```
/*****************************************************************
*
*     ROMPER NODO                                                 *
*                                                                 *
*     RUTINA QUE PARA ROMPER UN NODO CUANDO SU TAMAÑO ES          *
*     MAYOR QUE EL MAXIMO                                         *
*                                                                 *
*     RUTINAS QUE LLAMA:  -----                                   *
*                                                                 *
*     LLAMADA POR:     INSERTA                                    *
*                                                                 *
*     PARAMETROS:                                                 *
*       DE ENTRADA:                                               *
*       TESAURO: ARCHIVO TRATADO                                  *
*       NODO: ESTRUCTURA QUE CONTIENE INFORMACION SOBRE EL        *
*                NUMERO, NIVEL DENTRO DEL TESAURO, NUMERO DE CLA- *
*                VES(NC), TAMAÑO DEL NODO(TN), NODO VECINO IZQ., *
*                SU NODO VECINO DER., Y FINALMENTE LAS CLAVES Y  *
*               LOS APUNTADORES(ENTRADA)                          *
*         VIA: VECTOR QUE CONTIENE LAS DIRECCIONES DE LOS NODOS   *
*              INVOLUCRADOS EN LA BUSQUEDA                        *
*                                                                 *
*****************************************************************/
```

```
/*****************************************************************
*
* AYUDA:
*
* MUESTRA POR PANTALLA LA EXPLICACION DEL MANEJO DE LOS
* PRINCIPALES COMANDOS DEL SISTEMA.
*
* PARAMETROS:
*
* VARIABLES:
*
* LLAMADA POR:
*
*     SELECCION
*
* LLAMA A:
*
* ARCHIVOS:
*
*     ARCHIVO HELP
*
*****************************************************************/
```

[illegible]

RUTINA QUE SE ENCARGA DE LA [illegible] DE UN PROCESO

PARAMATROS:

J: INDICA SI EL PROCESO VA A SER INTERRUMPIDO
0: INTERUMPE EL PROCESO
1: NO INTERUMPE LA CREACION

VARIABLES: DIRE_PANTA: DIRECCION DE LA PANTALLA DE CREACION DE REINTEGROS EN EL ARCHIVO PANTALLAS
DIRE_COR: DIRECCION DONDE SE ENCUENTRAN GUARDADAS LAS COORDENAS PARA LA PANTALLA DEL REINTEGRO EN EL ARCHIVO COORDENADAS
IG: COMIENZO EN EL VECTOR DE COORDENADAS
FG: FIN EN EL VECTOR DE COORDENADAS
CONTADOR_REIN: INDICA QUE EL REINTEGRO ESTA EN MEMORIA

LLAMADA POR:

CREA_ACTIVIDAD

LLAMA A:

GENERA
BUSCA_MUEBLE
GOTOXY
CREA_INFORMACION

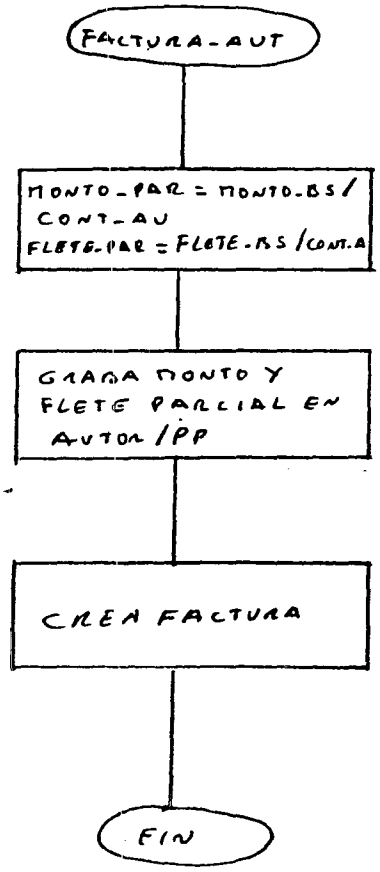
ARCHIVOS:

COORDENADAS: CONTIENE INFORMACION DE LAS COORDENADAS DE LOS DATOS DE LA PANTALLAS
PANTALLAS: CONTIENE LAS PANTALLAS DEL SISTEMA
REINTEGROS: GUARDA LOS REINTEGROS

MACRO ALGORITMO DEL PROGRAMA:

```
CREA_ACTIVIDAD:

 RUTINA ENCARGADA DE LA CREACION DE LOS PROCESOS DE
 PROFORMA,FACTURA,PEDIDO,PAGO Y RECIBIR

      PARAMETROS:

           J: INDICA EL PROCESO A SER CREADO

      VARIABLES:

           NUMERO_ARC: NUMERO DEL ARCHIVO DEL PROCESO
                       A CREAR

           NUMERO_INDICE: NUMERO DEL INDICE DEL PROCESO
                          A CREAR

           DIRE_PANTA: DIRECCION DE LA PANTALLA DEL PROCESO

           DIRE_COR: DIRECCION DE LAS COORDENADAS

      LLAMADA POR:

           SELECCION

      LLAMA A:

           GENERA_CAPTA

      ARCHIVOS:

           COORDENADAS: CONTIENE LAS COORDENADS DE LOS DATOS
                        DE LAS PANTALLAS

           PANTALLAS: CONTIENE LAS PANTALLA DEL SISTEMA

           CONTROL_PANTALLAS: CONTIENE LA DIRECCION DE LAS
                              PANTALLAS
```

```
/*****************************************************************************
*  FACTURA_AUT:
*
*  PROGRAMA QUE REALIZA AJUSTE(CUADRE) EN LA FACTURA EN
*  FORMA AUTOMATICA
*
*     PARAMETROS:
*
*          NUMERO_ACTIVIDAD: NUMERO DE LA FACTURA A LA CUAL
*                            SE LE ESTA REALIZANDO EL CUADRE
*          CADENA_LEIDA: CADENA QUE CONTIENE LA INFORMACION
*                        DE LA FACTURA
*
*     VARIABLES:
*
*          NUMERO_INDICE: NUMERO DEL INDICE DE LA FACTURA
*
*          NUMERO_ARC: NUMERO DEL ARCHIVO DE LA FACTURA
*
*          VECTOR_GUAR_NUM: VECTOR QUE CONTIENE LOS NUMEROS
*                           DE LOS AUTORES/PP INCLUIDOS EN
*                           LA FACTURA
*
*          VECTOR_GUAR_AP2: CONTIENE EL INDICADOR DE AUTOR
*                           O PP
*
*          MONTO_BS: MONTO TOTAL EN BOLIVARES
*
*          FLETE_BS: FLETE TOTAL EN BOLIVARES
*
*          MONT_PAR: MONTO POR AUTOR/PP
*
*          FLETE_PAR: FLETE POR AUTOR/PP
*
*     LLAMADA POR:
*
*          FINALIZA
*
*     LLAMA A:
*          GOTOXY
*          PIDE_VACIO
*          INSERTA
*          BUSCA
*
*****************************************************************************/
```


FACTURA-AUT
MONTO_PAR = MONTO-BS/
CONT-AU
FLETE-PAR = FLETE-BS /CONT.A
GRABA MONTO Y
FLETE PARCIAL EN
AUTOR /PP
CREA FACTURA
FIN

BIBLIOGRAFIA

1.- Miguel Ruiz, "Sistema Automatizado de Control de Usuarios", Informe de Pasantías, Escuela de Ing. de Sistemas, U.L.A., Mérida, 1982.

2.- Herbert Hoeger, "Sistema Automatizado para la Actualización del Catálogo de Publicaciones Periódicas", Informe de Pasantías, Escuela de Ing. de Sistemas, U.L.A., Mérida, 1983.

3.- "PL/I Reference Manual", Data General Corporation, 1981.

4.- Herbert Hoeger, "Sistema Automatizado de Control y Administración de Publicaciones Periódicas, Escuela de Ing. de Sistemas, 1984.

5.- Dulce M. Rivero Albarran, "Sistema Automatizado para las Adquisiciones Bibliográficas", Escuela de Ing. de Sistemas, 1984.

6.- Pablo Lischinsky, "Primera Optimización del Sistema Automatizado de Recuperación de Información", 1985.

LISTADO DE LAS "MACROS" (RUTINAS DE COMANDOS DEL SISTEMA OPERATIVO CLI) UTILIZADAS.

CHAR/OFF/EBO

X ENTRADA_DOC

CHAR/ON/EBO

ENTRADA

LISTADOS DE ALGUNOS PROGRAMAS AUXILIARES QUE SE UTILIZAN PARA INICIAR EL SISTEMA

```
          [illegible]
    [illegible] PANTA([illegible]
    [illegible]                                                  [illegible]
    02 NUM_TAB    FIXED                                          [illegible]
    02 DIRE_P                                                    [illegible]
    02 DIRE_C1                                                   [illegible]
    02 NOMBRE_P                                                  [illegible]
DCL
DCL NUMERO,I                                                     BIN FIXED,
DCL [illegible]                                                  CHAR([illegible]) VAR;

OPEN FILE(IN) TITLE('DISPAN') STREAM INPUT;
OPEN FILE(OUT) TITLE('MONITOR')  STREAM OUTPUT PRINT;
OPEN FILE(CONTROL_PANTALLAS) RECORD OUTPUT DIRECT KEYED;

DO I = 1 TO 10;
   PUT FILE(OUT) SKIP EDIT('NUM_ARC     ')(A);
   IF I=1 THEN
   GET FILE(IN) EDIT(NUMERO)(F(4));
   ELSE
   GET FILE(IN) SKIP EDIT(NUMERO)(F(4));
   CONTROL_PANTA(I).NUM_ARC = NUMERO;
   PUT FILE(OUT) SKIP EDIT('NUM_TAB    ')(A);
   GET FILE(IN) SKIP EDIT(NUMERO)(F(4));
   CONTROL_PANTA(I).NUM_TAB = NUMERO;
   PUT FILE(OUT) SKIP EDIT('DIRE_P     ')(A);
   GET FILE(IN) SKIP EDIT(NUMERO)(F(4));
   CONTROL_PANTA(I).DIRE_P = NUMERO;
   PUT FILE(OUT) SKIP EDIT('DIRE_C1    ')(A);
   GET FILE(IN) SKIP EDIT(NUMERO)(F(4));
   CONTROL_PANTA(I).DIRE_C1 = NUMERO;
   PUT FILE(OUT) SKIP EDIT('DIRE_C2    ')(A);
   READ FILE(IN) INTO(TEXTO1);
   CONTROL_PANTA(I).NOMBRE_P = TEXTO1;
END;

WRITE FILE(CONTROL_PANTALLAS) FROM(CONTROL_PANTA) KEYFROM(I);
CLOSE FILE(CONTROL_PANTALLAS);

END; /* CONTROL OF PANTALLAS */
```

```
   [illegible]
   [illegible]
   [illegible]
   [illegible]                                         [illegible]
   0? LON_REG                                          [illegible]
   0? ULT_VACIO                                        [illegible]
   02 MAX_DIR                                          BIN FIXED,
   02 NOMBRE_ARC                                       CHAR([illegible]
01 CONTROL_[illegible],
   02 NUM_[illegible]                                  [illegible] FIXED,
   02 NUM_PR[illegible]                                [illegible]
   02 NUM_REG                                          BIN FIXED,
   0[illegible]                                        [illegible]
   0[illegible]                                        [illegible]
01 CONTROL_[illegible]
   02 NUM_ARCHIVO                                      [illegible]
   02 ULT_VACIO_[illegible]                            BIN FIXED,
   02 [illegible]                                      CHAR(1),
01 CONTROL_PROCESO(2),
   02 NOMBRE                                           CHAR(2),
   02 NUM_ARCHIVO                                      BIN FIXED,
   02 PROCESO                                          BIN FIXED,
   02 PRO_CAMBIO                                       BIN FIXED;
DCL
NUMERO [illegible]                                     BIN FIXED,
TEXTO1                                                 CHAR(80) VAR;

OPEN FILE(IN) [illegible]
OPEN FILE(OUT) [illegible]
OPEN FILE(CONTROL_ARCHIVO) RECORD OUTPUT DIRECT KEYED
ENV(RECSIZE(1024));
DO I = 1 TO 16;
   PUT FILE(OUT) SKIP EDIT('TIPO:')(A);
   READ FILE(IN) INTO(TEXTO1);
   CONTROL_ARCH(I).TIPO = TEXTO1;
   PUT FILE(OUT) SKIP EDIT('N ARC:')(A);
   GET FILE(IN) SKIP EDIT(NUMERO)(F(2));
   CONTROL_ARCH(I).N_ARC = NUMERO;
   PUT FILE(OUT) SKIP EDIT('LON REG:')(A);
   GET FILE(IN) SKIP EDIT(NUMERO)(F(4));
   PUT FILE(OUT) SKIP EDIT('LONGITUD=',NUMERO)([illegible]A);
   CONTROL_ARCH(I).LON_REG = NUMERO;
   PUT FILE(OUT) SKIP EDIT('ULT VACIO:')(A);
   GET FILE(IN) SKIP EDIT(NUMERO)(F(4));
   CONTROL_ARCH(I).ULT_VACIO = NUMERO;
   PUT FILE(OUT) SKIP EDIT('MAX DIR:')(A);
   GET FILE(IN) SKIP EDIT(NUMERO)(F(4));
   CONTROL_ARCH(I).MAX_DIR = NUMERO;
   PUT FILE(OUT) SKIP EDIT('NOMBRE ARC:')(A);
   READ FILE(IN) INTO(TEXTO1);
   CONTROL_ARCH(I).NOMBRE_ARC = TEXTO1;
END;
WRITE FILE(CONTROL_ARCHIVO) FROM(CONTROL_ARCH) KEYFROM(3);
DO I = 1 TO 2;
   NOMBRE(I) = 'A'||I;
   PROCESO(I) = 0;
   PRO_CAMBIO(I) = 0;
END;
WRITE FILE(CONTROL_ARCHIVO) FROM(CONTROL_PROCESO) KEYFROM(4);
NUM_AB = 0;
```

```
            [illegible]
         [illegible]
            02 [illegible] PROV
      01 REG_PAGO,                                   [illegible]
            02 CADENA_PAGO                           CHAR(1000) VAR,
      01 REG_PED,
            02 CADENA_PED                            CHAR(1000) VAR,
      01 REG_CHEQ,
            02 CADENA_CHEQ                           CHAR(1000) VAR,
      01 REG_FAC,
            02 CADENA_FAC                            CHAR(1000) VAR,
      01 REG_REC,
            02 CADENA_REC                            CHAR(1000) VAR,
DCL (COMPRAS,FACTURA,PEDIDO,CHEQUE,PAGO,RECIBIR)     FILE;
DCL [illegible]                                      FILE;
DCL      PROVEEDOR                                   FILE;
DCL      COMPRADORAS_PP                              FILE;
DCL
      01 REG_COMPRAS_PP,
            02 CADENA_[illegible]                    CHAR(1000) VAR;

OPEN FILE(PROVEEDOR) RECORD OUTPUT DIRECT KEYED
ENV(RECSIZE(1024));
OPEN FILE(COMPRADORAS_PP) [illegible]
[illegible]
OPEN FILE(COMPRAS) RECORD OUTPUT DIRECT KEYED
ENV(RECSIZE(1024));
OPEN FILE(FACTURA) RECORD OUTPUT DIRECT KEYED
ENV(RECSIZE(1024));
OPEN FILE(PEDIDO) RECORD OUTPUT DIRECT KEYED
ENV(RECSIZE(1024));
OPEN FILE(CHEQUE) RECORD OUTPUT DIRECT KEYED
ENV(RECSIZE(1024));
OPEN FILE(PAGO) RECORD OUTPUT DIRECT KEYED
ENV(RECSIZE(1024));
OPEN FILE(RECIBIR) RECORD OUTPUT DIRECT KEYED
ENV(RECSIZE(1024));

CLOSE FILE(PROVEEDOR);
CLOSE FILE(COMPRADORAS_PP);
CLOSE FILE(COMPRAS);
CLOSE FILE(FACTURA);
CLOSE FILE(PEDIDO);
CLOSE FILE(CHEQUE);
CLOSE FILE(PAGO);
CLOSE FILE(RECIBIR);

END; /* CREA_PROVEEDOR_AUTOR */
```

```
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
  1 [illegible],
      2 [illegible]                                  [illegible]
      2 [illegible]                                  BIN FIXED,
      2 [illegible]                                  BIN FIXED,
      2 [illegible]                                  BIN FIXED,
;                                                    BIN FIXED,
   1 [illegible],
      2 [illegible],
          [illegible]                                BIT FIXED,
          [illegible]                                BIT FIXED,
          [illegible]                                BIT FIXED,
          [illegible]                                BIT FIXED,
          [illegible]                                BIT FIXED,
          [illegible]                                BIN FIXED
      2 [illegible]                                  CHARACTER VAR,
[illegible]                                  FILE VARIABLE;
DO I = 1 TO 8;
[illegible] [illegible]
    [illegible]
    [illegible]
    [illegible]
    [illegible]
    [illegible]
    [illegible]
    [illegible]
    [illegible]
END;
OPEN FILE (IFS) OUTPUT RECORD KEYED DIRECT ENVIRONMENT (REC[illegible]);
[illegible] = 2;
[illegible] = 1;
[illegible] = 2;
[illegible] = 0;
READ FILE (IFS) FROM (INF) KEYFROM(I);
[illegible] = 2;
[illegible] = 1;
[illegible] = 1;
[illegible] = 0;
[illegible] = 0;
[illegible] = [illegible]||ASCII(0)||ASCII(1)||ASCII(0)||ASCII(0)||ASCII(0)||ASCII(0)||
        ASCII(0)||ASCII(0)||ASCII(0)||ASCII(0)||ASCII(2)||[illegible];
WRITE FILE (IFS) FROM (OUTD) KEYFROM (I);
END;
END;
```

[illegible]

[illegible]

[illegible] MEJU

[illegible] PTR

[illegible] MLS

[illegible] DRSRFUACTIOX [illegible]

[illegible] CONTROL MEJU.

[illegible] MAX DIR

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible] DIR - [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

LISTADO DE LAS RUTINAS QUE FORMAN EL SISTEMA

```
COMENZAR: PROCEDURE;

/*****************************************************************
 *                                                               *
 * COMENZAR:                                                     *
 *          PROGRAMA PRINCIPAL, ESTA SIEMPRE EN MEMORIA,         *
 *   SE ENCARGA DE IDENTIFICAR LOS COMANDOS QUE SE               *
 *   ENCUENTRAN EN LINEA.                                        *
 *                                                               *
 * PARAMETROS:                                                   *
 * LLAMADA POR:                                                  *
 * LLAMA A: GOTOXY,ERROR,SELECCION                               *
 * VARIABLES:                                                    *
 *     LINEA:        VARIABLE PARA LA LECTURA DEL TECLADO        *
 *     LON:          LONGITUD DE LINEA                           *
 *     LON_1:        LONGITUD DEL COMANDO                        *
 *     COMANDO:      COMANDO QUE VA HA SER IDENTIFICADO,         *
 *                   EXTRAIDO DE LINEA                           *
 *     J:            CODIGO DEL COMANDO INTERPRETADO             *
 *     TCOM_O:       TABLA DE COMANDOS                           *
 * ARCHIVOS:                                                     *
 *     OUT:          PANTALLA                                    *
 *     IN:           TECLADO                                     *
 *     PANTALLAS:    CONTIENE TODAS LAS PANTALLAS UTILIZADAS     *
 *                   EN EL SISTEMA                               *
 *     COORDENADAS:  CONTIENE LA POSICION DE LOS DATOS EN LA     *
 *                   PANTALLA                                    *
 *     CONTROL_ARCHIVO:SE ENCUENTRAN LA INFORMACION DE CADA      *
 *                     UNO DE LOS ARCHIVOS DEL SISTEMA           *
 *     CONTROL_PANTALLAS: CONTIENE INFORMACION DE LA DIRECCION   *
 *                        DE CADA UNA DE LAS PANTALLAS Y CADA    *
 *                        UNA DE LAS COORDENADAS                 *
 *                                                               *
 *****************************************************************/


DCL
CLAVE_AU_PP                          CHAR(120) VAR EXT STATIC,
(CONT_AU_PROF,CONT_AU_FACT,CONT_AU_PAG,
 CONT_AU_RECI,CONT_AU_CHEQ,CHEQ_AUX,CONTADOR_TIT,
 PROF_AUX,FACT_AUX,PAG_AUX,RECI_AUX,TITULO_AUX,
 CONT_AU_PED,PED_AUX,OB,CONTADOR_REIN) BIN FIXED EXT STATIC,
(CONTADOR_PR,CONTADOR_AU,CONTADOR_PP)  BIN FIXED EXT STATIC,
CK                                                    CHAR(1),
VERI_NUM                                              CHAR(10),
 LINEA                                                CHAR(70) VAR,
(IN,OUT,PANTALLAS,CONTROL_ARCHIVO,COORDENADAS,
 CONTROL_PANTALLAS,REINTFGRD)                         FILE,
(I,J,LON,LON_1,MINCOM,K)                              BIN FIXED,
(AUTORI,PPI)                                 CHAR(120) VAR EXT STATIC,
(PROFORMAL,FACTURAL,PEDIDOL,CHEQUEL,PAGOL) CHAR(12) VAR EXT STATIC,
RECIBIRI                                     CHAR(12) VAR EXT STATIC,
COMANDO                                               CHAR(14) VAR,
 COMANDO1                                             CHAR(50) VAR;
DCL
GOTOXY  ENTRY(BIN FIXED,BIN FIXED),
ERROR   ENTRY,
SELECCION ENTRY(BIN FIXED,BIN FIXED,CHAR(50) VAR);
DCL
   TCOM_O(18,2) CHAR(14) STATIC INITIAL(
                '+LINEAS       ','3',
                '-LINEAS       ','3',
                'CHEQUE        ','3',
```

```
                    "FACTURA          ", "3",
                    "IMPRIMIR         ", "3",
                    "TITULOS          ", "3",
                    "+AUTOR           ", "3",
                    "+CHEQUE          ", "3",
                    "+FACTURA         ", "3",
                    "-AUTOR           ", "3",
                    "-CHEQUE          ", "3",
                    "-FACTURA         ", "3",
                    "AJUSTE           ", "3",
                    "AUTOR            ", "3",
                    "AYUDA            ", "3",
                    "BANCO            ", "3",
                    "ELIMINAR         ", "1",
                    "ESTADISTICA      ", "2",
                    "NAUTOR           ", "3",
                    "NCHEQUE          ", "3",
                    "NFACTURA         ", "3",
                    "NOBSERVACIONES   ", "3",
                    "PAGO             ", "3",
                    "PEDIDO           ", "3",
                    "PP               ", "3",
                    "+PAGO            ", "3",
                    "+PEDIDO          ", "3",
                    "+PP              ", "3",
                    "-PAGO            ", "3",
                    "-PEDIDO          ", "3",
                    "-PP              ", "3",
                    "MODIFICAR        ", "1",
                    "MONEDA           ", "3",
                    "NPAGO            ", "3",
                    "NPEDIDO          ", "3",
                    "NPP              ", "3",
                    "PRESUPUESTO      ", "2",
                    "RECIBIR          ", "3",
                    "REINTEGRO        ", "3",
                    "FIN!             ", "3",
                    "NREINTEGRO       ", "3",
                    "PROFORMA         ", "3",
                    "PROVEEDOR        ", "3",
                    "+PROFORMA        ", "3",
                    "+PROVEEDOR       ", "3",
                    "-PROFORMA        ", "3",
                    "-PROVEEDOR       ", "3",
                    "+RECIBIR         ", "3",
                    "-RECIBIR         ", "3",
                    "NPROFORMA        ", "3",
                    "NPROVEEDOR       ", "3",
                    "NRECIBIR         ", "3",
                    "NRECLAMO         ", "3");

VERI_NUM = "0123456789";

OPEN FILE(IN) TITLE("PINPUT") STREAM INPUT ENV(SCREENEDIT);
OPEN FILE(OUT) TITLE("POUTPUT") STREAM OUTPUT PRINT;
OPEN FILE(REINTEGRO) RECORD UPDATE DIRECT KEYED
ENV(RECSIZE(1009));
OPEN FILE(CONTROL_ARCHIVO) RECORD UPDATE DIRECT KEYED
ENV(RECSIZE(1024));
OPEN FILE(COORDENADAS) RECORD INPUT DIRECT KEYED;
OPEN FILE(PANTALLAS) RECORD INPUT DIRECT KEYED
ENV(RECSIZE(1009));
OPEN FILE(CONTROL_PANTALLAS) RECORD INPUT DIRECT KEYED;
```

```
CONTADOR_RETN = 0;
CONT_AU_CHEQ = 0;
CONT_AU_PED = 0;
PED_AUX = 1;
CHEQ_AUX = 1;
CONTADOR_PR = 0;
CONTADOR_PP = 0;
CONTADOR_AU = 0;
CONT_AU_PROF = 0;
CONT_AU_FACT = 0;
CONT_AU_PAG = 0;
CONT_AU_RECI = 0;
OB = 0;
PROF_AUX = 1;
FACT_AUX = 1;
RECI_AUX = 1;
PAG_AUX = 1;
CONTADOR_TIT = 0;
TITULO_AUX = 1;
AUTOR1 = "AA";
PP1 = "A";
PROFORMA1 = "";
FACTURA1 = "";
PEDIDO1 = "";
CHEQUE1 = "";
PAGO1 = "";
RECIBIR1 = "";
CLAVE_AU_PP = "A";
CK = ASCII(11);
  LINEA = "PROVEEDOR AAA";
PUT FILE(OUT) PAGE;
  GOTO A;
PRINCIPIO:
   CALL GOTOXY(0,21);
   PUT FILE(OUT) EDIT("?: ",CK)(A,A);
   READ FILE(IN) INTO(LINEA);
   DO I = 21 TO 22;
      CALL GOTOXY(0,I);
      PUT FILE(OUT) EDIT(CK)(A);
   END;
A: LINEA = LINEA!!" ";
   LON = LENGTH(LINEA);
   I = INDEX(LINEA," ");
   IF I = 1 THEN GOTO PRINCIPIO;
   COMANDO = SUBSTR(LINEA,1,I-1);
   LON_1 = LENGTH(COMANDO);
   IF LON_1 = 1 THEN MINCOM = 6;
   IF LON_1 = 2 THEN MINCOM = 26;
      [illegible] = 3 THEN MINCOM = 40;
      [illegible] THEN MINCOM = 44;
      [illegible] 1 [illegible] 14) THEN
      [illegible]
   IF LON_[illegible]
      CALL ERROR;
      GOTO PRINCI[illegible]
   END;
   J = MINCOM;
   DO WHILE(COMANDO ^= SUBSTR(TCOM_[illegible]
      J = J - 1;
   END;
   IF (J = 1 & COMANDO ^= SUBSTR(TCOM_O(J,1),1,LON_1)) THEN DO;
    /*CALL ERROR;*/
```

```
        PUT FILE(OUT) SKIP EDIT(CK,"COMANDO NO EXISTE")(A);
        GOTO PRINCIPIO;
    END;
    IF ((LON_1 + 1) = LON) THEN DO;
        K = 0;
        COMANDO1 = "";
    END;
    ELSE DO;
        LINEA = SUBSTR(LINEA,J);
        I = 1;
        DO WHILE(SUBSTR(LINEA,I,1) = " " & I < LENGTH(LINEA)+1);
            I = I + 1;
        END;
        IF (I ^= LENGTH(LINEA)+1) THEN DO;
            COMANDO1 = SUBSTR(LINEA,I);
            LON_1 = LENGTH(COMANDO1);
            COMANDO1 = SUBSTR(COMANDO1,1,LON_1-1);
            IF VERIFY(COMANDO1,VERI_NUM) ^= 0 THEN
                COMANDO1=TRANSLATE(COMANDO1,
                "ABCDEFGHIJKLMNÑOPQRSTUVWXYZÁÉÍÓÚÄÏÖÜ",
                "abcdefghijklmnñopqrstuvwxyzáéíóúäïöü");
            ELSE DO;
                /* NORMALIZACION A DOSEVE CARACTERES DE LA CLAVE NUMERICA */
                DO I = 1 TO 9-LENGTH(COMANDO1);
                    COMANDO1 = " "||COMANDO1;
                END;
            END;
            K = 1;
        END;
        ELSE DO;
            K = 0;
            COMANDO1 = "";
        END;
    END;
    CALL SELECCION(J,K,COMANDO1);
    IF J ^= 40 THEN DO;
        IF J = 0 THEN DO;
            /* ESTO ES PARA QUE FUNCIONE EL CONTROL D */
            CLOSE FILE(IN);
            OPEN FILE(IN) TITLE("@INPUT") STREAM INPUT;
        END;
        GOTO PRINCIPIO;
    END;

CLOSE FILE(CONTROL_ARCHIVO);
CLOSE FILE(COORDENADAS);
CLOSE FILE(PANTALLAS);
CLOSE FILE(CONTROL_PANTALLAS);

END;   /*          COMENZAR                    */
```

```
SELECCION: PROCEDURE(J,K,COMANDO1);

/*****************************************************************
 *                                                               *
 *     RUTINA QUE SE ENCARGA DE SELECCIONAR LA RUTINA            *
 *     CORRESPONDIENTE AL COMANDO DADO                           *
 *                                                               *
 *        PARAMETROS:                                            *
 *             J: INDICA EL NUMERO DE LA RUTINA A SELECCIONAR    *
 *             K: INDICA EL NUMERO DEL COMANDO SECUNDARIO        *
 *             COMANDO1: COMANDO SECUNDARIO                      *
 *                                                               *
 *        VARIABLES:                                             *
 *                                                               *
 *        LLAMADA POR:                                           *
 *             COMENZAR                                          *
 *                                                               *
 *        LLAMA A:                                               *
 *             MAS_MENOS_LINEAS                                  *
 *             TITULOS                                           *
 *             BUSCA_AU_PROV_PP                                  *
 *             BUSCA_ACTIVIDAD                                   *
 *             AYUDA                                             *
 *             PRIMERO                                           *
 *             CREA_AU_PROV_PP                                   *
 *             CREA_ACTIVIDAD                                    *
 *             CREA_OBSERVACION                                  *
 *             CONSULTA_REINTEGRO                                *
 *             CREA_REINTEGRO                                    *
 *                                                               *
 *        ARCHIVOS:                                              *
 *                                                               *
 *****************************************************************/

DCL
 OUT                                                    FILE,
(J,K,RUT_A_LLAMAR)                                      BIN FIXED,
COMANDO1                                                CHAR(50) VAR;
DCL
  MAS_MENOS_LINEAS ENTRY(BIN FIXED),
  TITULOS ENTRY(BIN FIXED),
  BUSCA_AU_PROV_PP ENTRY(BIN FIXED,BIN FIXED,CHAR(50) VAR),
  BUSCA_ACTIVIDAD ENTRY(BIN FIXED,BIN FIXED,CHAR(50) VAR),
  AYUDA ENTRY,
  PRIMERO ENTRY,
  CREA_AU_PROV_PP ENTRY(BIN FIXED),
  CREA_ACTIVIDAD ENTRY(BIN FIXED),
  CREA_OBSERVACION ENTRY,
  CONSULTA_REINTEGRO ENTRY,
  CREA_REINTEGRO ENTRY(BIN FIXED);

RUT_A_LLAMAR = SUBSTR(
"1 1 10102 3 4 5 5 7 6 6 8 9 111213141516161710109 5 5 4 6 6 7 "||
"181216161519102023211095 4 6 7 5 6 14151622",J*2-1,2);
DO CASE(RUT_A_LLAMAR);
    CALL MAS_MENOS_LINEAS(J);
    CALL TITULOS(J);
    CALL TITULOS(J);
    CALL BUSCA_AU_PROV_PP(J,K,COMANDO1);
    CALL BUSCA_ACTIVIDAD(J,K,COMANDO1);
    CALL BUSCA_ACTIVIDAD(J,K,COMANDO1);
    CALL BUSCA_AU_PROV_PP(J,K,COMANDO1);
 ;/*CALL AJUSTAR;*/
```

```
    CALL BUSCA_AU_PROV_PP(J,K,COMAND01);
    CALL BUSCA_ACTIVIDAD(J,K,COMAND01);
    CALL AYUDA;
    CALL PRIMERO;
  ;/*CALL ELIMINAR;*/
  ;/*CALL ESTADISTICA;*/
    CALL CREA_AU_PROV_PP(J);
    CALL CREA_ACTIVIDAD(J);
    CALL CREA_OBSERVACION;
  ;/*CALL MODIFICAR;*/
  ;/*CALL PRESUPUESTO;*/
    CALL CONSULTA_REINTEGRO;
    CALL CREA_REINTEGRO(J);
  ;/*CALL CREA_RECLAMO;*/
  ;/*CALL FINALIZACION;*/
END;
END;     /*        SELECCION          */
```

```
      N = 0;
REP:  N = N + 1;
      CALL GOTOXY(9,13+N);
      READ FILE(IN) INTO(DATO);
      IF (N = 1 & DATO = "") THEN
         GOTO WW;
      CONTADOR_NOTA = CONTADOR_NOTA + 1;
      VECTOR_NOTA(CONTADOR_NOTA) = DATO;
      IF N < 3 THEN
         GOTO REP;
      WW:
      GOTO AA;
   END;
   IF P = 8 THEN DO;
      BLOQUEADO_AC = "0"B;
      BLOQUEADO_OM = "0"B;
      PRI = "0"B;
      AC = "0"B;
      OM = "0"B;
      AD = "1"B;
      RE = "0"B;
      MAS = "1"B;
      MENOS = "0"B;
      PROF = "0"B;
      IF DIRE_PANTA = 5 THEN DO;
         NUEVO = "0"B;
         IF NUEVO_AU = "1"B THEN DO;
            NUEVO_AU = "0"B;
            DO I = 1 TO 5;
               CALL GOTOXY(0,1+6);
               PUT FILE(OUT) EDIT(CX)(A);
            END;
            CALL GOTOXY(1,11);
            PUT FILE(OUT) EDIT(ASCII(29),"MONTO:",ASCII(28))(A,A,A);
            CALL GOTOXY(3A,11);
            PUT FILE(OUT) EDIT(ASCII(29),"HIERE:",ASCII(28))(A,A,A);
         END;
      END;
      MAS_FACT = "0"B;
      MAS_PED = "0"B;
      CALL CONTINUACION;
   END;
   IF (P = 6 & MENOS = "0"B) THEN DO;
      BLOQUEADO_AC = "1"B;
      BLOQUEADO_OM = "0"B;
      AD = "1"B;
      OM = "1"B;
      RE = "1"B;
      MAS = "0"B;
      CALL MENOS_NUEVA_ACTIVIDAD;
   END;
   IF (P = 5 & MAS = "0"B) THEN DO;
      BLOQUEADO_OM = "0"B;
      CALL MAS_NUEVA_ACTIVIDAD;
   END;
   IF (P = 10 & PROF = "0"B) THEN DO;
      IF DIRE_PANTA = 6 THEN DO;
         PROF = "1"B;
         GOTO AA;
      END;
      CALL LEA_ACTIVIDAD;
      NUMERO_INDICE = 13;
      NUEVO = "1"B;
```

```
            ACTIVIDAD_LEIDA = ' ' || ACTIVIDAD_LEIDA;
         END;
      END;
      CLAVE = ACTIVIDAD_LEIDA;
      IF DIRE_PANTA = 3 THEN DO;
         O = '1'B;
         CALL BUSCA_INDICE;
CALL BUSCA(TESAURO,CLAVE,BA_ENC,POS_ABS,RES_CM,PF_CM,ION_CM,NODO,
PF_CB,VIA,ULT_N,POS_RE,INF);
IF BA_ENC = '0'B THEN DO;
   BLOQUEADO_AC = '1'B;
   BLOQUEADO_OM = '1'B;
   MAS = '1'B;
   MENOS = '1'B;
   AD = '1'B;
   RE = '1'B;
   MAS_FACT = '0'B;
   MAS_PED = '0'B;
   CALL GOTOXY(0,22);
   PUT FILE(OUT) EDIT('NO EXISTE')(A);
   GOTO FIN;
END;
      CALL NUEVO_PEDIDO;
END;
   ELSE
      GOTO ENC;
   END;
   IF (P = 14 & MAS_PED = '0'B) THEN DO;
      MAS_PED = '1'B;
      MAS_FACT = '1'B;
      IF DIRE_PANTA = 5 THEN
         GOTO AA;
      IF DIRE_PANTA = 6 THEN GOTO AA;
      F = '1'B;
      BLOQUEADO_AC = '0'B;
      BLOQUEADO_OM = '0'B;
      NUMERO_INDICE = 15;
      IF DIRE_PANTA = 8 THEN DO;
         CALL BORRA_INFORMACION(12,12,14,13,0);
         CALL GOTOXY(14,13);
         READ FILE(IN) INTO(ACTIVIDAD_LEIDA);
         IF ACTIVIDAD_LEIDA = '' THEN DO;
            BLOQUEADO_AC = '1'B;
            BLOQUEADO_OM = '1'B;
            MAS_PED = '0'B;
            MAS_FACT= '0'B;
            GOTO AA;
         END;
         DO I = 1 TO 9 - LENGTH(ACTIVIDAD_LEIDA);
            ACTIVIDAD_LEIDA = ' ' || ACTIVIDAD_LEIDA;
         END;
         CLAVE = ACTIVIDAD_LEIDA;
         O = '0'B;
         F = '1'B;
         CALL BUSCA_INDICE;
CALL BUSCA(TESAURO,CLAVE,BA_ENC,POS_ABS,RES_CM,PF_CM,ION_CM,NODO,
PF_CB,VIA,ULT_N,POS_RE,INF);
IF BA_ENC = '0'B THEN DO;
   BLOQUEADO_AC = '1'B;
   BLOQUEADO_OM = '1'B;
   MAS = '1'B;
   MENOS = '1'B;
   AD = '1'B;
```

```
        PROF = '1'B;
        BLOQUEADO_AC = '0'B;
        BLOQUEADO_OM = '0'B;
        GOTO FNC;
     END;
     IF ((P=1) | P=12) & NUEVO='0'B) THEN DO;
        PROF = '1'B;
        NUEVO = '1'B;
        NUEVO_AU = '1'B;
        BLOQUEADO_AC = '0'B;
        IF F = 1 THEN
           NNA = '0'B;
        ELSE
           NNA = '1'B;
        NUMERO_ARC = 1;
        NUMERO_INDICE = 11;
     CALL NUEVO_AUTOR(NNA,NUMERO_INDICE,NUMERO_ARC,NUM_AUT,DTR,AP2,D);
        IF D = 1 THEN DO;
/* ESTO ES PARA QUE FUNCIONE EL CONTROL D */
           CLOSE FILE(IN);
           OPEN FILE(IN) TITLE('@INPUT') STREAM INPUT;
           GOTO AA;
        END;
        CALL GOTOXY(0,11);
        PUT FILE(OUT) EDIT(CK)(A);
        CALL GOTOXY(1,11);
        PUT FILE(OUT) EDIT("PUNTO:")(A);
        CALL GOTOXY(36,11);
        PUT FILE(OUT) EDIT("FLETE:")(A);
        CALL MUESTRA_AUTOR;
     END;
     IF (P = 13 & MAS_FACT = '0'B)THEN DO;
        IF DIRE_PANTA = 5 THEN DO;
           MAS_FACT = '1'B;
           MAS_PED = '1'B;
           GOTO AA;
        END;
        BLOQUEADO_AC = '0'B;
        BLOQUEADO_OM = '0'B;
        MAS_FACT = '1'B;
        F = '0'B;
        IF DIRE_PANTA = 8 THEN DO;
           CALL BORRA_INFORMACION(12,12,40,13,0);
           CALL GOTOXY(40,13);
        END;
        ELSE DO;
           CALL BORRA_INFORMACION(13,12,14,13,0);
           CALL GOTOXY(14,13);
        END;
        READ FILE(IN) INTO(ACTIVIDAD_LEIDA);
        IF ACTIVIDAD_LEIDA = '' THEN DO;
           BLOQUEADO_AC = '1'B;
           BLOQUEADO_OM = '1'B;
           MAS_PED = '0'B;
           MAS_FACT = '0'B;
           F = '1'B;
           GOTO AA;
        END;
        IF J ^= 0 THEN
           I = I + 1;
        NUMERO_INDICE = 14;
        IF VERIFY(ACTIVIDAD_LEIDA,'1234567890') = 0 THEN DO;
           DO I = 1 TO 9 - LENGTH(ACTIVIDAD_LEIDA);
```

```
        NF = '1'B;
        NAS_PANT = '0'B;
        NAS_PED = '0'B;
        CALL GOTOXY(0,22);
        PUT STR(RCL) EDIT('NO EXISTE')(A);
        [illegible];
    [illegible];
        [illegible] FB.BB;
    [illegible];
        [illegible];
        [illegible];
        [illegible];
        END;
    END;
    IF P = 9 THEN DO;
        IF DIRE_PANTA = 4 THEN [illegible];
            IF CONT_AU_ACEP_AUX ^= 0 THEN [illegible];
  CADENA_LEIDA = CADENA_LEIDA !! ASCII(LENGTH([illegible]
  NOMBRE_PROVEEDOR !! ASCII(LENGTH(MONEDA)+1) !! MONEDA;
  POS = 1;
  LON = RANK(SUBSTR(CADENA_PED,POS,1));
  DO I = 1 TO 7;
CADENA_LEIDA = CADENA_LEIDA !! ASCII(LENGTH(SUBSTR(CADENA_PED,POS+1,LON-1))+1)!!
SUBSTR(CADENA_PED,POS+1,LON-1);
  POS = POS + LON;
  LON = RANK(SUBSTR(CADENA_PED,POS,1));
END;
CADENA_LEIDA = CADENA_LEIDA !! ASCII(LENGTH(CHAR(A_TOT_$))+1)!!
CHAR(A_TOT_$) !! ASCII(LENGTH(CHAR(A_TOT_B))+1)!! CHAR(A_TOT_B)!!
ASCII(LENGTH(CHAR(P_TOT_$))+1) !! CHAR(P_TOT_$) !!
ASCII(LENGTH(CHAR(P_TOT_B))+1) !! CHAR(P_TOT_B)!! ASCII(253) !!
ASCII(LENGTH(CHAR(CONT_AU_ACEP_AUX))+1) !! CHAR(CONT_AU_ACEP_AUX)!!
ASCII(0,4);
  POS = 1;
  LON = RANK(SUBSTR(CADENA_PED,POS,1));
  DO I = 1 TO 4;
     POS = POS + LON;
     LON = RANK(SUBSTR(CADENA_PED,POS,1));
  END;
  A_TOTAL_$=A_TOTAL_$+DECIMAL(SUBSTR(CADENA_PED,POS+1,LON-1),10,0);
  POS = POS + LON;
  LON = RANK(SUBSTR(CADENA_PED,POS,1));
A_TOTAL_BS=A_TOTAL_BS+DECIMAL(SUBSTR(CADENA_PED,POS+1,LON-1),10,0);
END;
A_TOT_$_F = A_TOTAL_$;
A_TOT_B_F = A_TOTAL_BS;
END;
        IF DIRE_PANTA = 8 THEN DO;
            IF CONT_AU_ACEP ^= 0 THEN DO;
                CADENA_LEIDA = CADENA_LEIDA !!
                ASCII(LENGTH(PROVEEDOR)+1)!!PROVEEDOR;
            END;
        END;
```

```
          IF (DIRE_PANTA = 5 & CONT_AU_ACEP ^= 0)THEN DO;
             CALL CUADRE_ACTIVIDAD(CONT_AU_ACEP,VECTOR_GUAR_NUM,
CADENA_LEIDA,DIRE_PANTA,NUMERO_ACTIVIDAD,CUADRE);
             IF CUADRE = "0"B THEN DO;
                CALL GOTOXY(0,22);
                PUT FILE(OUT) EDIT("LA FACTURA NO CUADRA")(A);
                GOTO AA;
             END;
          END;
          CALL FINALIZA(CONT_AU_ACEP,DIRE_PANTA,NITTIH,NF_HMP,FECHA,
          CADENA_LEIDA,A_TOT_$_F,A_TOT_B_F,P_TOT_$_F,P_TOT_B_F,
          NUMERO_ACTIVIDAD,VECTOR_GUAR_NUM,VECTOR_GUAR_AP2);
          IF DIRE_PANTA = 4 THEN
             CALL DESACTIVA;
       END;
    END;
    ELSE
       GOTO AA;

END;   /*    INTERPRETA    */

DESACTIVA: PROCEDURE;

NUMERO_INDICE = 12;
CALL BUSCA_INDICE;
CONT_AUX_AU = CONT_AU;
I = 1;
DO WHILE(I < CONT+1);
   K = 1;
   DO WHILE(K < CONT_AU_ACEP+1);
   IF ((VECTOR(I)=VECTOR_GUAR_NUM(K))&VECTOR_GUAR_AP2(K)=1) THEN DO;
      CONT_AUX_AU = CONT_AUX_AU - 1;
      CLAVE2 = VECTOR_NUM(I);
      CALL ELIMINA_ARB(TESAURO,CLAVE2);
      K = CONT_AU_ACEP;
   END;
   K = K + 1;
   END;
   I = I + 1;
END;
CONT_AU = CONT_AUX_AU;
CALL BUSCA(TESAURO,CLAVE1,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,NODO,
PF_CB,VIA,ULT_N,POS_RE,INF);
SUBSTR(ENTRADA,POS_ABS+5,1) = ASCII(CONT_AU/256);
SUBSTR(ENTRADA,POS_ABS+6,1) = ASCII(CONT_AU);
REWRITE FILE(INDICE) FROM(NODO) KEY(NODO.CAB_NODO.NUM_N);

END;   /*    DESACTIVA    */

MUESTRA_AUTOR: PROCEDURE;

      READ FILE(ARCH_DATOS) INTO(BLQ_MONO_PP) KEY(DIR);
/*    IF DIRE_PANTA = 4 THEN DO;
         POS = 1;
         LON = RANK(SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS,1));
         CLAVE_AUX = "FA"||CLAVE;
         DO WHILE(CLAVE_AUX ^= SUBSTR(CADENA_LEIDA2,POS+1,LON-1));
            POS = POS + LON;
            LON = RANK(SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS,1));
         END;
         POS = POS + LON;
         LON = RANK(SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS,1));
         MONTO = SUBSTR(CADENA_LEIDA2,POS+1,LON-1);
```

```
        END; */
        POS = 1;
        I = 1;
REPI: LON = RANK(SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS,1));
        CLAVE2 = SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS+1,LON-1);
        POS = POS + LON;
        LON = RANK(SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS,1));
        LON = LON - 1;
        IF LON ^= 0 THEN
           CLAVE2 = CLAVE2||SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS+1,LON);
        IF AP2 = 0 THEN DO;
           CALL GOTOXY(8,14);
           PUT FILE(OUT) EDIT(CK)(A);
           CALL GOTOXY(65,14);
           PUT FILE(OUT) EDIT("!")(A);
           GOTO XX;
        END;
        IF I = 1 THEN DO;
           I = I + 1;
           POS = POS + LON + 1;
           AUTOR = CLAVE2;
           GOTO REPI;
        END;
        CALL GOTOXY(8,14);
        PUT FILE(OUT) EDIT(CK)(A);
        PUT FILE(OUT) EDIT(ASCII(29),AUTOR,ASCII(28))(A,A,A);
        CALL GOTOXY(65,14);
        PUT FILE(OUT) EDIT("!")(A);
   XX: CALL GOTOXY(9,16);
        PUT FILE(OUT) EDIT(CK)(A);
        PUT FILE(OUT) EDIT(ASCII(29),CLAVE2,ASCII(28))(A,A,A);
        CALL GOTOXY(65,16);
        PUT FILE(OUT) EDIT("!")(A);

END; /*     MUESTRA_AUTOR     */


CONTINUACION: PROCEDURE;

      IF DIRE_PANTA = 6 THEN DO;
         IF CONT_AU_ACEP_AUX = 0 THEN
            CON = '0'B;
      END;
      IF (GUAR_K = 0 & GUAR_I = 0) THEN
         GOTO RR;
      IF TEXTO = '1'B THEN DO;
         TEXTO = '0'B;
         CALL GOTOXY(2,14);
         PUT FILE(OUT) EDIT('AUTOR:',CK)(A,A);
         CALL GOTOXY(2,15);
         PUT FILE(OUT) EDIT(CK)(A);
         CALL GOTOXY(2,16);
         PUT FILE(OUT) EDIT('TITULO:',CK)(A,A);
         CALL GOTOXY(2,17);
         PUT FILE(OUT) EDIT(CK)(A);
         I = 14;
         DO N = 1 TO 4;
            CALL GOTOXY(65,I);
            PUT FILE(OUT) EDIT("!")(A);
            I = I + 1;
         END;
      END;
      K = GUAR_K;
```

```
      J = GUAR_J;
      IF (NUMERO_INDICE=12 ! NUMERO_INDICE=13 ! NUMERO_INDICE=14 ! NUMERO_INDICE
= 15 ! NUMERO_INDICE = 17)THEN
         GOTO NE;
      CALL ASIGNACION_INDICE;
NE:   CLAVE2 = CLAVE !! CHAR(GUAR_K);
      CALL BUSCA(TESAURO,CLAVE2,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,NODO,
      PF_CB,VTA,UET_N,POS_RE,INF);
      NUM_AUT = RANK(SUBSTR(ENTRADA,POS_ABS+5,1))*256 +
      RANK(SUBSTR(ENTRADA,POS_ABS+6,1));
      AP2 = DEC(RANK(SUBSTR(ENTRADA,POS_ABS+7,1)))*10000 +
      RANK(SUBSTR(ENTRADA,POS_ABS+8,1))*100 +
      RANK(SUBSTR(ENTRADA,POS_ABS+9,1));
      NUMERO_INDICE = 11;
      CALL BUSCA_INDICE;
      CLAVE2 = CHAR(NUM_AUT);
      CALL BUSCA(TESAURO,CLAVE2,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,NODO,
      PF_CB,VTA,UET_N,POS_RE,INF);
      DIR = DEC(RANK(SUBSTR(ENTRADA,POS_ABS+2,1)))*10000 +
      RANK(SUBSTR(ENTRADA,POS_ABS+3,1))*100 +
      RANK(SUBSTR(ENTRADA,POS_ABS+4,1));
      CALL MUESTRA_AUTOR;
      GUAR_J = 0;
      GUAR_K = 0;

   NE:

END;    /*   CONTINUACION    */

MENOS_NUEVA_ACTIVIDAD: PROCEDURE;

IF PRI = '0'B THEN DO;
   PRI = '1'B;
   GUAR_J = J;
   GUAR_K = K;
   AUXILIAR = CONT_AU_ACEP + 1;
END;
AUXILIAR = AUXILIAR - 1;
IF AUXILIAR > 0 THEN DO;
   CLAVE2 = VECTOR_GUAR_NUM(AUXILIAR);
   AP2 = VECTOR_GUAR_AP2(AUXILIAR);
   IF NUMERO_INDICE = 11 THEN
      GOTO LE;
      NUMERO_INDICE = 11;
      CALL BUSCA_INDICE;
LE:   CALL BUSCA(TESAURO,CLAVE2,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,NODO,
      PF_CB,VTA,UET_N,POS_RE,INF);
      DIR = DEC(RANK(SUBSTR(ENTRADA,POS_ABS+2,1)))*10000 +
      RANK(SUBSTR(ENTRADA,POS_ABS+3,1))*100 +
      RANK(SUBSTR(ENTRADA,POS_ABS+4,1));
      CALL MUESTRA_AUTOR;
   END;
   ELSE
      AUXILIAR = 1;
   IF CONT_AU_ACEP = 0 THEN
      MAS = '1'B;

END;     /*  MENOS NUEVA ACTIVIDAD     */

MAS_NUEVA_ACTIVIDAD: PROCEDURE;

AUXILIAR = AUXILIAR + 1;
IF AUXILIAR >= CONT_AU_ACEP + 1 THEN
   AUXILIAR = CONT_AU_ACEP;
```

```
CLAVE2 = VECTOR_GUAR_NUM(AUXILIAR);
AP2 = VECTOR_GUAR_AP2(AUXILIAR);
IF NUMERO_INDICE = 11 THEN
   GOTO LI;
     NUMERO_INDICE = 11;
     CALL BUSCA_INDICE;
LI:  CALL BUSCA(TESAURO,CLAVE2,BA_FNC,POS_ABS,RES_CM,PF_CM,LON_CM,NODO,
     PF_CB,VTA,ULT_N,POS_RE,INF);
     DIR = DEC(RANK(SUBSTR(ENTRADA,POS_ABS+2,1)))*10000 +
     RANK(SUBSTR(ENTRADA,POS_ABS+3,1))*100 +
     RANK(SUBSTR(ENTRADA,POS_ABS+4,1));
     CALL MUESTRA_AUTOR;

END;  /*     MAS_NUEVA_ACTIVIDAD      */

BUSCA_INDICE: PROCEDURE;

   CLOSE FILE(INDICE);
   I = 1;
   DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_INDICE);
      I = I + 1;
   END;
   OPEN FILE(INDICE) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) UPDATE
   DIRECT KEYED ENV(RECSIZE(1024));
END; /*  BUSCA_INDICE  */

BUSCA_PROVEEDOR: PROCEDURE;

      IF (CONT_AU_ACEP_AUX = 0 & CREA = '1'B & CON = '1'B)THEN
         CALL QUITA_PROVEEDOR;
      IF CON = '0'B THEN DO;
         CON = '1'B;
         IF CREA = '1'B THEN
            GOTO MMM;
      END;
      IF T ^= 0 THEN DO;
         PRIMERA = '1'B;
  CADENA_LEIDA = CADENA_LEIDA !! ASCII(LENGTH(NOMBRE_PROVEEDOR)+1)!!
  NOMBRE_PROVEEDOR !! ASCII(LENGTH(MONEDA)+1) !! MONEDA;
         POS = 1;
         LON = RANK(SUBSTR(CADENA_PED,POS,1));
         DO I = 1 TO 7;
            CADENA_LEIDA = CADENA_LEIDA !!
            ASCII(LENGTH(SUBSTR(CADENA_PED,POS+1,LON-1))+1)!!
            SUBSTR(CADENA_PED,POS+1,LON-1);
            POS = POS + LON;
            LON = RANK(SUBSTR(CADENA_PED,POS,1));
         END;
         CADENA_LEIDA = CADENA_LEIDA !! ASCII(LENGTH(CHAR(A_TOT_$))+1)!!
         CHAR(A_TOT_$)!!ASCII(LENGTH(CHAR(A_TOT_B))+1)!!
         CHAR(A_TOT_B)!!ASCII(LENGTH(CHAR(P_TOT_$))+1)!!
         CHAR(P_TOT_$)!!ASCII(LENGTH(CHAR(P_TOT_B))+1)!!
CHAR(P_TOT_B)!!ASCII(253)!!ASCII(LENGTH(CHAR(CONT_AU_ACEP_AUX))+1)!!
         CHAR(CONT_AU_ACEP_AUX)!!ASCII(254);
   POS = 1;
   LON = RANK(SUBSTR(CADENA_PED,POS,1));
   DO I = 1 TO 4;
      POS = POS + LON;
      LON = RANK(SUBSTR(CADENA_PED,POS,1));
   END;
   A_TOTAL_$=A_TOTAL_$+DECIMAL(SUBSTR(CADENA_PED,POS+1,LON-1),10,2);
   POS = POS + LON;
   LON = RANK(SUBSTR(CADENA_PED,POS,1));
```

```
   A_TOTAL_BS = A_TOTAL_BS + DECIMAL(SUBSTR(CADENA_PED,POS+1,LON-1),10,2);
   CONT_AU_ACEP_AUX = 0;
   DO I = 1 TO CONT_AU_ACEP;
      VECTOR_MARCA(I) = 1;
   END;
   END;
   ELSE DO;
      CREA = '1'B;
      CADENA_LEIDA = ASCII(LENGTH(CHAR(NUMERO_ACTIVIDAD))+1)!!
      CHAR(NUMERO_ACTIVIDAD);
   END;
HHH: DIR = DEC(RANK(SUBSTR(ENTRADA,POS_ABS+2,1)))*10000 +
     RANK(SUBSTR(ENTRADA,POS_ABS+3,1))*100 +
     RANK(SUBSTR(ENTRADA,POS_ABS+4,1));
     NUMERO_ARC = 4;
     I = 1;
     DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_ARC);
        I = I + 1;
     END;
     LONGITUD = CONTROL_ARCH(I).LON_REG;
     CLOSE FILE(ARCH_DATOS);
     OPEN FILE(ARCH_DATOS) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) UPDATE
     DIRECT KEYED ENV(RECSIZE(LONGITUD));
     READ FILE(ARCH_DATOS) INTO(BLQ_FACT) KEY(DIR);
     POS = 1;
     LON = RANK(SUBSTR(CADENA_FACT,POS,1));
     NOMBRE_PROVEEDOR = SUBSTR(CADENA_FACT,POS+1,LON-1);
     DO I = 1 TO 2;
        POS = POS + LON;
        LON = RANK(SUBSTR(CADENA_FACT,POS,1));
     END;
     MONEDA = SUBSTR(CADENA_FACT,POS+1,LON-1);
     CALL GOTOXY(12,1);
     PUT FILE(OUT) EDIT(NOMBRE_PROVEEDOR)(A);
     CALL GOTOXY(1,5);
     PUT FILE(OUT) EDIT(MONEDA)(A);
     CALL GOTOXY(5,8);
     PUT FILE(OUT) EDIT(CK)(A);
     CALL GOTOXY(8,5);
     PUT FILE(OUT) EDIT(CK)(A);
   CALL CREA_INFORMACION(DIRE_PANTA,DIRE_COR1,IG,FG,IV,FV,CADENA_PED);
  END; /*   BUSCA_PROVEEDOR    */

LEA_ACTIVIDAD: PROCEDURE;

    CALL GOTOXY(14,13);
    READ FILE(IN) INTO(ACTIVIDAD_LEIDA);
    IF (DIRE_PANTA = 8 & ACTIVIDAD_LEIDA = '')THEN DO;
       CALL GOTOXY(40,23);
       READ FILE(IN) INTO(ACTIVIDAD_LEIDA);
       O = '1'B;
       IF VERIFY(ACTIVIDAD_LEIDA,"123456789C") = 0 THEN DO;
          DO I = 1 TO 9 - LENGTH(ACTIVIDAD_LEIDA);
             ACTIVIDAD_LEIDA = ' ' !! ACTIVIDAD_LEIDA;
          END;
       END;
    END;
    IF ACTIVIDAD_LEIDA = '' THEN DO;
       BLOQUEADO_AC = '1'B;
       BLOQUEADO_OK = '1'B;
       NUEVO = '0'B;
       MAS_PED = '0'B;
       MAS_FACT = '0'B;
```

```
        GOTO AA;
     END;
     IF (DIRE_PANTA ^= 4 & O = '0"B)THEN DO;
/* NORMALIZACION A NUEVE CARACTERES DEL NUMERO DE ACTIVIDAD LEIDA */
        DO I = 1 TO 9-LENGTH(ACTIVIDAD_LEIDA);
           ACTIVIDAD_LEIDA = " "||ACTIVIDAD_LEIDA;
        END;
      END;
      IF DIRE_PANTA = 4 THEN DO;
        IF VERIFY(ACTIVIDAD_LEIDA,"1234567890") = 0 THEN DO;
           DO I = 1 TO 9 - LENGTH(ACTIVIDAD_LEIDA);
              ACTIVIDAD_LEIDA = " " || ACTIVIDAD_LEIDA;
           END;
        END;
      END;
      CLAVE = ACTIVIDAD_LEIDA;

END;    /*    LEA_ACTIVIDAD      */

ASIGNACION_INDICE: PROCEDURE;

    IF DIRE_PANTA = 4 THEN
       NUMERO_INDICE = 12;
    IF DIRE_PANTA = 5 THEN
       NUMERO_INDICE = 13;
    IF DIRE_PANTA = 6 THEN
       NUMERO_INDICE = 14;
    IF DIRE_PANTA = 8 THEN DO;
       IF F = '0"B THEN
          NUMERO_INDICE = 14;
       ELSE
          NUMERO_INDICE = 15;
    END;
    IF DIRE_PANTA = 9 THEN
       NUMERO_INDICE = 17;
    CALL BUSCA_INDICE;

END;     /*    ASIGNACION     INDICE       */

NUEVO_PEDIDO: PROCEDURE;

      IF CONT_AU_ACEP = 0 THEN
         PROVEEDOR = "";
      DIR = DEC(RANK(SUBSTR(ENTRADA,POS_ARG+2,1)))*10000 +
      RANK(SUBSTR(ENTRADA,POS_ARG+3,1))*100 +
      RANK(SUBSTR(ENTRADA,POS_ARG+4,1));
      SW = 0;
      IF O = '1"B THEN
         NUMERO_ARC = 4;
      ELSE
         NUMERO_ARC = 5;
      I = 1;
      DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_ARC);
         I = I + 1;
      END;
      LONGITUD = CONTROL_ARCH(I).LON_REG;
      CLOSE FILE(ARCH_DATOS);
      OPEN FILE(ARCH_DATOS) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) UNDATE
      DIRECT KEYED ENV(RECSIZE(LONGITUD));
      IF O = '1"B THEN DO;
         READ FILE(ARCH_DATOS) INTO(BLQ_FACT) KEY(DIR);
         POS = 1;
         LON = RANK(SUBSTR(CADENA_FACT,POS,1));
```

```
        PROVEEDOR1 = SUBSTR(CADENA_FACT,POS+1,LON-1);
        CALL GOTOXY(16,2);
        PUT FILE(OUT) EDIT(PROVEEDOR1)(A);
        IF PROVEEDOR = '' THEN
           PROVEEDOR = PROVEEDOR1;
        ELSE DO;
           IF PROVEEDOR ^= PROVEEDOR1 THEN DO;
              CALL GOTOXY(0,22);
              PUT FILE(OUT) EDIT("PROVEEDOR EQUIVOCADO")(A);
              GOTO AA;
           END;
        END;
        CONT_AU = RANK(SUBSTR(ENTRADA,POS_ABS+5,1))*256 +
        RANK(SUBSTR(ENTRADA,POS_ABS+6,1));
    NUMERO_ARC = 1;
    I = 1;
    DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_ARC);
       I = I + 1;
    END;
    LONGITUD = CONTROL_ARCH(I).LON_REG;
    CLOSE FILE(ARCH_DATOS);
    OPEN FILE(ARCH_DATOS) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) UPDATE
    DIRECT KEYED ENV(RECSIZE(LONGITUD));
    K = 0;
    J = 0;
    CONT = 0;
    GOTO CONTINUE;
END;
ELSE DO;
    READ FILE(ARCH_DATOS) INTO(BLQ_PED) KEY(DIR);
    POS = 1;
    LON = RANK(SUBSTR(CADENA_PED,POS,1));
    POS = POS + LON;
    LON = RANK(SUBSTR(CADENA_PED,POS,1));
    PROVEEDOR1 = SUBSTR(CADENA_PED,POS+1,LON-1);
    CALL GOTOXY(16,2);
    PUT FILE(OUT) EDIT(PROVEEDOR1,CK)(A,A);
    CALL GOTOXY(61,18);
    READ FILE(IN) INTO(RESP);
    IF (RESP = 'A' | RESP = 'AC') THEN
       CALL ACEPTADO;
REPITA:
    DO WHILE(LON ^= 253 & LON ^= 255);
       POS = POS + LON;
       LON = RANK(SUBSTR(CADENA_PED,POS,1));
    END;
    IF LON = 253 THEN DO;
       POS = POS + 1;
       LON = RANK(SUBSTR(CADENA_PED,POS,1));
       RR = RR + BINARY(SUBSTR(CADENA_PED,POS+1,LON-1));
       POS = POS + LON + 1;
       LON = RANK(SUBSTR(CADENA_PED,POS,1));
       PROVEEDOR1 = SUBSTR(CADENA_PED,POS+1,LON-1);
       IF VERIFY(PROVEEDOR1,"098765432 ") = 0 THEN
          GOTO AA;
       CALL GOTOXY(16,2);
       PUT FILE(OUT) EDIT(PROVEEDOR1)(A);
       CALL GOTOXY(63,18);
       READ FILE(IN) INTO(RESP);
       IF (RESP = 'A' | RESP = 'AC') THEN
          CALL ACEPTADO;
       ELSE
          GOTO REPITA;
```

```
   END;
   IF LON = 255 THEN
      GOTO AA;
END;

END;    /*     NUEVO_PEDIDO      */

ACEPTADO: PROCEDURE;

IF PROVEEDOR = "" THEN
   PROVEEDOR = PROVEEDOR1;
ELSE DO;
   IF PROVEEDOR ^= PROVEEDOR1 THEN DO;
      CALL GOTOXY(0,22);
      PUT FILE(OUT) EDIT("PROVEEDOR EQUIVOCADO")(A);
      GOTO AA;
   END;
END;
DO WHILE(LON ^= 255);
   POS = POS + LON;
   LON = RANK(SUBSTR(CADENA_PED,POS,1));
END;
POS = POS + 1;
LON = RANK(SUBSTR(CADENA_PED,POS,1));
CONT_AU_PROV = BINARY(SUBSTR(CADENA_PED,POS+1,LON-1));
CONT = 0;
J = RR;
K = RR;
CONT_AU = RR + CONT_AU_PROV;
NUMERO_ARC = 1;
I = 1;
DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_ARC);
   I = I + 1;
END;
LONGITUD = CONTROL_ARCH(I).LON_REG;
CLOSE FILE(ARCH_DATOS);
OPEN FILE(ARCH_DATOS) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) UPDATE
DIRECT KEYED ENV(RECSIZE(LONGITUD));
GOTO CONTINUE;

END;     /*    ACEPTADO    */

QUITA_PROVEEDOR: PROCEDURE;

IF PRIMERA = "C"B THEN DO;
   T = 0;
   CADENA_LEIDA = "";
END;
ELSE
   GOTO AA;

END; /*  QUITA_AUTOR  */
/*****************************************************************************/

FIN:
IF P ^= 9 THEN
   GOTO AA;

FINIO:

 CLOSE FILE(ARCH_DATOS);
 CLOSE FILE(INDICE);
```

```
IF ((DIRE_PANTA=4 | DIRE_PANTA=4 | DIRE_PANTA=8) & CONT_AU_ACEP ^= 0) THEN
   CALL REALIZA_CARTA(DIRE_PANTA,CONTADOR_NOTA,VECTOR_NOTA,
   NUMERO_ACTIVIDAD);

END;    /*     MUESTRA_INFORMACION      */
```

```
GENERA_CAPTA: PROCEDURE(DIRE_PANTA,DIRE_COR,IG,FG,IV,FV,CADENA_LEIDA);

/*****************************************************************
*                                                                *
* RUTINA QUE INDICA DE ACUERDO A LA DIRECCION DE LA PANTALLA     *
* (DIRE_PANTA) QUE ES LO QUE SE DESEA HACER, Y DE ACUERDO CON    *
* LO QUE SE DESEA LLAMA A LA RUTINA DETERMINADA                  *
*                                                                *
* PARAMETROS:                                                    *
*                                                                *
*     DIRE_PANTA: DIRECCION DE LA PANTALLA EN EL ARCHIVO         *
*     PANTALLAS                                                  *
*                                                                *
*     DIRE_COR: DIRECCION DE LAS COORDENADAS DE LOS DATOS EN     *
*     EL ARCHIVO COORDENADAS                                     *
*                                                                *
*     IG: INICIO DE LAS COORDENADS EN EL VECTOR DE COORDENADAS   *
*     DEL ARCHIVO COORDENADAS                                    *
*                                                                *
*     FG: FIN DE LAS COORDENADAS                                 *
*                                                                *
*     IV: INICIO DEL GRUPO REPETITIVO                            *
*                                                                *
*     FV: FIN DEL GRUPO REPETITIVO                               *
*                                                                *
* VARIABLES:                                                     *
*                                                                *
* LLAMA A:                                                       *
*                                                                *
*     CREA_INFORMACION                                           *
*                                                                *
*     MUESTRA_INFORMACION                                        *
*                                                                *
*     GENERA                                                     *
*                                                                *
*     CREA_CHEQUE                                                *
*                                                                *
* LLAMADA POR:                                                   *
*                                                                *
*     CREA_ACTIVIDAD                                             *
*                                                                *
*     CREA_AU_PROV_PP                                            *
*                                                                *
*****************************************************************/


DCL
(DIRE_PANTA,DIRE_COR,IV,FV,IG,FG)                    BIN FIXED,
CADENA_LEIDA                                         CHAR(1000) VAR,
(IN,OUT)                                             FILE,
GENERA ENTRY(BIN FIXED,BIN FIXED,BIN FIXED),
MUESTRA_INFORMACION ENTRY(BIN FIXED,BIN FIXED,BIN FIXED,BIN FIXED,BIN FIXED,BIN
FIXED),
CREA_INFORMACION ENTRY(BIN FIXED,BIN FIXED,BIN FIXED,BIN FIXED,BIN FIXED,BIN FI)
ED,CHAR(1000) VAR),
CREA_CHEQUE ENTRY;
PUT FILE(OUT) PAGE;
CALL GENERA(DIRE_PANTA,IV,FV);
IF (DIRE_PANTA = 4 ! DIRE_PANTA = 5 ! DIRE_PANTA = 6 ! DIRE_PANTA = 8 ! DIRE_PAN
TA = 9)THEN
   CALL MUESTRA_INFORMACION(DIRE_PANTA,DIRE_COR,IG,FG,IV,FV);
IF (DIRE_PANTA = 1 ! DIRE_PANTA = 2 ! DIRE_PANTA = 3) THEN
   CALL CREA_INFORMACION(DIRE_PANTA,DIRE_COR,IG,FG,IV,FV,CADENA_LEIDA);
IF DIRE_PANTA = 7 THEN
   CALL CREA_CHEQUE
```

```
MAS_MENOS_LINEAS: PROCEDURE(J);

/*****************************************************************************
 *                                                                           *
 *  RUTINA QUE SE UTILIZA PARA MOVERSE ENTRE LOS AUTORES DE LOS              *
 *  PROCESOS Y ENTRE LAS OBSERVACIONES                                       *
 *                                                                           *
 *     PARAMETROS:                                                           *
 *                                                                           *
 *         J: INDICA SI EL MOVIMIENTO ES HACIA DELANTE O HACIA               *
 *            ATRAZ                                                          *
 *                                                                           *
 *            1: MOVIMIENTO HACIA DELANTA (SI NO HAY MAS AUTORES             *
 *               MUESTRA LAS OBSERVACIONES)                                  *
 *            0: MOVIMIENTO HACIA ATRAZ                                      *
 *                                                                           *
 *     VARIABLES:                                                            *
 *                                                                           *
 *         DIRE_PANTA: DIRECCION DE LA PANTALLA                              *
 *                                                                           *
 *         DIRE_COR: DIRECCION DE LAS COORDENADAS DE LA PANTALLA             *
 *                                                                           *
 *         IG: INICIO DE COORDENADAS EN EL VECTOR DE COORDENADAS             *
 *                                                                           *
 *         FG: FIN DE COORDENADAS EN EL VECTOR DE COORDENADAS                *
 *                                                                           *
 *         CONTADOR_REIN: REINTEGRO EN MEMORIA                               *
 *                                                                           *
 *         CONTADOR_PROV: PROVEEDOR EN MEMORIA                               *
 *                                                                           *
 *         CONTADOR_AU: AUTOR EN MEMORIA                                     *
 *                                                                           *
 *         CONTADOR_PP: PP EN MEMORIA                                        *
 *                                                                           *
 *         CONT_AU_PROF: PROFORMA EN MEMORIA(IGUAL PARA LOS                  *
 *                       DEMAS PROCESOS)                                     *
 *                                                                           *
 *         CLAVE2: NOMBRE DEL PROCESO                                        *
 *                                                                           *
 *         NUMERO_INDICE: NUMERO DEL INDICE DONDE SE ESTAN                   *
 *                        CONSULTANDO LOS AUTORES                            *
 *                                                                           *
 *         NUMERO_ARC: NUMERO DEL ARCHIVO DE AUTORES                         *
 *                                                                           *
 *         OB: INDICA QUE NUMERO DE OBSERVACION ESTA EN MEMORIA              *
 *             1: PRIMERA OBSERVACION                                        *
 *             2: SEGUNDA OBSERVACION                                        *
 *                                                                           *
 *         ACTIVIDAD_AUX: CUALQUIERA ACTIVIDAD ADMINISTRATIVA                *
 *                                                                           *
 *     LLAMADA POR:                                                          *
 *                                                                           *
 *         SELECCION                                                         *
 *                                                                           *
 *     LLAMA A:                                                              *
 *                                                                           *
 *         MUEVE_REIN                                                        *
 *         RECONSTRUYE                                                       *
 *         BUSCA                                                             *
 *         GOTOXY                                                            *
 *         CONSULTA_OBSERVACION                                              *
 *                                                                           *
 *     ARCHIVOS:                                                             *
```

```
*                                                                    *
*         CONTROL_ARCHIVOS: CONTIENE INFORMACION DE LOS              *
*                            PRINCIPALES ARCHIVOS DEL SISTEMA        *
*                                                                    *
*         MACRO ALGORITMO DEL PROGRAMA:                              *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
 *********************************************************************/

DCL
(PROF_AUX,FACT_AUX,PAG_AUX,RECI_AUX,CONTADOR_TIT,
 CONT_AU_PROF,CONT_AU_FACT,CONT_AU_PAG,CONT_AU_PED,
 CONT_AU_RECI,CONT_AU_CHEQ,CHEQ_AUX,TITULO_AUX,
 CONT_AU_PED_ACT,PED_AUX,KK,TT,CONT_AU_TIT,JJ,JJ1,
  CONT_AU_PAG_ACT,OB,CONTADOR_REIN)     BIN FIXED EXT STATIC,
(CONTADOR_PR,CONTADOR_PP,CONTADOR_AU)   BIN FIXED EXT STATIC,
(CLAVE1,AUTOR1,PP1)                     CHAR(70) VAR EXT STATIC,
(J,REPII_AUX,CONT_AUX,K,LONGITUD,I,L)                  BIN FIXED,
(DIRE_PANTA,DIRE_COR,IV,IG,FV,FG,CONT_AU_PED_ACT_1)    BIN FIXED,
01 BLQ_PED,
   02 CADENA_PED                                       CHAR(1000) VAR,
CADENA_LEIDA                                           CHAR(1000) VAR,
TESAURO                                                FILE VARIABLE,
(PROFORMA1,FACTURA1,PEDIDO1,CHEQUE1,PAGO1)     CHAR(12) VAR EXT STATIC,
RECIBIR1                                       CHAR(12) VAR EXT STATIC,
NUM_AU                                                 BIN FIXED,
AP2                                                    DEC FIXED,
(MONTO,FLETE)                                          CHAR(10) VAR,
EJEM_PEDIDO                                            CHAR(4) VAR,
(NUMERO_ARC,NUMERO_INDICE)                             BIN FIXED,
(POS2,LON1)                                            BIN FIXED,
(NEJEMP_LE,ANO_SUB_LE,ISBN_LE)                         CHAR(12) VAR,
(POS,LON,CONT_AU,
 X,Y,X1,Y1,X2,Y2,X3,Y3,X4,Y4)                          BIN FIXED,
```

```
(CLAVE2,AUTOR,CLAVE,RES_CM,CLAVE2_AUX,PROVEEDOR1)      CHAR(120) VAR,
DIR                                                    DEC FIXED,
(OUT,IN,CONTROL_ARCHIVO,INDICE,
ARCH_DATOS)                                            FILE,
01 BLQ_MONO_PP,
   02 CADENA_LEIDA2                                    CHAR(1000) VAR,
01 BLQ_PROV,
   02 CADENA_LEIDA1                                    CHAR(1000) VAR,
01 BLQ_CHEQ,
   02 CADENA_CHEQ                                      CHAR(1000) VAR,
01 BLQ_PAG,
   02 CADENA_PAG                                       CHAR(1000) VAR,
01 BLQ_RECI,
   02 CADENA_RECI                                      CHAR(1000) VAR,
01 BLQ_PROF,
   02 CADENA_PROF                                      CHAR(1000) VAR,
01 BLQ_FACT,
   02 CADENA_FACT                                      CHAR(1000) VAR,
BA_ENC                                                 BIT(1),
(POS_ABS,PF_CM,LON_CM,PF_CB,ULT_N,POS_RE,VIA(5))       BIN FIXED,
01 INF,
   02 APU_RAIZ                                         BIN FIXED,
   02 NO_NIV                                           BIN FIXED,
   02 NO_NODO                                          BIN FIXED,
   02 APU_PROX_VACIO                                   BIN FIXED,
01 NODO,
   02 CAB_NODO,
      03 NUM_N                                         BIN FIXED,
      03 NIV                                           BIN FIXED,
      03 NC                                            BIN FIXED,
      03 TN                                            BIN FIXED,
      03 VECINA_I                                      BIN FIXED,
      03 VECINA_D                                      BIN FIXED,
   02 ENTRADA                                          CHAR(1009) VAR,
01 CONTROL_ARCH(16),
   02 TIPO                                             CHAR(1),
   02 N_ARC                                            BIN FIXED,
   02 LON_REG                                          BIN FIXED,
   02 ULT_VACIO                                        BIN FIXED,
   02 MAX_DIR                                          BIN FIXED,
   02 NOMBRE_ARC                                       CHAR(20) VAR,
CK                                                     CHAR(1);
DCL
MUEVE_REIN ENTRY(BIN FIXED),
RECONSTRUYE ENTRY(BIN FIXED,BIN FIXED,BIN FIXED,BIN FIXED,BIN FIXED,
BIN FIXED,CHAR(1000) VAR),
BUSCA ENTRY(FILE VARIABLE,CHAR(120) VAR,BIT(1),BIN FIXED,CHAR(120) VAR,
BIN FIXED,BIN FIXED,1,2,3 BIN FIXED,3 BIN FIXED,3 BIN FIXED,
3 BIN FIXED,3 BIN FIXED,3 BIN FIXED,2 CHAR(1009) VAR,BIN FIXED,
(5) BIN FIXED,BIN FIXED,BIN FIXED,1,2 BIN FIXED,2 BIN FIXED,2 BIN FIXED,
2 BIN FIXED),
GOTOXY ENTRY(BIN FIXED,BIN FIXED),
CONSULTA_OBSERVACION ENTRY(BIN FIXED,CHAR(1000) VAR);

TESAURO = INDICE;
CK = ASCII(11);
READ FILE(CONTROL_ARCHIVO) INTO(CONTROL_ARCH) KEY(1);

IG = 1;
FG = 9;
DIRE_PANTA = 16;
IV = 1;
FV = 100;
```

```
DIRE_COR = 28;

IF J = 1 THEN DO;
   IF (CONT_AU_PROF=0 & CONT_AU_FACT=0 & CONT_AU_PAG=0 & CONT_AU_RECT=0 & CONT_A
U_CHEQ=0 & CONT_AU_PED=0 &
CONTADOR_PR=0 & CONTADOR_PP=0 & CONTADOR_AU=0 & CONTADOR_REIN = 0 &
CONTADOR_TIT = 0) THEN
        CALL MENSAJE;
   IF CONTADOR_REIN ^= 0 THEN DO;
      CALL MUEVE_REIN(J);
      GOTO FIN;
   END;
   IF CONTADOR_PR ^= 0 THEN DO;
      NUMERO_INDICE = 12;
      NUMERO_ARC = 2;
      CLAVE2 = CLAVE1;
      CALL BUSCA_CADENA;
      GOTO FIN;
   END;
   IF (CONTADOR_PP ^= 0 ! CONTADOR_AU ^= 0) THEN DO;
      NUMERO_INDICE = 11;
      NUMERO_ARC = 1;
      IF CONTADOR_PP ^= 0 THEN
         CLAVE2 = PP1;
      ELSE
         CLAVE2 = AUTOR1;
      CALL BUSCA_CADENA;
      GOTO FIN;
   END;
   IF CONTADOR_TIT ^= 0 THEN DO;
      TITULO_AUX = TITULO_AUX + 1;
      IF TITULO_AUX < CONT_AU_TIT THEN DO;
         CONT_AUX = CONT_AU_TIT - TITULO_AUX + 1;
         NUMERO_INDICE = 12;
         CLAVE2 = CLAVE1;
         JJ = JJ - 1;
         CALL MUESTRA_AUTOR;
         GOTO FIN;
      END;
      ELSE DO;
         TITULO_AUX = TITULO_AUX - 1;
         CALL MENSAJE;
      END;
   END;
   IF CONT_AU_PROF ^= 0 THEN DO;
      IF OB = 0 THEN DO;
         PROF_AUX = PROF_AUX + 1;
         IF PROF_AUX < CONT_AU_PROF THEN DO;
            CONT_AUX = CONT_AU_PROF - PROF_AUX + 1;
            J = 42;
            NUMERO_INDICE = 13;
            CLAVE2 = PROFORMA1;
            REPII_AUX = PROF_AUX;
            CALL MUESTRA_AUTOR;
         END;
         ELSE DO;
            PROF_AUX = PROF_AUX - 1;
            NUMERO_INDICE = 13;
            NUMERO_ARC = 3;
            CLAVE2 = PROFORMA1;
            CALL BUSCA_CADENA;
            GOTO FIN;
         END;
      END;
```

```
      ELSE DO;
         NUMERO_INDICE = 13;
         NUMERO_ARC = 3;
         CLAVE2 = PROFORMA1;
         CALL BUSCA_CADENA;
      END;
   END;.
   IF CONT_AU_PAG ^= 0 THEN DO;
      IF OB = 0 THEN DO;
         PAG_AUX = PAG_AUX + 1;
         IF PAG_AUX < CONT_AU_PAG THEN DO;
            CONT_AUX = CONT_AU_PAG - PAG_AUX + 1;
            J = 23;
            NUMERO_INDICE = 17;
            CLAVE2 = PAG01;
            REPII_AUX = PAG_AUX;
            CALL MUESTRA_AUTOR;
         END;
         ELSE DO;
            PAG_AUX = PAG_AUX - 1;
            NUMERO_INDICE = 17;
            NUMERO_ARC = 7;
            CLAVE2 = PAG01;
            CALL BUSCA_CADENA;
            GOTO FIN;
         END;
      END;
   ELSE DO;
       NUMERO_INDICE = 17;
       NUMERO_ARC = 7;
       CLAVE2 = PAG01;
       CALL BUSCA_CADENA;
   END;
END;
   IF CONT_AU_FACT ^= 0 THEN DO;
      IF OB = 0 THEN DO;
         FACT_AUX = FACT_AUX + 1;
         IF FACT_AUX < CONT_AU_FACT THEN DO;
            CONT_AUX = CONT_AU_FACT - FACT_AUX + 1;
            J = 4;
            NUMERO_INDICE = 14;
            CLAVE2 = FACTURA1;
            REPII_AUX = FACT_AUX;
            CALL MUESTRA_AUTOR;
         END;
         ELSE DO;
            FACT_AUX = FACT_AUX - 1;
            NUMERO_INDICE = 14;
            NUMERO_ARC = 4;
            CLAVE2 = FACTURA1;
            CALL BUSCA_CADENA;
            GOTO FIN;
         END;
      END;
      ELSE DO;
           NUMERO_INDICE = 14;
           NUMERO_ARC = 4;
           CLAVE2 = FACTURA1;
           CALL BUSCA_CADENA;
      END;
   END;
   IF CONT_AU_RECI ^= 0 THEN DO;
      IF OB = 0 THEN DO;
```

```
        RECI_AUX = RECI_AUX + 1;
        IF RECI_AUX < CONT_AU_RECI THEN DO;
           CONT_AUX = CONT_AU_RECI - RECI_AUX + 1;
           J = 38;
           NUMERO_INDICE = 17;
           CLAVE2 = RECIBIR1;
           REPII_AUX = RECI_AUX;
           CALL MUESTRA_AUTOR;
        END;
        ELSE DO;
           RECI_AUX = RECI_AUX - 1;
           NUMERO_INDICE = 18;
           NUMERO_ARC = 8;
           CLAVE2 = RECIBIR1;
           CALL BUSCA_CADENA;
           GOTO FIN;
        END;
     END;
     ELSE DO;
         NUMERO_INDICE = 18;
         NUMERO_ARC = 8;
         CLAVE2 = RECIBIR1;
         CALL BUSCA_CADENA;
     END;
END;
IF CONT_AU_CHEQ ^= 0 THEN DO;
   IF OB = 0 THEN DO;
        CHEQ_AUX = CHEQ_AUX + 1;
        IF CHEQ_AUX < CONT_AU_CHEQ THEN DO;
           CONT_AUX = CONT_AU_CHEQ - CHEQ_AUX + 1;
           REPII_AUX = CHEQ_AUX;
           CALL MUESTRA_CHEQUE;
        END;
        ELSE DO;
           CHEQ_AUX = CHEQ_AUX - 1;
           NUMERO_INDICE = 16;
           NUMERO_ARC = 6;
           CLAVE2 = CHEQUE1;
           CALL BUSCA_CADENA;
           GOTO FIN;
        END;
  END;
        ELSE DO;
           NUMERO_INDICE = 16;
           NUMERO_ARC = 6;
           CLAVE2 = CHEQUE1;
           CALL BUSCA_CADENA;
        END;
   END;
IF CONT_AU_PED ^= 0 THEN DO;
   IF OB = 0 THEN DO;
        PED_AUX = PED_AUX + 1;
        IF PED_AUX < CONT_AU_PED_ACT THEN DO;
           CONT_AUX = CONT_AU_PED_ACT - PED_AUX + 1;
           J = 24;
           NUMERO_INDICE = 15;
           CLAVE2 = PEDIDO1;
           TT = TT + 1;
           REPII_AUX = TT;
           CALL MUESTRA_AUTOR;
        END;
        ELSE DO;
           IF CONT_AU_PED_ACT < CONT_AU_PED THEN DO;
```

```
            NUMERO_INDICE = 15;
            NUMERO_ARC = 5;
            CALL ABRE_INDICE;
            I = 1;
            DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_ARC);
               I = I + 1;
            END;
            LONGITUD = CONTROL_ARCH(I).LON_REG;
            CLOSE FILE(ARCH_DATOS);
          OPEN FILE(ARCH_DATOS) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) INPUT
            DIRECT KEYED ENV(RECSIZE(LONGITUD));
            CLAVE2 = PEDIDO1;
          CALL BUSCA(TESAURO,CLAVE2,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,
            NODO,PF_CB,VIA,ULT_N,POS_RE,INF);
            DIR = DEC(RANK(SUBSTR(ENTRADA,POS_ABS+2,1)))*10000 +
            RANK(SUBSTR(ENTRADA,POS_ABS+3,1))*100 +
            RANK(SUBSTR(ENTRADA,POS_ABS+4,1));
            READ FILE(ARCH_DATOS) INTO(BLQ_PED) KEY(DIR);
            CONT_AU_PED_ACT_1 = CONT_AU_PED_ACT;
            KK = KK + 1;
            CALL MUESTRA_PROVEEDOR;
 CALL RECONSTRUYE(DIRE_PANTA,DIRE_COR,IG,FG,IV,FV,BLQ_PED.CADENA_PED);
            CALL GOTOXY(12,0);
            PUT FILE(OUT) EDIT(PEDIDO1)(A);
            IF CONT_AU_PED_ACT_1 = 1 THEN
                TT = TT + 1;
            ELSE
               TT = TT + 2;
            PED_AUX = 1;
            J = 24;
            CONT_AUX = CONT_AU_PED_ACT;
            REPIT_AUX = TT;
            CALL MUESTRA_AUTOR;
         END;
         ELSE DO;
            PED_AUX = PED_AUX - 1;
            NUMERO_INDICE = 15;
            NUMERO_ARC = 5;
            CLAVE2 = PEDIDO1;
            CALL BUSCA_CADENA;
            GOTO FIN;
         END;
      END;
   END;
   ELSE DO;
     NUMERO_INDICE = 15;
     NUMERO_ARC = 5;
     CLAVE2 = PEDIDO1;
     CALL BUSCA_CADENA;
   END;
END;
END;
IF J = 2 THEN DO;
   IF (CONT_AU_PROF=0 & CONT_AU_FACT=0 & CONT_AU_PAG=0 & CONT_AU_RECI=0 & CONT_A
U_CHEQ=0 & CONT_AU_PED=0 &
CONTADOR_PR=0 & CONTADOR_PP=0 & CONTADOR_AU=0 & CONTADOR_REIN=0 &
CONTADOR_TIT = 0) THEN
        CALL MENSAJE;
   IF CONTADOR_REIN ^= 0 THEN DO;
      CALL MUEVE_REIN(J);
      GOTO FIN;
   END;
   IF CONTADOR_TIT ^= 0 THEN DO;
      TITULO_AUX = TITULO_AUX - 1;
```

```
    IF TITULO_AUX = 0 THEN DO;
       TITULO_AUX = TITULO_AUX + 1;
       CALL MENSAJE;
    END;
    NUMERO_INDICE = 12;
    CLAVE2 = CLAVE1;
    CONT_AUX = 2;
    JJ = JJ1;
    CALL ABRE_INDICE;
    BA_ENC = "0"B;
    DO WHILE(BA_ENC = "0"B);
       JJ = JJ - 1;
       CLAVE = CLAVE2 !! CHAR(JJ);
       CALL BUSCA(TESAURO,CLAVE,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,
       NODO,PF_CB,VIA,ULT_N,POS_RE,INF);
    END;
    JJ = JJ - 1;
    CALL MUESTRA_AUTOR;
    GOTO FIN;
 END;
 IF CONT_AU_PED ^= 0 THEN DO;
    IF OB >= 2 THEN DO;
       NUMERO_INDICE = 15;
       NUMERO_ARC = 5;
       CLAVE2 = PEDIDO1;
       CALL BUSCA_CADENA;
    END;
    ELSE DO;
       IF OB = 1 THEN DO;
          OB = 0;
          GOTO EE;
       END;
       PED_AUX = PED_AUX - 1;
       IF PED_AUX = 0 THEN DO;
          IF TT = 1 THEN DO;
             PED_AUX = PED_AUX + 1;
             CALL MENSAJE;
          END;
          KK = KK - 1;
          IF KK = 0 THEN DO;
             KK = KK + 1;
             PED_AUX = PED_AUX + 1;
             CALL MENSAJE;
          END;
          NUMERO_INDICE = 15;
          NUMERO_ARC = 5;
          CALL ABRE_INDICE;
          I = 1;
          DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_ARC);
             I = I + 1;
          END;
          LONGITUD = CONTROL_ARCH(I).LON_REG;
          CLOSE FILE(ARCH_DATOS);
       OPEN FILE(ARCH_DATOS) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) INPUT
          DIRECT KEYED ENV(RECSIZE(LONGITUD));
          CLAVE2 = PEDIDO1;
       CALL BUSCA(TESAURO,CLAVE2,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,
          NODO,PF_CB,VIA,ULT_N,POS_RE,INF);
          DIR = DEC(RANK(SUBSTR(ENTRADA,POS_ABS+2,1)))*10000 +
          RANK(SUBSTR(ENTRADA,POS_ABS+3,1))*100 +
          RANK(SUBSTR(ENTRADA,POS_ABS+4,1));
          READ FILE(ARCH_DATOS) INTO(BLQ_PED) KEY(DIR);
          CALL MUESTRA_PROVEEDOR;
```

```
CALL RECONSTRUYE(DIRE_PANTA,DIRE_COR,IG,FG,IV,FV,BLQ_PED,CADENA_PED);
            CALL GOTOXY(12,0);
            PUT FILE(OUT) EDIT(PEDIDO1)(A);
            IF CONT_AU_PED_ACT = 1 THEN DO;
               PED_AUX = 1;
               CONT_AUX = 1;
               TT = TT - 1;
            END;
            ELSE DO;
               TT = TT - 2;
               PED_AUX = CONT_AU_PED_ACT - 1;
               CONT_AUX = 2;
            END;
            REPII_AUX = TT;
            J = 24;
            CALL MUESTRA_AUTOR;
         END;
         ELSE DO;
           TT = TT - 1;
EF:      IF CONT_AU_PED_ACT = 1 THEN
            CONT_AUX = 1;
         ELSE
           CONT_AUX = 2;
           REPII_AUX = TT;
           J = 24;
           NUMERO_INDICE = 15;
           CLAVE2 = PEDIDO1;
           CALL MUESTRA_AUTOR;
        END;
     END;
   END;
   IF CONT_AU_PAG ^= 0 THEN DO;
      IF OB >= 2 THEN DO;
         NUMERO_INDICE = 17;
         NUMERO_ARC = 7;
         CLAVE2 = PAGO1;
         CALL BUSCA_CADENA;
      END;
      ELSE DO;
         IF OB = 1 THEN DO;
            OB = 0;
            J = 23;
            GOTO MM;
         END;
         J = 23;
         PAG_AUX = PAG_AUX - 1;
         IF PAG_AUX = 0 THEN DO;
               PAG_AUX = PAG_AUX + 1;
               CALL MENSAJE;
         END;
      MM:      NUMERO_INDICE = 17;
               J = 23;
               CLAVE2 = PAGO1;
               IF CONT_AU_PAG = 1 THEN
                  CONT_AUX = 1;
               ELSE
                  CONT_AUX = 2;
               REPII_AUX = PAG_AUX;
               CALL MUESTRA_AUTOR;
          END;
      END;
   IF CONT_AU_PROF ^= 0 THEN DO;
      IF OB >= 2 THEN DO;
```

```
      NUMERO_INDICE = 13;
      NUMERO_ARC = 3;
      CLAVE2 = PROFORMA1;
      CALL BUSCA_CADENA;
   END;
   ELSE DO;
      IF OB = 1 THEN DO;
         OB = 0;
         GOTO II;
      END;
      PROF_AUX = PROF_AUX - 1;
      IF PROF_AUX = 0 THEN DO;
         PROF_AUX = PROF_AUX + 1;
         CALL MENSAJE;
      END;
II:   J = 42;
      NUMERO_INDICE = 13;
      CLAVE2 = PROFORMA1;
```

```
        IF CONT_AU_PROF = 1 THEN
           CONT_AUX = 1;
        ELSE
           CONT_AUX = 2;
        REPII_AUX = PROF_AUX;
        CALL MUESTRA_AUTOR;
    END;
END;
IF CONT_AU_FACT ^= 0 THEN DO;
    IF OB >= 2 THEN DO;
        NUMERO_INDICE = 14;
        NUMERO_ARC = 4;
        CLAVE2 = FACTURA1;
        CALL BUSCA_CADENA;
    END;
    ELSE DO;
        IF OB = 1 THEN DO;
           OB = 0;
           GOTO LL;
        END;
        FACT_AUX = FACT_AUX - 1;
        IF FACT_AUX = 0 THEN DO;
           FACT_AUX = FACT_AUX + 1;
           CALL MENSAJE;
        END;
LL:     J = 4;
        NUMERO_INDICE = 14;
        CLAVE2 = FACTURA1;
        IF CONT_AU_FACT = 1 THEN
           CONT_AUX = 1;
        ELSE
           CONT_AUX = 2;
        REPII_AUX = FACT_AUX;
        CALL MUESTRA_AUTOR;
    END;
END;
IF CONT_AU_RECI ^= 0 THEN DO;
    IF OB >= 2 THEN DO;
        NUMERO_INDICE = 18;
        CLAVE2 = RECIBIR1;
        NUMERO_ARC = 8;
        CALL BUSCA_CADENA;
    END;
    ELSE DO;
        IF OB = 1 THEN DO;
           OB = 0;
           GOTO PP;
        END;
        RECI_AUX = RECI_AUX - 1;
        IF RECI_AUX = 0 THEN DO;
           RECI_AUX = RECI_AUX + 1;
           CALL MENSAJE;
        END;
PP:     J = 38;
        NUMERO_INDICE = 18;
        CLAVE2 = RECIBIR1;
        IF CONT_AU_RECI = 1 THEN
           CONT_AUX = 1;
        ELSE
           CONT_AUX = 2;
        REPII_AUX = RECI_AUX;
        CALL MUESTRA_AUTOR;
    END;
```

```
    END;
    IF CONT_AU_CHEQ ^= 0 THEN DO;
        IF OB >= 2 THEN DO;
            NUMERO_INDICE = 16;
            NUMERO_ARC = 6;
            CLAVE2 = CHEQUE1;
            CALL BUSCA_CADENA;
        END;
        ELSE DO;
            IF OB = 1 THEN DO;
                OB = 0;
                GOTO WW;
            END;
        CHEQ_AUX = CHEQ_AUX - 1;
        IF CHEQ_AUX = 0 THEN DO;
            CHEQ_AUX = CHEQ_AUX + 1;
            CALL MENSAJE;
        END;
 WW:    REPII_AUX = CHEQ_AUX;
        IF CONT_AU_CHEQ = 1 THEN
            CONT_AUX = 1;
        ELSE
            CONT_AUX = 2;
        CALL MUESTRA_CHEQUE;
    END;
END;
IF CONTADOR_PR ^= 0 THEN DO;
    IF OB >= 2 THEN DO;
        NUMERO_INDICE = 12;
        NUMERO_ARC = 2;
        CLAVE2 = CLAVE1;
        CALL BUSCA_CADENA;
    END;
    ELSE DO;
        IF OB = 0 THEN DO;
            CALL GOTOXY(0,22);
            PUT FILE(OUT) EDIT("NO HAY OBSERVACIONES",CK)(A,A);
        END;
        IF OB = 1 THEN DO;
            CALL GOTOXY(0,22);
            PUT FILE(OUT) EDIT("NO HAY MAS OBSERVACIONES",CK)(A,A);
            OB = 0;
        END;
  END;
END;
IF (CONTADOR_PP ^= 0 | CONTADOR_AU ^= 0) THEN DO;
    IF OB >= 2 THEN DO;
        NUMERO_INDICE = 11;
        NUMERO_ARC = 1;
        IF CONTADOR_PP ^= 0 THEN
            CLAVE2 = PP1;
        ELSE
            CLAVE2 = AUTOR1;
        CALL BUSCA_CADENA;
    END;
    ELSE DO;
        IF OB = 1 THEN
            OB = 0;
    END;
END;
END;

/*          RUTINAS        */
```

```
MENSAJE: PROCEDURE;

  CALL GOTOXY(0,22);
  PUT FILE(OUT) EDIT("NO HAY MAS/MENOS AUTOR/PP")(A);
  GOTO FIN;

END;   /*  MENSAJE    */

MUESTRA_AUTOR: PROCEDURE;

  CALL ABRE_INDICE;
  NUMERO_ARC = 1;
  I = 1;
  DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_ARC);
     I = I + 1;
  END;
  LONGITUD = CONTROL_ARCH(I).LON_REG;
  CLOSE FILE(ARCH_DATOS);
  OPEN FILE(ARCH_DATOS) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) INPUT
  DIRECT KEYED ENV(RECSIZE(LONGITUD));
  K = 0;
  X = 14;
  Y = 7;
  X1 = 15;
  Y1 = 9;
  X2 = 12;
  Y2 = 11;
  X3 = 34;
  Y3 = 11;
  X4 = 54;
  Y4 = 11;
REPI_AU:
  IF CONTADOR_TIT ^= 0 THEN DO;
     BA_ENC = "0"B;
     DO WHILE(BA_ENC = "0"B);
        JJ = JJ + 1;
        CLAVE = CLAVE2 !! CHAR(JJ);
        CALL BUSCA(TESAURO,CLAVE,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,
        NODO,PF_CB,VIA,ULT_N,POS_RE,INF);
     END;
 END;
 ELSE DO;
  CLAVE = CLAVE2 !! CHAR(K + REPIT_AUX);
  CALL BUSCA(TESAURO,CLAVE,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,
  NODO,PF_CB,VIA,ULT_N,POS_RE,INF);
END;
  NUM_AU = RANK(SUBSTR(ENTRADA,POS_ABS+5,1))*256 +
  RANK(SUBSTR(ENTRADA,POS_ABS+6,1));
  AP2 = DEC(RANK(SUBSTR(ENTRADA,POS_ABS+7,1)))*10000 +
  RANK(SUBSTR(ENTRADA,POS_ABS+8,1))*100 +
  RANK(SUBSTR(ENTRADA,POS_ABS+9,1));
  NUMERO_INDICE = 11;
  CALL ABRE_INDICE;
  CLAVE = CHAR(NUM_AU);
  CALL BUSCA(TESAURO,CLAVE,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,
  NODO,PF_CB,VIA,ULT_N,POS_RE,INF);
  DIR = DEC(RANK(SUBSTR(ENTRADA,POS_ABS+2,1)))*10000 +
  RANK(SUBSTR(ENTRADA,POS_ABS+3,1))*100 +
  RANK(SUBSTR(ENTRADA,POS_ABS+4,1));
  READ FILE(ARCH_DATOS) INTO(BLQ_MONO_PP) KEY(DIR);
  POS = 1;
  I = 1;
```

```
   REPI;
   LON = RANK(SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS,1));
   CLAVE = SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS+1,LON-1);
   POS = POS + LON;
   LON = RANK(SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS,1)) - 1;
   IF LON ^= 0 THEN
      CLAVE = CLAVE!!SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS+1,LON);
   IF AP2 = 0 THEN DO;
      CALL GOTOXY(X,Y);
      PUT FILE(OUT) EDIT(CK)(A);
      GOTO XX2;
   END;
   IF I = 1 THEN DO;
      I = I + 1;
      POS = POS + LON + 1;
      AUTOR = CLAVE;
      GOTO REPI;
   END;
   CALL GOTOXY(X,Y);
   PUT FILE(OUT) EDIT(ASCII(29),AUTOR,ASCII(28),CK)(A,A,A,A);
   XX2:CALL GOTOXY(X1,Y1);
   PUT FILE(OUT) EDIT(ASCII(29),CLAVE,ASCII(28),CK)(A,A,A,A);
   IF CONTADOR_TIT ^= 0 THEN GOTO YA;
   CALL BUSCA_POSICION;
   POS2 = POS2 + LON1;
   LON1 = RANK(SUBSTR(CADENA_LEIDA2,POS2,1));
   IF J = 42 THEN DO;
      NEJEMP_LE = SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS2+1,LON1-1);
      CALL GOTOXY(X2,Y2);
      PUT FILE(OUT) EDIT(NEJEMP_LE)(A(10));
      POS2 = POS2 + LON1;
      LON1 = RANK(SUBSTR(CADENA_LEIDA2,POS2,1));
      IF AP2 = 1 THEN
         GOTO XX3;
      ANO_SUB_LE = SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS2+1,LON1-1);
      CALL GOTOXY(X3,Y3);
      PUT FILE(OUT) EDIT(ANO_SUB_LE)(A(8));
      POS2 = POS2 + LON1;
      LON1 = RANK(SUBSTR(CADENA_LEIDA2,POS2,1));
      XX3: ISBN_LE = SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS2+1,LON1-1);
      CALL GOTOXY(X4,Y4);
      PUT FILE(OUT) EDIT(ISBN_LE)(A(20));
   END;
   IF J = 4 THEN DO;
      CALL GOTOXY(X2,Y2);
PUT FILE(OUT) EDIT(ASCII(29),SUBSTR(CADENA_LEIDA2,POS2+1,LON1-1),ASCII(28))(A,A,
A);
      POS2 = POS2 + LON1;
      LON1 = RANK(SUBSTR(CADENA_LEIDA2,POS2,1));
      FLETE = SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS2+1,LON1-1);
      CALL GOTOXY(X4,Y4);
PUT FILE(OUT) EDIT(ASCII(29),SUBSTR(CADENA_LEIDA2,POS2+1,LON1-1),ASCII(28))(A,A,
A);
   END;
   IF (J = 23 ! J = 38 ! J = 24)THEN DO;
       EJEM_PEDIDO = SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS2+1,LON1-1);
       CALL GOTOXY(X2,Y2);
       PUT FILE(OUT) EDIT(EJEM_PEDIDO)(A(4));
   END;
   IF CONT_AUX = 1 THEN DO;
      CALL GOTOXY(14,13);
      PUT FILE(OUT) EDIT(CK)(A);
      CALL GOTOXY(15,15);
      PUT FILE(OUT) EDIT(CK)(A);
```

```
YA:  IF (CONT_AUX ^= 1 & K = 0) THEN DO;
            K = K + 1;
            Y = 13;
            Y1 = 15;
            Y2 = 17;
            Y3 = 17;
            Y4 = 17;
            JJ1 = JJ;
            CLOSE FILE(INDICE);
            IF CONTADOR_TIT ^= 0 THEN
               NUMERO_INDICE = 12;
            IF J = 38 THEN
               NUMERO_INDICE = 18;
            IF J = 42 THEN
               NUMERO_INDICE = 13;
            IF J = 4 THEN
               NUMERO_INDICE = 14;
            IF J = 23 THEN
               NUMERO_INDICE = 17;
            IF J = 24 THEN
               NUMERO_INDICE = 15;
            CALL ABRE_INDICE;
            GOTO REPI_AU;
         END;

END;  /*  MUESTRA_AUTOR   */

BUSCA_POSICION: PROCEDURE;

IF J = 42 THEN
   CLAVE2_AUX = "PF"!!CLAVE2;
IF J = 4 THEN
   CLAVE2_AUX = "FA"!!CLAVE2;
IF J = 24 THEN
   CLAVE2_AUX = "PD"!!CLAVE2;
IF J = 23 THEN
   CLAVE2_AUX = "PA"!!CLAVE2;
IF J = 38 THEN
   CLAVE2_AUX = "RE"!!CLAVE2;
POS2 = 1;
LON1 = RANK(SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS2,1));
DO WHILE(CLAVE2_AUX ^= SUBSTR(CADENA_LEIDA2,POS2+1,LON1-1));
   POS2 = POS2 + LON1;
   LON1 = RANK(SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS2,1));
END;
END; /*      BUSCA_POSICION       */

ABRE_INDICE: PROCEDURE;

   CLOSE FILE(INDICE);
   I = 1;
   DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_INDICE);
      I = I + 1;
   END;
   OPEN FILE(INDICE) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) INPUT
   DIRECT KEYED ENV(RECSIZE(1024));
END;    /*  ABRE_INDICE;   */

MUESTRA_CHEQUE: PROCEDURE;

   X = 16;
   Y = 2;
   X1 = 12;
```

```
   Y1 = 4;
   X2 = 35;
   Y2 = 6;
   X3 = 54;
NUMERO_INDICE = 16;
CALL ABRE_INDICE;
NUMERO_ARC = 6;
I = 1;
DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_ARC);
   I = I + 1;
END;
LONGITUD = CONTROL_ARCH(I).LON_REG;
CLOSE FILE(ARCH_DATOS);
OPEN FILE(ARCH_DATOS) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) INPUT
DIRECT KEYED ENV(RECSIZE(LONGITUD));
CLAVE2 = CHEQUE1;
CALL BUSCA(TESAURO,CLAVE2,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,
NODO,PF_CB,VIA,ULT_N,POS_RE,INF);
DIR = DEC(RANK(SUBSTR(ENTRADA,POS_ABS+2,1)))*10000 +
RANK(SUBSTR(ENTRADA,POS_ABS+3,1))*100 +
RANK(SUBSTR(ENTRADA,POS_ABS+4,1));
READ FILE(ARCH_DATOS) INTO(BLQ_CHEQ) KEY(DIR);
L = 1;
POS = 1;
LON = RANK(SUBSTR(CADENA_CHEQ,POS,1));
DO I = 1 TO 4;
   POS = POS + LON;
   LON = RANK(SUBSTR(CADENA_CHEQ,POS,1));
END;
K = 1;
DO WHILE(K < REPII_AUX);
   DO I = 1 TO 4;
      POS = POS + LON;
      LON = RANK(SUBSTR(CADENA_CHEQ,POS,1));
   END;
   K = K + 1;
END;
OTRA_VEZ:
   CLAVE2 = SUBSTR(CADENA_CHEQ,POS+1,LON-1);
   POS = POS + LON;
   LON = RANK(SUBSTR(CADENA_CHEQ,POS,1));
   AUTOR = SUBSTR(CADENA_CHEQ,POS+1,LON-1);
   POS = POS + LON;
   LON = RANK(SUBSTR(CADENA_CHEQ,POS,1));
   MONTO = SUBSTR(CADENA_CHEQ,POS+1,LON-1);
   POS = POS + LON;
   LON = RANK(SUBSTR(CADENA_CHEQ,POS,1));
   FLETE = SUBSTR(CADENA_CHEQ,POS+1,LON-1);
   CALL GOTOXY(X,Y);
   PUT FILE(OUT) EDIT(CLAVE2,CK)(A,A);
   CALL GOTOXY(X1,Y1);
   PUT FILE(OUT) EDIT(AUTOR,CK)(A,A);
   CALL GOTOXY(X2,Y2);
   PUT FILE(OUT) EDIT(MONTO)(F(10,2));
   CALL GOTOXY(X3,Y2);
   PUT FILE(OUT) EDIT(FLETE)(F(10,2));
   IF (L = 1 & CONT_AUX ^= 1) THEN DO;
      L = L + 1;
      Y = 9;
      Y1 = 11;
      Y2 = 13;
      POS = POS + LON;
      LON = RANK(SUBSTR(CADENA_CHEQ,POS,1));
```

```
      GOTO OTRA_VEZ;
   END;

END; /*     MUESTRA_CHEQUE    */

MUESTRA_PROVEEDOR: PROCEDURE;

K = 1;
POS = 1;
LON = RANK(SUBSTR(CADENA_PED,POS,1));
POS = POS + LON;
LON = RANK(SUBSTR(CADENA_PED,POS,1));
BA_ENC = "0"B;
DO WHILE(BA_ENC = "0"B);
   IG = POS;
   DO WHILE(LON ^= 253 & LON ^= 255);
      POS = POS + LON;
      LON = RANK(SUBSTR(CADENA_PED,POS,1));
   END;
   IF LON = 255 THEN DO;
      KK = KK - 1;
      PED_AUX = PED_AUX - 1;
      NUMERO_INDICE = 15;
      NUMERO_ARC = 5;
      CLAVE2 = PEDID01;
      CALL BUSCA_CADENA;
      GOTO FIN;
   END;
   IF LON = 253 THEN DO;
      POS = POS + 1;
      LON = RANK(SUBSTR(CADENA_PED,POS,1));
      CONT_AU_PED_ACT = BINARY(SUBSTR(CADENA_PED,POS+1,LON-1));
      POS = POS + LON + 1;
      LON = RANK(SUBSTR(CADENA_PED,POS,1));
   END;
   IF K ^= KK THEN
      K = K + 1;
   ELSE
      BA_ENC = "1"B;
END;

END;     /*   MUESTRA_PROVEEDOR*/

BUSCA_CADENA: PROCEDURE;

   CALL ABRE_INDICE;
   I = 1;
   DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_ARC);
      I = I + 1;
   END;
   LONGITUD = CONTROL_ARCH(1).LON_REG;
   CLOSE FILE(ARCH_DATOS);
   OPEN FILE(ARCH_DATOS) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) INPUT
   DIRECT KEYED ENV(RECSIZE(LONGITUD));
   CALL BUSCA(TESAURO,CLAVE2,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,
   NODO,PF_CB,VIA,ULT_N,POS_RE,INF);
   DIR = DEC(RANK(SUBSTR(ENTRADA,POS_ABS+2,1)))*10000 +
   RANK(SUBSTR(ENTRADA,POS_ABS+3,1))*100 +
   RANK(SUBSTR(ENTRADA,POS_ABS+4,1));
   IF CONT_AU_PROF ^= 0 THEN DO;
      READ FILE(ARCH_DATOS) INTO(BLQ_PROF) KEY(DIR);
      CADENA_LEIDA = CADENA_PROF;
   END;
```

```
   IF CONT_AU_FACT ^= 0 THEN DO;
      READ FILE(ARCH_DATOS) INTO(BLQ_FACT) KEY(DIR);
      CADENA_LEIDA = CADENA_FACT;
   END;
   IF CONT_AU_RECI ^= 0 THEN DO;
      READ FILE(ARCH_DATOS) INTO(BLQ_RECI) KEY(DIR);
      CADENA_LEIDA = CADENA_RECI;
   END;
   IF CONT_AU_PED ^= 0 THEN DO;
      READ FILE(ARCH_DATOS) INTO(BLQ_PED) KEY(DIR);
      CADENA_LEIDA = CADENA_PED;
   END;
   IF CONT_AU_PAG ^= 0 THEN DO;
      READ FILE(ARCH_DATOS) INTO(BLQ_PAG) KEY(DIR);
      CADENA_LEIDA = CADENA_PAG;
   END;
   IF CONT_AU_CHEQ ^= 0 THEN DO;
      READ FILE(ARCH_DATOS) INTO(BLQ_CHEQ) KEY(DIR);
      CADENA_LEIDA = CADENA_CHEQ;
   END;
   IF CONTADOR_PR ^= 0 THEN DO;
      READ FILE(ARCH_DATOS) INTO(BLQ_PROV) KEY(DIR);
      CADENA_LEIDA = CADENA_LEIDA1;
   END;
   IF (CONTADOR_PP ^= 0 ! CONTADOR_AU ^= 0) THEN DO;
      READ FILE(ARCH_DATOS) INTO(BLQ_MONO_PP) KEY(DIR);
      CADENA_LEIDA = CADENA_LEIDA2;
   END;
   CALL CONSULTA_OBSERVACION(J,CADENA_LEIDA);
END;   /*          BUSCA_CADENA        */

FIN:

END;   /* MAS_MENOS_LINEAS   */
```

```
NUEVO_AUTOR: PROCEDURE(NNA,NUMERO_INDICE,NUMERO_ARC,NUM_AU1,DIR,AP2,T);

/*****************************************************************************
 *                                                                           *
 *  CREA UN AUTOR O PP CUANDO SE ESTA CREANDO UNA FACTURA                    *
 *                                                                           *
 *      PARAMETROS:                                                          *
 *                                                                           *
 *          NNA: INDICA SI LO QUE VA A SER CREADO ES UN                      *
 *               AUTOR O PP                                                  *
 *               0: PP                                                       *
 *               1: AUTOR                                                    *
 *                                                                           *
 *          NUMERO_INDICE: NUMERO DEL INDICE                                 *
 *                                                                           *
 *          NUMERO_ARC: NUMERO DEL ARCHIVO                                   *
 *                                                                           *
 *          NUM_AU1: NUMERO ASIGNADO AL AUTOR/PP                             *
 *                                                                           *
 *          AP2: INDICA SI ES UN AUTOR O PP                                  *
 *                                                                           *
 *      VARIABLES:                                                           *
 *                                                                           *
 *          DIRE_PANTA: DIRECCION DE LA PANTALLA DEL AUTOR/PP                *
 *                                                                           *
 *          DIRE_COR: DIRECCION DE LAS COORDENADAS                           *
 *                                                                           *
 *          IG: INICIO DE LAS COORDENADAS EN EL ARCHIVO DE                   *
 *              COORDENADAS                                                  *
 *                                                                           *
 *          FG: FIN DE LAS COORDENADAS                                       *
 *                                                                           *
 *      LLAMADA POR:                                                         *
 *                                                                           *
 *          MUESTRA_INFORMACION                                              *
 *                                                                           *
 *      LLAMA A:                                                             *
 *                                                                           *
 *          PIDE_VACIO                                                       *
 *          CREA_INFORMACION                                                 *
 *          INSERTA                                                          *
 *          BUSCA                                                            *
 *          GENERA                                                           *
 *          GOTOXY                                                           *
 *                                                                           *
 *      MACRO ALGORITMO DEL PROGRAMA                                         *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
```

```
 *                                                                    *
 *                                                                    *
 *                                                                    *
 *                                                                    *
 ********************************************************************/

DCL
FECHA                                                    CHAR(8),
AÑO                                                      CHAR(2),
CK                                                       CHAR(1),
BA_ENC                                                   BIT(1),
(POS_ABS,PF_CM,LON_CM,PF_CB,ULT_N,POS_RE,VIA(5),T)       BIN FIXED,
RES_CM                                                   CHAR(120) VAR,
NNA                                                      BIT(1),
(CLAVE2)                                                  CHAR(120) VAR,
CLAVE1                                        CHAR(120) VAR EXT STATIC,
NUM_AU1                                                  BIN FIXED,
(DIRE_COR1,DIRE_PANTA1,POS,LON,I,IG,FG,IV,FV,K)          BIN FIXED,
(AP2,APLINV1,APLINV2,DIR)                                DEC FIXED,
NOMBRE_ARCHIVO                                           CHAR(20) VAR,
01 BLQ_MONO_PP,
   02 CADENA_LEIDA2                                      CHAR(1000) VAR,
NUMERO_INDICE                                            BIN FIXED,
(CONTROL_ARCHIVO,INDICE,OUT,ARCH_DATOS)                  FILE,
TESAURO                                                  FILE VARIABLE,
(NREF,CONT_AU,NUMERO_ARC,CONT_AU_AUX)                    BIN FIXED;

/*                  RUTINAS EXTERNAS            */

DCL
PIDE_VACIO ENTRY(CHAR(20) VAR,DEC FIXED),
CREA_INFORMACION ENTRY(BIN FIXED,BIN FIXED,BIN FIXED,BIN FIXED,
BIN FIXED,BIN FIXED,CHAR(1000) VAR),
INSERTA ENTRY(FILE VARIABLE,CHAR(120) VAR,DEC FIXED,BIN FIXED,DEC FIXED),
BUSCA ENTRY(FILE VARIABLE,CHAR(120) VAR,BIT(1),BIN FIXED,CHAR(120) VAR,
BIN FIXED,BIN FIXED,1,2,3 BIN FIXED,3 BIN FIXED,3 BIN FIXED,
3 BIN FIXED,3 BIN FIXED,3 BIN FIXED,2 CHAR(1009) VAR,BIN FIXED,
(5) BIN FIXED,BIN FIXED,BIN FIXED,1,2 BIN FIXED,2 BIN FIXED,
2 BIN FIXED,2 BIN FIXED),
GENERA ENTRY(BIN FIXED,BIN FIXED,BIN FIXED),
GOTOXY ENTRY(BIN FIXED,BIN FIXED);

DCL
01 CONTROL_ARCH(16),
   02 TIPO                                               BIN FIXED,
   02 N_ARC                                              BIN FIXED,
   02 LON_REG                                            BIN FIXED,
   02 ULT_VACIO                                          BIN FIXED,
   02 MAX_DIR                                            BIN FIXED,
   02 NOMBRE_ARC                                       CHAR(20) VAR,
01 CONTROL_NUMERICO,
   02 NUM_AU                                             BIN FIXED,
   02 NUM_PROF                                           BIN FIXED,
   02 NUM_PED                                            BIN FIXED,
   02 NUM_CHEQ                                           BIN FIXED,
   02 NUM_PAG                                            BIN FIXED,
01 INF,
   02 APU_RAIZ                                           BIN FIXED,
   02 NO_NIV                                             BIN FIXED,
   02 NO_NODO                                            BIN FIXED,
   02 APU_PROX_VACIO                                     BIN FIXED,
01 NODO,
   02 CAB_NODO,
```

```
      03 NUM_N                                                  BIN FIXED,
      03 NIV                                                    BIN FIXED,
      03 NC                                                     BIN FIXED,
      03 TN                                                     BIN FIXED,
      03 VECINA_I                                               BIN FIXED,
      03 VECINA_D                                               BIN FIXED,
   02 ENTRADA                                                  CHAR(1009) VAR;

READ FILE(CONTROL_ARCHIVO) INTO(CONTROL_ARCH) KEY(1);
READ FILE(CONTROL_ARCHIVO) INTO(CONTROL_NUMERICO) KEY(2);

CK = ASCII(11);
T = 0;
IG = 1;
IV = 1;
FG = 100;
FV = 100;
TESAURO = INDICE;

  DO I = 1 TO 8;
   CALL GOTOXY(2,I+1);
   PUT FILE(OUT) SKIP EDIT(CK)(A);
END;
IF NNA = '0'B THEN DO;
   AP2 = 1;
   DIRE_COR1 = 7;
   DIRE_PANTA1 = 15;
END;
ELSE DO;
      FECHA = DATE();
FECHA = SUBSTR(FECHA,5,2)!!"/"!!SUBSTR(FECHA,3,2)!!"/"!!SUBSTR(FECHA,1,2);
AND = SUBSTR(FECHA,7,2);
   AP2 = 0;
   DIRE_COR1 = 19;
   DIRE_PANTA1 = 14;
END;
CALL GENERA(DIRE_PANTA1,IV,FV);
IF NNA = '1'B THEN DO;
   CALL GOTOXY(9,10);
   PUT FILE(OUT) EDIT(ASCII(29),AND,ASCII(28))(A,A,A);
END;
CALL CREA_INFORMACION(DIRE_PANTA1,DIRE_COR1,IG,FG,IV,FV,CADENA_LEIDA2);
IF FV = 0 THEN DO;
   T = 1;
   GOTO CONTINUE1;
END;
CADENA_LEIDA2 = CADENA_LEIDA2!!ASCII(255);
POS = 1;
LON = RANK(SUBSTR(CADENA_LEIDA2,POS,1));
CLAVE2 = SUBSTR(CADENA_LEIDA2,POS+1,LON-1);
POS = POS + LON;
LON = RANK(SUBSTR(CADENA_LEIDA2,POS,1)) - 1;
IF LON ^= 0 THEN
   CLAVE2 = CLAVE2 !! SUBSTR(CADENA_LEIDA2,POS+1,LON);
CLOSE FILE(INDICE);
I = 1;
DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_INDICE);
   I = I + 1;
END;
OPEN FILE(INDICE) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) UPDATE
DIRECT KEYED ENV(RECSIZE(1024));
NN:
CALL BUSCA(TESAURO,CLAVE2,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,NODO,
```

```
PF_CB,VIA,ULT_N,POS_RE,INF);
IF BA_ENC = "0"B THEN DO;
   CLOSE FILE(ARCH_DATOS);
   I = 1;
   DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_ARC);
   I = I + 1;
   END;
   LON = CONTROL_ARCH(I).LON_REG;
   NOMBRE_ARCHIVO = CONTROL_ARCH(I).NOMBRE_ARC;
   OPEN FILE(ARCH_DATOS) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) UPDATE
   DIRECT KEYED ENV(RECSIZE(LON));
   CALL FIDE_VACIO(NOMBRE_ARCHIVO,DIR);
   NUM_AU = NUM_AU + 1;
   WRITE FILE(CONTROL_ARCHIVO) FROM(CONTROL_NUMERICO) KEYFROM(2);
   WRITE FILE(ARCH_DATOS) FROM(BLQ_MONO_PP) KEYFROM(DIR);
   APLINV1 = DIR;
   NREF = NUM_AU;
   APLINV2 = AP2;
   CALL INSERTA(TESAURO,CLAVE2,APLINV1,NREF,APLINV2);
   CLAVE2 = CHAR(NUM_AU);
   CALL INSERTA(TESAURO,CLAVE2,APLINV1,NREF,APLINV2);
   IF AP2 = 0 THEN DO;
      CLOSE FILE(INDICE);
      NUMERO_INDICE = 12;
      I = 1;
      DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_INDICE);
         I = I + 1;
      END;
      OPEN FILE(INDICE) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) UPDATE
      DIRECT KEYED ENV(RECSIZE(1024));
      CALL BUSCA(TESAURO,CLAVE1,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,NODO,
      PF_CB,VIA,ULT_N,POS_RE,INF);
      CONT_AU = RANK(SUBSTR(ENTRADA,POS_ABS+5,1))*256 +
      RANK(SUBSTR(ENTRADA,POS_ABS+6,1));
      DIR = DEC(RANK(SUBSTR(ENTRADA,POS_ABS+2,1)))*10000 +
      RANK(SUBSTR(ENTRADA,POS_ABS+3,1))*100 +
      RANK(SUBSTR(ENTRADA,POS_ABS+4,1));
      CONT_AU_AUX = CONT_AU;
      CONT_AU = CONT_AU + 1;
      SUBSTR(ENTRADA,POS_ABS+5,1) = ASCII(CONT_AU/256);
      SUBSTR(ENTRADA,POS_ABS+6,1) = ASCII(CONT_AU);
      REWRITE FILE(INDICE) FROM(NODO) KEY(NODO.CAB_NODO.NUM_N);
      I = 1;
      K = 1;
      DO WHILE(K < CONT_AU_AUX + 1);
         CLAVE2 = CLAVE1!!CHAR(I);
         CALL BUSCA(TESAURO,CLAVE2,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,
         NODO,PF_CB,VIA,ULT_N,POS_RE,INF);
         IF BA_ENC = "1"B THEN
            K = K + 1;
         I = I + 1;
      END;
      CLAVE2 = CLAVE1 !! CHAR(I);
      APLINV1 = DIR;
      APLINV2 = AP2;
      NREF = NUM_AU;
      CALL INSERTA(TESAURO,CLAVE2,APLINV1,NREF,APLINV2);
   NUMERO_INDICE = 11;
   CLOSE FILE(INDICE);
   I = 1;
   DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_INDICE);
      I = I + 1;
   END;
```

```
   OPEN FILE(INDICE) TITLE(CONTROL_ARCH(1).NOMBRE_ARC) UPDATE
   DIRECT KEYED ENV(RECSIZE(1024));
END;
   CLAVE2 = CHAR(NUM_AU);
   CALL BUSCA(TESAURO,CLAVE2,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,NODO,
   PF_CB,VIA,ULT_N,POS_RE,INF);
END;
ELSE
   NUM_AU = RANK(SUBSTR(ENTRADA,POS_ABS+5,1))*256 +
   RANK(SUBSTR(ENTRADA,POS_ABS+6,1));
NUM_AU1 = NUM_AU;
DIR = DEC(RANK(SUBSTR(ENTRADA,POS_ABS+2,1)))*10000 +
RANK(SUBSTR(ENTRADA,POS_ABS+3,1))*100 +
RANK(SUBSTR(ENTRADA,POS_ABS+4,1));

CONTINUE1:

END;     /* NUEVO AUTOR  */
```

```
ACEPTAR: PROCEDURE(NUMERO_ACTIVIDAD,NUM_AU1,VECTOR_GUAR_NUM,
CONT_AU_ACEP,VECTOR_GUAR_AP2,AP2,DIRE_PANTA,AUTOR,NEJEMP,
NTITUL,DIR,CLAVE2,CONT_AU_ACEP_AUX);

/*****************************************************************
 *                                                               *
 *   RUTINA QUE INCLUYE LOS AUTORES Y LAS PUBLICACIONES EN LOS   *
 *   PROCESOS                                                    *
 *                                                               *
 *        PARAMETROS:                                            *
 *                                                               *
 *            NUMERO_ARCTIVIDAD: NUMERO DEL PROCESO QUE SE  ESTA *
 *            CREANDO                                            *
 *                                                               *
 *            NUM_AU1: NUMERO CORRESPONDIENTE DEL AUTOR/PP       *
 *                                                               *
 *            VECTOR_GUAR_NUM: VECTOR QUE CONTIENE LOS AUTORES/PP *
 *            A SER INCLUIDOS EL PROCESO                         *
 *                                                               *
 *            VECTOR_GUAR_AP2: VECTOR QUE CONTIENE LOS INDICADORES *
 *            DE LOS AUTORES O PP                                *
 *                                                               *
 *            DIRE_PANTA: DIRECCION DE LA PANTALLA               *
 *                                                               *
 *            NEJEMP: NUMERO DE EJEMPLARES                       *
 *                                                               *
 *            NTIT: NUMERO DE TITULOS                            *
 *                                                               *
 *        VARIABLES:                                             *
 *                                                               *
 *            CADENA_LEIDA: CADENA CON LA INFORMACION DE LOS     *
 *            AUTORES/PP                                         *
 *                                                               *
 *        LLAMADA POR:                                           *
 *                                                               *
 *            MUESTRA_INFORMCION                                 *
 *                                                               *
 *        LLAMA A:                                               *
 *                                                               *
 *            GOTOXY                                             *
 *            CREA INFORMACION                                   *
 *                                                               *
 *        ARCHIVOS:                                              *
 *                                                               *
 *            CONTROL_ARCHIVO: CONTIENE LA INFORMACION DE LOS    *
 *            ARCHIVOS PRINCIPALES DEL SISTEMA                   *
 *                                                               *
 *        MACRO ALGORITMO DEL PROGRAMA:                          *
 *                                                               *
 *                                                               *
 *                                                               *
 *                                                               *
 *                                                               *
 *                                                               *
 *                                                               *
 *                                                               *
 *                                                               *
 *                                                               *
 *                                                               *
 *                                                               *
 *                                                               *
 *                                                               *
 *                                                               *
 *                                                               *
```

```
 *                                                                      *
 *                                                                      *
 *                                                                      *
 *                                                                      *
 *                                                                      *
 *                                                                      *
 ************************************************************************/

DCL
(NEJEMP,NTITUL,CONT_AU_ACEP)                    BIN FIXED,
 CONT_AU_ACEP_AUX                               BIN FIXED,
(NUMERO_ACTIVIDAD,NUMERO_ACTIVIDAD_AUX)         CHAR(12) VAR,
 NUMERO_ACTIVIDAD_1                             CHAR(12) VAR,
DIR                                             DEC FIXED,
(IN,CONTROL_ARCHIVO,ARCH_DATOS)                 FILE,
NUMERO_PROF                                     CHAR(12) VAR,
FECHA                                            CHAR(8),
(I,POS,LON,L,P,M)                                BIN FIXED,
NUM_AU1                                          BIN FIXED,
AC                                               BIT(1),
VECTOR_GUAR_NUM(30)                              BIN FIXED,
VECTOR_GUAR_AP2(30)                              DEC FIXED,
AP2                                              DEC FIXED,
DIRE_PANTA                                       BIN FIXED,
OUT                                              FILE,
CK                                               CHAR(1),
(AUTOR,CLAVE2)                                   CHAR(120) VAR,
DIRE_COR                                         BIN FIXED,
CADENA_LEIDA                                     CHAR(1000) VAR,
(IG,FG,IV,FV)                                    BIN FIXED,
NEJEMP_LE                                        BIN FIXED,
(ANO_SUB,ISBN)                                   CHAR(12) VAR,
ANO                                              CHAR(2),
NEJEM                                            CHAR(3) VAR,
GOTOXY ENTRY(BIN FIXED,BIN FIXED),
CREA_INFORMACION ENTRY(BIN FIXED,BIN FIXED,BIN FIXED,BIN FIXED,
BIN FIXED,BIN FIXED,CHAR(1000) VAR),
01 BLQ_MONO_PP,
   02 CADENA_LEIDA2                                CHAR(1000) VAR,
01 CONTROL_ARCH(16),
   02 TIPO                                         BIN FIXED,
   02 N_ARC                                        BIN FIXED,
   02 LON_REG                                      BIN FIXED,
   02 ULT_VACIO                                    BIN FIXED,
   02 MAX_DIR                                      BIN FIXED,
   02 NOMBRE_ARC                                   CHAR(20) VAR;

READ FILE(CONTROL_ARCHIVO) INTO(CONTROL_ARCH) KEY(1);
READ FILE(ARCH_DATOS) INTO(BLQ_MONO_PP) KEY(DIR);

IF LENGTH(CADENA_LEIDA2) > 850 THEN DO;
   CALL GOTOXY(0,22);
   PUT FILE(OUT) EDIT("AGOTADA CADENA EN AUTOR")(A);
   GOTO FF;
END;

FECHA = DATE();
ANO = SUBSTR(FECHA,1,2);

CK = ASCII(11);
IG = 1;
IV = 1;
FG = 100;
```

```
FV = 100;

  IF CONT_AU_ACEP ^= 0 THEN DO;
      I = 1;
      DO WHILE(NUM_AU1 ^= VECTOR_GUAR_NUM(I) & I < CONT_AU_ACEP + 1);
         I = I + 1;
      END;
      IF I < CONT_AU_ACEP + 1 THEN
         GOTO FF;
   END;
   CONT_AU_ACEP = CONT_AU_ACEP + 1;
   CONT_AU_ACEP_AUX = CONT_AU_ACEP_AUX + 1;
   VECTOR_GUAR_NUM(CONT_AU_ACEP) = NUM_AU1;
   VECTOR_GUAR_AP2(CONT_AU_ACEP) = AP2;
   IF DIRE_PANTA = 5 THEN DO;
      I = 3;
      POS = 5;
   END;
   IF (DIRE_PANTA = 4 ! DIRE_PANTA = 8 ! DIRE_PANTA = 9) THEN DO;
      IF DIRE_PANTA = 8 THEN DO;
         I = 4;
         POS = 6;
      END;
      IF DIRE_PANTA = 9 THEN DO;
         I = 3;
         POS = 5;
      END;
      IF DIRE_PANTA = 4 THEN DO;
         I = 5;
         POS = 7;
      END;
   END;
   IF DIRE_PANTA = 6 THEN DO;
      I = 7;
      POS = 9;
   END;
   CALL GOTOXY(8,I);
   PUT FILE(OUT) EDIT(CK)(A);
   IF AP2 = 0 THEN
      GOTO X1;
   PUT FILE(OUT) EDIT(ASCII(29),AUTOR,ASCII(28),CK)(A,A,A,A);
X1:CALL GOTOXY(9,POS);
   PUT FILE(OUT) EDIT(CK)(A);
   PUT FILE(OUT) EDIT(ASCII(29),CLAVE2,ASCII(28),CK)(A,A,A,A);
   IF DIRE_PANTA = 4 THEN
      NUMERO_ACTIVIDAD_AUX = "PF"!!NUMERO_ACTIVIDAD;
   IF DIRE_PANTA = 5 THEN
      NUMERO_ACTIVIDAD_AUX = "FA"!!NUMERO_ACTIVIDAD;
   IF DIRE_PANTA = 6 THEN
      NUMERO_ACTIVIDAD_AUX = "PD"!!NUMERO_ACTIVIDAD;
   IF DIRE_PANTA = 8 THEN
      NUMERO_ACTIVIDAD_AUX = "PA"!!NUMERO_ACTIVIDAD;
   IF DIRE_PANTA = 9 THEN DO;
      NUMERO_ACTIVIDAD_1 = NUMERO_ACTIVIDAD;
      DO I = 1 TO 9 - LENGTH(NUMERO_ACTIVIDAD_1);
         NUMERO_ACTIVIDAD_1 = " " !! NUMERO_ACTIVIDAD_1;
      END;
      NUMERO_ACTIVIDAD_AUX = "RE"!!NUMERO_ACTIVIDAD_1;
   END;
   IF DIRE_PANTA = 4 THEN DO;
      IF AP2 = 1 THEN
         DIRE_COR = 22;
      ELSE
```

```
        DIRE_COR = 4;
    DO I = 1 TO 4;
       CALL GOTOXY(10+I,9);
      PUT FILE(OUT) EDIT(" ")(A(1));
   END;
   DO I = 1 TO 4;
      CALL GOTOXY(32+I,9);
      PUT FILE(OUT) EDIT(" ")(A(1));
   END;
   DO I = 1 TO 12;
      CALL GOTOXY(47+I,9);
      PUT FILE(OUT) EDIT(" ")(A(1));
   END;
     CALL CREA_INFORMACION(DIRE_PANTA,DIRE_COR,IG,FG,IV,FV,CADENA_LEIDA);
     POS = 1;
     LON = RANK(SUBSTR(CADENA_LEIDA,POS,1));
     NEJEMP_LE = BINARY(SUBSTR(CADENA_LEIDA,2,LON-1));
     POS = POS + LON;
     LON = RANK(SUBSTR(CADENA_LEIDA,POS,1));
     IF AP2 = 0 THEN DO;
        ANO_SUB = SUBSTR(CADENA_LEIDA,POS+1,LON-1);
        IF ANO_SUB = "" THEN
           ANO_SUB = ANO;
         POS = POS + LON;
        LON = RANK(SUBSTR(CADENA_LEIDA,POS,1));
     END;
     ISBN = SUBSTR(CADENA_LEIDA,POS+1,LON-1);
     IF NEJEMP_LE = 0 THEN
        NEJEMP_LE = 1;
     IF AP2 = 0 THEN DO;
        POS = 1;
        LON = RANK(SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS,1));
        DO WHILE(LON ^= 255);
           POS = POS + LON;
           LON = RANK(SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS,1));
        END;
        CADENA_LEIDA2 = SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,1,POS-1)!!
        ASCII(LENGTH(NUMERO_ACTIVIDAD_AUX)+1)!!NUMERO_ACTIVIDAD_AUX !!
        ASCII(LENGTH(CHAR(NEJEMP_LE))+1)
        !!CHAR(NEJEMP_LE)!!ASCII(LENGTH(CHAR(ANO_SUB))+1)!!
       CHAR(ANO_SUB)!!ASCII(LENGTH(CHAR(ISBN))+1)!!CHAR(ISBN)!!
        SUBSTR(CADENA_LEIDA2,POS);
    END;
    ELSE DO;
        POS = 1;
        LON = RANK(SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS,1));
        DO WHILE(LON ^= 255);
           POS = POS + LON;
           LON = RANK(SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS,1));
        END;
      CADENA_LEIDA2 = SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,1,POS-1)!!
      ASCII(LENGTH(NUMERO_ACTIVIDAD_AUX)+1)!!NUMERO_ACTIVIDAD_AUX !!
      ASCII(LENGTH(CHAR(NEJEMP_LE))+1)!!CHAR(NEJEMP_LE)!!
      ASCII(LENGTH(CHAR(ISBN))+1)!!CHAR(ISBN)!!
      SUBSTR(CADENA_LEIDA2,POS);
    END;
    NEJEMP = NEJEMP + NEJEMP_LE;
    NTITUL = NTITUL + 1;
 END;
 IF DIRE_PANTA = 5 THEN DO;
    M = 10;
    P = 6;
    L = 11;
```

```
      CALL BORRA1;
      M = 10;
      P = 44;
      L = 11;
      CALL BORRA1;
      DIRE_COR = 6;
      CALL CREA_INFORMACION(DIRE_PANTA,DIRE_COR,IG,FG,IV,FV,CADENA_LEIDA);
      POS = 1;
      LON = RANK(SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS,1));
      DO WHILE(LON ^= 255);
         POS = POS + LON;
         LON = RANK(SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS,1));
      END;
      CADENA_LEIDA2 = SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,1,POS-1)!!
      ASCII(LENGTH(NUMERO_ACTIVIDAD_AUX)+1)!!NUMERO_ACTIVIDAD_AUX !!
      CADENA_LEIDA!!SUBSTR(CADENA_LEIDA2,POS);
   END;
   IF (DIRE_PANTA = 6 ! DIRE_PANTA = 8 ! DIRE_PANTA = 9) THEN DO;
      IF DIRE_PANTA = 6 THEN
       POS = 11;
      IF DIRE_PANTA = 8 THEN
       POS = 8;
      IF DIRE_PANTA = 9 THEN
       POS = 7;
      M = 4;
      P = 18;
      L = POS;
      CALL BORRA1;
REE:  CALL GOTOXY(18,POS);
      READ FILE(IN) INTO(NEJEM);
      IF NEJEM = "" THEN
         NEJEM = 1;
    IF VERIFY(NEJEM,"0987654321. ") ^= 0 THEN DO;
       M = 4;
       P = 18;
       L = POS;
       CALL BORRA1;
       GOTO REE;
    END;
      POS = 1;
      LON = RANK(SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS,1));
      DO WHILE(LON ^= 255);
         POS = POS + LON;
         LON = RANK(SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS,1));
      END;
      CADENA_LEIDA2 = SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,1,POS-1)!!
      ASCII(LENGTH(NUMERO_ACTIVIDAD_AUX)+1)!!NUMERO_ACTIVIDAD_AUX !!
      ASCII(LENGTH(NEJEM)+1)!!NEJEM!!SUBSTR(CADENA_LEIDA2,POS);
   END;
   WRITE FILE(ARCH_DATOS) FROM(BLQ_MONO_PP) KEYFROM(DIR);

  BORRA1: PROCEDURE;

    DO I = 1 TO M;
       CALL GOTOXY(I+P,L);
       PUT FILE(OUT) EDIT(" ")(A(1));
    END;

  END;   /*   BORRA   */

FF:

END; /* ACEPTAR    */
```

```
MUESTRA_DATOS: PROCEDURE(DIRE_PANTA,NUMERO_ACTIVIDAD,CADENA_LEIDA,
NUMERO_INDICE,DIRE_COR1,IG,FG,IV,FV);

/*****************************************************************************
 *                                                                           *
 *   PROGRAMA ENCARGADO DE MOSTRAR Y PEDIR LOS PRINCIPALES                   *
 *   DATOS DEL PROCESO A SER CREADO                                          *
 *                                                                           *
 *   PARAMETROS:                                                             *
 *                                                                           *
 *       DIRE_PANTA: DIRECCION DE LA PANTALLA DEL PROCESO                    *
 *                                                                           *
 *       NUMERO_ACTIVIDAD: NUMERO DEL PROCESO A SER CREADO                   *
 *                                                                           *
 *       CADENA_LEIDA: CADENA DE INFORMACION DEL PROCESO                     *
 *                                                                           *
 *       NUMERO_INDICE: NUMERO DEL INDICE DEL PROCESO A CREAR                *
 *                                                                           *
 *       DIRE_COR1: DIRECCION DE LAS COORDENADAS DE LOS DATOS                *
 *                  DE LA PANTALLA                                           *
 *                                                                           *
 *       IG: INICIO DE COORDENADAS                                           *
 *                                                                           *
 *       FG: FIN DE COORDENADAS                                              *
 *                                                                           *
 *       IV: INICIO DEL GRUPO REPETITIVO                                     *
 *                                                                           *
 *       FV: FIN DEL GRUPO REPETITIVO                                        *
 *                                                                           *
 *   VARIABLES:                                                              *
 *                                                                           *
 *       FECHA: FECHA DEL DIA                                                *
 *                                                                           *
 *       ANO: ANO ACTUAL                                                     *
 *                                                                           *
 *   LLAMADA POR:                                                            *
 *                                                                           *
 *       MUESTRA_INFORMACION                                                 *
 *                                                                           *
 *   LLAMA A:                                                                *
 *                                                                           *
 *       GOTOXY                                                              *
 *       BUSCA_MONEDA                                                        *
 *       BUSCA_BANCO                                                         *
 *       CREA_INFORMACION                                                    *
 *                                                                           *
 *   ARCHIVOS:                                                               *
 *                                                                           *
 *       CONTROL_ARCHIVO: GUARDA LA INFORMACION DE LOS PRINCIPALES           *
 *                        ARCHIVOS DEL SISTEMAS                              *
 *                                                                           *
 *   MACRO ALGORITMO DEL PROGRAMA:                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
 *                                                                           *
```

```
 *                                                                                  *
 *                                                                                  *
 *                                                                                  *
 *                                                                                  *
 *                                                                                  *
 ************************************************************************************/

DCL
(I,L,P,M)                                                          BIN FIXED,
ENCONTRADO                                                          BIT(1),
MONED                                                               CHAR(5) VAR,
BANC                                                                CHAR(1) VAR,
(NUM_FACTURA,NUM_CHEQUE)                                            CHAR(20) VAR,
FECHA                                                                CHAR(8),
ANO                                                                  CHAR(2),
CLAVE1                                                     CHAR(120) VAR EXT STATIC,
(IG,FG,IV,FV,DIRE_PANTA,DIRE_COR1,NUMERO_INDICE,LON) BIN FIXED,
NUMERO_ACTIVIDAD                                                    CHAR(12) VAR,
CADENA_LEIDA                                                       CHAR(1000) VAR,
(OUT,CONTROL_ARCHIVO,IN)                                            FILE,
GOTOXY ENTRY(BIN FIXED,BIN FIXED);
DCL
01 CONTROL_NUMERICO,
   02 NUM_AU                                                         BIN FIXED,
   02 NUM_PROF                                                       BIN FIXED,
   02 NUM_PED                                                        BIN FIXED,
   02 NUM_CHEQ                                                       BIN FIXED,
   02 NUM_PAG                                                        BIN FIXED,
BUSCA_MONEDA ENTRY(CHAR(5) VAR,BIT(1)),
BUSCA_BANCO ENTRY(BIN FIXED,BIT(1)),
CREA_INFORMACION ENTRY(BIN FIXED,BIN FIXED,BIN FIXED,BIN FIXED,BIN FIXED,
BIN FIXED,CHAR(1000) VAR);

FECHA = DATE();
FECHA = SUBSTR(FECHA,5,2)!!"/"!!SUBSTR(FECHA,3,2)!!"/"!!SUBSTR(FECHA,1,2);
ANO = SUBSTR(FECHA,7,2);

READ FILE(CONTROL_ARCHIVO) INTO(CONTROL_NUMERICO) KEY(2);

IF DIRE_PANTA = 4 THEN DO;
   NUM_PROF = NUM_PROF + 1;
   WRITE FILE(CONTROL_ARCHIVO) FROM(CONTROL_NUMERICO) KEYFROM(2);
   CALL GOTOXY(12,0);
   PUT FILE(OUT) EDIT(ASCII(29),CLAVE1,ASCII(28))(A,A(70),A);
   CALL GOTOXY(0,3);
   PUT FILE(OUT) EDIT(ASCII(29),NUM_PROF,ASCII(28))(A,A(11),A);
   CALL GOTOXY(58,3);
   PUT FILE(OUT) EDIT(ASCII(29),FECHA,ASCII(28))(A,A(8),A);
   CALL GOTOXY(29,9);
   PUT FILE(OUT) EDIT(ASCII(29),ANO,ASCII(28))(A,A,A);
   CADENA_LEIDA = ASCII(LENGTH(CLAVE1)+1)!!CLAVE1!!
   ASCII(LENGTH(CHAR(NUM_PROF))+1)!!CHAR(NUM_PROF);
   NUMERO_INDICE = 12;
   NUMERO_ACTIVIDAD = NUM_PROF;
END;
IF DIRE_PANTA = 5 THEN DO;
   CALL GOTOXY(12,2);
   PUT FILE(OUT) EDIT(ASCII(29),CLAVE1,ASCII(28))(A,A(70),A);
CCC: CALL GOTOXY(1,1);
   READ FILE(IN) INTO(NUM_FACTURA);
   IF (NUM_FACTURA = "" ! LENGTH(NUM_FACTURA) > 9)THEN DO;
      M = 12;
      P = 1;
```