continuar ...');
03095      1 3 BORRADO;
03096      1 3 CLOSE (FNAV); CLOSE (NAVIMPORTE);CLOSE (NAVSALDO);
03097      1 3 NAVAL;
03098      1 2              END;      (* fin del BEGIN ... 'C' *)
03099      1 2              'H','h': BEGIN      AYUDA;      NAVAL;      END;
03100      1 2              'X','x': BORRADO;
03101      1 2              OTHERWISE BEGIN      WRITE (CHR(7));      GOTO 10;      END;
03102      0 0 END; END;
03103      C 0 0              (*****
03104      C 0 0              (*****      OBRAS      *****
03105      C 0 0              (*****
03106      1 0 PROCEDURE OBRAS;
03107      1 0 LABEL 10,11,AA,A,B,C,D,R;
03108      1 0 TYPE STRING= VARYING i3002 OF CHAR;
03109      1 0 VAR FOR: TEXT;
03110      1 0      OBSALDO,OBIMPORTE: FILE OF INTEGER;
03111      1 0      S,FECHA,NFACTURA,CONCEPTO: STRING;
03112      1 0      M,J,N,I,W,IMPORTE,SALDO: INTEGER;
03113      1 1 BEGIN
03114      1 1 COMENTARIO:= ' CUENTA DE OBRAS PUBLICAS ';
03115      1 1 OPCUENTAS;
03116      1 1 10: OPCION;
03117      1 2 CASE CH OF
03118      C 1 2 (*****,      Inicializaci,rn de la cuenta      *****)
03119      1 2      'I','i':
03120      1 3              BEGIN
03121      1 3 11: BORRADO;
03122      1 3 WRITE (CHR(27),'i11;3H');
03123      1 3 WRITELN (
03124      1 3 'Pylse la "S" para confirmar la inicializaci,rn, "N" en caso con
03125      1 3 WRITE (' ':2,'(recuerde que todas las operaciones son borradas)
03126      1 3 WRITE (CHR(27),'i0J');
03127      1 3 READLN (CH);
03128      1 4              CASE CH OF
03129      1 5              'S','s': BEGIN
03130      1 5                      ENCA;
03131      1 5      INTRODATOS;
03132      1 6      REPEAT
03133      1 6 AA:      WRITE (CHR(27),'i5;20H');      READLN (FECHA);
03134      1 6      IF FECHA = '' THEN BEGIN WRITE (CHR(7));GOTO AA;      END;
03135      1 5      UNTIL LENGTH (FECHA) < 10;

```

-LINE-IDC-PL-SL-

```
03136 1 5 WRITE (CHR(27),'(0)'); WRITE (CHR(27),'i5;29H'); WRITE (' x
03137 1 5 WRITE (CHR(27),'(B)');
03138 1 5 CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada
03139 1 6 REPEAT
03140 1 6 WRITE (CHR(27),'i5;55H'); WRITE (CHR(27),'(0)'); WRITE ('
03141 1 6 READ (IMPORTE);
03142 1 5 UNTIL IMPORTE < 100000000;
03143 1 5 WRITE (CHR(27),'(B)'); SALDO:= IMPORTE;
03144 1 5 OPEN (FOB,'FOB.DAT');
03145 1 5 OPEN (OBSALDO,'OBSALDO.DAT');
03146 1 5 OPEN (OBIMPORTE,'OBIMPORTE.DAT');
03147 1 5 REWRITE (FOB);
03148 1 5 REWRITE (OBIMPORTE);
03149 1 5 REWRITE (OBSALDO);
03150 1 5 WRITELN (FOB,NFACTURA);
03151 1 5 WRITELN (FOB,FECHA);
03152 1 5 WRITELN (FOB,CONCEPTO);
03153 1 5 WRITE (OBIMPORTE,IMPORTE);
03154 1 5 WRITE (OBSALDO,IMPORTE);
03155 1 5 CLOSE (FOB); CLOSE (OBIMPORTE); CLOSE (OBSALDO);
03156 1 4 END;
03157 1 4 'N','n': BEGIN WRITELN; END;
03158 1 4 'H','h': BEGIN AYUDA; GOTO 11; END;
03159 1 4 OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END;
03160 1 3 END;
03161 1 2 OBRAS; END; (* Desde Begin del 'i','I': *)
03162 C 1 2 (***** Agadir operaciones *****)
03163 1 2 'A','a':
03164 1 3 BEGIN
03165 1 3 ENCA;
03166 1 3 OPEN (FOB,'FOB.DAT',HISTORY:= OLD);
03167 1 3 OPEN (OBIMPORTE,'OBIMPORTE.DAT',HISTORY:= O
03168 1 3 OPEN (OBSALDO,'OBSALDO.DAT',HISTORY:= OLD);
03169 1 4 REPEAT
03170 1 4 WRITE (CHR(27),'i22;8H');
03171 1 4 WRITE ('Introduzca el nmero de operaciones: ');
03172 1 4 WRITE (CHR(27),'i0J');
03173 1 4 READLN (N);
03174 1 3 UNTIL N < 16;
03175 1 3 FOR I:= 1 TO N DO
03176 1 4 BEGIN
03177 1 5 REPEAT
03178 1 5 A: FILA:= I+4; COLUMNA:= 10;
03179 1 5 CURSOR;
03180 1 5 READLN (NFACTURA);
03181 1 6 IF NFACTURA = '' THEN BEGIN
03182 1 6 WRITE (CHR(7));
03183 1 6 GOTO A;
03184 1 5 END;
03185 1 4 UNTIL LENGTH (NFACTURA) < 10;
03186 1 5 REPEAT
03187 1 5 B: FILA:= I+4; COLUMNA:= 18;
03188 1 5 CURSOR;
03189 1 5 WRITE (CHR(27),'(0)');
03190 1 5 WRITE ('x ');
```

-LINE-IDC-PL-SL-

```
03191      1 5 WRITE (CHR(27),'(B)');
03192      1 5   READLN (FECHA);
03193      1 6     IF FECHA = '' THEN BEGIN
03194      1 6         WRITE (CHR(7));
03195      1 6         GOTO B;
03196      1 5     END;
03197      1 4 UNTIL LENGTH (FECHA) < 10;
03198      1 5 REPEAT
03199      1 5 C: FILA:= I+4; COLUMNA:= 28;
03200      1 5 CURSOR;
03201      1 5 WRITE (CHR(27),'(O)');
03202      1 5 WRITE (' x ');
03203      1 5 WRITE (CHR(27),'(B)');
03204      1 5   READLN (CONCEPTO);
03205      1 6     IF CONCEPTO = '' THEN BEGIN
03206      1 6         WRITE (CHR(7));
03207      1 6         GOTO C;
03208      1 5     END;
03209      1 4 UNTIL LENGTH (CONCEPTO) < 25;
03210      1 5 REPEAT
03211      1 5 D: FILA:= I+4; COLUMNA:= 55;
03212      1 5 CURSOR;
03213      1 5 WRITE (CHR(27),'(O)');
03214      1 5 WRITE (' x ');
03215      1 5 WRITE (CHR(27),'(B)');
03216      1 5   READLN (IMPORTE);
03217      1 6     IF IMPORTE < 0 THEN BEGIN
03218      1 6         WRITE (CHR(7));
03219      1 6         GOTO D;
03220      1 5     END;
03221      1 4 UNTIL IMPORTE < 100000000;
03222      1 4 R: WRITE (CHR(27),'i22;8H');
03223      1 4 WRITE ('Confirme s no la operacion ("S".s "N") : ');
03224      1 4 WRITE (CHR(27),'i0J');
03225      1 4 READLN (CH);
03226      1 5 CASE CH OF
03227      1 5         'N','n': FILA:= FILA-1;
03228      1 5         'S','s':
03229      1 6             BEGIN
03230      1 6                 EXTEND (FOB);
03231      1 6                 EXTEND (OBIMPORTE);
03232      1 6                 WRITELN (FOB,NFACTURA);
03233      1 6                 WRITELN (FOB,FECHA);
03234      1 6                 WRITELN (FOB,CONCEPTO);
03235      1 6                 WRITE (OBIMPORTE,IMPORTE);
03236      1 6                 RESET (OBSALDO);
03237      1 6                 WHILE NOT EOF (OBSALDO) DO
03238      1 6                     READ (OBSALDO,SALDO);
03239      1 6                 SALDO:= SALDO - IMPORTE;
03240      1 6                 EXTEND (OBSALDO);
03241      1 6                 WRITE (OBSALDO,SALDO);
03242      1 5                 END;
03243      1 5                 OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
03244      1 3                 END;           END;
03245      1 3                 CLOSE (FOB);
```

-LINE-IDC-PL-SL-

```
03246      1 3          CLOSE (OBIMPORTE);
03247      1 3          CLOSE (OBSALDO);
03248      1 2 OBRAS;
03249      C 1 2 (***** Consulta de la cuenta *****);
03250      1 2          'C', 'c';
03251      1 3          BEGIN
03252      1 3              ENCABEZAMIENTO;
03253      1 3              OPEN (FOB, 'FOB.DAT', HISTORY:=OLD);
03254      1 3              RESET (FOB);
03255      1 3              OPEN (OBIMPORTE, 'OBIMPORTE.DAT', HISTORY:=OLD);
03256      1 3              RESET (OBIMPORTE);
03257      1 3              OPEN (OBSALDO, 'OBSALDO.DAT', HISTORY:=OLD);
03258      1 3              RESET (OBSALDO);
03259      1 3              J:=5; M:=1; W:=1;
03260      1 3              WRITE (CHR(27), 'i5;1H');
03261      1 3          WHILE NOT EOF (FOB) DO
03262      1 4              BEGIN
03263      1 4                  WHILE NOT EOF (OBIMPORTE) DO
03264      1 5                      BEGIN
03265      1 5                          WHILE NOT EOF (OBSALDO) DO
03266      1 6                              BEGIN
03267      1 6                                  WRITE (CHR(27), ' (O)'); WRITE ('x'); WRITE (CHR(27), ' (B)');
03268      1 6                                      FILA:=J; COLUMNA:=0; CURSOR;
03269      1 6                                      WRITE (M);
03270      1 6                                      FILA:=J; COLUMNA:=0; CURSOR;
03271      1 6                                      WRITE (CHR(27), 'i1C');
03272      1 6                                  WRITE (CHR(27), ' (O)'); WRITE ('x'); WRITE (CHR(27), ' (B)');
03273      1 6                                  READLN (FOB, S); WRITE (S);
03274      1 6                                  WRITE (CHR(27), 'i12D');
03275      1 6                                  FILA:=J; COLUMNA:=0; CURSOR;
03276      1 6                                  WRITE (CHR(27), 'i12C');
03277      1 6                                  WRITE (CHR(27), ' (O)'); WRITE ('x'); WRITE (CHR(27), ' (B)');
03278      1 6                                  READLN (FOB, S); WRITE (S);
03279      1 6                                  WRITE (CHR(27), 'i25D');
03280      1 6                                  FILA:=J; COLUMNA:=0; CURSOR;
03281      1 6                                  WRITE (CHR(27), 'i25C');
03282      1 6                                  WRITE (CHR(27), ' (O)'); WRITE ('x'); WRITE (CHR(27), ' (B)');
03283      1 6                                  READLN (FOB, S); WRITE (S);
03284      1 6                                  WRITE (CHR(27), 'i40D');
03285      1 6                                  FILA:=J; COLUMNA:=0; CURSOR;
03286      1 6                                  WRITE (CHR(27), 'i40C');
03287      1 6                                  WRITE (CHR(27), ' (O)'); WRITE ('x');
03288      1 6                                  READ (OBIMPORTE, IMPORTE);
03289      1 6                                  WRITE (IMPORTE);
03290      1 6                                  WRITE (CHR(27), 'i50D');
03291      1 6                                  FILA:=J; COLUMNA:=0; CURSOR;
03292      1 6                                  WRITE (CHR(27), 'i50C'); WRITE ('x'); WRITE (CHR(27), ' (B)');
03293      1 6                                  READ (OBSALDO, SALDO); WRITELN (SALDO);
03294      1 6                                  J:=J+1; M:=M+1;
03295      1 7                                  IF M = 17*W THEN BEGIN
03296      1 7                                      WRITE (CHR(27), ' (O), 'm'); FOR I:= 1 TO 78 DO WRITE ('q');
03297      1 7                                      WRITELN ('j'); WRITE (CHR(27), ' (B)');
03298      1 7                                      WRITE ('Pzlse <RETURN> para continuar');
03299      1 7                                      WRITE (CHR(27), 'i0J'); READ (CH); W:=W+1;
03300      1 7                                      WRITE (CHR(27), 'i5;1H'); WRITE (CHR(27), 'i0J');
```

-LINE-IDC-PL-SL-

```
03301      1  5 END;          END;      (* fin del WHILE ... OBIMPORTE *)
03302      1  4      END;      (* fin del WHILE ... OBSALDO *)
03303      1  3 END;
03304      1  3 FILA:=22; COLUMNA:=3;
03305      1  3 CURSOR;
03306      1  3 WRITE ('Pulse <RETURN> para continuar ...');
03307      1  3 BORRADO;
03308      1  3 CLOSE (FOB); CLOSE (OBIMPORTE);CLOSE (OBSALDO);
03309      1  3 OBRAS;
03310      1  2          END;      (* fin del BEGIN ... 'C' *)
03311      1  2      'H','h': BEGIN AYUDA; OBRAS; END;
03312      1  2      'X','x': BORRADO;
03313      1  2          OTHERWISE BEGIN WRITE (CHR(7)); GOTO 10; END;
03314      0  0 END; END;
03315      C  0  0          (*****
03316      C  0  0          (***** CUENTAGEN *****)
03317      C  0  0          (*****
03318      1  0 PROCEDURE CUENTAGEN;
03319      1  0 LABEL 10,11,AA,A,B,C,D,R;
03320      1  0 TYPE STRING= VARYING (300) OF CHAR;
03321      1  0 VAR FDU: TEXT;
03322      1  0      CUSALDO,CUIMPORTE: FILE OF INTEGER;
03323      1  0      S,FECHA,NFACTURA,CONCEPTO: STRING;
03324      1  0      M,J,N,I,W,IMPORTE,SALDO: INTEGER;
03325      1  1 BEGIN
03326      1  1 COMENTARIO:= ' CUENTA GENERAL DE ACTIVIDADES DOCENTES ';
03327      1  1 OPCUENTAS;
03328      1  1 10: OPCION;
03329      1  2 CASE CH OF
03330      C  1  2 (*****, Inicializacion de la cuenta *****)
03331      1  2      'I','i':
03332      1  3          BEGIN
03333      1  3 11: BORRADO;
03334      1  3 WRITE (CHR(27),'i11;3H');
03335      1  3 WRITELN (
03336      1  3 'Pulse la "S" para confirmar la inicializacion, "N" en caso con
03337      1  3 WRITE (' ':2,'(recuerde que todas las operaciones son borradas);
03338      1  3 WRITE (CHR(27),'i0J');
03339      1  3 READLN (CH);
03340      1  4          CASE CH OF
03341      1  5      'S','s': BEGIN
03342      1  5          ENCA;
03343      1  5 INTRODATOS;
03344      1  6 REPEAT
03345      1  6 AA: WRITE (CHR(27),'i5;20H'); READLN (FECHA);
03346      1  6 IF FECHA = '' THEN BEGIN WRITE (CHR(7));GOTO AA; END;
03347      1  5 UNTIL LENGTH (FECHA) < 10;
03348      1  5 WRITE (CHR(27),'(0)'); WRITE (CHR(27),'i5;28H'); WRITE (' x
03349      1  5 WRITE (CHR(27),'(B)');
03350      1  5 CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada
03351      1  6 REPEAT
03352      1  6 WRITE (CHR(27),'i5;55H'); WRITE (CHR(27),'(0)'); WRITE ('
03353      1  6 READ (IMPORTE);
03354      1  5 UNTIL IMPORTE < 100000000;
03355      1  5 WRITE (CHR(27),'(B)'); SALDO:= IMPORTE;
```

-LINE-IDC-PL-SL-

```
03356      1  5      OPEN (FCU,'FCU.DAT');
03357      1  5      OPEN (CUSALDO,'CUSALDO.DAT');
03358      1  5      OPEN (CUIMPORTE,'CUIMPORTE.DAT');
03359      1  5      REWRITE (FCU);
03360      1  5      REWRITE (CUIMPORTE);
03361      1  5      REWRITE (CUSALDO);
03362      1  5      WRITELN (FCU,NFACTURA);
03363      1  5      WRITELN (FCU,FECHA);
03364      1  5      WRITELN (FCU,CONCEPTO);
03365      1  5      WRITE (CUIMPORTE,IMPORTE);
03366      1  5      WRITE (CUSALDO,IMPORTE);
03367      1  5      CLOSE (FCU); CLOSE (CUIMPORTE); CLOSE (CUSALDO);
03368      1  4      END;
03369      1  4      'N','n': BEGIN WRITELN; END;
03370      1  4      'H','h': BEGIN AYUDA; GOTO 11; END;
03371      1  4      OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END;
03372      1  3      END;
03373      1  2 CUENTAGEN;      END; (* Desde      Begin del 'i','I': *)
03374      C 1  2 (*****      Agadir operaciones      *****)
03375      1  2      'A','a':
03376      1  3      BEGIN
03377      1  3      ENCA;
03378      1  3      OPEN (FCU,'FCU.DAT',HISTORY:= OLD);
03379      1  3      OPEN (CUIMPORTE,'CUIMPORTE.DAT',HISTORY:= OLD);
03380      1  3      OPEN (CUSALDO,'CUSALDO.DAT',HISTORY:= OLD);
03381      1  4 REPEAT
03382      1  4 WRITE (CHR(27),'i22;BH');
03383      1  4 WRITE ('Introduzca el nzmero de operaciones: ');
03384      1  4 WRITE (CHR(27),'i0J');
03385      1  4 READLN (N);
03386      1  3 UNTIL N < 16;
03387      1  3 FOR I:= 1 TO N DO
03388      1  4      BEGIN
03389      1  5 REPEAT
03390      1  5 A: FILA:= I+4; COLUMNA:= 10;
03391      1  5 CURSOR;
03392      1  5 READLN (NFACTURA);
03393      1  6 IF NFACTURA = '' THEN BEGIN
03394      1  6      WRITE (CHR(7));
03395      1  6      GOTO A;
03396      1  5      END;
03397      1  4 UNTIL LENGTH (NFACTURA) < 10;
03398      1  5 REPEAT
03399      1  5 B: FILA:= I+4; COLUMNA:= 18;
03400      1  5 CURSOR;
03401      1  5 WRITE (CHR(27),'(O)');
03402      1  5 WRITE ('x ');
03403      1  5 WRITE (CHR(27),'(B)');
03404      1  5 READLN (FECHA);
03405      1  6 IF FECHA = '' THEN BEGIN
03406      1  6      WRITE (CHR(7));
03407      1  6      GOTO B;
03408      1  5      END;
03409      1  4 UNTIL LENGTH (FECHA) < 10;
03410      1  5 REPEAT
```

-LINE-IDC-PL-SL-

```
03411      1  5 C: FILA:= I+4; COLUMNA:= 28;
03412      1  5 CURSOR;
03413      1  5 WRITE (CHR(27),'(0)');
03414      1  5 WRITE (' x ');
03415      1  5 WRITE (CHR(27),'(B)');
03416      1  5 READLN (CONCEPTO);
03417      1  6     IF CONCEPTO = '' THEN BEGIN
03418      1  6         WRITE (CHR(7));
03419      1  6         GOTO C;
03420      1  5     END;
03421      1  4 UNTIL LENGTH (CONCEPTO) < 25;
03422      1  5 REPEAT
03423      1  5 D: FILA:= I+4; COLUMNA:= 55;
03424      1  5 CURSOR;
03425      1  5 WRITE (CHR(27),'(0)');
03426      1  5 WRITE (' x ');
03427      1  5 WRITE (CHR(27),'(B)');
03428      1  5 READLN (IMPORTE);
03429      1  6     IF IMPORTE < 0 THEN BEGIN
03430      1  6         WRITE (CHR(7));
03431      1  6         GOTO D;
03432      1  5     END;
03433      1  4 UNTIL IMPORTE < 1000000000;
03434      1  4 R: WRITE (CHR(27),'i22;BH');
03435      1  4 WRITE ('Confirme s no la operacion ("S" s "N") : ');
03436      1  4 WRITE (CHR(27),'i0J');
03437      1  4 READLN (CH);
03438      1  5 CASE CH OF
03439      1  5     'N','n': FILA:= FILA-1;
03440      1  5     'S','s':
03441      1  6         BEGIN
03442      1  6     EXTEND (FCU);
03443      1  6     EXTEND (CUIMPORTE);
03444      1  6     WRITELN (FCU,NFACTURA);
03445      1  6     WRITELN (FCU,FECHA);
03446      1  6     WRITELN (FCU,CONCEPTO);
03447      1  6     WRITE (CUIMPORTE,IMPORTE);
03448      1  6     RESET (CUSALDO);
03449      1  6     WHILE NOT EOF (CUSALDO) DO
03450      1  6         READ (CUSALDO,SALDO);
03451      1  6         SALDO:= SALDO - IMPORTE;
03452      1  6         EXTEND (CUSALDO);
03453      1  6         WRITE (CUSALDO,SALDO);
03454      1  5     END;
03455      1  5     OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
03456      1  3     END;
03457      1  3     CLOSE (FCU);
03458      1  3     CLOSE (CUIMPORTE);
03459      1  3     CLOSE (CUSALDO);
03460      1  2 CUENTAGEN;
03461      1  2 C ***** Consulta de la cuenta *****
03462      1  2     'C','c':
03463      1  3     BEGIN
03464      1  3     ENCABEZAMIENTO;
03465      1  3     OPEN (FCU,'FCU.DAT',HISTORY:=OLD);
```

-LINE-IDC-PL-SL-

```
03466      1 3          RESET (FCU);
03467      1 3          OPEN (CUIMPORTE,'CUIMPORTE.DAT',HISTORY:=OLD);
03468      1 3          RESET (CUIMPORTE);
03469      1 3          OPEN (CUSALDO,'CUSALDO.DAT',HISTORY:=OLD);
03470      1 3          RESET (CUSALDO);
03471      1 3          J:=5; M:=1; W:=1;
03472      1 3          WRITE (CHR(27),'i5;1H');
03473      1 3 WHILE NOT EOF (FCU) DO
03474      1 4     BEGIN
03475      1 4     WHILE NOT EOF (CUIMPORTE) DO
03476      1 5         BEGIN
03477      1 5         WHILE NOT EOF (CUSALDO) DO
03478      1 6             BEGIN
03479      1 6             WRITE (CHR(27),'(0)'); WRITE ('x'); WRITE (CHR(27),'(B)');
03480      1 6             FILA:=J; COLUMNA:=0; CURSOR;
03481      1 6             WRITE (M);
03482      1 6             FILA:=J; COLUMNA:=0; CURSOR;
03483      1 6             WRITE (CHR(27),'i1C');
03484      1 6             WRITE (CHR(27),'(0)'); WRITE ('x'); WRITE (CHR(27),'(B)');
03485      1 6             READLN (FCU,S); WRITE (S);
03486      1 6             WRITE (CHR(27),'i12D');
03487      1 6             FILA:=J; COLUMNA:=0; CURSOR;
03488      1 6             WRITE (CHR(27),'i12C');
03489      1 6             WRITE (CHR(27),'(0)'); WRITE ('x'); WRITE (CHR(27),'(B)');
03490      1 6             READLN (FCU,S); WRITE (S);
03491      1 6             WRITE (CHR(27),'i25D');
03492      1 6             FILA:=J; COLUMNA:=0; CURSOR;
03493      1 6             WRITE (CHR(27),'i25C');
03494      1 6             WRITE (CHR(27),'(0)'); WRITE ('x'); WRITE (CHR(27),'(B)');
03495      1 6             READLN (FCU,S); WRITE (S);
03496      1 6             WRITE (CHR(27),'i40D');
03497      1 6             FILA:=J; COLUMNA:=0; CURSOR;
03498      1 6             WRITE (CHR(27),'i40C');
03499      1 6             WRITE (CHR(27),'(0)'); WRITE ('x');
03500      1 6             READ (CUIMPORTE,IMPORTE);
03501      1 6             WRITE (IMPORTE);
03502      1 6             WRITE (CHR(27),'i50D');
03503      1 6             FILA:=J; COLUMNA:=0; CURSOR;
03504      1 6             WRITE (CHR(27),'i50C'); WRITE ('x'); WRITE (CHR(27),'(B)');
03505      1 6             READ (CUSALDO,SALDO); WRITELN (SALDO);
03506      1 6             J:=J+1; M:=M+1;
03507      1 7             IF M = 17*W THEN BEGIN
03508      1 7             WRITE (CHR(27),'(0','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
03509      1 7             WRITELN ('j'); WRITE (CHR(27),'(B)');
03510      1 7             WRITE ('Prlse <RETURN> para continuar
03511      1 7             WRITE (CHR(27),'iOJ'); READ (CH); W:=W+1;
03512      1 7             WRITE (CHR(27),'i5;1H'); WRITE (CHR(27),'iOJ')
03513      1 5 END;          END;          (* fin del WHILE ... *)
03514      1 4 END;          (* fin del WHILE .. *)
03515      1 3 END;
03516      1 3 FILA:=22; COLUMNA:=3;
03517      1 3 CURSOR;
03518      1 3 WRITE ('Prlse <RETURN> para continuar ...');
03519      1 3 BORRADO;
03520      1 3 CLOSE (FCU); CLOSE (CUIMPORTE);CLOSE (CUSALDO);
```

-LINE-IDC-FL-SL-

```

03521      1  3 CUENTAGEN;
03522      1  2          END;          (* fin del BEGIN ... 'C' *)
03523      1  2          'H','h': BEGIN AYUDA; CUENTAGEN; END;
03524      1  2          'X','x': BORRADO;
03525      1  2          OTHERWISE BEGIN WRITE (CHR(7)); GOTO 10; END;
03526      0  0 END; END;
03527      C  0  0          (*****
03528      C  0  0          (***** CUENTASIN *****
03529      C  0  0          (*****

03530      1  0 PROCEDURE CUENTASIN;
03531      1  0 LABEL 10,11,AA,A,B,C,D,R;
03532      1  0 TYPE STRING= VARYING (300) OF CHAR;
03533      1  0 VAR FSIN: TEXT;
03534      1  0 SINSALDO,SINIMPORTE: FILE OF INTEGER;
03535      1  0 S,FECHA,NFACTURA,CONCEPTO: STRING;
03536      1  0 M,J,N,I,W,IMPORTE,SALDO: INTEGER;
03537      1  1 BEGIN
03538      1  1 COMENTARIO:= ' CUENTA DE LA CUENTA SIN CARGO A CATEDRA ';
03539      1  1 OPCUENTAS;
03540      1  1 10: OPCION;
03541      1  2 CASE CH OF
03542      C  1  2 (*****, Inicializaci3n de la cuenta *****
03543      1  2 'I','i':
03544      1  3 BEGIN
03545      1  3 11: BORRADO;
03546      1  3 WRITE (CHR(27),'i11;3H');
03547      1  3 WRITELN (
03548      1  3 'Pulse la "S" para confirmar la inicializaci3n, "N" en caso con
03549      1  3 WRITE (' ':2,'(recuerde que todas las operaciones son borradas)
03550      1  3 WRITE (CHR(27),'i0J');
03551      1  3 READLN (CH);
03552      1  4 CASE CH OF
03553      1  5 'S','s': BEGIN
03554      1  5 ENCA;
03555      1  5 INTRODATOS;
03556      1  6 REPEAT
03557      1  6 AA: WRITE (CHR(27),'i5;20H'); READLN (FECHA);
03558      1  6 IF FECHA = '' THEN BEGIN WRITE (CHR(7));GOTO AA; END;
03559      1  5 UNTIL LENGTH (FECHA) < 10;
03560      1  5 WRITE (CHR(27),'(0'); WRITE (CHR(27),'i5;28H'); WRITE (' x
03561      1  5 WRITE (CHR(27),'(B');
03562      1  5 CONCEPTO:= 'Cantidad asignada'; WRITELN (' Cantidad asignada
03563      1  6 REPEAT
03564      1  6 WRITE (CHR(27),'i5;55H'); WRITE (CHR(27),'(0'); WRITE ('
03565      1  6 READ (IMPORTE);
03566      1  5 UNTIL IMPORTE < 100000000;
03567      1  5 WRITE (CHR(27),'(B'); SALDO:= IMPORTE;
03568      1  5 OPEN (FSIN,'FSIN.DAT');
03569      1  5 OPEN (SINSALDO,'SINSALDO.DAT');
03570      1  5 OPEN (SINIMPORTE,'SINIMPORTE.DAT');
03571      1  5 REWRITE (FSIN);
03572      1  5 REWRITE (SINIMPORTE);
03573      1  5 REWRITE (SINSALDO);
03574      1  5 WRITELN (FSIN,NFACTURA);
03575      1  5 WRITELN (FSIN,FECHA);

```

-LINE-IDC-PL-SL-

```
03576      1  5          WRITELN (FSIN,CONCEPTO);
03577      1  5          WRITE (SINIMPORTE,IMPORTE);
03578      1  5          WRITE (SINSALDO,IMPORTE);
03579      1  5          CLOSE (FSIN); CLOSE (SINIMPORTE); CLOSE (SINSALDO);
03580      1  4          END;
03581      1  4          'N','n': BEGIN WRITELN; END;
03582      1  4          'H','h': BEGIN AYUDA; GOTO 11; END;
03583      1  4          OTHERWISE BEGIN WRITE (CHR(7)); GOTO 11; END;
03584      1  3          END;
03585      1  2 CUENTASIN;      END; (* Desde Begin del 'i','I': *)
03586      C 1  2 (***** Agadir operaciones *****
03587      1  2          'A','a':
03588      1  3          BEGIN
03589      1  3          ENCA;
03590      1  3          OPEN (FSIN,'FSIN.DAT',HISTORY:= OLD);
03591      1  3          OPEN (SINIMPORTE,'SINIMPORTE.DAT',HISTORY:=
03592      1  3          OPEN (SINSALDO,'SINSALDO.DAT',HISTORY:= OLD
03593      1  4 REPEAT
03594      1  4 WRITE (CHR(27),'i22;8H');
03595      1  4 WRITE ('Introduzca el nzmero de operaciones: ');
03596      1  4 WRITE (CHR(27),'i0J');
03597      1  4 READLN (N);
03598      1  3 UNTIL N < 16;
03599      1  3 FOR I:= 1 TO N DO
03600      1  4          BEGIN
03601      1  5 REPEAT
03602      1  5 A: FILA:= I+4; COLUMNA:= 10;
03603      1  5 CURSOR;
03604      1  5 READLN (NFACTURA);
03605      1  6 IF NFACTURA = '' THEN BEGIN
03606      1  6          WRITE (CHR(7));
03607      1  6          GOTO A;
03608      1  5          END;
03609      1  4 UNTIL LENGTH (NFACTURA) < 10;
03610      1  5 REPEAT
03611      1  5 B: FILA:= I+4; COLUMNA:= 18;
03612      1  5 CURSOR;
03613      1  5 WRITE (CHR(27),'(0)');
03614      1  5 WRITE (' x ');
03615      1  5 WRITE (CHR(27),'(B)');
03616      1  5 READLN (FECHA);
03617      1  6 IF FECHA = '' THEN BEGIN
03618      1  6          WRITE (CHR(7));
03619      1  6          GOTO B;
03620      1  5          END;
03621      1  4 UNTIL LENGTH (FECHA) < 10;
03622      1  5 REPEAT
03623      1  5 C: FILA:= I+4; COLUMNA:= 28;
03624      1  5 CURSOR;
03625      1  5 WRITE (CHR(27),'(0)');
03626      1  5 WRITE (' x ');
03627      1  5 WRITE (CHR(27),'(B)');
03628      1  5 READLN (CONCEPTO);
03629      1  6 IF CONCEPTO = '' THEN BEGIN
03630      1  6          WRITE (CHR(7));
```

-LINE-IDC-PL-SL-

```
03631      1  6          GOTO C;
03632      1  5          END;
03633      1  4 UNTIL LENGTH (CONCEPTO) < 25;
03634      1  5 REPEAT
03635      1  5 D: FILA:= I+4; COLUMNA:= 55;
03636      1  5 CURSOR;
03637      1  5 WRITE (CHR(27),'(O)');
03638      1  5 WRITE (' x ');
03639      1  5 WRITE (CHR(27),'(B)');
03640      1  5 READLN (IMPORTE);
03641      1  6      IF IMPORTE < 0 THEN BEGIN
03642      1  6          WRITE (CHR(7));
03643      1  6          GOTO D;
03644      1  5      END;
03645      1  4 UNTIL IMPORTE < 100000000;
03646      1  4 R: WRITE (CHR(27),'i22;8H');
03647      1  4 WRITE ('Confirme s no la operacion ("S" s "N") : ');
03648      1  4 WRITE (CHR(27),'i0J');
03649      1  4 READLN (CH);
03650      1  5 CASE CH OF
03651      1  5         'N','n': FILA:= FILA-1;
03652      1  5         'S','s':
03653      1  6             BEGIN
03654      1  6                 EXTEND (FSIN);
03655      1  6                 EXTEND (SINIMPORTE);
03656      1  6                 WRITELN (FSIN,NFACTURA);
03657      1  6                 WRITELN (FSIN,FECHA);
03658      1  6                 WRITELN (FSIN,CONCEPTO);
03659      1  6                 WRITE (SINIMPORTE,IMPORTE);
03660      1  6                 RESET (SINSALDO);
03661      1  6                 WHILE NOT EOF (SINSALDO) DO
03662      1  6                     READ (SINSALDO,SALDO);
03663      1  6                 SALDO:= SALDO - IMPORTE;
03664      1  6                 EXTEND (SINSALDO);
03665      1  6                 WRITE (SINSALDO,SALDO);
03666      1  5             END;
03667      1  5             OTHERWISE BEGIN WRITE (CHR(7)); GOTO R; END;
03668      1  3             END;
03669      1  3             CLOSE (FSIN);
03670      1  3             CLOSE (SINIMPORTE);
03671      1  3             CLOSE (SINSALDO);
03672      1  2 CUENTASIN;          END;
03673      C 1  2 (***** Consulta de la cuenta *****);
03674      1  2         'C','c':
03675      1  3             BEGIN
03676      1  3                 ENCABEZAMIENTO;
03677      1  3                 OPEN (FSIN,'FSIN.DAT',HISTORY:=OLD);
03678      1  3                 RESET (FSIN);
03679      1  3                 OPEN (SINIMPORTE,'SINIMPORTE.DAT',HISTORY:=
03680      1  3                 RESET (SINIMPORTE);
03681      1  3                 OPEN (SINSALDO,'SINSALDO.DAT',HISTORY:=OLD);
03682      1  3                 RESET (SINSALDO);
03683      1  3                 J:=5; M:=1; W:=1;
03684      1  3                 WRITE (CHR(27),'i5;1H');
03685      1  3                 WHILE NOT EOF (FSIN) DO
```

-LINE-IDC-PL-SL-

```
03686      1  4      BEGIN
03687      1  4          WHILE NOT EOF (SINIMPORTE) DO
03688      1  5              BEGIN
03689      1  5                  WHILE NOT EOF (SINSALDO) DO
03690      1  6                      BEGIN
03691      1  6                          WRITE (CHR(27),'(0)'); WRITE ('x'); WRITE (CHR(27),'(B)');
03692      1  6                          FILA:=J; COLUMNA:=0; CURSOR;
03693      1  6                          WRITE (M);
03694      1  6                          FILA:=J; COLUMNA:=0; CURSOR;
03695      1  6                          WRITE (CHR(27),'i10');
03696      1  6                          WRITE (CHR(27),'(0)'); WRITE ('x'); WRITE (CHR(27),'(B)');
03697      1  6                          READLN (FSIN,S); WRITE (S);
03698      1  6                          WRITE (CHR(27),'i12D');
03699      1  6                          FILA:=J; COLUMNA:=0; CURSOR;
03700      1  6                          WRITE (CHR(27),'i12C');
03701      1  6                          WRITE (CHR(27),'(0)'); WRITE ('x'); WRITE (CHR(27),'(B)');
03702      1  6                          READLN (FSIN,S); WRITE (S);
03703      1  6                          WRITE (CHR(27),'i25D');
03704      1  6                          FILA:=J; COLUMNA:=0; CURSOR;
03705      1  6                          WRITE (CHR(27),'i25C');
03706      1  6                          WRITE (CHR(27),'(0)'); WRITE ('x'); WRITE (CHR(27),'(B)');
03707      1  6                          READLN (FSIN,S); WRITE (S);
03708      1  6                          WRITE (CHR(27),'i40D');
03709      1  6                          FILA:=J; COLUMNA:=0; CURSOR;
03710      1  6                          WRITE (CHR(27),'i40C');
03711      1  6                          WRITE (CHR(27),'(0)'); WRITE ('x');
03712      1  6                          READ (SINIMPORTE,IMPORTE);
03713      1  6                          WRITE (IMPORTE);
03714      1  6                          WRITE (CHR(27),'i50D');
03715      1  6                          FILA:=J; COLUMNA:=0; CURSOR;
03716      1  6                          WRITE (CHR(27),'i50C'); WRITE ('x'); WRITE (CHR(27),'(B)');
03717      1  6                          READ (SINSALDO,SALDO); WRITELN (SALDO);
03718      1  6                          J:=J+1; M:=M+1;
03719      1  7                              IF M = 17*W THEN BEGIN
03720      1  7                                  WRITE (CHR(27),'(0','m)'); FOR I:= 1 TO 78 DO WRITE ('q');
03721      1  7                                  WRITELN ('j'); WRITE (CHR(27),'(B)');
03722      1  7                                  WRITE ('Pzlse <RETURN> para continuar');
03723      1  7                                  WRITE (CHR(27),'i0J'); READ (CH); M:=M+1;
03724      1  7                                  WRITE (CHR(27),'i5;1H'); WRITE (CHR(27),'i0J');
03725      1  5      END; END; (* fin del WHILE ... *)
03726      1  4      END; (* fin del WHILE ... *)
03727      1  3      END;
03728      1  3      FILA:=22; COLUMNA:=3;
03729      1  3      CURSOR;
03730      1  3      WRITE ('Pzlse <RETURN> para continuar ...');
03731      1  3      BORRADO;
03732      1  3      CLOSE (FSIN); CLOSE (SINIMPORTE);CLOSE (SINSALDO);
03733      1  3      CUENTASIN;
03734      1  2          END; (* fin del BEGIN ... 'C' *)
03735      1  2          'H','h': BEGIN AYUDA; CUENTASIN; END;
03736      1  2          'X','x': BORRADO;
03737      1  2          OTHERWISE BEGIN WRITE (CHR(7)); GOTO 10; END;
03738      0  0      END; END;
03739      C  0  0          (*****
03740      C  0  0          (***** CARRERAS *****)
```

-LINE-IDC-FL-SL-

```

03741      C  0  0      (*****
03742      1  0  0 PROCEDURE CARRERAS;
03743      1  0  0 LABEL 3;
03744      1  1  1 BEGIN
03745      1  1  1 BORRADO;
03746      1  1  1 COMENTARIO:= ' Carreras de la Escuela Universitaria ';
03747      1  1  1 CENIN; WRITELN; WRITE (' ':5);
03748      1  1  1 WRITE (CHR(27),'(0)');
03749      1  1  1 FOR I:= 1 TO 70 DO
03750      1  1  1   WRITE ('q');
03751      1  1  1 WRITELN (CHR(27),'(B)');
03752      1  1  1 WRITELN (' ':20,'1. TELECOMUNICACIONES');
03753      1  1  1 WRITELN;WRITELN;
03754      1  1  1 WRITELN (' ':20,'2. INDUSTRIALES');
03755      1  1  1 WRITELN;WRITELN;
03756      1  1  1 WRITELN (' ':20,'3. TOPOGRAFIA');
03757      1  1  1 WRITELN;WRITELN;
03758      1  1  1 WRITELN (' ':20,'4. NAVAL');
03759      1  1  1 WRITELN;WRITELN;
03760      1  1  1 WRITELN (' ':20,'5. OBRAS PUBLICAS');
03761      1  1  1 WRITELN;
03762      1  1  1 WRITELN (' ':20,'X. Salida del Menz');
03763      1  1  1 WRITE (' ':5);
03764      1  1  1 WRITE (CHR(27),'(0)');
03765      1  1  1 FOR I:= 1 TO 70 DO
03766      1  1  1   WRITE ('q');
03767      1  1  1 WRITELN (CHR(27),'(B)');
03768      1  1  1 3: OPCION;
03769      1  2  2 CASE CH OF
03770      1  2  2 '1': BEGIN TELECO; CARRERAS; END;
03771      1  2  2 '2': BEGIN INDUS; CARRERAS; END;
03772      1  2  2 '3': BEGIN TOPOG; CARRERAS; END;
03773      1  2  2 '4': BEGIN NAVAL; CARRERAS; END;
03774      1  2  2 '5': BEGIN OBRAS; CARRERAS; END;
03775      1  2  2 'X','x': BORRADO;
03776      1  2  2 'H','h': BEGIN AYUDA; CARRERAS; END;
03777      1  2  2 OTHERWISE BEGIN WRITE (CHR(7)); GOTO 3; END;
03778      0  0  0 END;END;
03779      C  0  0      (*****
03780      C  0  0      (*****      ACTIV      *****
03781      C  0  0      (*****
03782      1  0  0 PROCEDURE ACTIV;
03783      1  0  0 LABEL 5;
03784      1  1  1 BEGIN
03785      1  1  1 BORRADO;
03786      1  1  1 COMENTARIO:= ' Cuentas correspondientes a la seccisn de Activid
03787      1  1  1 CENIN;WRITELN;WRITE (' ':5);
03788      1  1  1 WRITE (CHR(27),'(0)');
03789      1  1  1 FOR I:= 1 TO 70 DO WRITE ('q');
03790      1  1  1 WRITELN (CHR(27),'(B)');
03791      1  1  1 WRITELN;
03792      1  1  1 WRITELN (' ':15,'1. CUENTA GENERAL DE ACTIVIDADES DOCENTES');
03793      1  1  1 WRITELN;WRITELN;
03794      1  1  1 WRITELN (' ':15,'2. MENU DE LAS CUENTAS DE LAS CARRERAS');
03795      1  1  1 WRITELN;WRITELN;

```

-LINE-IDC-PL-SL-

```
03796      1 1 WRITELN (' ':15,'3. CUENTA SIN CARGO A CATEDRA');
03797      1 1 WRITELN;WRITELN;
03798      1 1 WRITELN (' ':15,'X. Salida del Menz');
03799      1 1 WRITELN;
03800      1 1 WRITE (' ':5);
03801      1 1 WRITE (CHR(27),'(0)');
03802      1 1 FOR I:= 1 TO 70 DO
03803      1 1   WRITE ('q');
03804      1 1 WRITELN (CHR(27),'(B)');
03805      1 2 5: OPCION; CASE CH OF
03806      1 2 '1': BEGIN CUENTAGEN; ACTIV; END;
03807      1 2 '2': BEGIN CARRERAS; ACTIV; END;
03808      1 2 '3': BEGIN CUENTASIN; ACTIV; END;
03809      1 2 'X','x': BORRADO;
03810      1 2 'H','h': BEGIN AYUDA; ACTIV; END;
03811      1 2 OTHERWISE BEGIN WRITE (CHR(7)); GOTO 5; END;
03812      0 0 END;END;
03813      C 0 0          (*****
03814      C 0 0          (***** INICIAL *****)
03815      C 0 0          (*****
03816      1 0 PROCEDURE INICIAL;
03817      1 0 LABEL 6;
03818      1 1 BEGIN
03819      1 1 BORRADO;
03820      1 1 COMENTARIO:= ' Menz de las cuentas principales ';
03821      1 1 CENIN;WRITELN;WRITE (' ':5);
03822      1 1 WRITE (CHR(27),'(0)');
03823      1 1 FOR I:= 1 TO 70 DO WRITE ('q');
03824      1 1 WRITELN (CHR(27),'(B)');
03825      1 1 WRITELN;WRITELN;
03826      1 1 WRITELN (' ':10,'1. GENERAL DE LA E.U.P. ');
03827      1 1 WRITELN (' ':44,'2. ACTIVIDADES DOCENTES ');
03828      1 1 WRITELN (' ':10,'3. MATERIAL INVENTARIABLE ');
03829      1 1 WRITELN (' ':44,'4. CULTURA GENERAL ');
03830      1 1 WRITELN (' ':10,'5. EDUCACION FISICA Y DEPORTES ');
03831      1 1 WRITELN (' ':44,'6. EXTENSION UNIVERSITARIA ');
03832      1 1 WRITELN (' ':10,'7. TRANSPORTES ');
03833      1 1 WRITELN (' ':44,'8. MANTENIMIENTO ');
03834      1 1 WRITELN (' ':10,'9. REUNIONES Y CONFERENCIAS ');
03835      1 1 WRITELN (' ':44,'A. MOBILIARIO Y EQ. INVENTARIAL ');
03836      1 1 WRITELN (' ':10,'B. ACTOS DE CARACTER SOCIAL ');
03837      1 1 WRITELN (' ':44,'C. DIETAS ');
03838      1 1 WRITELN (' ':10,'D. GASTOS DE OFICINA ');
03839      1 1 WRITELN;
03840      1 1 WRITE (' ':5);
03841      1 1 WRITE (CHR(27),'(0)');
03842      1 1 FOR I:= 1 TO 70 DO
03843      1 1   WRITE ('q');
03844      1 1 WRITELN (CHR(27),'(B)');
03845      1 1 6: OPCION;
03846      1 2 CASE CH OF
03847      1 2 '1': EUP;
03848      1 2 '2': ACTIV;
03849      1 2 '3': MATER;
03850      1 2 '4': CULTUR;
```

--LINE--IDC--PL--SL--

```

03851      1  2  '5': EDUCA;
03852      1  2  '6': EXTEN;
03853      1  2  '7': TRANS;
03854      1  2  '8': MANTE;
03855      1  2  '9': REUNION;
03856      1  2  'A','a': MOBILIA;
03857      1  2  'B','b': ACTOS;
03858      1  2  'C','c': DIETAS;
03859      1  2  'D','d': GASTOS;
03860      1  2  'H','h': AYUDA;
03861      1  2  OTHERWISE BEGIN WRITE (CHR(7)); GOTO 6; END;
03862      0  0  END;END;
03863      C  0  0      (*****
03864      C  0  0      (***** CONSULTA *****)
03865      C  0  0      (*****
03866      1  0  PROCEDURE CONSULTA;
03867      1  0  LABEL 7;
03868      1  0  VAR HORA,DIA: PACKED ARRAY [1..11] OF CHAR;
03869      1  1  BEGIN
03870      1  1  7: WRITE (CHR(27),'I22;5H');
03871      1  1  WRITE (' Si continua pulse la "S", en caso negativo la "N": ');
03872      1  1  WRITE (CHR(27),'IOJ');
03873      1  1  READLN (CHA); CH:=CHA;
03874      1  2  CASE CHA OF
03875      1  2  'S','s': INICIAL;
03876      1  2  'H','h': BEGIN AYUDA; CONSULTA; END;
03877      1  3  'N','n': BEGIN
03878      1  3      BORRADO;
03879      1  3      WRITE (CHR(27),'I10;12H');
03880      1  3      WRITE (CHR(27),'R6');
03881      1  3      FOR I:= 1 TO 3 DO
03882      1  3          WRITE (CHR(7));
03883      1  3      WRITELN (' FIN DE EJECUCION ');
03884      1  3      WRITELN;WRITELN;WRITELN;
03885      1  3      TIME (HORA);
03886      1  3      DATE (DIA);
03887      1  3      WRITELN (' Se despide a las: ',HORA);
03888      1  3      WRITELN (' del dia: ',DIA);
03889      1  3      WRITELN;WRITELN;
03890      1  2      END;
03891      1  2  OTHERWISE BEGIN WRITE (CHR(7)); GOTO 7; END;
03892      0  0  END; END;
03893      C  0  0      (*****
03894      C  0  0      (***** MODULO I. *****)
03895      C  0  0      (*****
03896      0  1  BEGIN
03897      0  1  BORRADO;
03898      0  1  COMENTARIO:= ' ARCONT ';
03899      0  1  WRITE (CHR(27),'R6');
03900      0  1  FILA:=1; COLUMNA:=17; INVERSO;
03901      0  1  WRITE (CHR(27),'IOm');
03902      0  1  WRITE (CHR(27),'I10;28H');
03903      0  1  WRITELN ('H: Instrucciones del programa');
03904      0  1  WRITELN;WRITELN;
03905      0  1  WRITELN (' ':27,'S: Ejecucion de ARCONT');

```

C:\Dati\documentos\les\alumnos\Digitalizado\realizado\UJEC - Biblioteca Universitaria - 2016

-LINE-IDC-FL-SL-

```
03906      0  1 100: OPCION;  
03907      0  2 CASE CH OF  
03908      0  3      'S','s': BEGIN  
03909      0  3          CHA:= 'S';  
03910      0  3          INICIAL;  
03911      0  4          REPEAT  
03912      0  5              REPEAT  
03913      0  5              CONSULTA;  
03914      0  4              UNTIL CHA <>'S';  
03915      0  3              UNTIL CHA <>'s';  
03916      0  2          END;  
03917      0  2      'H','h': BEGIN AYUDA; INICIAL; END;  
03918      0  2      OTHERWISE GOTO 100;  
03919      0  1 END;  
03920      0  0 END.
```

BIBLIOGRAFIA

1. "VAX PASCAL User's Guide": Knuth, Donald E; Digital Equipment Corporation. Massachusetts 1.985.
2. "Programming in Vax Pascal": Knuth, Donald E; Digital Equipment Corporation. Massachusetts, 1.985.
3. "Algoritmos + Estructuras de datos =PROGRAMAS": Wirth, Niklaus; Ed. Castillo, Madrid 1.980.
4. "Programación en PASCAL": Keller, Arthur M; Ed. Mc Graw-Hill, Madrid, 1.983.