

Registro Observacional a través de Computadora

A computer software program for the coding of behavioral categories.

Alvaro Torres, Florente López y Daniel Zarabozo
Universidad Nacional Autónoma de México¹

RESUMEN

Se presenta un programa de cómputo ("software"), diseñado en lenguaje Pascal, que apoya tanto la recolección de observaciones como a la sistematización de los datos para su transferencia directa a "paquetes" de graficación y de análisis estadísticos.

Palabras clave: Observación del comportamiento, catálogo de comportamiento, codificación.

ABSTRACT

A Pascal software computer program, for the coding of behavior is presented. Data retrieval is arranged in such a way as to optimize its further analysis on other software packages.

Key words: Observational methodology, behavior catalogues, coding.

Los métodos de observación conductual constituyen una de las técnicas utilizadas para el estudio de diversos fenómenos de interacción social, como aquellos que se suscitan en ambientes escolares (Smith, 1978), en el medio familiar (Patterson, 1982), o en relaciones maritales (Gottman, 1979). La observación directa de comportamiento efectivo en los propios ambientes naturales, posibilita la comprensión de la eficacia adaptativa de las interacciones y las transformaciones de los patrones de interacción social a lo largo del tiempo.

1 Favor de enviar toda correspondencia a cualquiera de los autores a: Laboratorio de Psicología. Ap. Post. 21-182. Coyoacán, 04200. México, D.F. México. Esta investigación fue financiada por la Dirección General de Asuntos del Personal Académico, U.N.A.M. Proyecto: IN 500-391.

Tradicionalmente las observaciones conductuales se han realizado por medio de formatos prediseñados que contienen las categorías de observación, usando "tablas" de registro y cronómetros, o marcadores de tiempo auditivos (véase por ejemplo, López, 1987) luego de lo cual la información se "vacía" en hojas de "concentración" y posteriormente se introduce de modo manual en bases de datos o paquetes estadísticos. Este proceso es largo, poco económico, y se presta a equívocos y transformaciones de los datos. Sin embargo, haciendo uso de las micro-computadoras (algunas tan pequeñas y manuales como el tamaño de una hoja carta), la información puede capturarse directamente a través del teclado de las mismas y transferirse sin distorsiones y en un tiempo mínimo a sistemas de análisis de datos.

El programa Sistema de Registro Conductual Computarizado (SIRECC) se puede utilizar en cualquier computadora personal o "notebook" portátil. El SIRECC permite identificar los datos del registro (fecha y evento) y del registrador, así como acceder las categorías observacionales pertinentes al evento observado (hasta 9 categorías distintas y mantiene fija una décima categoría de "OTRAS CONDUCTAS"; véase figuras # 1), estas categorías son mostradas en la pantalla en la ubicación espacial correspondiente con el teclado numérico. Así, por ejemplo, si el observador define a la categoría # 1 como "hablar", ésta aparecerá en la parte inferior izquierda del recuadro de categorías, que corresponde con el número 1 del teclado numérico. Cada vez que el observador marque la ocurrencia de una categoría, el nombre correspondiente a la misma empezará a cintilar.

Registro Conductual Computarizado		
7	8	9
Categoría 7	Categoría 8	Categoría 9
Juego grupal	Etiqueta 8	Fuera de foco
4	5	6
Categoría 4	Categoría 5	Categoría 6
juego programado	juego diádico	inter momentánea
1	2	3
Categoría 1	Categoría 2	Categoría 3
conversación	inter normativa	inter negativa
0		
Categoría 0		
otras conductas		

D= definir etiquetas / C= continuar/ <Esc>= terminar

Figura 1. Muestra la disposición en pantalla de las teclas alfanuméricas correspondientes. La categoría seleccionada se mantiene cintilando hasta la selección de una nueva categoría. Las etiquetas que se presentan fueron previamente designadas para un registro particular, sin embargo, el investigador puede asignar las etiquetas pertinentes a su estudio.

Esta forma de presentación permanente, conspicua (cintilante) y en la misma ubicación dentro de la pantalla del monitor, permite que los observadores aprendan y memoricen de modo más rápido las categorías conductuales que se registran, ya que están en una relación unívoca con el teclado numérico de la computadora; asimismo impide las pérdidas de observación debidas al movimiento que realiza el registrador para anotar sus observaciones en la hoja de registro comúnmente utilizada.

El SIRECC contiene un archivo que permite utilizar un conjunto de categorías previamente definido y utilizado o bien accesar un nuevo conjunto de categorías cada vez que sea operado.

Asimismo, las teclas identificadas como (/), (*), (-) y (+) de mismo teclado numérico, también se pueden utilizar para el registro de eventos conductuales, en el uso particular de este programa permiten cualificar como "activo", "interrupción", "pausa" y "pasivo" respectivamente a cada una de las categorías conductuales que se registren. Por ejemplo, si el observador marca una ocurrencia de la categoría # 1 "hablar", puede cualificarla como "activo" presionando la tecla (/) lo cual hará que se muestre cintilando en el renglón inferior del recuadro de categorías la palabra ACTIVO.

El Sistema de Registro Conductual Computarizado contiene un reloj de tiempo real que automáticamente registra el momento de ocurrencia y la duración de una determinada categoría conductual, de igual forma registra el momento y la duración de una cualificación (véase figura # 2).

Este reloj interno de la computadora permite que el observador prescinda del cronómetro y elimina las pérdidas de observación de eventos del flujo conductual debidas a la lectura del cronómetro.

El SIRECC archiva los datos organizados de tal modo que permite su transferencia directa (en código ASCII) a diversos procesadores de texto, graficadores y paquetes estadísticos (p.e. Word Perfect, Harvard Graphics, Systat).

Un rasgo más del SIRECC es que analiza de modo inmediato diversas propiedades de los datos como la frecuencia de ocurrencia de las categorías, la frecuencia acumulada, la duración y la duración acumulada de cada una de ellas, y así como la secuencia de las conductas.

Por último, cabe destacar que este programa es altamente versátil, ya que puede servir para otros tipos de registro observacional, por ejemplo, como en el caso del registro de conducta animal en cuanto permite el registro de conducta "geográfica" en cajas de campo abierto, o para el registro de interacciones lingüísticas entre dos o más hablantes.

El programa está constituido por dos grandes secciones y un programa principal. La sección de rutinas generales contiene:

- 1.- Function ConstStr. Esta rutina permite disponer diversas líneas que enmarcan al conjunto de categorías que aparece en pantalla.

#	cat	f/a	dur
1	0	1	0.88
2	1	1	0.280
3	2	1	0.22
4	3	1	0.22
5	4	1	0.22
6	5	1	0.17
7	6	1	0.16
8	7	1	0.28
9	8	1	0.21
10	9	1	3.74
11	^	1	0.71
12	-	1	0.39
13	+	1	0.38
14	.	1	8.40
15	0	2	0.22
16	0	3	0.16
17	0	4	0.16
18	0	5	0.17
19	0	6	0.65
20	1	2	0.22
21	1	3	0.17
22	1	4	0.16
23	1	5	0.50
24	5	2	0.11
25	5	3	0.11
26	5	4	0.28
27	9	2	0.16
28	9	3	0.00
29	9	4	0.-05
30	8	2	0.00
31	8	3	0.17
32	^	2	0.33
33	^	3	0.16
34	^	4	0.11
35	^	5	1.21
36	5	5	0.33
37	1	6	0.33
38	0	7	1.10

Figura 2. Se presenta un listado de los registros de una breve sesión de observación. Los datos son almacenados en disco para su análisis posterior, con formato congruente con la mayoría de paquetes estadísticos procesadores de palabras o graficadores

2.- Function Existe. Esta rutina pregunta si existe un archivo en disco que ya contenga las categorías observacionales que se utilicen en un registro, regresa verdad si el archivo existe e ingresa dichas categorías en pantalla.

3.- Function Mayúsculas. Esta rutina, convierte una cadena de caracteres en mayúsculas.

4.- Procedure Encabezado. Esta rutina, junto con "Function ConstStr", permite disponer el marco visual del conjunto de categorías que aparece en pantalla.

5.- Procedure LimpiaP. Esta rutina permite limpiar el interior enmarcado de la pantalla donde aparecen las categorías.

6.- Procedure Suena. Esta rutina produce un tono de .85 segundos, y se utiliza para reportar errores de elección de acciones o de teclas a operar, o de rangos espaciales permitidos para movimiento de cursor, o de rangos permitidos para un máximo de caracteres.

7.- Procedure Pregunta. Esta rutina lee carácter en la última línea de la pantalla; su función primordial es permitir que el programa sea interactivo con el usuario, planteándole preguntas como: 'Ya existe archivo [B]orrar o [E]legir otro'.

8.- Procedure InputStr. Esta rutina permite acceder un nombre para el archivo de almacenaje de datos producidos durante la sesión de registro.

9.- Procedure Portada. Esta rutina permite imponer la portada inicial del programa en que se indican: la institución generada, el nombre del programa, los autores, y la función y capacidad del programa.

La sección de rutinas específicas contiene:

1.- Procedure Transcurrido. Esta rutina hace una resta entre un "gettime" inicial y uno final, la diferencia resultante constituye la duración de la categoría en cuestión.

2.- Procedure NombreDeArchivo. Esta rutina, junto con InputStr, permite acceder un nombre para el archivo de almacenaje de datos producidos durante la sesión de registro.

3.- Procedure Pantalla. Esta rutina tiene como función primordial disponer en pantalla información fija (números correspondientes a las teclas numéricas y leyendas "etiqueta #"), así como las categorías conductuales a registrar; en conjunción con otras rutinas accesa las categorías existentes en archivo o las categorías definidas en el momento.

4.- Procedure Catego. Esta rutina permite registrar el inicio de una categoría conductual y disponer a ésta en pantalla en una modalidad conspicua (cintilante). Asimismo, contiene diversos contadores que permiten generar un número ordinal progresiva para indicar el número de evento registrado, y un número acumulado que indica la frecuencia acumulada de la categoría en cuestión.

5.- Procedure Inicialización. Esta rutina permite disponer a ceros a todas las variables en uso, con el propósito de eliminar datos acumulados no deseados, en conjunción con otras rutinas permite asignar para uso de las categorías contenidas en el archivo "ReginCond.Txt" que contiene las categorías definidas en un uso previo del programa.

6.- Procedure DefinirEtiquetas. Esta rutina permite, si así se desea, definir nuevas categorías de registro a usar en la sesión en curso.

6.1- Procedure CentrarEtiqueta. Esta rutina permite centrar a las categorías definidas en el momento o a las predefinidas en archivo.

El listado del programa se presenta en el anexo 1.

**LISTADO DEL PROGRAMA "SISTEMA DE REGISTRO CONDUCTUAL
COMPUTARIZADO" EN LENGUAJE PASCAL.**
{ Sistema de Registro Conductual Computarizado }

PROGRAM SIReCCO;

[.....]

SIReCCO.PAS

REGISTRO CONDUCTUAL COMPUTARIZADO.

ÁLVARO TORRES CHÁVEZ
FLORENTE LÓPEZ RODRÍGUEZ
DANIEL ZARABOZO E. DE R.

DEPARTAMENTO DE ANÁLISIS EXPERIMENTAL DE LA CONDUCTA.
UNIDAD DE CÓMPUTO, INFORMÁTICA E INVESTIGACIÓN.
FACULTAD DE PSICOLOGÍA.
U.N.A.M.

FEBRERO DE 1991.

.....)

USES

Dos, CRT;

LABEL

ESCAPE;

CONST

MAXETI = 10;

TYPE

STR80 = STRING[80];
CHARSET = SET OF CHAR;

VAR

ETIQUETA	: ARRAY [0..MAXETI] OF STR80;
CH,DICE	: CHAR;
IG,JG,KG, C1,C2,C3,C4,C5,C6,C7,C8, C9,C10,C11,C12,C13,C14, CONTADOR, CONT	: INTEGER;
SUMA	: REAL;
NOMARCH:S	: STR80;
HRS5,MIN5,SEG5,CTS5,H2,M2,S2,CTS2, SUMA1,SUMA2	: WORD;
F	: TEXT;

FUNCTION CONSTRSTR (C : CHAR ; N : INTEGER) : STR80;

{ CONSTRSTR REGRESA UNA CADENA CON N CARACTERES DE VALOR C }

VAR S: STRING[80];

BEGIN

IF N < 0 THEN N := 0; S[0] := CHR(N); FILLCHAR(S[1],N,C); CONSTRSTR := S;

END; { CONSTRSTR }

FUNCTION EXISTE (NOMARCHIVO: STR80) : BOOLEAN;

{ REGRESA VERDAD SI EL ARCHIVO DE TEXTO LLAMADO ARCHIVO EXISTE EN DISCO. }

```

VAR OK : BOOLEAN; FKK : TEXT;
BEGIN
  ASSING (FKK.NOMARCHIVO);   {$!-}; RESET (FKK);   {$!+}; OK:= (IORESULT = 0);
  IF OK THEN CLOSE (FKK);   EXISTE:= OK;
END; {EXISTE}

FUNCTION MAYUSCULAS (S : STR80) : STR80;
{ CONVIERTE UNA CADENA DE CARACTERES EN MAYÚSCULAS }
VAR P : INTEGER;
BEGIN
  FOR P:= 1 TO LENGTH (S) DO S[P]:= UpCase (S[P] ; MAYUSCULAS:= S;
END; {MAYÚSCULAS}

PROCEDURE ENCABEZADO (TIPO: INTEGER; LETRERO: STR80);
VAR GUION : CHAR;
    AUX. COL : INTEGER;
BEGIN
  GUION := '=';
  IF TIPO IN [1,2]
    THEN BEGIN
      IF TIPO = 1
        THEN BEGIN
          AUX := (78 - LENGTH(LETRERO)) DIV 2;
          GoToXY (1,1);
          TEXTCOLOR (YELLOW);
        END
      ELSE BEGIN
          AUX := (40 -LENGTH(LETRERO)) DIV 2;
          LETRERO := MAYUSCULAS (LETRERO);
          COL := (80 - (2*AUX + LENGTH (LETRERO))) DIV 2;
          GoToXY (COL, 1);
          TEXTCOLOR (YELLOW);
        END;
      CLREOL;
      WRITE (CONSTSTR (GUION, AUX), "");
      TEXTCOLOR (WHITE);
      WRITE (LETRERO, "");
      TEXTCOLOR (YELLOW);
      WRITE (CONSTSTR(GUION,AUX));
    END
  ELSE BEGIN
    GoToXY (1,1); CLREOL;
    WRITE (LETRERO);
  END;
  TEXTCOLOR (YELLOW);
  GoToXY (1,24); WRITE (CONSTSTR(GUION,79));
  TEXTCOLOR (LIGHTGRAY);
END;

PROCEDURE LIMPIAP (LINI,LFIN: BYTE);
VAR I : BYTE;
BEGIN
  FOR I:= LINI TO LFIN DO
    BEGIN
      GoToXY (1,i). CLREOL;
    END;
END; {LIMPIAP}

PROCEDURE SUENA; {PRODUCE TONO.}
BEGIN SOUND (850); DELAY (150); NOSOUND; END;

```

```

PROCEDURE PREGUNTA (LINEA: STR80; TERM: CHARSET; VAR TC: CHAR);
{LEE CARÁCTER EN LA ÚLTIMA LÍNEA DE LA PANTALLA.}
VAR CH: CHAR;
BEGIN
  TEXTCOLOR (WHITE);
  GoToXY (1,25); CLREOL; WRITE (LINEA);
REPEAT
  CH:= READKEY; TC:= UPCASE (CH); IF NOT (TC IN TERM) THEN SUENA;
UNTIL TC IN TERM;
WRITE (TC); GoToXY(1,25); CLREOL;
TEXTCOLOR (LIGTHGRAY);
END; {PREGUNTA}

PROCEDURE INPUTSTR (LO: CHAR;
VAR S: STR80;
L, Y, X: INTEGER;
TERM: CHARSET;
VAR TC: CHAR);

CONST RAYA = '_';
VAR FKEY: BOOLEAN; P: INTEGER; CH: CHAR; VALIDOS: CHARSET;
BEGIN
  TEXTCOLOR (WHITE);
  CASE LO OF
    'L': VALIDOS:= [#32,#40,#41,#44..46,#48..#58,#65..#90,#92,#97..#122,
#130,160..#165];
    'N': VALIDOS:= [#45,#46,#48..#57];
  END; {CASE}
  GoToXY(Y,X); WRITE(S,CONSTSTR(RAYA,L - LENGTH (S)));
  P:= 0;
  FKEY:= FALSE;
REPEAT
  GoToXY(Y + P, X);
  CH:= READKEY;
  IF CH:= #0 {TECLA ESPECIAL}
  THEN
  BEGIN
    FKEY:= TRUE;
    CH:= READKEY;
    CASE CH OF
      #15: CH:= ^A; {SHIFT TAB}
      #71: CH:= ^O; {HOME}
      #72: BEGIN CH:= ^E; {FL.ARR.} FKEY:= FALSE; END;
      #75: BEGIN CH:= ^S; {FL.IZO.} FKEY:= FALSE; END;
      #77: BEGIN CH:= ^D; {FL.DER.} FKEY:= FALSE; END;
      #79: CH:= ^L; {END}
      #80: BEGIN CH:= ^X; {FL.ABA.} FKEY:= FALSE; END;
      #83: BEGIN CH:= ^G; {DEL} FKEY:= FALSE; END;
    END; {CASE}
  END;
  CASE CH OF
    #9: CH:= ^F; {TAB}
    #32. #165: IF NOT FKEY
      THEN
        IF CH IN VALIDOS
          THEN
            IF P < L
              THEN BEGIN
                IF LENGTH(S) = L THEN DELETE (S,L,1);
                P:= P + 1;
                INSERT(CH,S,P);
                WRITE(COPY(S,P,L));
                CH:= '';
              END;
            {OJO!}
          END;
        END;
      END;
    END;
  END;

```

```

        END
    ELSE SUENA;
^S: IF P > 0 THEN P:= P - 1 ELSE SUENA;
^D: IF P > LENGTH(S) THEN P:= P + 1 ELSE SUENA;
^G: IF P > LENGTH(S)
    THEN BEGIN
        DELETE(S,P + 1,1);
        WRITE (COPY (S,P + 1, L) RAYA);
    END;
^H, #127: IF P > 0
    THEN BEGIN
        DELETE(S,P,1);
        WRITE(^H,COPY(S,P,L),RAYA);
        P:= P - 1;
    END
    ELSE SUENA;
^Y: BEGIN
    WRITE(CONSTSTR(RAYA,LENGTH(S) - P));
    DELETE(S,P + 1,L);
    END;
    ELSE IF (NOT (CH IN TERM)) AND NOT FKEY THEN SUENA;
    END; {OF CASE}
UNTIL FKEY OR (CH IN TERM);
P:= LENGTH(S);
GoToXY(Y + P, X);
WRITE(" :L - P);
TC:= CH;
TEXTCOLOR (LIGHTGRAY);
END; {INPUTSTR}

PROCEDURE PORTADA;
CONST COL = 18;
VAR REN : BYTE;
BEGIN
CLRSCR; TEXTCOLOR (YELLOW); TEXTBACKGROUND (BLACK);
REN := 1;

GoToXY (COL.REN);      WRITE ('... ..')
GoToXY (COL.REN+1);   WRITE ('... ..')
GoToXY (COL.REN+2);   WRITE ('... ..')
GoToXY (COL.REN+3);   WRITE ('... ..')
GoToXY (COL.REN+4);   WRITE ('... ..')
GoToXY (COL.REN+5);   WRITE ('... ..')
TEXTCOLOR (WHITE); TEXTBACKGROUND (BLACK)
REN :=8;

GoToXY (COL+13,REN);  WRITE ('FACULTAD DE PSICOLOGIA');
TEXTCOLOR (WHITE); TEXTBACKGROUND (BLUE);
REN := 11;

GoToXY (COL.REN);      WRITE(' ');
GoToXY (COL.REN+1);    WRITE(' SISTEMA DE REGISTRO CONDUCTUAL COMPUTARIZADO ');
GoToXY (COL.REN+2);    WRITE(' ');
REN :=15;

GoToXY (COL.REN);      WRITE(' ');
GoToXY (COL.REN + 1);  WRITE(' REGISTRO DE HASTA 10 CATEGORÍAS ');
GoToXY (COL.REN + 2);  WRITE(' CONDUCTUALES EN EL TIEMPO ');
GoToXY (COL.REN + 3);  WRITE(' ');
GoToXY (COL.REN + 4);  WRITE(' FEBRERO DE 1991 ');
GoToXY (COL.REN + 5);  WRITE(' ');
GoToXY (COL.REN + 6);  WRITE(' ');
TEXTCOLOR (LIGHTGRAY); TEXTBACKGROUND (BLACK);
REN :=22;
    
```

```

GoToXY (COL,REN); WRITE(
GoToXY (COL,REN+ 1); WRITE(
GoToXY (COL,REN+ 2); WRITE(
GoToXY (COL,REN+ 3); WRITE(
GoToXY (73,1); TEXTCOLOR (LIGHTGRAY+BLINK); WRITE (' <ENTER>');
REPEAT CH := READKEY; UNTIL CH = #13;
TEXTCOLOR (LIGHTGRAY);
CLRSCR;
END;

{RUTINAS ESPECIFICAS DEL SIRECCO. PAS}

PROCEDURE TRANSCURRIDO (SUMA1 : WORD;
VAR SUMA2 : WORD);
BEGIN
SUMA2 := SUMA2-SUMA1;
END; {TRANSCURRIDO}

PROCEDURE NOMBRE DE ARCHIVO:
LABEL ESCAPE;
BEGIN
NOMARCH := "";
REPEAT
GOTOXY (1,25);
WRITE ('NOMBRE DEL ARCHIVO PARA LOS DATOS:');
REPEAT
INPUTSTR ('L', NOMARCH, 27,36,25,[#13, #27], CH);
NOMARCH := MAYUSCULAS (NOMARCH);
UNTIL NOMARCH "";
IF EXISTE (NOMARCH)
THEN
PREGUNTA ('YA EXISTE '+NOMARCH+' [B]ORRAR O [E]LEGIR OTRO:');
IF CH = #27 THEN GOTO ESCAPE;
UNTIL (NOT EXISTE (NOMARCH) OR (CH = 'B');
S := 'ARCHIVO DE DATOS: 'NOMARCH';
GOTOXY ((79-LENGTH(S)) DIV 2,22);
TEXTCOLOR (WHITE);
WRITE (S);
TEXTCOLOR (LIGHTGRAY);
ASSIGN (F, NOMARCH);
REWRITE(F);
ESCAPE:
END: (DATOS)

PROCEDURE PANTALLA:
CONST
COL1 = 10;
COL2 = 9;
COL3 = 2;
ESP1 = 26;
VAR
HA YETI : BOOLEAN;
COL : BYTE;
BEGIN
LIMPIAP (2,23);
TEXTCOLOR (LIGHTGRAY);
GOTOXY (5,3); WRITELN (CONSTSTR ('-', 70);
TEXTCOLOR (WHITE);
GOTOXY (COL1,5); WRITELN ('7:5:8:ESP1,9:ESP1);
GOTOXY (COL1,9); WRITELN ('4:5:5:ESP1,6:ESP1);
GOTOXY (COL1,13); WRITELN ('1:5:2:ESP1,3:ESP1);
GOTOXY (COL1,17); WRITELN ('0:5:

```

ALVARO TORRES CHÁVEZ / DANIEL ZARABOZO
FLORENTE LÓPEZ

```

GoToXY (COL,REN); WRITE(
GoToXY (COL,REN+ 1); WRITE(
GoToXY (COL,REN+ 2); WRITE(
GoToXY (COL,REN+ 3); WRITE(
GoToXY (73,1); TEXTCOLOR (LIGHTGRAY+BLINK); WRITE (' <ENTER>');
REPEAT CH := READKEY; UNTIL CH = #13;
TEXTCOLOR (LIGHTGRAY);
CLRSCR;
END;

{RUTINAS ESPECIFICAS DEL SIRECCO. PAS}

PROCEDURE TRANSCURRIDO (SUMA1 : WORD;
VAR SUMA2 : WORD);
BEGIN
SUMA2 := SUMA2-SUMA1;
END; {TRANSCURRIDO}

PROCEDURE NOMBRE DE ARCHIVO:
LABEL ESCAPE;
BEGIN
NOMARCH := "";
REPEAT
GOTOXY (1,25);
WRITE ('NOMBRE DEL ARCHIVO PARA LOS DATOS:');
REPEAT
INPUTSTR ('L', NOMARCH, 27,36,25,[#13, #27], CH);
NOMARCH := MAYUSCULAS (NOMARCH);
UNTIL NOMARCH "";
IF EXISTE (NOMARCH)
THEN
PREGUNTA ('YA EXISTE '+NOMARCH+' [B]ORRAR O [E]LEGIR OTRO:'
['B', 'E', #27], CH);
IF CH = #27 THEN GOTO ESCAPE;
UNTIL (NOT EXISTE (NOMARCH) OR (CH = 'B');
S := 'ARCHIVO DE DATOS: 'NOMARCH';
GOTOXY ((79-LENGTH(S)) DIV 2,22);
TEXTCOLOR (WHITE);
WRITE (S);
TEXTCOLOR (LIGHTGRAY);
ASSIGN (F, NOMARCH);
REWRITE(F);
ESCAPE:
END: (DATOS)

PROCEDURE PANTALLA:
CONST
COL1 = 10;
COL2 = 9;
COL3 = 2;
ESP1 = 26;
VAR
HA YETI : BOOLEAN;
COL : BYTE;
BEGIN
LIMPIAP (2,23);
TEXTCOLOR (LIGHTGRAY);
GOTOXY (5,3); WRITELN (CONSTSTR ('-', 70);
TEXTCOLOR (WHITE);
GOTOXY (COL1,5); WRITELN ('7':5,'8':ESP1,'9':ESP1);
GOTOXY (COL1,9); WRITELN ('4':5,'5':ESP1,'6':ESP1);
GOTOXY (COL1,13); WRITELN ('1':5,'2':ESP1,'3':ESP1);
GOTOXY (COL1,17); WRITELN ('0':5;

```

```

CONTADOR := CONTADOR+1;
S1 := ETIQUETA[ORD(NUMCAT) - ORD('0')];

CASE NUMCAT OF
  '0' : COL := COL3;
  '1','4','7' : COL := COL3;
  '2','5','8' : COL := COL3+26;
  '3','6','9' : COL := COL3+52;
  '/' : COL := COL3+25;
  '*' : COL := COL3+35;
  '.' : COL := COL3+49;
  '+' : COL := COL3+63;
END;

CASE NUMCAT OF
  '0' : REN := 19;
  '1'..'3' : REN := 15;
  '4'..'6' : REN := 11;
  '7'..'9' : REN := 7;
  '/'..'+' : REN := 19;
END;

CASE NUMCAT OF
  '0' : C1:=C1+1;
  '1' : C2:=C2+1;
  '2' : C3:=C3+1;
  '3' : C4:=C4+1;
  '4' : C5:=C5+1;
  '5' : C6:=C6+1;
  '6' : C7:=C7+1;
  '7' : C8:=C8+1;
  '8' : C9:=C9+1;
  '9' : C10:=C10+1;
  '/' : C11:=C11+1;
  '*' : C12:=C12+1;
  '.' : C13:=C13+1;
  '+' : C14:=C14+1;
END;

CASE NUMCAT OF
  '0' : CONT :=C1;
  '1' : CONT :=C2;
  '2' : CONT :=C3;
  '3' : CONT :=C4;
  '4' : CONT :=C5;
  '5' : CONT :=C6;
  '6' : CONT :=C7;
  '7' : CONT :=C8;
  '8' : CONT :=C9;
  '9' : CONT :=C10;
  '/' : CONT :=C11;
  '*' : CONT :=C12;
  '.' : CONT :=C13;
  '+' : CONT :=C14;
END;

IF DICE = '/' THEN S1 := ' ACTIVO ';
IF DICE = '*' THEN S1 := ' PASIVO ';
IF DICE = '.' THEN S1 := ' INTERRUPCION ';
IF DICE = '+' THEN S1 := ' PAUSA ';
TEXTCOLOR (14+BLINK);
GOTOXY(COL,REN); WRITE (S1);
TEXTCOLOR (WHITE);
LECTURA := UPCASE(READKEY);

```

```

TEXTCOLOR (LIGHTGRAY);
GOTOXY (COL2,6);
WRITELN ('CATEGORIA 7', 'CATEGORIA 8':ESP1, 'CATEGORIA 9':ESP1);
GOTOXY (COL2,10);
WRITELN ('CATEGORIA 4', 'CATEGORIA 5':ESP1, 'CATEGORIA 6':ESP1);
GOTOXY (COL2,14);
WRITELN ('CATEGORIA 1', 'CATEGORIA 2', 'CATEGORIA 3':ESP1);
GOTOXY (COL2, 18);
WRITELN ('CATEGORIA 0');
GOTOXY (5,21); WRITELN (CONSTSTR ('-', 70);
HAYETI := FALSE;
FOR IG := 0 TO MAXETI DO
  IF ETIQUETA[IG] ""
    THEN HAYETI := TRUE;
IF HAYETI
  THEN
    BEGIN
      TEXTCOLOR (WHITE);
      COL := COL3;
      FOR IG := 7 TO 9 DO
        BEGIN
          GOTOXY (COL, 7);
          WRITE (ETIQUETA[IG]);
          COL := COL+26;
        END;
      COL := COL3;
      FOR IG := 4 TO 6 DO
        BEGIN
          GOTOXY (COL, 11);
          WRITE (ETIQUETA[IG]);
          COL := COL+26;
        END;
      COL := COL3;
      FOR IG := 1 TO 3 DO
        BEGIN
          GOTOXY (COL, 15);
          WRITE (ETIQUETA[IG]);
          COL := COL+26;
        END;
      COL := COL3;
      FOR IG := 0 TO 0 DO
        BEGIN
          GOTOXY (COL, 19);
          WRITE (ETIQUETA[IG]);
          COL := COL+26;
        END;
      TEXTCOLOR (LIHTGRAY);
    END;
END; {PANTALLA}

PROCEDURE CATEGO (NUMCAT:CHAR
                  VAR H.M.S.C: WORD;
                  VAR LECTURA: CHAR);

CONST COL3 = 2;
VAR COL.REN : BYTE;
    S1 : STR80;
BEGIN
  GETTIME (H.M.S.C);
  H := H*60*60*100;
  M := M*60*100;
  S := S*100;
  SUMA1 := (H+M+S+C);

```

```

CASE CH OF
  ^X,#13 : IF (REN-4) MAXE TI
    THEN REN := REN + 1
    ELSE SUENA;
  ^E : IF (REN-4) 1
    THEN REN := REN - 1
    ELSE SUENA;
END;
UNTIL CH = #27;
FOR IG := 0 TO MAXE TI DO
  CENTRAR E TIQUETA (ETIQUETA[IG]);
PREGUNTA ('GUARDAR ETIQUETAS EN DISCO / S = SI / N = NO ', ['N', 'S'], CH);
UNTIL CH IN ['N', 'S'];
IF CH = 'S'
  THEN
    BEGIN
      ASSIGN (F, 'REGI COND. TXT'); REWRITE (F);
      FOR IG := 0 TO MAXE TI DO
        WRITELN (F, ETIQUETA[IG]);
      CLOSE (F);
    END;
END; {DEFINIR ETIQUETAS}

PROCEDURE CODIGOS;
BEGIN
  CASE DICE OF
    '0' : WRITE (F, 'OC':5);
    '1' : WRITE (F, 'II': 5);
    '2' : WRITE (F, 'IN': 5);
    '3' : WRITE (F, 'I-': 5);
    '4' : WRITE (F, 'JP': 5);
    '5' : WRITE (F, 'JJ': 5);
    '6' : WRITE (F, 'OS':5);
    '7' : WRITE (F, 'AA':5);
    '8' : WRITE (F, 'NU':5);
    '9' : WRITE (F, 'CO':5);
    '/' : WRITE (F, 'AC':5);
    '*' : WRITE (F, 'PA':5);
    '-' : WRITE (F, 'BK':5);
    '+' : WRITE (F, 'DE':5);
  END;
END; {CODIGOS}

{PROGRAMA PRINCIPAL}
BEGIN
  CLRSCR;
  INICIALIZACION;
  PORTADA;
  ENCABEZADO (1, 'REGISTRO CONDUCTUAL COMPUTARIZADO');
  REPEAT
    PANTALLA;
    PREGUNTA ('D= DEFINIR ETIQUETAS / C= CONTINUAR / <ESC> = TERMINAR',
      ['C', 'D', #27], CH);
    IF CH = #27 THEN GOTO ESCAPE;
    IF CH = 'D' THEN DEFINIR ETIQUETAS;
  UNTIL CH = 'C';
  NOMBRE DE ARCHIVO;
  PREGUNTA ('ENTER = PRINCIPIAR / <ESC> = TERMINAR ', [#13, #27], CH);
  IF CH = #27 THEN GOTO ESCAPE;
  WRITELN (F, ' # CAT F/A DUR'); WRITELN (F);
  REPEAT
    IF DICE IN [ '0' .. '9', '/', '*', '-', '+' ]
      THEN

```

```
BEGIN
  WRITE (F,CONTADOR+1:3);
  CODIGOS;
  TEXTCOLOR (WHITE);
  GOTOXY (60,25); WRITE (' <ESC> PARA TERMINAR:');
  TEXTCOLOR (LIGHTGRAY);
  CATEGO (DICE,H2,M2,S2,CTS2,DICE);
  GETTIME(HRS5,MIN5,SEG5,CTS5);
  HRS5 := HRS5*60*60*100;
  MIN5 := MIN5*60*100;
  SEG5 := SEG5*100;
  SUMA2:= (HRS5+MIN5+SEG5+CTS5);
  TRANSCURRIDO (SUMA1, SUMA2);
  SUMA:=SUMA2;
  WRITELN(F,CONT:5,SUMA/100:8:2);
  END
ELSE DICE := UPCase(READKEY);
UNTIL DICE = #27;
CLOSE(F);
ESCAPE: CLRSCR;
END.
```

REFERENCIAS

- Gottman, J.M. (1979) *Marital interaction: Experimental investigations*. Nueva York: Academic Press.
- López R., Florente (1987) Un indicador de tiempo auditivo, con base temporal ajustable y dos opciones de salida. *Revista Mexicana de Análisis de la Conducta*, 13, 71-73.
- Patterson, G.R. (1982) *Coercive family process*. Eugene, OR: Castalia Press.
- Smith, P.K. (1978) A longitudinal study of social participation in preschool children: Solitary and parallel play reexamined. *Developmental Psychology*, 14, 517-523.