

```

L = 1;
CALL BORRA;
GOTO CCC;
END;
IF VERIFY(NUM_FACTURA,"1234567890") = 0 THEN DO;
DO I = 1 TO 9 - LENGTH(NUM_FACTURA);
NUM_FACTURA = " " !! NUM_FACTURA;
END;
END;
CALL GOTOXY(1,3);
PUT FILE(OUT) EDIT(ASCII(29),"A",ASCII(28))(A(1));
CCCC: CALL GOTOXY(14,1);
READ FILE(IN) INTO(MONED);
CALL BUSCA_MONEDA(MONED,ENCENTRADO);
IF ENCENTRADO = "0"B THEN DO;
M = 6;
P = 14;
L = 1;
CALL BORRA;
GOTO CCCC;
END;
CALL CREA_INFORMACION(DIRE_PANTA,DIRE_COR1,IG,FG,IV,FV,(CADENA_LEIDA);
NUMERO_ACTIVIDAD = NUM_FACTURA;
CADENA_LEIDA = ASCII(LENGTH(CLAVE1)+1)!!CLAVE1!!
ASCII(LENGTH(NUM_FACTURA)+1) !! NUM_FACTURA !!
ASCII(LENGTH(MONED)+1) !! MONED !!
CADENA_LEIDA;
NUMERO_INDICE = 14;
END;
IF DIRE_PANTA = 6 THEN DO;
NUM_PED = NUM_PED + 1;
WRITE FILE(CONTROL_ARCHIVO) FROM(CONTROL_NUMERICO) KEYFROM(2);
NUMERO_ACTIVIDAD = NUM_PED;
CALL GOTOXY(10,0);
PUT FILE(OUT) EDIT(ASCII(29),NUM_PED,ASCII(28))(A,A,A);
NUMERO_INDICE = 14;
END;
IF DIRE_PANTA = 8 THEN DO;
NUM_PAG = NUM_PAG + 1;
WRITE FILE(CONTROL_ARCHIVO) FROM(CONTROL_NUMERICO) KEYFROM(2);
NUMERO_ACTIVIDAD = NUM_PAG;
CALL GOTOXY(0,1);
PUT FILE(OUT) EDIT(ASCII(29),NUM_PAG,ASCII(28))(A,A,A);
SS:CALL GOTOXY(13,1);
READ FILE(IN) INTO(NUM_CHEQUE);
IF NUM_CHEQUE = "" THEN GOTO SS;
IF LENGTH(NUM_CHEQUE) > 12 THEN DO;
DO I = 1 TO LENGTH(NUM_CHEQUE);
CALL GOTOXY(I+12,1);
PUT FILE(OUT) EDIT(" ")(A(1));
END;
GOTO SS;
END;
SSS: CALL GOTOXY(27,1);
READ FILE(IN) INTO(BANC);
IF VERIFY(BANC,"1234567890") ^= 0 THEN DO;
M = 3;
P = 27;
L = 1;
CALL BORRA;
GOTO SSS;
END;
CALL BUSCA_BANCO(BINARY(BANC),ENCENTRADO);

```

```

IF ENCONTRADO = "0" B THEN DO;
  M = 3;
  P = 27;
  L = 1;
  CALL BORRA;
  GOTO SSS;
END;
CALL CREA_INFORMACION(DIRE_PANTA,DIRE_COR1,IG,FG,IV,FV,CADENA_LEIDA);
CADENA_LEIDA = ASCII(LENGTH(CHAR(NUM_PAG))+1)!!CHAR(NUM_PAG)!!
ASCII(LENGTH(NUM_CHEQUE)+1) !! NUM_CHEQUE !!
ASCII(LENGTH(BANC)+1) !! BANC !!
CADENA_LEIDA;
END;
IF DIRE_PANTA = 9 THEN DO;
  CALL CREA_INFORMACION(DIRE_PANTA,DIRE_COR1,IG,FG,IV,FV,CADENA_LEIDA);
  LON = RANK(SUBSTR(CADENA_LEIDA,1,1));
  NUMERO_ACTIVIDAD = SUBSTR(CADENA_LEIDA,2,LON-1);
  NUMERO_INDICE = 17;
END;

BORRA: PROCEDURE;

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

END; /* BORRA */

END; /* MUESTRA_DATOS */

```

BUSCA_ACTIVIDAD: PROCEDURE(J,K,COMANDO1);

```
/*
*
* PROGRAMA QUE SE ENCARGA DE BUSCAR EL PROCESO DESEADO,
* BUSCA EN EL ARCHIVO PANTALLAS LA DIRECCION DEL REGISTRO
* DONDE SE ENCUENTRA LA PANTALLA Y LA DIRECCION DEL REGISTRO
* DONDE ESTAN LAS COORDENADAS DE LOS DATOS EN EL ARCHIVO
* PANTALLAS Y COORDENADAS RESPECTIVAMENTE, LUEGO VA A LA
* TABLA DE ARCHIVOS Y ABRE EL ARCHIVO Y EL INDICE RESPECTIVO
* PARAMETROS:
*   J: INDICA EL NUMERO DEL PROCESO A BUSCAR
*   K: CONTROL DE INTERUPCION DEL PROCESO
*     1: EL PROCESO A SIDO INTERUMPIDO
*     0: NO HAY INTERUPCION
*   COMANDO1: COMANDO SECUNDARIO
* VARIABLES:
*   CONT_AU_PROF: INDICA EL NUMERO DE AUTORES/PP QUE
*   HAY EN LA PROFORMA
*   CONT_AU_FACT: INDICA EL NUMERO DE AUTORES/PP QUE
*   HAY EN LA FACTURA
*   CONT_AU_RECT: INDICA EL NUMERO DE AUTORES/PP
*   RECIBIDOS
*   CONT_AU_PED: INDICA EL NUMERO DE AUTORES/PP QUE
*   HAY EN EL PEDIDO
*   CONT_AU_PAG: INDICA EL NUMERO DE AUTORES/PP
*   PAGOS
*   CONT_AU_CHEQ: INDICA EL NUMERO DE PROVEEDORES
*   QUE HAY EN LA SOLICITUD DE CHEQUE
*   CONTADOR_REIN: INDICA SI EL REINTEGRO ESTA EN
*   CONSULTA
*     1: REINTEGRO EN CONSULTA
*     0: NO ESTA EN CONSULTA
*   CONTADOR_TIT: NUMERO DE AUTORES/PP QUE SE
*   ENCUENTRAN BAJO EL PROVEEDOR
*   TITULO_AUX:
*     1: TITULO EN CONSULTA
*     0: TITULO NO ESTA EN CONSULTA
*   PROF_AUX:
*     1: PROFORMA EN CONSULTA
*     0: NO ESTA EN CONSULTA
*   (IGUAL PARA LOS DEMAS PROCESOS)
*   J1:
*     INDICA SI HAY QUE ADELANTAR O RETROCEDER
*     EN ALGUNA ACTIVIDAD ADMINISTRATIVA
*     0: NO HAY QUE ADELANTAR NI RETROCEDER
*     (8,44,9,26,27,48) ADELANTA
*     (46,11,29,12,30,49) RETROCEDE
*
* LLAMADA POR:
*   COMENZAR
*
* LLAMA A:
*   BUSCA
*   GOTOXY
*   RECONSTRUYE
*
* ARCHIVOS:
*   PANTALLAS: GUARDA LAS PANTALLAS DEL SISTEMA
*   COORDENADAS: GUARDA LAS COORDENADAS DE LOS DATOS
*   DE LAS PANTALLAS
*   CONTROL_PANTALLAS: GUARDA LAS DIRECCIONES DE LAS
*   PANTALLAS Y COORDENADAS RESPECTIVAS
*   CONTROL_ARCHIVO: GUARDA INFORMACION DE LOS ARCHIVOS
*   PRINCIPALES DEL SISTEMA
*
*/
```



```

01 BLQ_CHEQ,
  02 CADENA_CHEQ CHAR(1000) VAR,
01 EST_LEC,
  02 VECTOR_COOR(500) CHAR(1),
TESAURO FILE VARIABLE,
BA_ENC BIT(1),
(PDS_ABS,PF_CN,LON_CN,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_1 BIN FIXED,
    03 VECINA_D BIN FIXED,
  02 ENTRADA CHAR(1009) VAR,
01 CONTROL_PANTA(16),
  02 NUM_ARC BIN FIXED,
  02 NUM_INDICE BIN FIXED,
  02 DIRE_P BIN FIXED,
  02 DIRE_C1 BIN FIXED,
  02 NOMBRE_P CHAR(4) 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;
DCI
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),
RECONSTRUYE ENTRY(BIN FIXED,BIN FIXED,BIN FIXED,BIN FIXED,BIN FIXED,
BIN FIXED,CHAR(1000) VAR);

```

```

CK = ASCII(11);
OB = 0;
CONTADOR_REIN = 0;
CONTADOR_TIT = 0;
TITULO_AUX = 1;
CONT_AU_PROF = 0;
CONT_AU_PED = 0;
PED_AUX = 1;
CONTADOR_FR = 0;
CONTADOR_AU = 0;
CONTADOR_PP = 0;
CONT_AU_CHEQ = 0;
CHEQ_AUX = 1;
CONT_AU_FACT = 0;
CONT_AU_REC1 = 0;
CONT_AU_PAG = 0;
PROF_AUX = 1;
FACT_AUX = 1;
PAG_AUX = 1;

```

```
RECT_AUX = 1;
J1 = 0;
TESAURO = INDICE;
CONT_AUX = 0;
READ FILE(CONTROL_ARCHIVO) INTO((CONTROL_ARCH) KEY(1);
READ FILE(CONTROL_PANTALLAS) INTO((CONTROL_PANTA) KEY(1);
```

```
IF COMANDO1 = "" THEN DO;
  IF (J = 42 & PROFORMA1 ^= "") THEN
    COMANDO1 = PROFORMA1;
  IF (J = 4 & FACTURA1 ^= "") THEN
    COMANDO1 = FACTURA1;
  IF (J = 24 & PEDIDO1 ^= "") THEN
    COMANDO1 = PEDIDO1;
  IF (J = 3 & CHEQUE1 ^= "") THEN
    COMANDO1 = CHEQUE1;
  IF (J = 23 & PAGO1 ^= "") THEN
    COMANDO1 = PAGO1;
  IF (J = 38 & RECIBIR1 ^= "") THEN
    COMANDO1 = RECIBIR1;
```

```
END;
```

```
IF (J = 44 ! J = 46) THEN DO;
```

```
  J1 = J;
```

```
  J = 42;
```

```
  IF PROFORMA1 ^= "" THEN
```

```
    COMANDO1 = PROFORMA1;
```

```
END;
```

```
IF (J = 9 ! J = 12) THEN DO;
```

```
  J1 = J;
```

```
  J = 4;
```

```
  IF FACTURA1 ^= "" THEN
```

```
    COMANDO1 = FACTURA1;
```

```
END;
```

```
IF (J = 27 ! J = 30) THEN DO;
```

```
  J1 = J;
```

```
  J = 24;
```

```
  IF PEDIDO1 ^= "" THEN
```

```
    COMANDO1 = PEDIDO1;
```

```
END;
```

```
IF (J = 8 ! J = 11) THEN DO;
```

```
  J1 = J;
```

```
  J = 3;
```

```
  IF CHEQUE1 ^= "" THEN
```

```
    COMANDO1 = CHEQUE1;
```

```
END;
```

```
IF (J = 26 ! J = 29) THEN DO;
```

```
  J1 = J;
```

```
  J = 23;
```

```
  IF PAGO1 ^= "" THEN
```

```
    COMANDO1 = PAGO1;
```

```
END;
```

```
IF (J = 48 ! J = 49) THEN DO;
```

```
  J1 = J;
```

```
  J = 38;
```

```
  IF RECIBIR1 ^= "" THEN
```

```
    COMANDO1 = RECIBIR1;
```

```
END;
```

```
IF J = 38 THEN
```

```
  NUMERO_ARC = 35;
```

```
IF J = 42 THEN
```

```
  NUMERO_ARC = 30;
```

```
IF J = 4 THEN
```

```
  NUMERO_ARC = 31;
```

```

IF J = 24 THEN
  NUMERO_ARC = 32;
IF J = 23 THEN
  NUMERO_ARC = 33;
IF J = 3 THEN
  NUMERO_ARC = 34;
  I = 1;
  DO WHILE (CONTROL_PANTA(I).NUM_ARC ^= NUMERO_ARC);
    I = I + 1;
  END;
  DIRE_PANTA = CONTROL_PANTA(I).DIRE_F;
  FG = 100;
  DIRE_COR = CONTROL_PANTA(I).DIRE_C1;
  FV = 100;
  IG = 1;
  IV = 1;
  NUMERO_INDICE = CONTROL_PANTA(I).NUM_INDICE;
  I = 1;
  IF J = 38 THEN
    NUMERO_ARC = 8;
  IF J = 42 THEN
    NUMERO_ARC = 3;
  IF J = 4 THEN
    NUMERO_ARC = 4;
  IF J = 24 THEN
    NUMERO_ARC = 5;
  IF J = 23 THEN
    NUMERO_ARC = 7;
  IF J = 3 THEN
    NUMERO_ARC = 6;
  DO WHILE (CONTROL_ARCH(I).N_ARC ^= NUMERO_INDICE);
    I = I + 1;
  END;
  CLOSE FILE(INDICE);
  OPEN FILE(INDICE) TITLE (CONTROL_ARCH(I).NOMBRE_ARC) INPUT
  DIRECT KEYED ENV(RECSIZE(1024));
  IF COMANDO1 ^= "" THEN DO;
    I = 1;
    DO WHILE (CONTROL_ARCH(I).N_ARC ^= NUMERO_ARC);
      I = I + 1;
    END;
    LONGITUD = CONTROL_ARCH(I).LON_RFG;
    CLOSE FILE(ARCH_DATOS);
    OPEN FILE(ARCH_DATOS) TITLE (CONTROL_ARCH(I).NOMBRE_ARC) INPUT
    DIRECT KEYED ENV(RECSIZE(LONGITUD));
    CLAVE2 = COMANDO1;
    CALL BUSCA(TESAURO,CLAVE2,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,
    NODO,PF_CB,VIA,ULT_N,POS_RE,(NF));
    IF BA_ENC ^= "0"B THEN DO;
      IF J1 = 8 THEN
        CALL ADELANTA;
      IF (J1=44 ! J1=9 ! J1=27 ! J1=26 ! J1=48) THEN DO;
        CONT_AU = RANK(SUBSTR(ENTRADA,POS_ABS+5,1))*256 +
        RANK(SUBSTR(ENTRADA,POS_ABS+6,1));
        K = 0;
        I = 0;
        DO WHILE (I < CONT_AU);
          BA_ENC = "0"B;
          DO WHILE (BA_ENC = "0"B);
            K = K + 1;
            CLAVE = CLAVE2 !! CHAR(K);
            CALL BUSCA(TESAURO,CLAVE,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,
            NODO,PF_CB,VIA,ULT_N,POS_RE,(NF));

```

```

        END;
        I = I + 1;
    END;
    CALL ADELANTA;
    END;
    IF (J1=46 ! J1=11 ! J1=29 ! J1=12 ! J1=30 ! J1=49). THEN
        CALL RETROCEDE;
        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 = RANK(SUBSTR(ENTRADA,POS_ABS+5,1))*256 +
            RANK(SUBSTR(ENTRADA,POS_ABS+6,1));
        IF J = 42 THEN DO;
            READ FILE(ARCH_DATOS) INTO(BLQ_PROF) KEY(DIR);
            CALL RECONSTRUYE(DIRE_PANTA,DIRE_COR,IG,FG,IV,FV,BLQ_PROF.CADENA_PR
F);
            IF J1 ^= 0 THEN
                CALL BUSCA_NUEVA_ACTIVIDAD;
            PROFMA1 = COMANDO1;
        END;
        IF J = 4 THEN DO;
            READ FILE(ARCH_DATOS) INTO(BLQ_FACT) KEY(DIR);
            CALL RECONSTRUYE(DIRE_PANTA,DIRE_COR,IG,FG,IV,FV,BLQ_FACT.CADENA_FA
T);
            IF J1 ^= 0 THEN
                CALL BUSCA_NUEVA_ACTIVIDAD;
            FACTURA1 = COMANDO1;
        END;
        IF J = 24 THEN DO;
            FG = 10;
            READ FILE(ARCH_DATOS) INTO(BLQ_PED) KEY(DIR);
            POS = 1;
            LON = RANK(SUBSTR(CADENA_PED,POS,1));
            DO WHILE(LON ^= 253 & LON ^= 255);
                POS = POS + LON;
                LON = RANK(SUBSTR(CADENA_PED,POS,1));
            END;
            IF LON ^= 255 THEN DO;
                POS = POS + 1;
                LON = RANK(SUBSTR(CADENA_PED,POS,1));
                CONT_AU_PED_ACT = BINARY(SUBSTR(CADENA_PED,POS+1,LON-1));
            END;
            CALL RECONSTRUYE(DIRE_PANTA,DIRE_COR,IG,FG,IV,FV,BLQ_PED.CADENA_PED
);
            IF J1 ^= 0 THEN
                CALL BUSCA_NUEVA_ACTIVIDAD;
            PEDIDO1 = COMANDO1;
            CONT_AU_PED = CONT_AU;
            CONT_AU = CONT_AU_PED_ACT;
            IT = 1;
            KK = 1;
            GOTO XX;
        END;
        IF J = 23 THEN DO;
            FG = 8;
            READ FILE(ARCH_DATOS) INTO(BLQ_PAG) KEY(DIR);
            CALL RECONSTRUYE(DIRE_PANTA,DIRE_COR,IG,FG,IV,FV,BLQ_PAG.CADENA_PAG
);
            CONT_AU_PAG = CONT_AU;
            IF J1 ^= 0 THEN
                CALL BUSCA_NUEVA_ACTIVIDAD;
            PAG01 = COMANDO1;
            GOTO XX;
        END;

```


CALL BUSCA_NUEVA_ACTIVIDAD;

```

    CHEQUE1 = COMANDO1;
    GOTO TERMINA;
END;
IF J = 38 THEN DO;
    READ FILE(ARCH_DATOS) INTO(BLQ_REC1) KEY(DIR);
    CALL RECONSTRUYE(DIRE_PANTA,DIRE_COR,IG,FG,IV,FV,BLQ_REC1,CADENA RI
I);
    IF J1 ^= 0 THEN
        CALL BUSCA_NUEVA_ACTIVIDAD;
        DO I = 1 TO 9 - LENGTH(COMANDO1);
            COMANDO1 = " " !! COMANDO1;
        END;
        RECIBIR1 = COMANDO1;
    END;
    CLAVE = COMANDO1;
    CALL BUSCA(TESAURO,CLAVE,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));
XX: R = 1;
    X = 14;
    Y = 7;
    X1 = 15;
    Y1 = 9;
    X2 = 12;
    Y2 = 11;
    X3 = 34;
    Y3 = 11;
    X4 = 54;
    Y4 = 11;
REPI_AU:
    CLAVE=COMANDO1!!CHAR(R);
    CALL BUSCA(TESAURO,CLAVE,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,
NODO,PF_CB,VIA,ULT_N,POS_RE,INF);
    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)));
    IF R = 1 THEN DO;
        CLOSE FILE(ARCH_DATOS);
        NUMERO_ARC = 1;
        I = 1;
        DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_ARC);
            I = I + 1;
        END;
        LONGITUD = CONTROL_ARCH(I).LON_REG;
        OPEN FILE(ARCH_DATOS) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) INPUT
DIRECT KEYED ENV(RECSIZE(LONGITUD));
    END;
    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(I).NOMBRE_ARC) INPUT
DIRECT KEYED ENV(RECSIZE(1024));
    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_DADOS) 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
    GOTO XX2;
  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))(A,A,A);
XX2:CALL GOTOXY(X1,Y1);
  PUT FILE(OUT) EDIT(ASCII(29),CLAVE,ASCII(28))(A,A,A);
  CALL BUSCA_POSICION;
  POS2 = POS2 + LON1;
  LON1 = RANK(SUBSTR(CADENA_LEIDA2,POS2,1));
  IF J = 42 THEN DO;
    CONT_AU_PROF = CONT_AU;
    NEJEMP_LE = SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS2+1,LON1-1);
    CALL GOTOXY(X2,Y2);
    PUT FILE(OUT) EDIT(ASCII(29),NEJEMP_LE,ASCII(28))(A,A(9),A);
    POS2 = POS2 + LON1;
    LON1 = RANK(SUBSTR(CADENA_LEIDA2,POS2,1));
    IF AP2 = 1 THEN
      GOTO XX3;
    AND_SUB_LE = SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS2+1,LON1-1);
    CALL GOTOXY(X3,Y3);
    PUT FILE(OUT) EDIT(ASCII(29),AND_SUB_LE,ASCII(28))(A,A(8),A);
    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(ASCII(29),ISBN_LE,ASCII(28))(A,A(20),A);
  END;
  IF J = 4 THEN DO;
    CONT_AU_FACT = CONT_AU;
    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));
    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 = 24 ! J = 23 ! J = 38) THEN DO;
    IF J = 38 THEN
      CONT_AU_REC1 = CONT_AU;
      EJEM_PEDIDO = SUBSTR(BLQ_MONO_PP.CADENA_LEIDA2,POS2+1,LON1-1);
      CALL GOTOXY(X2,Y2);
      PUT FILE(OUT) EDIT(ASCII(29),EJEM_PEDIDO,ASCII(28))(A,A(9),A);
    END;
    IF (CONT_AU ^= 1 & R = 1) THEN DO;
      R = R + 1;
      Y = 13;

```

```

Y2 = 17;
Y3 = 17;
Y4 = 17;
CLOSE FILE(INDICE);
IF J = 38 THEN
    NUMERO_INDICE = 18;
IF J = 42 THEN
    NUMERO_INDICE = 13;
IF J = 4 THEN
    NUMERO_INDICE = 14;
IF J = 24 THEN
    NUMERO_INDICE = 15;
IF J = 23 THEN
    NUMERO_INDICE = 17;
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));
GOTO REPI_AU;
END;
END;
ELSE DO;
    CALL GOTOXY(0,22);
    IF J = 42 THEN
PUT FILE(OUT) EDIT("PROFORMA NO EXISTE",CK)(A,A);
    IF J = 4 THEN
PUT FILE(OUT) EDIT("FACTURA NO EXISTE",CK)(A,A);
    IF J = 24 THEN
PUT FILE(OUT) EDIT("PEDIDO NO EXISTE",CK)(A,A);
    IF J = 3 THEN
PUT FILE(OUT) EDIT("CHEQUE NO EXISTE",CK)(A,A);
    IF J = 23 THEN
PUT FILE(OUT) EDIT("PAGO NO EXISTE",CK)(A,A);
    IF J = 38 THEN
PUT FILE(OUT) EDIT("RECIBIR NO EXISTE",CK)(A,A);
END;
END;

BUSCA_POSICION: PROCEDURE;

IF J = 42 THEN
    COMANDO1_AUX = "PF"!!COMANDO1;
IF J = 4 THEN
    COMANDO1_AUX = "FA"!!COMANDO1;
IF J = 24 THEN
    COMANDO1_AUX = "PD"!!COMANDO1;
IF J = 23 THEN
    COMANDO1_AUX = "PA"!!COMANDO1;
IF J = 38 THEN
    COMANDO1_AUX = "RE"!!COMANDO1;
POS2 = 1;
LON1 = RANK(SUBSTR(BLQ_MONO_PF.CADENA_LEIDA2,POS2,1));
DO WHILE(COMANDO1_AUX ^= SUBSTR(CADENA_LEIDA2,POS2+1,LON1-1));
    POS2 = POS2 + LON1;
    LON1 = RANK(SUBSTR(BLQ_MONO_PF.CADENA_LEIDA2,POS2,1));
END;
END; /*          BUSCA_POSICION          */

/* EN EL CASO DEL PEDIDO AL DAR MAS LINEAS SE SEGUIRAN MOSTRANDO
LOS AUTOK BAJO ESE PEDIDO Y ESE PROVEEDOR.SI SE TERMINAN LOS
AUTORES SE PARA,SI SE DA MAS LINEAS MOSTRARA EL SIGUIENTE

```

PROVEEDOR(SI EXISTE Y SUS RESPECTIVOS AUTORES */

MUESTRA_CHEQUE: PROCEDURE;

```
    READ FILE(ARCH_DATOS) INTO(BLR_CHEQ) KEY(DIR);
    FG = 3;
CALL RECONSTRUYE(DIRE_PANTA,DIRE_COR,IG,FG,IV,FV,BLR_CHEQ,CADENA_CHEQ);
    X = 16;
    Y = 2;
    X1 = 12;
    Y1 = 4;
    X2 = 35;
    Y2 = 6;
    X3 = 54;
    POS = 1;
    LON = RANK(SUBSTR(CADENA_CHEQ,POS,1));
    DO I = 1 TO 3;
        POS = POS + LON;
        LON = RANK(SUBSTR(CADENA_CHEQ,POS,1));
    END;
    CONT_AU = BINARY(SUBSTR(CADENA_CHEQ,POS+1,LON-1));
    I = 1;
    OTRA_VEZ:
    POS = POS + LON;
    LON = RANK(SUBSTR(CADENA_CHEQ,POS,1));
    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)(A);
    CALL GOTOXY(X1,Y1);
    PUT FILE(OUT) EDIT(AUTOR)(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 (I = 1 & CONT_AU ^= 1) THEN DO;
        I = I + 1;
        Y = 9;
        Y1 = 11;
        Y2 = 13;
        GOTO OTRA_VEZ;
    END;
    CONT_AU_CHEQ = CONT_AU;

END; /*          MUESTRA_CHEQUE          */
```

BUSCA_NUEVA_ACTIVIDAD: PROCEDURE;

```
IF J = 38 THEN DO;
    POS = 1;
    LON = RANK(SUBSTR(CADENA_REC1,POS,1));
    COMANDO1 = SUBSTR(CADENA_REC1,POS+1,LON-1);
END;
IF J = 42 THEN DO;
    POS = 1;
```

```

LON = RANK(SUBSTR(CADENA_PROF, POS, 1));
POS = POS + LON;
LON = RANK(SUBSTR(CADENA_PROF, POS, 1));
COMANDO1 = SUBSTR(CADENA_PROF, POS+1, LON-1);
END;
IF J = 4 THEN DO;
  POS = 1;
  LON = RANK(SUBSTR(CADENA_FACT, POS, 1));
  POS = POS + LON;
  LON = RANK(SUBSTR(CADENA_FACT, POS, 1));
  COMANDO1 = SUBSTR(CADENA_FACT, POS+1, LON-1);
END;
IF J = 24 THEN DO;
  POS = 1;
  LON = RANK(SUBSTR(CADENA_PED, POS, 1));
  COMANDO1 = SUBSTR(CADENA_PED, POS+1, LON-1);
END;
IF J = 3 THEN DO;
  POS = 1;
  LON = RANK(SUBSTR(CADENA_CHEQ, POS, 1));
  COMANDO1 = SUBSTR(CADENA_CHEQ, POS+1, LON-1);
END;
IF J = 23 THEN DO;
  POS = 1;
  LON = RANK(SUBSTR(CADENA_PAG, POS, 1));
  COMANDO1 = SUBSTR(CADENA_PAG, POS+1, LON-1);
END;

END; /* BUSCA_NUEVA_ACTIVIDAD */

ADELANTA: PROCEDURE;

/* RUTINA QUE ADELANTA EN UN INDICE INDICADO */

IF POS_RE = NC - 1 THEN DO;
  IF VECINA_D ^= 0 THEN DO;
    POS_ABS = 1;
    POS_RE = 1;
    READ FILE(TESAURO) INTO(NODO) KEY(VECINA_D);
  END;
  ELSE DO;
    CALL GOTOXY(0, 22);
    IF J = 42 THEN
      PUT FILE(OUT) EDIT("ULTIMA PROFORMA DEL ARCHIVO", CK)(A, A);
    IF J = 4 THEN
      PUT FILE(OUT) EDIT("ULTIMA FACTURA DEL ARCHIVO", CK)(A, A);
    IF J = 24 THEN
      PUT FILE(OUT) EDIT("ULTIMO PEDIDO DEL ARCHIVO", CK)(A, A);
    IF J = 3 THEN
      PUT FILE(OUT) EDIT("ULTIMO CHEQUE DEL ARCHIVO", CK)(A, A);
    IF J = 23 THEN
      PUT FILE(OUT) EDIT("ULTIMO PAGO DEL ARCHIVO", CK)(A, A);
    IF J = 38 THEN
      PUT FILE(OUT) EDIT("ULTIMO RECIBIDO DEL ARCHIVO", CK)(A, A);
    GOTO TERMINA;
  END;
  END;
  ELSE DO;
    POS_ABS = POS_ABS + LON_CM;
    POS_RE = POS_RE + 1;
  END;

END;

END; /* ADELANTA */

```

RETROCEDE: PROCEDURE;

/* RUTINA QUE RETROCEDE EN UN INDICE INDICADO */

```
POS_RE_A = POS_RE - 1;
IF POS_RE = 1 THEN DO;
  IF VECINA_I ^= 0 THEN DO;
    READ FILE(TESAURO) INTO(NODO) KEY(VECINA_I);
    POS_ABS = 1;
    POS_RE = 1;
    LON_CM = RANK(SUBSTR(ENTRADA,POS_ABS+1,1));
    DO WHILE(POS_RE < NC);
      POS_ABS = POS_ABS + LON_CM;
      LON_CM = RANK(SUBSTR(ENTRADA,POS_ABS+1,1));
      POS_RE = POS_RE + 1;
    END;
  END;
ELSE DO;
  CALL GOTOXY(0,22);
  IF J = 42 THEN
  PUT FILE(OUT) EDIT("PRIMERA PROFORMA DEL ARCHIVO",CK)(A,A);
  IF J = 4 THEN
  PUT FILE(OUT) EDIT("PRIMERA FACTURA DEL ARCHIVO",CK)(A,A);
  IF J = 24 THEN
  PUT FILE(OUT) EDIT("PRIMER PEDIDO DEL ARCHIVO",CK)(A,A);
  IF J = 3 THEN
  PUT FILE(OUT) EDIT("PRIMER CHEQUE DEL ARCHIVO",CK)(A,A);
  IF J = 23 THEN
  PUT FILE(OUT) EDIT("PRIMER PAGO DEL ARCHIVO",CK)(A,A);
  IF J = 38 THEN
  PUT FILE(OUT) EDIT("PRIMER RECIBIDO DEL ARCHIVO",CK)(A,A);
  GOTO TERMINA;
END;
END;
ELSE DO;
  POS_ABS = 1;
  POS_RE = 1;
  LON_CM = RANK(SUBSTR(ENTRADA,POS_ABS+1,1));
  DO WHILE(POS_RE < POS_RE_A);
    POS_ABS = POS_ABS + LON_CM;
    LON_CM = RANK(SUBSTR(ENTRADA,POS_ABS+1,1));
    POS_RE = POS_RE + 1;
  END;
END;
END;
/* RETROCEDE */

ZZ:
TERMINA:

END; /* BUSCA_ACTIVIDAD */
```

CONSULTA_REINTEGRO: PROCEDURE;

```
/*
*   RUTINA QUE MUESTRA EL PRIMER REINTEGRO DEL ARCHIVO
*
*   PARAMETROS:
*
*   VARIABLES:
*       CONTADOR_REIN: REINTEGRO EN MEMORIA
*       DIRE_PANTA: DIRECCION DE LA PANTALLA
*       CONSULTA REINTEGRO EN EL ARCHIVO
*       PANTALLAS
*       DIRE_COR: DIRECCION EN EL ARCHIVO COODENADAS
*       DE LAS COORDENADAS DE LOS DATOS
*       IG: INICIO DE LAS COORDENADAS
*       FG: FIN DE LAS COORDENADAS
*
*   LLAMADO POR:
*       SELECCION
*
*   LLAMA A:
*       RECONSTRUYE
*       GOTOXY
*
*   ARCHIVOS:
*       PANTALLAS: CONTIENE LAS PANTALLAS DEL SISTEMA
*       COORDENADAS: CONTIENE LAS COORDENADAS DE LOS
*       DATOS DE LAS PANTALLAS
*       REINTEGROS: CONTIENE LOS REINTEGROS
*
*/
```

```
DCL
CADENA_LEIDA CHAR(1000) VAR,
CONTADOR_REIN BIN FIXED EXT STATIC,
(DIRE_PANTA,DIRE_COR,IG,FG,IV,FV,I) BIN FIXED,
(REINTEGRO,IN,OUT,PANTALLAS,COORDENADAS) FILE,
01 BLQ_REIN,
   02 CADENA_REIN CHAR(500) VAR,
   02 BLA BIN FIXED,
   02 BLS BIN FIXED,
   02 OBSERVACION_REIN(3) CHAR(70) VAR,
RECONSTRUYE ENTRY(BIN FIXED,BIN FIXED,BIN FIXED,BIN FIXED,BIN FIXED,
BIN FIXED,CHAR(1000) VAR),
GOTOXY ENTRY(BIN FIXED,BIN FIXED);
```

```
CONTADOR_REIN = 1;
DIRE_PANTA = 10;
DIRE_COR = 30;
IV = 1;
FV = 100;
IG = 1;
FG = 100;
READ FILE(REINTEGRO) INTO(BLQ_REIN) KEY(1);
CADENA_LEIDA = CADENA_REIN;
CALL RECONSTRUYE(DIRE_PANTA,DIRE_COR,IG,FG,IV,FV,CADENA_LEIDA);

DO I = 1 TO 3;
   CALL GOTOXY(0,18+I);
   PUT FILE(OUT) EDIT(OBSERVACION_REIN(I))(A(70));
END;

END; /* CONSULTA_REINTEGRO */
```


BUSCA_AU_PROV_PP: PROCEDURE(J,K,COMANDO1);

/*****

* BUSCA_AU_PROV_PP: *

* PROGRAMA QUE SE ENCARGA DE BUSCAR EL AUTOR/PP O PROVEEDOR, *
* VA A EL ARCHIVO CREA_PANTALLAS Y SUBSTRAE LA DIRECCION DE *
* LA PANTALLA Y LA DIRECCION DE LAS COORDENADAS DE LA PANTA- *
* LLAS EN EL ARCHIVO PANTALLAS Y COORDENADAS RESPECTIVAMENTE *
* TAMBIEN SUBSTRAE EL NUMERO DEL INDICE CORRESPONDIENTE AL *
* ARCHIVO DE INFORMACION, LUEGO VA A TABLA DE ARCHIVOS Y *
* BUSCA LA INFORMACION NECESARIA PARA ABIR LOS ARCHIVOS *
* DE INFORMACION E INDICE *

* PARAMETROS: *

* J: INDICA SI SE QUIERE CONSULTAR UN AUTOR, PP O *

* PROVEEDOR *

* K: INDICADOR DE INTERUPCION DEL ACTIVIDAD *

* 0: PROCESO INTERUMPIDO *

* ^0: NO INTERUMPIDO *

* COMANDO1: COMANDO SECUNDARIO QUE CONTIENE EL *
* NOMBRE DEL PROVEEDOR, PUBLICACION *
* O REVISTA *

* VARIABLES: *

* CONTADOR_PROV: PROVEEDOR EN MEMORIA *

* CONTADOR_PP: PP EN MEMORIA *

* CONTADOR_AU: AUTOR EN MEMORIA *

* NUMERO_INDICE: NUMERO DEL INDICE DEL AUTOR/PP O *
* PROVEEDOR EN CONSULTA *

* NUMERO_ARC: NUMERO DEL ARCHIVO DEL AUTOR/PP O *
* PROVEEDOR EN CONSULTA *

* CLAVEL: PROVEEDOR EN MEMORIA *

* AUTORI: AUTOR EN MEMORIA *

* PPI: PUBLICACION EN MEMORIA *

* J1: INDICA LO QUE SE VA A CONSULTAR : EL SIGUIENTE *
* O ANTERIOR AUTOR/PP O PROVEEDOR EN EL ARCHIVO *
* DE AUTORES PP O PROVEEDORES RESPECTIVAMENTE *

* LLAMADA POR: *

* SELECCION *

* LLAMA A: *

* BUSCA *

* GOTOXY *

* RECONSTRUYE *

* ARCHIVOS: *

* **


```

01 BLQ_MONO_PP,
  02 CADENA_LEIDA2 CHAR(1000) VAR,
01 EST_LEC,
  02 VECTOR_COOR(500) CHAR(1),
TESAURO FILE VARIABLE,
  BA_ENC BIT(1),
(POS_ABS,PF_CM,LON_CM,PF_CB,ULT_N,POS_RE,VIA(5),CL) BIN FIXED,
(X,Y) BIN FIXED;
DCI
(CLAVE,RES_CM) CHAR(120) VAR,
(CLAVE1,AUTOR1,PF1,CLAVE2) CHAR(120) VAR EXT STATIC,
DIR DEC 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_PANTA(16),
  02 NUM_ARC BIN FIXED,
  02 NUM_INDICE BIN FIXED,
  02 DIRE_P BIN FIXED,
  02 DIRE_C1 BIN FIXED,
  02 NOMBRE_P CHAR(4) 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;
DCI
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),
RECONSTRUYE ENTRY(BIN FIXED,BIN FIXED,BIN FIXED,BIN FIXED,BIN FIXED,
BIN FIXED,CHAR(1000) VAR);

CK = ASCII(11);
OB = 0;
CONTADOR_TIT = 0;
TITULO_AUX = 1;
CONTADOR_REIN = 0;
CONTADOR_PR = 0;
CONTADOR_PF = 0;
CONTADOR_AU = 0;
CONT_AU_PROF = 0;
CONT_AU_PED = 0;
PED_AUX = 1;
CONT_AU_CHEQ = 0;
CHEQ_AUX = 1;
CONT_AU_FACT = 0;

```

```
CONT_AU_RECT = 0;
CONT_AU_PAG = 0;
PROF_AUX = 1;
FACT_AUX = 1;
PAG_AUX = 1;
RECT_AUX = 1;
JI = 0;
TESAURO = INDICE;
FG = 16;
CI = 0;
ENCONTRE="0"B;
NUMERO_ARC = 2;
```

```
READ FILE(CONTROL_ARCHIVO) INTO(CONTROL_ARCH) KEY(1);
READ FILE(CONTROL_PANTALLAS) INTO(CONTROL_PANTA) KEY(1);
```

```
IF (J=45 ! J=47) THEN DO;
  JI = J;
  J = 43;
```

```
END;
```

```
IF (J=28 ! J=31) THEN DO;
  JI = J;
  J = 25;
```

```
END;
```

```
IF (J=7 ! J=10) THEN DO;
  JI = J;
  J = 14;
```

```
END;
```

```
IF J ^= 43 THEN
  NUMERO_ARC = 1;
```

```
I = 1;
```

```
DO WHILE(CONTROL_PANTA(I).NUM_ARC ^= NUMERO_ARC);
  I = I + 1;
```

```
END;
```

```
IF J = 25 THEN DO;
  I = I + 1;
  FG = 13;
```

```
END;
```

```
DIRE_PANTA = CONTROL_PANTA(I).DIRE_P;
```

```
NUMERO_INDICE = CONTROL_PANTA(I).NUM_INDICE;
```

```
/* BUSCA PROV TIENE UNA DIR DE COORDENADAS Y CREA_PROV TIENE OTRA */
```

```
IF J = 43 THEN
```

```
  DIRE_COR = 18;
```

```
ELSE DO;
```

```
/*DIRE_COR = CONTROL_PANTA(I).DIRF_C1;*/
```

```
  IF J = 25 THEN
```

```
    DIRE_COR = 21;
```

```
  ELSE
```

```
    DIRE_COR = 20;
```

```
END;
```

```
FV = 100;
```

```
IG = 1;
```

```
IV = 1;
```

```
IF (JI ^= 0 & JI ^= 45 & JI ^= 47) THEN DO;
```

```
  CALL NUEVE_AU_PP;
```

```
  GOTO TERMINA;
```

```
END;
```

```
I = 1;
```

```
A: IF (J ^= 43 & COMANDO1 = "") THEN
```

```
  NUMERO_INDICE = 12;
```

```
DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_INDICE);
```

```
  I = I + 1;
```

```
END;
```

```

CLOSE FILE(INDICE);
OPEN FILE(INDICE) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) INPUT
DIRECT KEYED ENV(RECSIZE(1024));
  I = 1;
  DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_ARC);
    I = I + 1;
  END;
LONGITUD = CONTROL_ARCH(I).LON_RFG;
CLOSE FILE(ARCH_DATOS);
OPEN FILE(ARCH_DATOS) TITLE(CONTROL_ARCH.NOMBRE_ARC(I))
INPUT DIRECT KEYED ENV(RECSIZE(LONGITUD));
IF COMANDO1 ^= "" THEN
  CLAVE = COMANDO1;
/* MUESTRA EL PROVEEDOR QUE ESTA EN MEMORIA O EL PRIMER AU/PP */
ELSE
  CLAVE = CLAVE1;
CALL BUSCA(TESAURO,CLAVE,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,
NODO,PF_CB,VIA,ULT_N,POS_RE,INF);
IF (J ^= 43 & COMANDO1 = "") THEN DO;
  CONT_AU = RANK(SUBSTR(ENTRADA,POS_ABS+5,1))*256 +
  RANK(SUBSTR(ENTRADA,POS_ABS+6,1));
  I = 0;
  BA_ENC = "0"B;
  DO WHILE(ENCONTRE="0"B & CL<CONT_AU);
    DO WHILE(BA_ENC = "0"B);
      I = I + 1;
      CLAVE = CLAVE1 !! CHAR(I);
      CALL BUSCA(TESAURO,CLAVE,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,
      NODO,PF_CB,VIA,ULT_N,POS_RE,INF);
    END;
    APLINV2 = DEC(RANK(SUBSTR(ENTRADA,POS_ABS+7,1))*10000 +
    RANK(SUBSTR(ENTRADA,POS_ABS+8,1))*100 +
    RANK(SUBSTR(ENTRADA,POS_ABS+9,1)));
    IF ((APLINV2=0 & J=25) ! (APLINV2=1 & J=14)) THEN DO;
      ENCONTRE="1"B;
      NUM_AU = RANK(SUBSTR(ENTRADA,POS_ABS+5,1))*256 +
      RANK(SUBSTR(ENTRADA,POS_ABS+6,1));
      NUMERO_INDICE = 11;
      I = 1;
      DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_INDICE);
        I = I + 1;
      END;
      CLOSE FILE(INDICE);
      OPEN FILE(INDICE) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) INPUT
      DIRECT KEYED ENV(RECSIZE(1024));
      CLAVF = 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);
    END;
    CL = CL + 1;
    BA_ENC = "0"B;
  END;
  IF ENCONTRE="0"B THEN DO;
    CALL GOTEXY(0,22);
    IF J=25 THEN
      PUT FILE(OUT) EDIT("NO HAY PP BAJO EL PROVEEDOR: ",CLAVE1,CK)(A,A,A);
    ELSE
      PUT FILE(OUT) EDIT("NO HAY AUTOR BAJO EL PROVEEDOR: ",CLAVE1,CK)(A,A,A);
    END;
  END;
END;
IF (J=43 ! ENCONTRE="1"B ! COMANDO1 ^= "") THEN DO;
  IF J1 = 45 THEN DO;
    K = 0;

```

```

I = 0;
CONT_AU = RANK(SUBSTR(ENTRADA,POS_ABS+5,1))*256 +
RANK(SUBSTR(ENTRADA,POS_ABS+6,1));
DO WHILE(I < CONT_AU);
  BA_ENC = "0"B;
  DO WHILE(BA_ENC = "0"B);
    K = K + 1;
    CLAVE2 = CLAVE !! CHAR(K);
    CALL BUSCA(TESAURO,CLAVE2,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,
NODO,PF_CB,VIA,ULT_N,POS_RE,INF);
  END;
  I = I + 1;
END;
CALL ADELANTA;
END;
IF J1 = 47 THEN
  CALL RETROCEDE;
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 J = 43 THEN DO;
  FG = 100;
  READ FILE(ARCH_DATOS) INTO(BLR_PROV) KEY(DIR);
  POS = 1;
  LON = RANK(SUBSTR(CADENA_LEIDA1,POS,1));
  CLAVE = SUBSTR(CADENA_LEIDA1,POS+1,LON-1);
  POS = POS + LON;
  LON = RANK(SUBSTR(CADENA_LEIDA1,POS,1)) - 1;
  IF LON ^= 0 THEN
    CLAVE = CLAVE !! " " !! SUBSTR(CADENA_LEIDA1,POS+1,LON);
  CLAVE1 = CLAVE;
  CALL RECONSTRUYE(DIRE_PANTA,DIRE_COR,IG,FG,IV,FV,BLR_PROV,CADENA_LEIDA1);
  CONTADOR_PR = 1;
END;
ELSE DO;
  APLINV2 = DEC(RANK(SUBSTR(ENTRADA,POS_ABS+7,1))*10000 +
RANK(SUBSTR(ENTRADA,POS_ABS+8,1))*100 +
RANK(SUBSTR(ENTRADA,POS_ABS+9,1)));
/* IF ((APLINV2=0 & J=25) ! (APLINV2=1 & J=14)) THEN DO; */
  READ FILE(ARCH_DATOS) INTO(BLR_MONO_PP) KEY(DIR);
  POS = 1;
  LON = RANK(SUBSTR(CADENA_LEIDA2,POS,1));
  CLAVE = SUBSTR(CADENA_LEIDA2,POS+1,LON-1);
  POS = POS + LON;
  LON = RANK(SUBSTR(CADENA_LEIDA2,POS,1)) - 1;
  IF LON ^= 0 THEN
    CLAVE = CLAVE !! " " !! SUBSTR(CADENA_LEIDA2,POS+1,LON);
/* IF J = 14 THEN DO; */
  IF APLINV2 = 1 THEN DO;
    DIRE_PANTA = 2;
    DIRE_COR = 20;
    FG = 16;
    AUTOR1 = CLAVE;
    CONTADOR_AU = 1;
    CLAVE_AU_PP = AUTOR1;
  END;
ELSE DO;
  DIRE_PANTA = 3;
  DIRE_COR = 21;
  FG = 13;
  PP1 = CLAVE;
  CONTADOR_PP = 1;
  CLAVE_AU_PP = PP1;

```

```

        END;
        CALL RECONSTRUYE(DIRE_PANTA,DIRE_COR,IG,FG,IV,FV,BLQ_MONO_PP,(CADENA_I F
DA2));
/*      END;
      ELSE DO;
        CALL GOTOXY(0,22);
        IF J = 25 THEN
          PUT FILE(OUT) EDIT("PP NO EXISTE",CK)(A,A);
        ELSE
          PUT FILE(OUT) EDIT("AUTOR NO EXISTE ",CK)(A,A);
        END; */
      END;
    END;
  END;

```

```

/*****
                                     SUBROUTINAS
*****/

```

```

MUEVE_AU_PP: PROCEDURE;

```

```

  I = 1;
  DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_INDICE);
    I = I + 1;
  END;
  CLOSE FILE(INDICE);
  OPEN FILE(INDICE) TITLE(CONTROL_ARCH(I).NOMBRE_ARCH) INPUT
  DIRECT KEYED ENV(RECSIZE(1024));
  I = 1;
  DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_ARCH);
    I = I + 1;
  END;
  LONGITUD = CONTROL_ARCH(I).LON_REG;
  CLOSE FILE(ARCH_DATOS);
  OPEN FILE(ARCH_DATOS) TITLE(CONTROL_ARCH(I).NOMBRE_ARCH)
  INPUT DIRECT KEYED ENV(RECSIZE(LONGITUD));
  /*IF J = 14 THEN
    CLAVE = AUTORI;
  ELSE
    CLAVE = PP1; */
  CLAVE = CLAVE_AU_PP;
  CLAVE_E = CLAVE_AU_PP;
  CALL BUSCA(TESAURO,CLAVE,BA_ENC,POS_ABS,RES_CM,PF_CM,LON_CM,
  NODO,PF_CB,VIA,ULT_N,POS_RE,INF);
  IF (J1=7 ! J1=28) THEN
    CALL ADELANTA;
  IF (J1=10 ! J1=31) THEN
    CALL RETROCEDE;
  APLINV2 = DEC(RANK(SUBSTR(ENTRADA,POS_ABS+7,1))*10000 +
  RANK(SUBSTR(ENTRADA,POS_ABS+8,1))*100 +
  RANK(SUBSTR(ENTRADA,POS_ABS+9,1)));
  /*DO WHILE(ENCONTRE = "0"B);
    IF ((APLINV2=0 & J=25) ! (APLINV2=1 & J=14)) THEN
      ENCONTRE = "1"B;
    ELSE DO;
      LON_CM = RANK(SUBSTR(ENTRADA,POS_ABS+1,1));
      IF (J1=7 ! J1=28) THEN
        CALL ADELANTA;
      IF (J1=10 ! J1=31) THEN
        CALL RETROCEDE;
      APLINV2 = DEC(RANK(SUBSTR(ENTRADA,POS_ABS+7,1))*10000 +
      RANK(SUBSTR(ENTRADA,POS_ABS+8,1))*100 +
      RANK(SUBSTR(ENTRADA,POS_ABS+9,1)));
    END;
  END; */
  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;
LON = RANK(SUBSTR(CADENA_LEIDA2,POS,1));
CLAVE = SUBSTR(CADENA_LEIDA2,POS+1,LON-1);
POS = POS + LON;
LON = RANK(SUBSTR(CADENA_LEIDA2,POS,1)) - 1;
IF LON ^= 0 THEN
  CLAVE = CLAVE !! " " !! SUBSTR(CADENA_LEIDA2,POS+1,LON);
IF APLINV2 = 1 THEN DO;
  DIRE_PANTA = 2;
  DIRE_COR = 20;
  FG = 16;
  AUTORI = CLAVE;
  CONTADOR_AU = 1;
  CLAVE_AU_PP = AUTORI;
END;
IF APLINV2 = 0 THEN DO;
  DIRE_PANTA = 3;
  DIRE_COR = 21;
  FG = 13;
  PP1 = CLAVE;
  CONTADOR_PP = 1;
  CLAVE_AU_PP = PP1;
END;
CALL RECONSTRUYE(DIRE_PANTA,DIRE_COR,IG,FG,IV,FV,BLQ_MONO_PP,CADENA_LEIDA2);
END;      /*  MUEVE_AU_PP      */

ADELANTA: PROCEDURE;

/*  RUTINA QUE ADELANTA EN UN INDICE INDICADO  */

IF POS_RE = NC - 1 THEN DO;
  IF VECINA_D ^= 0 THEN DO;
    POS_ABS = 1;
    POS_RE = 1;
    READ FILE(TESAURO) INTO(NODO) KEY(VECINA_D);
  END;
  ELSE DO;
    CALL GOTOXY(0,22);
    IF J = 43 THEN
      PUT FILE(OUT) EDIT("ULTIMO PROVEEDOR DEL ARCHIVO",CK)(A,A);
    IF J = 14 THEN
      PUT FILE(OUT) EDIT("ULTIMO AUTOR DEL ARCHIVO",CK)(A,A);
    IF J = 25 THEN
      PUT FILE(OUT) EDIT("ULTIMO PP DEL ARCHIVO",CK)(A,A);
    GOTO TERMINA;
  END;
END;
ELSE DO;
  POS_ABS = POS_ABS + LON_CK;
  POS_RE = POS_RE + 1;
END;

END;      /*  ADELANTA      */

RETROCEDE: PROCEDURE;

/*  RUTINA QUE RETROCEDE EN UN INDICE INDICADO  */
POS_RE_A = POS_RE - 1;
/*PUT FILE(OUT) SKIP EDIT("POS_RE= ",POS_RE)(A,A);*/

```



```

IF POS_RE = 1 THEN DO;
  IF VECINA_I ^= 0 THEN DO;
    READ FILE(TESAURO) INTO(NODO) KEY(VECINA_I);
    POS_ABS = 1;
    POS_RE = 1;
    LON_CM = RANK(SUBSTR(ENTRADA,POS_ABS+1,1));
    DO WHILE(POS_RE < NC - 1);
      POS_ABS = POS_ABS + LON_CM;
      LON_CM = RANK(SUBSTR(ENTRADA,POS_ABS+1,1));
      POS_RE = POS_RE + 1;
    END;
  END;
ELSE DO;
  CALL GOTOXY(0,22);
  IF J = 43 THEN
    PUT FILE(OUT) EDIT("PRIMER PROVEEDOR DEL ARCHIVO",CK)(A,A);
  IF J = 14 THEN
    PUT FILE(OUT) EDIT("PRIMER AUTOR DEL ARCHIVO",CK)(A,A);
  IF J = 25 THEN
    PUT FILE(OUT) EDIT("PRIMERA PP DEL ARCHIVO",CK)(A,A);
  GOTO TERMINA;
END;
END;
ELSE DO;
  POS_ABS = 1;
  POS_RE = 1;
  LON_CM = RANK(SUBSTR(ENTRADA,POS_ABS+1,1));
  DO WHILE(POS_RE < POS_RE_A);
    POS_ABS = POS_ABS + LON_CM;
    LON_CM = RANK(SUBSTR(ENTRADA,POS_ABS+1,1));
    POS_RE = POS_RE + 1;
  END;
END;
END;

END; /* RETROCEDE */

TERMINA:

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

END; /* BUSCA_AU_PROV_PP */

```



```

02 NUM_PAG_ BIN FIXED,
REALIZA_CARTA_ENTRY(BIN FIXED,BIN FIXED,(15)CHAR(20) VAR,
CHAR(12) VAR),
BUSCA_BANDA_ENTRY(BIN FIXED,BIT(1)),
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),
INSERTA_ENTRY(FILE VARIABLE,CHAR(120) VAR,DEC FIXED,BIN FIXED,DEC FIXED),
ACTIVIDAD_LEIDA CHAR(12) VAR,
NUMERO_ACTIVIDAD CHAR(12) VAR,
AUXILIAR BIN FIXED,
CK CHAR(1),
(AFL INVL,AFL INV2) DEC FIXED,
DIR DEC FIXED,
(BLOQUEADO_ON,BLOQUEADO_AC,AC,ON,MAS,MENOS,AD,RE,
PRI,PRI AC,MAS PED) BIT(1),
BA_ENC BIT(1),
(POR_ABS,POR,PE_CM,ON_CM,PE_CB,ULT_N,POR_RE,VIA(5)) BIN FIXED,
PIDE_VACTIL_ENTRY(CHAR(20) VAR,DEC FIXED),
DIRECCION DEC FIXED,
NUMERO_ARCHIVO CHAR(20) VAR,
01 REQ_PED,
02 CADENA_PED CHAR(1000) VAR,
01 REQ_CHEQ,
02 CADENA_CHEQ CHAR(1000) VAR,
TCOR 2(10) CHAR(20) STATIC INITIAL
"AC",
"ON",
"AD",
"RE",
"+",
"-",
"PD",
"DI",
"NO",
"FI");
TEXTIO = "O"R;
DB = 0;
CONTADOR_TIT = 0;
TITULO_AUX = 1;
CONTADOR_REFIN = 0;
CONTADOR_NOTA = 0;
CONTADOR_FR = 0;
CONTADOR_PP = 0;
CONTADOR_AI = 0;
CONT_AI_PROB = 0;
CONT_AI_PED = 0;
PED_AUX = 1;
CONT_AI_CHEQ = 0;
CHEQ_AUX = 1;
CONT_AI_FACT = 0;
CONT_AI_PAG = 0;
CONT_AI_RECT = 0;
FRON_AUX = 1;
FACT_AUX = 1;
PAG_AUX = 1;
RECT_AUX = 1;
BLOQUEADO_AC = "O"R;
BLOQUEADO_ON = "O"R;
AC = "O"R;

```

```

OM = "O"R;
MAÑ = "1"R;
MENOS = "1"R;
AD = "1"R;
RF = "1"R;
PRI = "0"R;
PRI_AC = "0"R;
BAS_FED = "1"R;

FECHA = DATE();
FECHA = SUBSTR(FECHA,5,2) || "/" || SUBSTR(FECHA,8,2) || "/" || SUBSTR(FECHA,1,2);
OK = ASCII(11);
NUM_CHEQ = 0;
NUM_B = 0;
J = 0;
CONT_PRODU = 0;
GUAR_J = 0;
CONT_PROD_ACEP = 0;
TESAURO = "INDICE";

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

    NUM_CHEQ = NUM_CHEQ + 1;
    WRITE FILE(CONTROL_ARCHIVO) FROM(CONTROL_NUMERICO) KEYFROM(2);
    NUMERO_ACTIVIDAD = NUM_CHEQ;
    CALL BOTOXY(14,0);
    PUT FILE(IND) EDIT(ASCII(29),NUM_CHEQ,ASCII(28))(A,A,A);
    CALL BOTOXY(40,0);
    PUT FILE(IND) EDIT(ASCII(29),FECHA,ASCII(28))(A,A,A);
REF: CALL BOTOXY(47,0);
READ FILE(IN) INTO(BAN);
IF VERIFY(BAN,"12345A/890") ^= 0 THEN DO;
    B = 3;
    PP = 47;
    L = 0;
    CALL BORRA;
    GO TO REF;
END;
CALL BUSCA_BANCO(ETHARY(BAN),ENCONTRADO);
IF ENCONTRADO = "0"R THEN DO;
    B = 3;
    PP = 47;
    L = 0;
    CALL BORRA;
    GO TO REF;
END;
CADENA_CHEQ = ASCII(LENGTH(CHAR(NUM_CHEQ))+1) || CHAR(NUM_CHEQ) ||
ASCII(LENGTH(BAN)+1) || BAN || ASCII(LENGTH(FECHA)+1) || FECHA;
NUMERO_ARCH = 5;
NUMERO_INDICE = 15;
CALL ARRE_ARCHIVOS;
END;
J = 0;
CALL BOTOXY(14,13);
READ FILE(IN) INTO(ACTIVIDAD_LETDA);
IF ACTIVIDAD_LETDA = "" THEN DO;
    BLOQUEADO_AC = "1"R;
    BLOQUEADO_DM = "1"R;
    BAS_FED = "0"R;
    GO TO AA;
END;
/* MOPRA LIZACION A NUEVE CARACTERES DEL NUMERO DE ACTIVIDAD */

```

```

DO I = 1 TO 9 = LENGTH(CACTUIDAD_LETDA);
  ACTIVIDAD_LETDA = ' 'ACTIVIDAD_LETDA;
END;
CLAVE = ACTIVIDAD_LETDA;
CALL BUSCACTESAURIO,CLAVE,RA_FNC,POS_ABS,RES_CM,FE_CM,LON_CM,MODD,
PE_DB,DTA,DTI,UN,POS_RF,INF);
IF RA_FNC = '0'R THEN DO;
  RLDORFADD_AC = '1'R;
  RLDORFADD_OM = '1'R;
  RA5 = '1'R;
  REMOS = '1'R;
  AD = '1'R;
  RE = '1'R;
  RA5_FED = '0'R;
  CALL GOTOXY(0,22);
  PUT FILE(OUT) EDIT('NO EXISTE')(A);
  GOTO AA;
END;
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(BLD_FED) KEY(DIR);
CALL GOTOXY(0,20);
CONT_PROV = 0;
POS = 1;
LON = RANK(SUBSTR(CADENA_PED,POS,1));
POS = POS + LON;
LON = RANK(SUBSTR(CADENA_PED,POS,1));
REPEAT;
CONT_PROV = CONT_PROV + 1;
VECTOR_PROD(CONT_PROV) = SUBSTR(CADENA_PED,POS+1,LON-1);
DO I = 1 TO 3;
  POS = POS + LON;
  LON = RANK(SUBSTR(CADENA_PED,POS,1));
END;
VECTOR_PAIS(CONT_PROV) = SUBSTR(CADENA_PED,POS+1,LON-1);
POS = POS + LON;
LON = RANK(SUBSTR(CADENA_PED,POS,1));
DO WHILE(LON ^= 253 & LON ^= 250);
  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));
  POS = POS + LON + 1;
  LON = RANK(SUBSTR(CADENA_PED,POS,1));
  POS_AUX = POS + LON;
  LON_AUX = RANK(SUBSTR(CADENA_PED,POS_AUX,1));
  POS_AUX = POS_AUX + LON_AUX;
  LON_AUX = RANK(SUBSTR(CADENA_PED,POS_AUX,1));
  IF LON_AUX ^= 255 THEN
    GOTO REPEAT;
END;
CONTINUE;
I = I + 1;
CALL MUESTRA_PROVEEDOR;
AA;DO I = 1 TO 2;
  CALL GOTOXY(A3-1,18);
  PUT FILE(OUT) EDIT('')(A(1));
END;
CALL GOTOXY(A1,18);
READ FILE(IN) INTO(RESXP);

```



```

IF P = 10 THEN DO;
  IF (P = 1 & BLOQUEADO_AC = '0'R) THEN DO;
    IF PRI_AC = '0'R THEN DO;
      PRI_AC = '1'B;
      MAS = '1'R;
      MENOS = '0'R;
      AD = '1'R;
      RF = '0'R;
    END;
    CALL ACEPTAR;
    IF (J < CONT_PROV & AC = '0'B) THEN
      GOTO CONTINUE;
    IF J = CONT_PROV THEN DO;
      BLOQUEADO_AC = '1'R;
      BLOQUEADO_OM = '1'R;
    END;
  END;
  IF (P = 2 & BLOQUEADO_OM = '0'R) THEN DO;
    IF OM = '1'R THEN DO;
      IF AUXILIAR < CONT_PROV_ACEP THEN DO;
        DO WHILE(AUXILIAR < CONT_PROV_ACEP);
          VECTOR_GUAR_PROV(AUXILIAR) = VECTOR_GUAR_PROV(AUXILIAR+1);
          VECTOR_GUAR_PATS(AUXILIAR) = VECTOR_GUAR_PATS(AUXILIAR+1);
          VECTOR_MONTO_A(AUXILIAR) = VECTOR_MONTO_A(AUXILIAR+1);
          VECTOR_MONTO_B(AUXILIAR) = VECTOR_MONTO_B(AUXILIAR+1);
          AUXILIAR = AUXILIAR + 1;
        END;
        CONT_PROV_ACEP = CONT_PROV_ACEP - 1;
        AUXILIAR = AUXILIAR - 1;
        BLOQUEADO_OM = '1'R;
        IF CONT_PROV_ACEP = 0 THEN
          AUXILIAR = AUXILIAR + 1;
        END;
      IF ((J < CONT_PROV) & OM = '0'R) THEN
        GOTO CONTINUE;
      IF J = CONT_PROV THEN DO;
        BLOQUEADO_AC = '1'R;
        BLOQUEADO_OM = '1'R;
      END;
    END;
  END;
  IF (P = 3 & AD = '0'R) THEN DO;
    IF J > GUAR_J THEN
      AD = '1'R;
    ELSE
      J = J + 1;
    CALL MUESTRA_PROVEEDOR;
  END;
  IF (P = 4 & RF = '0'R) THEN DO;
    AD = '0'R;
    MAS = '1'R;
    MENOS = '1'R;
    AC = '1'R;
    BLOQUEADO_OM = '1'R;
    BLOQUEADO_AC = '0'R;
    CALL RETRUCEDIR;
  END;
  IF (P = 5 & MAS = '0'R) THEN DO;
    BLOQUEADO_OM = '0'R;
    CALL MAS_NUEVA_ACTIVIDAD;
  END;
  IF (P = 6 & MENOS = '0'R) THEN DO;
    BLOQUEADO_AC = '1'R;

```

```

SI QUEFADD_DM = "0"R;
AD = "1"R;
DM = "1"R;
RF = "1"R;
MAS = "0"R;
CALL MENOS NUEVA_ACTIVIDAD;
END;
IF (P = 7 & MAS_PED = "0"R) THEN DO;
MAS_PED = "1"R;
SI QUEFADD_AC = "0"R;
SI QUEFADD_DM = "0"R;
N = 10;
FR = 14;
L = 13;
CALL BORRA;
GOTO FIN;
END;
IF P = 9 THEN DO;
SI QUEFADD_AC = "1"R;
SI QUEFADD_DM = "1"R;
RF = "1"R;
AD = "1"R;
MAS = "1"R;
MENOS = "1"R;
TEXT0 = "1"R;
GUAR_1 = 1;
DO N = 1 TO 3;
CALL GOTOXY(2, 13+N);
IF N = 1 THEN
PUT FILE(OUT) EDY("NOTA: ", CK)(A, A);
ELSE
PUT FILE(OUT) EDIT(CK)(A);
END;
L = 14;
DO N = 1 TO 3;
CALL GOTOXY(45, 1);
PUT FILE(OUT) EDIT("")(A);
L = L + 1;
END;
N = 0;
REP: N = N + 1;
CALL GOTOXY(9, 13+N);
READ FILE(CTN) INTO(DATO);
IF (N = 1 & DATO = "") THEN
GOTO MM;
CONTADOR_NOTA = CONTADOR_NOTA + 1;
VECTOR_NOTA(CONTADOR_NOTA) = DATO;
IF N < 3 THEN
GOTO REP;
MM:
GOTO AA;
END;
IF P = 8 THEN DO;
SI QUEFADD_AC = "1"R;
SI QUEFADD_DM = "1"R;
FR = "0"R;
AC = "0"R;
DM = "0"R;
AD = "1"R;
RF = "0"R;
MAS = "1"R;
MENOS = "0"R;
MAS_PED = "0"R;

```



```

IF TEXTO = "I R" THEN DO;
  REQUERIDO_AC = "0"R;
  REQUERIDO_B = "0"R;
  TEXTO = "0"R;
  CALL GOTOXY(2,14);
  PUT FILE(OUT) EDIT("PROVEEDOR:" ,CK) (A,A);
  CALL GOTOXY(2,15);
  PUT FILE(OUT) EDIT(CK) (A);
  CALL GOTOXY(2,16);
  PUT FILE(OUT) EDIT("PAIS:" ,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("I") (A);
    I = I + 1;
  END;
END;
IF GUAR_1 /= 0 THEN DO;
  J = GUAR_1;
  CALL MUESTRA_PROVEEDOR;
  GUAR_1 = 0;
END;
END;
IF P = 10 THEN
  CALL FINAL;
END;
ELSE
  GO TO 4A;
END; /* INTERPRETA */

ACEPTAR: PROCEDURE;
IF CONT_PROV_ACEP /= 0 THEN DO;
  I = 1;
  DO WHILE VECTOR_PROV(I) /= VECTOR_GUAR_PROV(I) & (CONT_PROV_ACEP+1);
    I = I + 1;
  END;
  IF I < CONT_PROV_ACEP + 1 THEN
    GO TO FF;
  CONT_PROV_ACEP = CONT_PROV_ACEP + 1;
  VECTOR_GUAR_PROV(CONT_PROV_ACEP) = VECTOR_PROV(I);
  VECTOR_GUAR_PAIS(CONT_PROV_ACEP) = VECTOR_PAIS(I);
  CALL GOTOXY(12,2);
  PUT FILE(OUT) EDIT(CK) (A);
  PUT FILE(OUT) EDIT(ASCT(29),VECTOR_PROV(I),ASCT(28)) (A,A,A);
  CALL GOTOXY(10,4);
  PUT FILE(OUT) EDIT(CK) (A);
  PUT FILE(OUT) EDIT(ASCT(29),VECTOR_PAIS(I),ASCT(28)) (A,A,A);
  M = 10;
  PP = 24;
  I = 6;
  CALL BORRA;
SSS: CALL GOTOXY(24,6);
  READ FILE(IN) INTO(MONTO,*);
  IF (SUBSTR(MONTO,5,"0987654321,") /= 0 & MONTO,8 = ".") THEN DO;
    M = 10;
    PP = 24;
    I = 6;
    CALL BORRA;

```

```

        GO TO SSK;
    END;
    N = 10;
    PP = 46;
    I = 6;
    CALL BORRA;
SS:CALL DIBUXXY(46,A);
    READ FILE(IN) INTO(MONTO_B);
    IF (VERIFY(MONTO_B,'0987654321.') ^= 0 / MONTO_B = '') THEN DO;
        N = 10;
        PP = 46;
        I = 6;
        CALL BORRA;
        GO TO SS;
    END;
    VECTOR_MONTO_A(CONT_PROV_ACEP) = MONTO_A;
    VECTOR_MONTO_B(CONT_PROV_ACEP) = MONTO_B;
FF:
END; /*          ACEPTAR          */

RETRUCEDE: PROCEDURE;

    IF PRI = '0'*R THEN DO;
        PRI = '1'*R;
        GUAR_J = J;
    END;
    J = J - 1;
    IF J = 0 THEN
        J = 1;
    CALL MUESTRA_PROVEEDOR;

END; /*          RETRUCEDE          */

MENOS_NUEVA_ACTIVIDAD: PROCEDURE;

    IF PRI = '0'*R THEN DO;
        PRI = '1'*R;
        GUAR_J = J;
        AUXILIAR = CONT_PROV_ACEP + 1;
    END;
    AUXILIAR = AUXILIAR - 1;
    IF AUXILIAR ^= 0 THEN DO;
        J = AUXILIAR;
        CALL MUESTRA_PROVEEDOR_1;
    END;
    IF AUXILIAR = 0 THEN
        AUXILIAR = AUXILIAR + 1;
    IF CONT_PROV_ACEP = 0 THEN
        MAS = '1'*R;

END; /*          MENOS NUEVA ACTIVIDAD          */

MAS_NUEVA_ACTIVIDAD: PROCEDURE;

    AUXILIAR = AUXILIAR + 1;
    IF AUXILIAR = CONT_PROV_ACEP + 1 THEN
        AUXILIAR = AUXILIAR - 1;
    J = AUXILIAR;
    CALL MUESTRA_PROVEEDOR_1;

END; /*          MAS NUEVA ACTIVIDAD          */

```

```

FINAL = PROCEDURE ;

CLOSE FILE(ARCH DATOS) ;
CLOSE FILE(INDICE) ;
IF CONT_PROV_ACEP > 0 THEN DO ;
    NUMERO_AÑO = 6 ;
    NUMERO_INDICE = 16 ;
    CALL APPEND ARCHIVOS ;
    CALL PUEB_MACTO(GBPRE_ARCHIVO,DIRECCION) ;
    CAUE = NUMERO_ACTIVIDAD ;
    CHEQUEO = NUMERO_ACTIVIDAD ;
    APLINUM1 = DIRECCION ;
    APLINUM2 = 1 ;
    DREF = CONT_PROV_ACEP ;
    CALL INSERTA(RESAÑO,CLAVE,APLINUM1,NREF,APLINUM2) ;
    CADENA_CHEQ = CADENA_CHEQ(ASCII(LENGTH(CHAR(CONT_PROV_ACEP))+1)) ||
    CHAR(CONT_PROV_ACEP) ;
    DO I = 1 TO CONT_PROV_ACEP ;
        CADENA_CHEQ = CADENA_CHEQ(ASCII(LENGTH(VECTOR GUAR_PRODCT))+1) ||
        VECTOR_GUAR_PRODCT(I) || ASCII(LENGTH(VECTOR GUAR_PAIS(I))+1) ||
        VECTOR_GUAR_PAIS(I) || ASCII(LENGTH(VECTOR MONTO_$(I))+1) ||
        VECTOR_MONTO_$(I) || ASCII(LENGTH(VECTOR MONTO_R(I))+1) ||
        VECTOR_MONTO_R(I) ;
    END ;
    MON_$(I) = MON_$(I) + DECIMAL(VECTOR_MONTO_$(I),7,2) ;
    MON_R = MON_R + DECIMAL(VECTOR_MONTO_R(I),7,2) ;
END ;
CADENA_CHEQ = CADENA_CHEQ(ASCII(255)) ;
WRITE FILE(ARCH DATOS) FROM(OLD CHEQ) KEYFROM(DIRECCION) ;
CLOSE FILE(ARCH DATOS) ;
CLOSE FILE(INDICE) ;
CALL GOTOXY(11,8) ;
PUT FILE(OLD) EDIT('TOTAL')(A) ;
CALL GOTOXY(18,8) ;
PUT FILE(OLD) EDIT(MON_$(F(10,2))) ;
CALL GOTOXY(41,8) ;
PUT FILE(OLD) EDIT(MON_R)(F(10,2)) ;
CALL GOTOXY(11,10) ;
PUT FILE(OLD) EDIT('MONTO COMPROMETIDO')(A) ;
CALL GOTOXY(41,10) ;
PUT FILE(OLD) EDIT(MON_R)(F(10,2)) ;
END ;
ELSE DO ;
    NUM_CHEQ = NUM_CHEQ - 1 ;
    WRITE FILE(CONTROL_ARCHIVO) FROM(CONTROL_NUMER(I)) KEYFROM(2) ;
    CONT_PP ;
END ;

END ; /* FINALIZA */

IF P <= 10 THEN
    GOTO A6 ;

DIRE_PANTA = 7 ;
CALL REALIZA_CARTA(DIRE_PANTA,CONTADOR_NOTA,VECTOR_NOTA,
NUMERO_ACTIVIDAD) ;
BORRA = PROCEDURE ;

DO I = 1 TO N ;
    CALL GOTOXY(I+PP,1) ;
    PUT FILE(OLD) EDIT('')(A(1)) ;
END ;

```

END; /* RIVERA */
FF;

END; /* CREA_CHEQUE */


```

*
*
*
*
*
*****

```

```

DCI
(A_TOT_#_F,A_TOT_B_F,F_TOT_#_F,F_TOT_B_F) DEC FIXED(10,2),
LUNGTID) BIN FIXED,
CK CHAR(1),
DIR DEC FIXED,
(CONT_AU,I,K,NUM_AU) BIN FIXED,
VECTOR_GUAR_NUM(30) BIN FIXED,
VECTOR_GUAR_AP2(30) BIN FIXED,
(NUMERO_ARCH,NUMERO_INDICE) DEC FIXED,
(INDICE_ARCH,DATOS,OUT,CONTROL_ARCHIVO,IN) BIN FIXED,
(CONT_AU_ACEP,I,CONT_AU_PAG) FILE,
NUMERO_ACTIVIDAD BIN FIXED,
DIRE_PANTA CHAR(20) VAR,
(NTTID,NF,IMP) BIN FIXED,
FECHA CHAR(8),
CADENA_LEIDA CHAR(1000) VAR,
01 BLQ_PROF,
02 CADENA_PROF CHAR(1000) VAR,
01 BLQ_FACT,
02 CADENA_FACT CHAR(1000) VAR,
01 BLQ_PED,
02 CADENA_PED CHAR(1000) VAR,
01 BLQ_PAG,
02 CADENA_PAG CHAR(1000) VAR,
01 BLQ_RECT,
02 CADENA_RECT CHAR(1000) VAR,
ESTADO CHAR(1),
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,
(PERFORMA,FACTURAL,PEDIDO1,PAGO1,RECIBTRI) CHAR(12) VAR EXT STATIC,
(CI,PA) CHAR(3) VAR,
NUMERO_ARCHIVO CHAR(20) VAR,
(APL INV1,APL INV2) DEC FIXED,
NREF BIN FIXED,
(CLAUF,CLAUF2,RES,CK) CHAR(120) VAR,
TESAURO FILE VARIABLE,
RA_FNC BIT(1),
(POS_ABS,FE_CA,LON_CM,FE_CB,ILL_N,POS_RE,UTA(C)) BIN FIXED,
01 INF,
02 APL RATZ BIN FIXED,
02 NO NTU BIN FIXED,
02 NO NODU BIN FIXED,
02 APL PROX VACTO BIN FIXED,
01 NODU,
02 CAR NODU,
03 IUM N BIN FIXED,
03 NTU BIN FIXED,
03 NC BIN FIXED,
03 IN BIN FIXED,
03 VECTNA_I BIN FIXED,
03 VECTNA_D BIN FIXED,
02 ENTRADA CHAR(1009) VAR,

```

```

01 CONTROL_ARCHIVO),
02 TIPO) BIN FIXED,
02 N_ARC) BIN FIXED,
02 LON_REG) BIN FIXED,
02 HIT_VACTO) BIN FIXED,
02 MAX_DTR) BIN FIXED,
02 NUMBRE_ARC) CHAR(20) VAR;

```

```

/* RUTINAS EXTERNAS */

```

```

DCI
INSERTA_ENTRY(FILE VARIABLE, CHAR(120) VAR, DEC FIXED, BIN FIXED, DEC FIXED),
PIDE_VACTO_ENTRY(CHAR(20) VAR, 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,
(1) 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);

```

```

OK = ASCII(11);

```

```

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

```

```

CLOSE FILE(ARCH_DATOS);
CLOSE FILE(INDICE);
INDICADO = INDICE;
IF CONT_ARCHIVO = 0 THEN DO;
  IF DIRE_PANTA = 4 THEN DO;
    CALL GOTOXY(18, 3);
    PUT FILE(OUT) EDIT(NTITUL)(F(4));
    CALL GOTOXY(30, 3);
    PUT FILE(OUT) EDIT(NE.IEMP)(F(6));
    CADENA_PROF = CADENA_LEIDA || ASCII(LENGTH(CHAR(NTITUL))+1) ||
    CHAR(NTITUL) || ASCII(LENGTH(CHAR(NE.IEMP))+1) || CHAR(NE.IEMP) ||
    ASCII(LENGTH(CHAR(FECHA))+1) || CHAR(FECHA) || ASCII(255);
    NUMERO_ARC = 3;
    NUMERO_INDICE = 13;
    PROFORMAL = NUMERO_ACTIVIDAD;
  END;
  IF DIRE_PANTA = 5 THEN DO;
    NUMERO_ARC = 4;
    NUMERO_INDICE = 14;
    CADENA_FACT = CADENA_LEIDA || ASCII(255);
    FACTURAL = NUMERO_ACTIVIDAD;
  END;
  IF DIRE_PANTA = 6 THEN DO;
    CALL GOTOXY(40, 6);
    PUT FILE(OUT) EDIT(CK)(6);
    PUT FILE(OUT) EDIT("MONTO TOTAL $ MONTO TOTAL BS") (6);
    CALL GOTOXY(40, 7);
    PUT FILE(OUT) EDIT(A_TOT $ F)(F(10, 2));
    CALL GOTOXY(60, 7);
    PUT FILE(OUT) EDIT(A_TOT_B $ F)(F(10, 2));
    CALL GOTOXY(40, 10);
    PUT FILE(OUT) EDIT("CANTIDAD A REPORTAR: ") (6);
    READ FILE(IN) INTO(CI);
    CALL GOTOXY(42, 11);
    PUT FILE(OUT) EDIT("PESO APROXIMADO: ") (6);
    READ FILE(IN) INTO(PA);
    CADENA_PED = CADENA_LEIDA ||
    ASCII(LENGTH(CI)+1) || CI ||
    ASCII(LENGTH(PA)+1) || PA || ASCII(255);
  END;

```

```

NUMERO_ARC = 5;
NUMERO_INDICE = 15;
PEDIDO = NUMERO_ACTIVIDAD;
END;
IF DIRE_PANTA = 8 THEN DO;
  CONTROL_ARCH = CONTROL_ARCH(ARCHIVO);
  CALL GOTOXY(32,10);
  READ FILE(IN) INTO(CI);
  NUMERO_ARC = 7;
  NUMERO_INDICE = 17;
  PAGUI = NUMERO_ACTIVIDAD;
END;
IF DIRE_PANTA = 9 THEN DO;
  CLAVE2 = NUMERO_ACTIVIDAD;
  DO I = 1 TO 9 - LENGTH(CLAVE2);
    CLAVE2 = ' ' || CLAVE2;
  END;
  NUMERO_INDICE = 18;
  NUMERO_ARC = 8;
  CLOSE FILE(INDICE);
  CLOSE FILE(ARCH_DATOS);
  I = 1;
  DO WHILE (CONTROL_ARCH(I).N_ARC ^= NUMERO_ARC);
    I = I + 1;
  END;
  NOMBRE_ARCHIVO = CONTROL_ARCH(I).NOMBRE_ARC;
  LONGITUD = CONTROL_ARCH(I).LON_REG;
  OPEN FILE(ARCH_DATOS) TITLE (CONTROL_ARCH(I).NOMBRE_ARC) UPDATE
  DIRECT KEYED ENV(RESIZE(LONGITUD));
  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(RESIZE(1024));
  CALL BUSCA(TESAURO,CLAVE2,RA_ENC,POS_ABS,RES_CM,PE_CM,LON_CM,
  NDDO,PE_DB,VIA,ULT_N,POS_RE,(NF));
  IF RA_ENC = '0'B THEN DO;
    IF NEJEMP = CONT_AU_ACEP THEN DO;
      NUMERO_INDICE = 17;
      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(RESIZE(1024));
      CALL BUSCA(TESAURO,CLAVE2,RA_ENC,POS_ABS,RES_CM,PE_CM,LON_CM,
      NDDO,PE_DB,VIA,ULT_N,POS_RE,(NF));
      APIINU2 = 0;
      SUBSTR(ENTRADA,POS_ABS+7,1) = ASCII(APIINU2/10000);
      SUBSTR(ENTRADA,POS_ABS+8,1) = ASCII(APIINU2/100);
      SUBSTR(ENTRADA,POS_ABS+9,1) = ASCII(APIINU2);
      REWRITE FILE(INDICE) FROM(NDDO) KEY(NDDO,CAR_NDDO,NUR_N);
      ESTADO = '0';
    END;
  ELSE
    ESTADO = '1';
  CALL GOTOXY(27,1);
  PUT FILE(OUT) FDLT(CONT_AU_ACEP)(F(4));
  CALL GOTOXY(42,1);
  PUT FILE(OUT) FDLT(ESTADO)(A(1));
  NUMERO_INDICE = 18;

```



```

CADENA_RECT = CADENA_LEIDA !! ASCII(LENGTH(CHAR(CONT_AU_ACEP))+1)!!
CHAR(CONT_AU_ACEP) !! ASCII(LENGTH(ESTADO)+1) !! ESTADO !!
ASCII(255);
CALL PTDE_VACIO(NUMBRE_ARCHIVO, APLINV1);
WRITE FILE(ARCH_DATOS) FROM(BLD_RECT) KEYFROM(APLINV1);
CLOSE FILE(INDICE);
I = 1;
DO WHILE(CONTROL_ARCH(C).N_ARC ^= NUMERO_INDICE);
    I = I + 1;
END;
OPEN FILE(INDICE) TITLE(CONTROL_ARCH(C).NOMBRE_ARC) UPDATE
DIRECT KEYED ENV(RESIZE(1024));
APLINV2 = 1;
NREF = CONT_AU_ACEP;
CALL INSERTA(TESAURO, CLAVE2, APLINV1, NREF, APLINV2);
DO I = 1 TO CONT_AU_ACEP;
    CLAVE = CLAVE2 !! CHAR(I);
    NREF = VECTOR_GUAR_NUM(I);
    APLINV2 = VECTOR_GUAR_AP2(I);
    CALL INSERTA(TESAURO, CLAVE, APLINV1, NREF, APLINV2);
END;
END;
ELSE DO;
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 = RANK(SUBSTR(ENTRADA, POS_ABS+5, 1))*256 +
RANK(SUBSTR(ENTRADA, POS_ABS+6, 1));
J = 0;
K = 0;
CONTINUE;
    J = J + 1;
    BA_FNC = '0*B';
    DO WHILE(BA_FNC = '0*B');
        K = K + 1;
        CLAVE = CLAVE2 !! CHAR(K);
        CALL BUSCA(TESAURO, CLAVE, BA_FNC, POS_ABS, RES_CN_FF_CN, LON_CN,
        MODD, PE_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));
        I = 1;
DO WHILE(COND_AU1 ^= VECTOR_GUAR_NUM(I) & I < (CONT_AU_ACEP + 1);
    I = I + 1;
END;
        IF J < CONT_AU_ACEP + 1 THEN DO;
            IF I < CONT_AU_ACEP THEN DO;
                DO WHILE(I < CONT_AU_ACEP);
                    VECTOR_GUAR_NUM(I) = VECTOR_GUAR_NUM(I+1);
                    VECTOR_GUAR_AP2(I) = VECTOR_GUAR_AP2(I+1);
                    I = I + 1;
                END;
            END;
            CONT_AU_ACEP = CONT_AU_ACEP - 1;
            I = I - 1;
        END;
        IF J < CONT_AU THEN
            GOTO CONTINUE;
        IF CONT_AU_ACEP ^= 0 THEN DO;
            APLINV1 = DIR;
            CONT_AU = CONT_AU + CONT_AU_ACEP;
            CALL BUSCA(TESAURO, CLAVE2, BA_FNC, POS_ABS, RES_CN_FF_CN, LON_CN,
            MODD, PE_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);
      DO I = 1 TO CONT_AU_ACEP;
        K = K + 1;
        CLAVE = CLAVE2 !! CHAR(K);
        NREF = VECTOR_DUAR_NUM(I);
        APLINU2 = VECTOR_DUAR_AP2(I);
        CALL INSERTA(TESAURO,CLAVE,APLINU1,NREF,APLINU2);
      END;
NUMERO_INDICE = 17;
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).NUMBRE_ARC) UPDATE
DIRECT KEYED ENU(RESIZE(1024));
CALL BUSCA(TESAURO,CLAVE2,BA_ENC,POS_ABS,RES_CB,FE_CB,LON_CB,
NODO,FE_CB,VIA,UTL_N,POS_RE,INF);
CONT_AU_PAG = RANK(SUBSTR(ENTRADA,POS_ABS+5,1))*256 +
RANK(SUBSTR(ENTRADA,POS_ABS+6,1));
IF CONT_AU_PAG = CONT_AU THEN DO;
  APLINU2 = 0;
  SUBSTR(ENTRADA,POS_ABS+7,1) = ASCII(APLINU2/10000);
  SUBSTR(ENTRADA,POS_ABS+8,1) = ASCII(APLINU2/100);
  SUBSTR(ENTRADA,POS_ABS+9,1) = ASCII(APLINU2);
  REWRITE FILE(INDICE) FROM(NODO) KEY(NODO,CAB_NODO,NUM_N);
  ESTADO = "C";
END;
ELSE
  ESTADO = "T";
CADENA_RECT = CADENA_LEIDA !! ASCII(LENGTH(CHAR(CONT_AU))+1) !!
CHAR(CONT_AU) !! ASCII(LENGTH(ESTADO)+1) !! ESTADO !! ASCII(255);
  WRITE FILE(ARCH_DATOS) FROM(BLU_RECT) KEYFROM(DIR);
  CALL GOTOXY(27,1);
  PUT FILE(OUT) EDIT(CONT_AU)(F(4));
  CALL GOTOXY(42,1);
  PUT FILE(OUT) EDIT(ESTADO)(A(1));
  END;
  END;
  GOTO FIN;
END;

I = 1;
DO WHILE (CONTROL_ARCH(I).N_ARC ^= NUMERO_ARC);
  I = I + 1;
END;
NUMBRE_ARCHIVO = CONTROL_ARCH(I).NUMBRE_ARC;
LONGTUD = CONTROL_ARCH(I).LON_REG;
OPEN FILE(ARCH_DATOS) TITLE (CONTROL_ARCH(I).NUMBRE_ARC) UPDATE
DIRECT KEYED ENU(RESIZE(LONGTUD));
I = 1;
DO WHILE (CONTROL_ARCH(I).N_ARC ^= NUMERO_INDICE);
  I = I + 1;
END;
OPEN FILE(INDICE) TITLE (CONTROL_ARCH(I).NUMBRE_ARC) UPDATE
DIRECT KEYED ENU(RESIZE(1024));
CALL PIDE_VACIO(NUMBRE_ARCHIVO,APLINU1);
IF DIRE_PANTA = 4 THEN
  WRITE FILE(ARCH_DATOS) FROM(BLU_PROF) KEYFROM(APLINU1);
IF DIRE_PANTA = 5 THEN
  WRITE FILE(ARCH_DATOS) FROM(BLU_FACT) KEYFROM(APLINU1);
IF DIRE_PANTA = 6 THEN

```

```

        WRITE FILE(ARCH_DATOS) FROM(BLO_PED) KEYFROM(API_INV1);
    IF DIRE_PANTA = 8 THEN
        WRITE FILE(ARCH_DATOS) FROM(BLO_PAG) KEYFROM(API_INV1);
    CLAVE2 = NUMERO_ACTIVIDAD;
    API_INV2 = 1;
    NREF = CONT_AU_ACEP;
    CALL INSERTA(TESAURO,CLAVE2,API_INV1,NREF,API_INV2);
    DO I = 1 TO CONT_AU_ACEP;
        CLAVE = CLAVE2 || CHAR(I);
        NREF = VECTOR_GUAR_NUM(I);
        API_INV2 = VECTOR_GUAR_AP2(I);
        CALL INSERTA(TESAURO,CLAVE,API_INV1,NREF,API_INV2);
    END;
END;
ELSE DO;
    IF (DIRE_PANTA = 4 || DIRE_PANTA = 6 || DIRE_PANTA = 8) THEN DO;
    READ FILE(CONTROL_ARCHIVO) INTO(CONTROL_NUMERTCO) KEY(2);
    IF DIRE_PANTA = 4 THEN
        NUM_PROF = NUM_PROF - 1;
    IF DIRE_PANTA = 6 THEN
        NUM_PED = NUM_PED - 1;
    IF DIRE_PANTA = 8 THEN
        NUM_PAG = NUM_PAG - 1;
    WRITE FILE(CONTROL_ARCHIVO) FROM(CONTROL_NUMERTCO) KEYFROM(2);
    END;
END;

FIN;
END; /*      FINALIZA      */

```



```

01 CONTROL_PANTA(16),
  02 NUM_ARC
  02 NUM_INDICE
  02 DIRE_P
  02 DIRE_CL
  02 NOMBRE_P
DCL
01 CONTROL_ARCH(16),
  02 TIPO
  02 N_ARC
  02 LON_REG
  02 ULT_VACIO
  02 MAX_DIR
  02 NOMBRE_ARCH
01 CONTROL_NUMERICO,
  02 NUM_AU
  02 NUM_PROF
  02 NUM_PED
  02 NUM_CHEQ
  02 NUM_PAG
01 CONTROL_VECTOR,
  02 NUM_ARCHIVO
  02 ULT_VACIO_INV
  02 IN_EL
01 CONTROL_PROCESO(6),
  02 NOMBRE
  02 NUM_ARCHIVO
  02 PROCESO
  02 PRO_CAMBIO
DCL
01 EST_LEC,
  02 VECTOR_COOR(500)
DCL
PIDE_VACIO ENTRY(CHAR(20) VAR,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),
GOTOXY ENTRY(BIN FIXED,BIN FIXED),
INSERTA ENTRY(FILE VARIABLE,CHAR(120) VAR,DEC FIXED,BIN FIXED,DEC FIXED),
GENERA_CAPTA ENTRY(BIN FIXED,BIN FIXED,BIN FIXED,BIN FIXED,BIN FIXED,
BIN FIXED,CHAR(1000) VAR);
READ FILE(CONTROL_ARCHIVO) INTO(CONTROL_ARCH) KEY(1);
READ FILE(CONTROL_PANTALLAS) INTO(CONTROL_PANTA) KEY(1);

CK = ASCII(11);
CONTADOR_REIN = 0;
CONTADOR_TIT = 0;
TITULO_AUX = 1;
CONTADOR_PR = 0;
CONTADOR_PP = 0;
CONTADOR_AU = 0;
APLINV2 = 1;
CONT_AU_PROF = 0;
CONT_AU_PED = 0;
PED_AUX = 1;
CONT_AU_CHEQ = 0;
CHEQ_AUX = 1;
CONT_AU_FACT = 0;
CONT_AU_PAG = 0;
CONT_AU_RECT = 0;
PROF_AUX = 1;

```

```

FACT_AUX = 1;
PAG_AUX = 1;
RECI_AUX = 1;
OB = 0;
IF J = 51 THEN
    NUMERO_ARC = 2;
ELSE
    NUMERO_ARC = 1;
    = 1;
DO WHILE (CONTROL_PANTA(I).NUM_ARC ^= NUMERO_ARC);
    I = I + 1;
END;
NUMERO_INDICE = CONTROL_PANTA(I).NUM_INDICE;
IF J = 36 THEN DO;
    I = I + 1;
    APLINV2 = 0;
END;
DIRE_PANTA = CONTROL_PANTA(I).DIRE_F;
FG = 100;
DIRE_COR = CONTROL_PANTA(I).DIRE_C1;
FV = 100;
IG = 1;
IV = 1;
I = 1;
DO WHILE (CONTROL_ARCH(I).N_ARC ^= NUMERO_ARC);
    I = I + 1;
END;
LONGITUD = CONTROL_ARCH.LON_REG(I);
CLOSE FILE (ARCH_DATOS);
OPEN FILE (ARCH_DATOS) TITLE (CONTROL_ARCH.NOMBRE_ARC(I)) UPDATE
DIRECT KEYED ENV (RECSIZE (LONGITUD));
NOMBRE_ARCHIVO = CONTROL_ARCH.NOMBRE_ARC(I);
I = 1;
DO WHILE (CONTROL_ARCH(I).N_ARC ^= NUMERO_INDICE);
    I = I + 1;
END;
CLOSE FILE (INDICE);
OPEN FILE (INDICE) TITLE (CONTROL_ARCH.NOMBRE_ARC(I)) UPDATE
DIRECT KEYED ENV (RECSIZE (1024));
CALL GENERA_CAPTA (DIRE_PANTA, DIRE_COR, IG, FG, IV, FV, CADENA_LEIDA);
IF FV = 0 THEN DO;
    J = 0;
    GOTO CONTINUE1;
END;
POS = 1;
LON = RANK (SUBSTR (CADENA_LEIDA, POS, 1));
CLAVE = SUBSTR (CADENA_LEIDA, POS+1, LON-1);
POS = POS + LON;
LON = RANK (SUBSTR (CADENA_LEIDA, POS, 1)) - 1;
IF LON ^= 0 THEN
    CLAVE = CLAVE || SUBSTR (CADENA_LEIDA, POS+1, LON);
TESAURO = INDICE;
/* TRANSLATE: SE ENCARGA DE TRANSFORMAR LAS LETRAS MINUSCULAS
EN MAYUSCULAS */
CLAVE=TRANSLATE (CLAVE,
    "ABCDEFGHIJKLMN\&OPQRSTUVWXYZAEIOUAEIOU",
    "abcdefghijklmno pqrstuvwxyzáéíóúáéíóú");
CALL BUSCA (TESAURO, CLAVE, BA_ENC, POS_ABS, RES_CM, PF_CM, LON_CM, NODO,
    PF_CB, VTA, ULT_N, POS_RE, TNF);
IF BA_ENC = "0"B THEN DO;
    CALL PIDE_VACIO (NOMBRE_ARCHIVO, DIRECCION);
    CONTROL_PROCESO.NOMBRE(1) = "A1";
    CONTROL_PROCESO.PROCESO(1) = 0;

```

```

CONTROL_PROCESO.PROC_CAMBIO(1) = 1;
CONTROL_PROCESO.NOMBRE(2) = '11';
CONTROL_PROCESO.NUM_ARCHIVO(2) = NUMERO_INDICE;
DO I = 3 TO 6;
    CONTROL_PROCESO.NUM_ARCHIVO(I) = 0;
    CONTROL_PROCESO.PROCESO(I) = 0;
    CONTROL_PROCESO.PROC_CAMBIO(I) = 0;
END;
CONTROL_VECTOR.ULT_VACIO_INV = DIRECCION;
IN_EL = 'I';
APLINV1 = DIRECCION;
WRITE FILE(CONTROL_ARCHIVO) FROM(CONTROL_PROCESO) KEYFROM(4);
IF J = 51 THEN DO;
    CADENA_LEIDA1 = CADENA_LEIDA;
    CONTROL_VECTOR.NUM_ARCHIVO = 2;
    WRITE FILE(CONTROL_ARCHIVO) FROM(CONTROL_VECTOR) KEYFROM(3);
    WRITE FILE(ARCH_DATOS) FROM(BLQ_PROV) KEYFROM(DIRECCION);
    CLOSE FILE(ARCH_DATOS);
    NREF = 0;
    CALL INSERTA(TESAURO,CLAVE,APLINV1,NREF,APLINV2);
    CLAVE1 = CLAVE;
    CONTADOR_PR = 1;
END;
ELSE DO;
    IF J = 36 THEN
        CONTADOR_PP = 1;
    ELSE
        CONTADOR_AU = 1;
    CADENA_LEIDA2 = CADENA_LEIDA;
    CONTROL_VECTOR.NUM_ARCHIVO = 1;
    WRITE FILE(CONTROL_ARCHIVO) FROM(CONTROL_VECTOR) KEYFROM(3);
    WRITE FILE(ARCH_DATOS) FROM(BLQ_MOND_PP) KEYFROM(DIRECCION);
    CLOSE FILE(ARCH_DATOS);
    CONTROL_PROCESO.PROCESO(2) = 1;
    CONTROL_PROCESO.PROC_CAMBIO(2) = 0;
    CONTROL_PROCESO.NUM_ARCHIVO(1) = 1;
    READ FILE(CONTROL_ARCHIVO) INTO(CONTROL_NUMERICO) KEY(2);
    NUM_AU = NUM_AU + 1;
    WRITE FILE(CONTROL_ARCHIVO) FROM(CONTROL_NUMERICO) KEYFROM(2);
    NREF = NUM_AU;
    CALL INSERTA(TESAURO,CLAVE,APLINV1,NREF,APLINV2);
    CLAVE = CHAR(NUM_AU);
    CALL INSERTA(TESAURO,CLAVE,APLINV1,NREF,APLINV2);
    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,ION_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));
    CONT_AU = RANK(SUBSTR(ENTRADA,POS_ABS+5,1))*256 +
    RANK(SUBSTR(ENTRADA,POS_ABS+6,1));
    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);
    CLAVE = CLAVE1 !! CHAR(I);
    CALL BUSCA(TESAURO,CLAVE,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;
    APLINV1 = DIR;
    CLAVE = CLAVE1 !! CHAR(I);
    CALL INSERTA(TESAURO,CLAVE,APLINV1,NREF,APLINV2);
END;
END;
ELSE DO;
    IF J ^= 51 THEN DO;
/* EL AUTOR/PP EXISTE */
    IF J = 36 THEN DO;
        CALL GOTOXY(0,22);
        PUT FILE(OUT) EDIT("PP YA EXISTE",CK)(A,A);
    END;
    ELSE DO;
        CLAVE2 = CLAVE;
        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;
        LON = RANK(SUBSTR(CADENA_LEIDA2,POS,1));
        POS = POS + LON;
        LON = RANK(SUBSTR(CADENA_LEIDA2,POS,1));
        POS = POS + LON;
        LON = RANK(SUBSTR(CADENA_LEIDA2,POS,1));
        TITULO1 = SUBSTR(CADENA_LEIDA2,POS+1,LON-1);
        POS = POS + LON;
        LON = RANK(SUBSTR(CADENA_LEIDA2,POS,1)) - 1;
        IF LON ^= 0 THEN
            TITULO1 = TITULO1 !! SUBSTR(CADENA_LEIDA2,POS+1,LON);
        POS = 1;
        LON = RANK(SUBSTR(CADENA_LEIDA,POS,1));
        POS = POS + LON;
        LON = RANK(SUBSTR(CADENA_LEIDA,POS,1));
        POS = POS + LON;
        LON = RANK(SUBSTR(CADENA_LEIDA,POS,1));
        TITULO = SUBSTR(CADENA_LEIDA,POS+1,LON-1);
        POS = POS + LON;
        LON = RANK(SUBSTR(CADENA_LEIDA,POS,1)) - 1;
        IF LON ^= 0 THEN
            TITULO = TITULO !! SUBSTR(CADENA_LEIDA,POS+1,LON);
        IF TITULO = TITULO1 THEN DO;
            CALL GOTOXY(0,22);
            PUT FILE(OUT) EDIT("AUTOR YA EXISTE",CK)(A,A);
        END;
    ELSE DO;
        I = 1;
        ENCONTRADO = "1"B;
        DO WHILE(ENCONTRADO = "1"B);
            CLAVE = CLAVE2 !! CHAR(I);
            CALL BUSCA(TESAURO,CLAVE,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
                ENCONTRADO = "0"B;

```

```

ELSE DO;
  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(BLU_MONO_PP) KEY(DIR);
  POS = 1;
  LON = RANK(SUBSTR(CADENA_LEIDA2,POS,1));
  POS = POS + LON;
  LON = RANK(SUBSTR(CADENA_LEIDA2,POS,1));
  POS = POS + LON;
  LON = RANK(SUBSTR(CADENA_LEIDA2,POS,1));
  TITULO1 = SUBSTR(CADENA_LEIDA2,POS+1,LON-1);
  POS = POS + LON;
  LON = RANK(SUBSTR(CADENA_LEIDA2,POS,1)) - 1;
  IF LON ^= 0 THEN
  TITULO1 = TITULO1 !! SUBSTR(CADENA_LEIDA2,POS+1,LON);
  IF TITULO1 = TITULO THEN DO;
    CALL GOTDXY(0,22);
    PUT FILE(OUT) EDIT("AUTOR YA EXISTE,CK")(A,A);
    GOTD CONTINUE1;
  END;
  ELSE
    I = I + 1;
  END;
END;
CONTADOR_AU = 1;
CALL PIDE_VACIO(NOMBRE_ARCHIVO,DIRECCION);
APLINV1 = DIRECCION;
CADENA_LEIDA2 = CADENA_LEIDA;
WRITE FILE(ARCH_DATOS) FROM(BLU_MONO_PP) KEYFROM(DIRECCION);
CLOSE FILE(ARCH_DATOS);
READ FILE(CONTROL_ARCHIVO) INTO(CONTROL_NUMERICO) KEY(2);
NUM_AU = NUM_AU + 1;
WRITE FILE(CONTROL_ARCHIVO) FROM(CONTROL_NUMERICO) KEYFROM(2);
NREF = NUM_AU;
CALL INSERTA(TESAURO,CLAVE,APLINV1,NREF,APLINV2);
CLAVE = CHAR(NUM_AU);
CALL INSERTA(TESAURO,CLAVE,APLINV1,NREF,APLINV2);
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_CN,PE_CN,LON_CN,
NODO,PE_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));
CONT_AU = RANK(SUBSTR(ENTRADA,POS_ABS+5,1))*256 +
RANK(SUBSTR(ENTRADA,POS_ABS+6,1));
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);
  CLAVE = CLAVE1 !! CHAR(I);
  CALL BUSCA(TESAURO,CLAVE,BA_ENC,POS_ABS,RES_CN,PE_CN,LON_CN,NODO,

```

```
PE_CB,UTA,ULT_N,POS_RE,INF);
IF BA_ENC = '1'B THEN
    K = K + 1;
    I = I + 1;
END;
APLINV1 = DIR;
CLAVE = CLAVE1||CHAR(I);
CALL INSERTA(TESAURD,CLAVE,APLINV1,NREF,APLINV2);
END;
END;
END;
ELSE DO;
    CALL GOTOXY(0,22);
    PUT FILE(OUT) SKIP EDIT('PROVEEDOR YA EXISTE',CK)(A,A);
END;
END;
CONTINUE1;
CLOSE FILE(INDICE);

END;      /*      CREA_AL_PROV_PP      */
```



```

*
*
*****/

```

```

DCI
(CONTADOR_TIT,TITULO_AUX,CONT_AU_TIT)          BIN FIXED EXT STATIC,
  CONTADOR_PR                                  BIN FIXED EXT STATIC,
(CONT_AU_PROF,CONT_AU_FACT,CONT_AU_PAG,
  CONT_AU_RECT,CONT_AU_CHEQ,CHEQ_AUX,JJ,
  PROF_AUX,FACT_AUX,PAG_AUX,RECT_AUX,
  CONT_AU_PED,PED_AUX,OB,CONTADOR_REIN) BIN FIXED EXT STATIC,
(CONTADOR_AU,CONTADOR_PP)  BIN FIXED EXT STATIC,
CK                                              CHAR(1) VAR,
APLINU2                                        DEC FIXED,
(NUMERO_INDICE,I,NUMERO_ARC,IV,FU,DIRE_PANTA)  BIN FIXED,
(OUT,IN,PANTALLAS,CONTROL_ARCHIVO,INDICE,IMP,
  ARCH_DATOS,COORDENADAS)                    FILE,
01 BLQ_MONO_PP,
  02 CADENA_LEIDA?                            CHAR(1000) VAR,
TESAURO                                        FILE VARIABLE,
  BA_ENC                                      BIT(1),
(POS_ABS,PF_CM,LON_CM,PF_CB,ULT_N,POS_RE,VIA(S),CI)  BIN FIXED,
(X,Y,X1,Y1,CONT_AU,NUM_AU1,R,LONGITUD,POS,LON,J)  BIN FIXED,
(CLAVE,RES_CM,AUTOR)                          CHAR(120) VAR,
CLAVE1                                        CHAR(120) VAR EXT STATIC,
DIR                                           DEC FIXED,
01 INF,
  02 APU_RATZ                                  BIN FIXED,
  02 NO_NIV                                    BIN FIXED,
  02 NO_NODO                                   BIN FIXED,
  02 APU_PROX_VACTO                            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_VACTO                               BIN FIXED,
  02 MAX_DIR                                 BIN FIXED,
  02 NOMBRE_ARC                              CHAR(20) VAR,

```

```

DCI
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,
(S) 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),
GENERA ENTRY(BIN FIXED,BIN FIXED,BIN FIXED);

```

```

CK = ASCII(11);
JJ = 0;
TESAURO = INDICE;
CONTADOR_REIN = 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_RECT = 0;
OR = 0;
PROF_AUX = 1;
FACT_AUX = 1;
RECT_AUX = 1;
PAG_AUX = 1;
NUMERO_INDICE = 12;
DIRE_PANTA = 20;
CONT_AU = 0;
IV = 1;
FV = 100;
READ FILE(CONTROL_ARCHIVO) INTO(CONTROL_ARCH) KEY(1);

IF J = 5 THEN DO;
  OPEN FILE(IMP) TITLE("%0LPT") STREAM OUTPUT PRINT;
  PUT FILE(IMP) SKIP EDIT(CLAVE1)(A);
  GOTO AA;
END;
PUT FILE(OUT) PAGE;
CALL GENERA(DIRE_PANTA,IV,FV);
CALL GOTOXY(15,0);
PUT FILE(OUT) EDIT(ASCII(29),CLAVE1,ASCII(28))(A,A,A);
AA: I = 1;
DO WHILE (CONTROL_ARCH(I).N_ARC ^= NUMERO_INDICE);
  I = I + 1;
END;
CLOSE FILE(INDICE);
OPEN FILE(INDICE) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) INPUT
DIRECT KEYED ENV(RECSIZE(1024));
CLAVE = CLAVE1;
CALL BUSCA(TESAURO,CLAVE,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));
IF CONT_AU = 0 THEN DO;
  CALL GOTOXY(0,22);
  PUT FILE(OUT) EDIT("NO HAY AUTORES NI PP BAJO ESTE PROVEEDOR")(A);
  GOTO FIN;
END;
CLOSE FILE(ARCH_DATOS);
NUMERO_ARC = 1;
I = 1;
DO WHILE (CONTROL_ARCH(I).N_ARC ^= NUMERO_ARC);
  I = I + 1;
END;
LONGITUD = CONTROL_ARCH(I).LON_REG;
OPEN FILE(ARCH_DATOS) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) INPUT
DIRECT KEYED ENV(RECSIZE(LONGITUD));
R = 1;
X = 14;
Y = 7;
X1 = 15;
Y1 = 9;
REPT_AU:
BA_ENC = "0"B;
DO WHILE (BA_ENC = "0"B);

```

```

JJ = JJ + 1;
CLAVE=CLAVE||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;
NUM_AU1 = RANK(SUBSTR(ENTRADA,POS_ABS+5,1))*256 +
RANK(SUBSTR(ENTRADA,POS_ABS+6,1));
APLINV2 = 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;
CLAVE = CHAR(NUM_AU1);
I = 1;
DO WHILE(CONTROL_ARCH(I).N_ARC ^= NUMERO_INDICE);
I = I + 1;
END;
CLOSE FILE(INDICE);
OPEN FILE(INDICE) TITLE(CONTROL_ARCH(I).NOMBRE_ARC) INPUT
DIRECT KEYED ENV(RECSTZE(1024));
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);
I = 1;
POS = 1;
REPT:
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 (APLINV2 = 1 & I = 1) THEN DO;
I = I + 1;
POS = POS + LON + 1;
AUTOR = CLAVE;
GOTO REPT;
END;
IF J = 5 THEN DO;
IF APLINV2 = 1 THEN
PUT FILE(IMP) SKIP EDIT('AUTOR: ',AUTOR)(A,A);
PUT FILE(IMP) SKIP EDIT('TITULO: ',CLAVE)(A,A);
GOTO BB;
END;
IF APLINV2 = 0 THEN GOTO XX;
CALL GOTOXY(X,Y);
PUT FILE(OUT) EDIT(ASCII(29),AUTOR,ASCII(28))(A,A,A);
XX: CALL GOTOXY(X1,Y1);
PUT FILE(OUT) EDIT(ASCII(29),CLAVE,ASCII(28))(A,A,A);
BB:
IF (CONT_AU ^= 1 & R = 1) THEN DO;
R = R + 1;
Y = 13;
Y1 = 15;
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) INPUT

```

```
DIRECT KEYED ENV(RECIZE(1024));  
GOTO REPI_AU;  
END;  
CONTADOR_TIT = 1;  
TITULO_AUX = 1;  
CONT_AU_TIT = CONT_AU;  
  
FIN;  
CLOSE FILE(IMP);  
END; /* TITULOS */
```


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

```

/*****
*
* PROGRAMA QUE SE ENCARGA DE RECIBIR INFORMACION Y NORMALIZARLA
* PARA LA CREACION DE PROVEEDORES,AUTORES Y PP
*
* PARAMETROS:
*
* DIRE_PANTA: DIRECCION DE LA PANTALLA DE LA ACTIVIDAD
* QUE SE ESTA CREANDO EN EL ARCHIVO PANTALLAS
*
* DIRE_COR: DIRECCION DONDE SE ENCUENTRAN LAS COORDENADAS
* CORRESPONDIENTE A LOS DATOS DE LA PANTALLA EN EL
* ARCHIVO COORDENADAS
*
* IG: INICIO DE COORDENADAS
*
* FG: FIN DE COORDENADAS
*
* CADENA_LEIDA: CADENA DONDE SE GUARDA LA INFORMACION
* CREADA
*
* VARIABLES:
*
* MODIFICAR:
* S: SI HAY QUE MODIFICAR
* N: NO HAY QUE MODIFICAR
*
* CLAVE1: PROVEEDOR EN MEMORIA
*
* VALOR: TEXTO DEL CAMPO
*
* LONGITUD_1: LONGITUD DEL CAMPO
*
* X: POSICION HORIZONTAL
*
* Y: POSICION VERTICAL
*
* LLAMADA POR:
*
* MUESTRA_DATOS
* CREA_REINTEGRO
* MUESTRA_INFORMACION
*
* LLAMA A:
*
* GENERA
* BORRA_INFORMACION
* GOTOXY
* VALIDAR
* CORRIGE
*
* ARCHIVOS:
* COORDENADAS: CONTIENE LAS COORDENADAS DE LOS DATOS DE
* LAS PANTALLA DEL SISTEMA
*
*****/

```

DCI
CK
(LONGITUD, LONGITUD_1, I1, J1)
CLAVE1

CHAR(1) VAR,
BIN FIXED,
CHAR(120) VAR EXT STATIC,

```

UBICACION(4)
(NUM, NUM1, NUM2)
(POS, IG, FG, IV, FU, DIRE_PANTA, DIRE_COR, X, Y, TC, PG, T)
MODIFICAR
(TOPE_REPETITIVO, FIN_REP, CL)
CADENA_LEIDA
(IN, OUT, POSICION, COORDENADAS)
01 EST_LEC,
    02 VECTOR_COOR(500)
POS FIN_REP(10)
01 POSICIONES(10),
    02 IG
    02 FG
    02 VENTANA
    02 IV
    02 FU
(BA_COM_GR, POS1, LON, A, B)
(TIPO, COD, VALIDO)
GENERA_ENTRY(BIN FIXED, BIN FIXED, BIN FIXED),
BORRA_INFORMACION_ENTRY(BIN FIXED, BIN FIXED, BIN FIXED, BIN FIXED, BIN FIXED),
(VLOR_V, VALOR)
GOTOXY_ENTRY(BIN FIXED, BIN FIXED),
VALIDAR_ENTRY(CHAR(70) VAR, CHAR(1), CHAR(1)),
(K, LON1, LON2, T)
CLAVE
CORRIGE_ENTRY(CHAR(120) VAR, BIN FIXED);

ON ENDFILE(IN) REBTN;
    FU = 0;
    GOTO FINLO;
END;
POST = 1;

JJ = 0;
CK = ASCII(11);
JJ = 1;
POS = 10;
MODIFICAR = "N";
TOPE_REPETITIVO = 0;
FIN_REP = 0;
CADENA_LEIDA = "";
CL = 0;
BA_COM_GR = 0;
UBICACION(1) = 1;
UBICACION(2) = 3;
UBICACION(3) = 8;
UBICACION(4) = 13;

OPEN FILE(POSICION) RECORD INPUT DIRECT KEYED;
READ FILE(COORDENADAS) INTO(EST_LEC) KEY(DIRE_COR);

IF DIRE_PANTA = 2 THEN DO;
    CALL GOTOXY(12, 7);
    PUT FILE(OUT) EDIT(CLAVE1)(A);
END;
IF DIRE_PANTA = 3 THEN DO;
    CALL GOTOXY(12, 3);
    PUT FILE(OUT) EDIT(CLAVE1)(A);
END;
INI WHILE;
DO WHILE(VECTOR_COOR(POS) ^= ASCII(255) & POS < FG);
/* AVERTIGUAR SI HAY CONTENZO DE GRUPO REPETITIVO */

```

```

BIN FIXED,
CHAR(1) VAR,
BIN FIXED,
CHAR(1) VAR,
BIN FIXED,
CHAR(1000) VAR,
FILE,

```

```

CHAR(1),
CHAR(1),

```

```

BIN FIXED,
BIN FIXED,
BIN FIXED,
BIN FIXED,
BIN FIXED,
BIN FIXED,
CHAR(1),

```

```

BIN FIXED, BIN FIXED, BIN FIXED, BIN FIXED, BIN FIXED, BIN FIXED),
CHAR(70) VAR,

```

```

BIN FIXED,
CHAR(120) VAR,

```

```

X = RANK(VECTOR_COOR(POS));
/* IF X = 253 THEN DO;
    TOPE_REPETITIVO = TOPE_REPETITIVO + 1;
    BA_COM_GR = 1;
    POSICION DE INICIO DEL GRUPO REPETITIVO
    POSICIONES.IG(TOPE_REPETITIVO) = POS + 5;
    POSICIONES.IV(TOPE_REPETITIVO) = RANK(VECTOR_COOR(POS+3));
    POSICIONES.FU(TOPE_REPETITIVO) = RANK(VECTOR_COOR(POS+4));
    POSICIONES.VENTANA(TOPE_REPETITIVO) = RANK(VECTOR_COOR(POS+2));
    FIN_REP = FIN_REP + 1;
    POSICIONES.FG(TOPE_REPETITIVO) = RANK(POS_FIN_REP(FIN_REP));
    POS = POS + 5;
END;
IF X = 254 THEN
    POS = POSICIONES.IG(TOPE_REPETITIVO);
X = RANK(VECTOR_COOR(POS)); */
IF X = 255 THEN
    GOTO FIN_WHILE;
Y = RANK(VECTOR_COOR(POS+1));
IF MODIFICAR = "S" THEN
    TIPO = "N";
ELSE
    TIPO = VECTOR_COOR(POS+2);
    COD = VECTOR_COOR(POS+3);
    LONGITUD = RANK(VECTOR_COOR(POS+4));
/* IF BA_COM_GR = 1 THEN
    CALL GENERA(DIRE_PANTA,POSICIONES(TOPE_REPETITIVO).IV,POSICIONES(TOPE_REPE
TITIVO).FU);*/
VALIDO = "N";
IF MODIFICAR = "S" THEN DO;
/* IF SUBSTR(CADENA_LEIDA,POS1,1) = ASCII(253) THEN DO;
    POS = RANK(SUBSTR(CADENA_LEIDA,POS1+1,1));
    POS1 = POS1 + 2;
END; */
LON = RANK(SUBSTR(CADENA_LEIDA,POS1,1));
VALOR_V = SUBSTR(CADENA_LEIDA,POS1+1,LON-1);
END;
DO WHILE(VALIDO = "N");
REPT: CALL GOTOXY(X,Y);
READ FILE(IN) INTO(VALOR);
LONGITUD_1 = LENGTH(VALOR);
IF LONGITUD_1 > LONGITUD THEN DO;
    CALL GOTOXY(0,22);
PUT FILE(OUT) EDIT("LA LONGITUD DEL DATO EXEDE LA LONGITUD ASIGNADA",CK)(A,A);
JJ = 1;
CALL BORRA_RESTO_LINEA;
GOTO REPT;
END;
/* IF (BA_COM_GR = 1 & (VALOR = "" ! LENGTH(VALOR) = 0)) THEN DO;
    POS = POSICIONES.FG(TOPE_REPETITIVO);
    TOPE_REPETITIVO = TOPE_REPETITIVO - 1;
    CADENA_LEIDA = CADENA_LEIDA !! ASCII(254);
    GOTO FIN_WHILE;
END; */
IF LENGTH(VALOR) ^= 0 THEN
    CALL VALIDAR(VALOR,COD,VALIDO);
IF (TIPO="N" ! MODIFICAR="S") & (VALOR="" ! LENGTH(VALOR)=0) THEN
    VALIDO = "S";
IF (TIPO="O" & (VALOR="" ! LENGTH(VALOR)=0)) THEN DO;
    CALL GOTOXY(0,22);
    PUT FILE(OUT) EDIT("LA INFORMACION ES REQUERIDA")(A);
    GOTO TTT;
END;
IF VALIDO = "N" THEN DO;

```

```

        JI = 1;
        CALL BORRA_RESTO_LINEA;
    END;
TTT:
    END;
    CI = CI + 1;
    IF MODIFICAR="S" & LENGTH(VALOR) < LENGTH(VALOR_V) THEN
        CALL BORRA_RESTO_LINEA;
    IF MODIFICAR = "N" THEN
        CALL ASIGNA;
    ELSE
        CALL REASIGNA;
    POS = POS + 5;
    FIN_WHILE;
    BA_COM_GR = 0;
END;
CALL GOTOXY(0,21);
PUT FILE(OUT) EDIT("DESEA MODIFICAR S/N                ")(A);
READ FILE(IN) INTO(MODIFICAR);
IF MODIFICAR = "S" THEN DO;
    POS = 1;
    POS1 = 1;
    GOTO INL_WHILE;
END;
IF (DIRE_PANTA=1!DIRE_PANTA=2!DIRE_PANTA=3!DIRE_PANTA=15!DIRE_PANTA=14) THEN DO;
    IF DIRE_PANTA = 1 THEN DO;
        DO I = 1 TO 4;
            CALL NORMALIZA;
        END;
        POS = 1;
        IC = 0;
        K = 1;
        I = 3;
AA: DO WHILE(IC < UBICACION(I));
        LON = RANK(SUBSTR(CADENA_LEIDA,POS,1));
        VALOR = SUBSTR(CADENA_LEIDA,POS+1,LON-1);
        POS = POS + LON;
        IC = IC + 1;
    END;
    IF K = 1 THEN DO;
        K = K + 1;
        CLAVE = VALOR;
        I = 4;
        GOTO AA;
    END;
    K = 1;
    I = 10;
BB: CALL GOTOXY(I,2);
    READ FILE(IN) INTO(NUM);
    IF LENGTH(NUM) ^= 1 THEN GOTO BB;
    IF NUM="*" THEN NUM="*";
    IF VERIFY(NUM,"123") ^= 0 THEN
        GOTO BB;
    IF ((NUM="2" & CLAVE="") | (NUM="3" & VALOR="")) THEN
        GOTO BB;
    IF K = 1 THEN DO;
        NUM1 = NUM;
        I = 27;
        K = K + 1;
        GOTO BB;
    END;
    IF K = 2 THEN DO;
        I = 47;

```