Psicothema, 1994. Vol 6, n.º 1, pp. 105-118
ISSN 0214 - 9915 CODEN PSOTEG

SOFTWARE, INSTRUMENTACION Y METODOLOGIA

FRA3: UN PROGRAMA EN PASCAL PARA LA INVESTIGACION SOBRE LOS PROCESOS DE COMPRENSION EN LA LECTURA

Ramón López Sánchez
Departamento de Psicología Básica II: Procesos Cognitivos. Universidad Complutense de Madrid

FRA3 puede considerarse como una aplicación (en lenguage Pascal) de una técnica de lectura autoespaciada no-acumulativa que resulta especialmente útil en la investigación sobre el proceso de comprensión lectora. FRA3 presenta el material verbal dividido en fragmentos. También puede incorporar una tarea de decisión léxica en un punto crítico de estos materiales. El fichero de datos de cada sujeto incluye los tiempos de lectura de cada uno de los fragmentos y el tiempo de la decisión léxica. El código fuente se ha dividido en procedimientos para facilitar la descripción de FRA3.

Palabras clave: lenguaje Pascal, lectura autoespaciada no acumulativa, decisión léxica, material verbal, tiempos de lectura, tiempo de decisión.

FRA3: A Pascal application for reading-comprehension research. FRA3 could be considered as an application (in Pascal language) of a no-cummulative self-paced reading task that is specially useful in reading-comprehension process research. FRA3 displays verbal material divided in fragments. It can incorporate a lexical decision task in a critical point of these materials too. Each subject's data file includes reading times for each of the fragments and the lexical decision time. The program's original code has been divided in procedures to facilitate FRA3 description.

Key words: Pascal language, no-accumulative self-paced reading task, lexical decision, verbal materials, reading times, decision time.

FRA3 se diseñó con el propósito de utilizar una técnica de lectura auto-espaciada no-acumulativa en las investigaciones sobre el procesamiento de frases localmente ambiguas en la lectura (Rayner et al., 1989; López Sánchez, 1992). Los enunciados ambiguos funcionan como *unidades neutralizadas* que permiten estudiar la fuerza de los factores que condicionan la desambiguación (Mayor, 1979, 1980; Mayor y Moya, 1991). Los estudios realizados en este ámbito han empleado generalmente frases que contienen una *ambigüedad sintáctica local* (frases que son ambiguas hasta un punto pero que incluyen información posterior que las desambigua). Los estudios sobre el procesamiento de la ambigüedad sintáctica han puesto de manifiesto lo que hacen los lectores ante una situación de incertidumbre, y han proporcionado distintas respuestas, desde distintas posiciones teóricas, en relación con tres cuestiones críticas sobre la arquitectura funcional del analizador del lenguaje natural: la estructura del procesador, la disposición temporal de los procesos, y el modo de comunicación entre los componentes del sistema (Frazier, 1987; Taraban y McClelland, 1988; Steedman y Altmann, 1989). En general las hipótesis sobre la arquitectura funcional del procesador se pueden reducir a dos: la concepción modular (Forster, 1979; Fodor,

Correspondencia a: Ramón López Sánchez
Dpto. Psicología Básica II: Procesos Cognitivos.
Facultad de Psicología. Universidad Complutense.
Campus de Somosaguas. 28223 Madrid. España

1983) y la concepción interactiva (Tyler y Marslen-Wilson, 1977; Marslen-Wislon y Tyler, 1980, 1987; McClelland, 1987).

La representación final de una frase o de un texto es insuficiente para conocer la forma en que se construye la estructura y el significado de una frase palabra a palabra. Por ello se han desarrollado una gran cantidad de técnicas para explorar los procesos que tienen lugar durante la lectura. Una de estas tareas es *la lectura auto-espaciada palabra a palabra,* en la que los sujetos presionan un botón para leer cada nueva palabra (Mitchell y Green, 1978; Just, et al., 1982; Ferreira y Henderson, 1990). La presentación de las frases en esta modalidad puede ser *acumulativa* (las palabras se van acumulando sucesivamente en la pantalla a medida que los sujetos pulsan el botón) o *no acumulativa* (las que ya han sido presentadas desaparecen de la pantalla). Otras modalidades de lectura auto-espaciada pueden incluir la presentación de unidades más largas de un texto, como una cláusula o una frase. Una variedad de esta técnica es la que combina la *lectura auto-espaciada* con una *tarea secundaria,* en la que los sujetos leen un fragmento hasta una palabra y en este punto deben denominar la palabra o tomar una decisión léxica sobre la misma (Clifton, Frazier y Connine, 1984; Marslen-Wilson, Brown y Tyler, 1988). La lectura auto-espaciada acumulativa ha sido frecuentemente criticada puesto que los sujetos adoptan estrategias especificas de la tarea (por ejemplo, presentar rápidamente la frase y leerla posteriormente una vez que están disponibles todas las palabras) (Just et al., 1982; Ferreira y Henderson, 1990). Los resultados más próximos a los obtenidos con el registro de los movimientos oculares, que hoy por hoy resultan los más completos y precisos (Ferreira y Henderson, 1990), se han observado utilizando la técnica de lectura auto-espaciada no acumulativa palabra a palabra y la RSVP (Just, Carpenter y Woolley, 1982; Ferreira y Clifton, 1986; Ferreira y Henderson, 1990).

Existen en el mercado algunos programas y paquetes de programas que permiten la implementación de las técnicas de lectura auto-espaciada acumulativa y no acumulativa. Algunos de estos son el MEL (Micro Experimental Laboratory) desarrollado por *Psychology Software Tools Inc.* y el Mouselab System Program (Johnson et al., 1989). El último programa presenta algunos problemas puesto que los sujetos deben desplazarse a través del texto utilizando un ratón, y lógicamente, la habilidad de los mismos en su manejo puede influir claramente en los resultados. El MEL es un paquete integrado de programas que permite implementar una variedad de tareas clásicas para la investigación en Psicología Cognitiva. Sin embargo, a parte de su coste económico, resulta difícil establecer los parámetros del diseño que permiten llevar a cabo un experimento particular, y no presenta la posibilidad de incluir una tarea secundaria en un punto de las frases.

La aplicación desarrollada en este caso es un caso particular de la técnica de lectura auto-espaciada cláusula a cláusula, donde se presenta una frase básica (dos primeros fragmentos) que puede tener tres terminaciones posibles (tercer fragmento). El programa ofrece también la posibilidad de incluir una tarea de decisión léxica en el punto donde aparece la ambigüedad local de las frases (después del segundo fragmento). La estructuración del programa en procedimientos, hace que FRA3 pueda adaptarse fácilmente a las necesidades de investigación de cada usuario, y también permite una fácil adaptación a cualquiera de las modalidades de lectura auto-espaciada.

I. DESCRIPCION DEL PROGRAMA

La compilación del código fuente que aparece descrito en este apartado se realizó con el Turbo-Pascal v. 3.01.

Los comentarios en algunas lineas del programa aparecen encerrados entre asteris

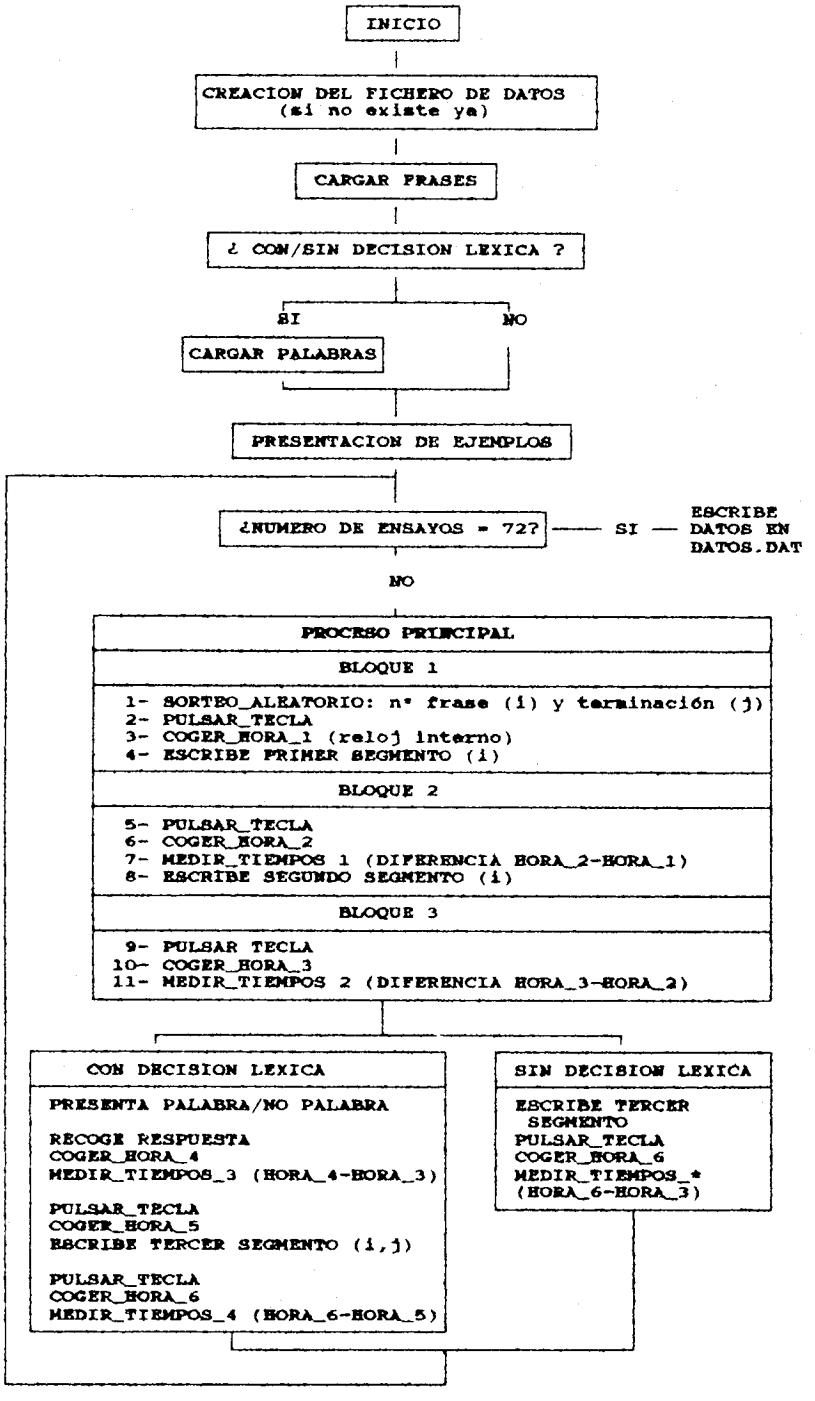

Figura 1: Diagrama de flujo que describe el funcionamiento general de FRA3.

cos y paréntesis, respetando así el modo de aparición de los mismos en Pascal. Tengase en cuenta que el orden de los procedimientos ha sido alterado para conseguir una mayor claridad expositiva y para ajustarse al diagrama de funcionamiento general del programa.

Los procedimientos que aparecen tras la definición de las variables ocuparían de forma obligatoria la última parte del programa en su versión ejecutable.

La Figura 1 muestra un diagrama de flujo del funcionamiento de FRA3.

1.1.–Definición de las variables y sus tipos.

A continuación aparecen la definición de los tipos y las variables utilizadas por el programa junto con anotaciones aclaratorias para identificarlas de una forma más precisa.

```
Program Frases;
Type
  tiempo = record
         seg : integer;(* segundos *)
         cent: integer;(* centésimas de segundo *)
        end;
  numero = record
         fra : integer;(* Número de frase (1-24) *)
         term: integer;(* Número de terminacion (1,2,3) *)
        end;
  reg    = record
         c_s  : char;(* 1-Con palabras, 2-Sin palabras *)
         num  : array [1..72] of numero;(* Nº de frase y
terminac. *)
         med1 : array [1..72] of tiempo;
         med2 : array [1..72] of tiempo;
         med3 : array [1..72] of tiempo;
         med4 : array [1..72] of tiempo;
         bien : integer;
         mal  : integer;
        end;
var
datos:file of reg;
aux:reg;
T1: array [1..24] of string[49];   (* 1ª parte de cada
    frase *)
T2: array [1..24] of string[28];   (* 2ª *)
T3: array [1..24,1..3] of string[30];(* 3ª *)
E1: array [1..4] of string[49];   (* 1ª parte ejemplos *)
E2: array [1..4] of string[28];   (* 2ª *)
E3: array [1..4] of string[30];(* 3ª *)
PALABRAS:array [1..24,1..3] of string[9];(*Palabras
                de cada frase*)
T : array [1..4] of string[9];(* Palabras que salen en
    ejemplos *)
SUCESO: array [1..24,1..3] of integer;(* Control de las
            que han salido *)
con,ok:boolean;
cs,correcto,palabra,resp: char;
x,y,h1,h2,h3,h4,h,m1,m2,m3,m4,m,s1,s2,s3,s4,s,d1,d2
,d3,d4,d: integer;
h5,m5,s5,d5,h6,m6,s6,d6: integer;
```

1.2.–Procedimientos principales.

1.2.1.–*Inicio del programa.*

A continuación aparece el cuerpo del programa, que hace una llamada al procedimiento *no_existe_fichero* para crear el fichero de datos, si es que este no existe. Por el contrario, si el fichero de datos ya existe llama al procedimiento *entrar_en_programa.*

```
Begin
  textmode(bw80);
  randomize;
  textcolor(9);
  assign (datos,'datos.dat');
  (*$I-*)reset(datos)(*$I+*);
  ok:=(ioresult=0);
  if not ok then no_existe_fichero;
  (*$I-*)reset(datos)(*$I+*);
  ok:=(ioresult=0);
  if ok then entrar_en_programa;
End.
```

1.2.2.–*Creación del fichero de datos si no existe.*

Si el fichero de datos no exite todavia este procedimiento presenta un mensaje al usuario para ver si desea crearlo. Si la respuesta es afirmativa crea el fichero donde se almacenaran todos los datos que se recojan con el programa.

```
Procedure no_existe_fichero;
var crea:char;
begin
  clrscr;textcolor(15);
```

```
 sound(2000);delay(40);nosound;
 gotoxy(30,9);write('A T E N C I O N !');
 gotoxy(20,12);write('El fichero de datos no ha sido en-
contrado');
 gotoxy(27,14);write('Desea crearlo (S/N) :');
 repeat
  gotoxy(49,14);write(' ');
  buflen:=1;
  gotoxy(49,14);read(kbd,crea);
  if (upcase(crea)<>'S') and (upcase(crea)<>'N') hen
   begin
    sound(2000);delay(50);nosound;
   end;
 until (upcase(crea)='S') or (upcase(crea)='N');
 if (upcase(crea)='S') then
 begin
  gotoxy(27,17);write('creando fichero...');
  rewrite(datos);
 end
 else clrscr;
 textcolor(9);
end;
```

1.2.3.–*Entrar en el programa.*

Este procedimiento determina la secuencia de llamada a otros procedimientos. Los procedimientos a los que se hace referencia en este procedimiento aparecen después de este apartado en el orden en que son llamados por *entrar_en_programa.* La última operación tras ejecutar la secuencia de procedimientos a la que se hace mención es la escritura de los datos del experimento en el fichero donde se van almacenando los datos de los sujetos *(datos.dat).*

```
Procedure entrar_en_programa;

var tecla:char;
   i:integer;
Begin
 inicializar;
 cargar_textos;
 con_sin;
 ejemplos;
 proceso;
 reset(datos);
 seek(datos,filesize(datos));
 write(datos,aux);
 close(datos);
 clrscr;
 gotoxy(37,12);textcolor(15);write('F I N');
 textcolor(9);
end;
```

1.2.4.–*Inicialización y carga de los segmentos de las frases.*

En este apartado se incluyen dos procedimientos, *inicializar y cargar_textos.* El primer procedimiento inicializa (a) las tablas correspondientes a los dos primeros segmentos de las frases experimentales, (b) la matriz que almacena la ocurrencia o no ocurrencia de cada una de las combinaciones de las frases básicas más la terminación, y (c) la matriz de terminaciones que corresponden a cada una de las frases experimentales. El procedimiento *cargar_textos* asigna cadenas de caracteres (1) a cada una de las tablas creadas para los dos primeros segmentos de las frases experimentales, (2) a la matriz de combinaciones de las frases básicas más tipo de terminación, y (3) a las tablas de las frases que aparecen en los ejemplos y a los estimulos que aparecen en la tarea de decisión léxica de los mismos. En este último procedimiento se presentan sólo cuatro ejemplos de las frases utilizadas para no alargarlo en exceso.

```
Procedure inicializar;
var i,j:integer;
begin
 for i:=1 to 24 do
  begin
   T1[i]:='                                    ';
   T2[i]:='                    ';
    for j:=1 to 3 do
     begin
      SUCESO[i,j]:=0;
      T3[i,j]:='          ';
     end;
  end;
end;

Procedure cargar_textos;

begin
 (*** Primera parte de cada frase ***)
 T1[1]:='El reportero fotografió al chofer del capi-
tán';
 T1[2]:='El artista pintó a la mula del campesino';
 T1[3]:='El perro ladraba al gato del fiscal';
                    .
                    .
                    .
```

```
T1[24]:='La prensa criticó al abogado del terrorista';
(*** Segunda parte de cada frase ***)
T2[1]:='que tuvo el accidente';
T2[2]:='que estaba junto a la charca';
T2[3]:='que estaba en la terraza';
                .
                .
                .
T2[24]:='que escribió una carta';
(*** Primeras terminaciones de las frases ***)
T3[1,1]:='circulando por Madrid';
T3[2,1]:='trotando alegremente';
T3[3,1]:='lamiendose las patas';
                .
                .
                .
T3[24,1]:='defendiendo a su cliente';
(*** Segundas terminaciones ***)
T3[1,2]:='disparando su pistola';
T3[2,2]:='liando un cigarrillo';
T3[3,2]:='viendo la televisión';
                .
                .
                .
T3[24,2]:='amenazando al presidente';
(*** Terceras terminaciones ***)
T3[1,3]:='bajando las escaleras';
T3[2,3]:='bebiendo agua fresca';
T3[3,3]:='mirando a las gentes';
                .
                .
                .
T3[24,3]:='expresando sus opiniones';
(*** Frases de ejemplo ***)
E1[1]:='El taxista atropelló al perro del frutero';
E1[2]:='El guardia encerró al caballo del gitano';
E1[3]:='El profesor examinó a la hermana del delan-
tero';
E1[4]:='El tendero saludó al hermano del fraile';
E2[1]:='que estaba en el mercado';
E2[2]:='que estaba junto al puente';
E2[3]:='que estuvo en Melilla';
E2[4]:='que estuvo en el colegio';
E3[1]:='vendiendo naranjas';
E3[2]:='comiendo hierba';
E3[3]:='jugando al futbol';
E3[4]:='visitando las instalaciones';
T[1]:='LADRIDO';
T[2]:='ABREIH';
T[3]:='GOL';
T[4]:='MISA';
end;
```

1.2.5.–*Con o sin decisión léxica.*

El procedimiento con_sin permite al usuario realizar un experimento de lectura auto-espaciada que incluye (opción 1) o no incluye (opción 2) una tarea de decisión léxica.

```
Procedure con_sin;
begin
 clrscr;
 gotoxy(30,8);write('1 - Con palabras');
 gotoxy(30,10);write('2 - Sin palabras');
 gotoxy(30,22);write('Pulse opción (1-2) :');
 repeat
  cs:=' ';
  gotoxy(51,22);write(' ');
  buflen:=1;
  gotoxy(51,22);(*$I-*)read(kbd,cs)(*$I+*);
  ok:=(ioresult=0);
 until (ok=true) and ((cs='1') or (cs='2'));
 con:=false;
 if cs='1' then con:=true;
 if con then cargar_palabras;
 aux.c_s:=cs;
end;
```

Si el usuario elige la opción 1 (con palabras) el paso siguiente sería cargar la matriz de estímulos de la tarea de decisión léxica con las palabras y no palabras correspondientes a las combinaciones frase básica más terminación. Obsérvese que la disposición de las palabras y las no palabras responde a una regla especial (Ver apartado 1.3.5).

```
Procedure cargar_palabras;
begin
 PALABRAS[1,1]:='CONDUCIR';
 PALABRAS[1,2]:='RALIFSED';
 PALABRAS[2,1]:='RAECOC';
 PALABRAS[2,2]:='LABRAR';
 PALABRAS[3,1]:='ARAÑAR';
 PALABRAS[3,2]:='RASUAC';
                .                .
                .                .
                .                .
 PALABRAS[24,1]:='RAETIELP';
 PALABRAS[24,2]:='ASESINAR';
 PALABRAS[1,3]:='BENDECIR';
```

```
 PALABRAS[2,3]:='RAJROF';
 PALABRAS[3,3]:='AMASAR';
          .
          .
          .
 PALABRAS[24,3]:='RACILBUP';
end;
```

1.2.6.–*Proceso para la presentación de los ejemplos.*

Este proceso es básicamente igual al que tiene lugar para las frases experimentales (la única diferencia es que no se miden los tiempos de lectura y decisión léxica), por lo que se remite al lector a la descripción que aparece en el siguiente apartado.

```
Procedure ejemplos;
var i,p:integer;
tecla:char;
begin
 clrscr;
 gotoxy(20,12);textcolor(15);
 write('Pulse una tecla para comenzar los  ejemplos...');
 textcolor(9);
 read(kbd,tecla);
 for i:=1 to 4 do
  begin
   clrscr;
   gotoxy(15,6);textcolor(15);write('*');textcolor(9);
   pulsar_tecla(16,6);
   gotoxy(15,6);write(E1[i]);
   p:=length(E1[i]);
   gotoxy(p+16,6);textcolor(15);write('*');textcolor(9);
   pulsar_tecla(p+17,6);
   gotoxy(15,6);write('                                    ');
   gotoxy(p+16,6);write(' ');
   gotoxy(15,8);write(E2[i]);
   p:=length(E2[i]);
   gotoxy(p+16,8);textcolor(15);
   if con then write('#')
        else write('*');
   textcolor(9);
   pulsar_tecla(p+17,8);
   gotoxy(15,8);write('                                    ');
   gotoxy(p+16,8);write(' ');
   if con then
    begin
     sacar_ejemplo(i);
     respuesta;
     gotoxy(p+16,8); textcolor (15);write('*'); text-
     color(9);
     pulsar_tecla(p+17,8);
     gotoxy(p+16,8);write(' ');
    end;
   gotoxy(p+16,8);
   write(E3[i]);
   p:=p+length(E3[i]);
   gotoxy(p+17,8);textcolor(15);write('*');textcolor(9);
   pulsar_tecla(p+18,8);
  end;
end;

Procedure sacar_ejemplo(i:integer);

begin
 gotoxy(35,12);sound(2000);delay(50);nosound;de-
lay(50);
 textcolor(15);
 write(T[i]);
 delay(100);gotoxy(35,12);textcolor(9);write('          ');
end;
```

1.2.7.–*Proceso para la presentación los ensayos experimentales.*

El diagrama de flujo presentado anteriormente describe, bajo el encabezamiento «PROCESO PRINCIPAL», la secuencia de procesos que se realizan dentro de este procedimiento (Ver Figura 1). El procedimiento proceso hace llamadas a una serie de rutinas secundarias que apareceran bajo el apartado 1.3 de este artículo.

Cuando comienza la presentación del material experimental el programa sortea un número de frase y un número de terminación (Véase apartado 1.3.1). La primera vez que el sujeto pulsa la tecla definida por el procedimiento secundario pulsar_tecla (cursor_hacia-abajo) se toma el tiempo del reloj interno (Ver apartado 1.3.3) y después, aparece el primer fragmento de las frases. Cuando el sujeto pulsa por segunda vez esta tecla, desaparece el primer fragmento, se toma otra vez el tiempo del reloj interno, se calcula la diferencia entre los dos tiempos mencionados (Véase el apartado 1.3.4), y se presenta el segundo fragmento.

Cuando la opción elegida es la 2 (sin decisión léxica) pulsar de nuevo la tecla mencionada causa que desaparezca el segundo fragmento, se toma otra vez el tiempo y se calcula la diferencia entre este tiempo y el anterior, y por último, se presenta el tercer

fragmento. Pulsando otra vez la tecla definida previamente desaparece el tercer fragmento, se toma de nuevo el tiempo, se calcula una nueva diferencia entre los tiempos y, comienza un nuevo ensayo (aparece en la pantalla un asterisco).

Si se ha elegido la opción que incluye la tarea de decisión léxica, después del segundo fragmento aparece un estimulo durante 150 milisegundos (Ver apartado 1.3.5) y, el sujeto debe decidir si este es una palabra o una no-palabra (Véase el apartado 1.3.6). Cuando el sujeto da la respuesta se toma el tiempo del reloj interno y se calcula la diferencia entre el tiempo anterior y el presente. Después de la respuesta aparece de forma inmediata un asterisco en la pantalla, cuando el sujeto pulsa de nuevo la tecla cursor-hacia-abajo se toma el tiempo del reloj interno y aparece el tercer fragmento de las frases (terminación). Por último, cuando vuelve a pulsar esta tecla desaparece el tercer fragmento, se toma de nuevo el tiempo, se calcula la diferencia entre el tiempo previo y el que se ha tomado en última instancia y, aparece el asterisco que señala el inicio de otro ensayo.

```
Procedure proceso;
var i,p:integer;
tecla:char;
begin
 clrscr;
 gotoxy(25,12);textcolor(15);
 write('Pulse una tecla para comenzar...');textcolor(9);
 read(kbd,tecla);
 aux.mal:=0;
 aux.bien:=0;
 for i:=1 to 72 do
  begin
   clrscr;
   aux.med1[i].seg:=0;aux.med1[i].cent:=0;
   aux.med2[i].seg:=0;aux.med2[i].cent:=0;
   aux.med3[i].seg:=0;aux.med3[i].cent:=0;
   aux.med4[i].seg:=0;aux.med4[i].cent:=0;
   aux.num[i].fra:=0;
   aux.num[i].term:=0;
   repeat
    correcto:='s';
    coge_aleatorio;
   aux.num[i].fra:=x;
   aux.num[i].term:=y;
   until (correcto='s');
   gotoxy(65,3);write(i:2);
   gotoxy(35,3);write(x:2,'-',y:1);
   gotoxy(15,6);textcolor(15);write('*');textcolor(9);
   pulsar_tecla(16,6);
   coger_hora(h1,m1,s1,d1);
   gotoxy(15,6);write(T1[x]);
   p:=length(T1[x]);
   gotoxy(p+16,6);textcolor(15);write('*');textcolor(9);
   pulsar_tecla(p+17,6);
   gotoxy(15,6);write('                                    ');
   coger_hora(h2,m2,s2,d2);
   medir_tiempos(h1,m1,s1,d1,h2,m2,s2,d2,h,m,s,d);
   aux.med1[i].seg:=s;
   aux.med1[i].cent:=d;
   gotoxy(p+16,6);write(' ');
   gotoxy(15,8);write(T2[x]);
   p:=length(T2[x]);
   gotoxy(p+16,8);textcolor(15);
   if con then write('#')
      else write('*');
   textcolor(9);
   pulsar_tecla(p+17,8);
   gotoxy(15,8);write('                                    ');
   coger_hora(h3,m3,s3,d3);
   medir_tiempos(h2,m2,s2,d2,h3,m3,s3,d3,h,m,s,d);
   aux.med2[i].seg:=s;
   aux.med2[i].cent:=d;
   gotoxy(p+16,8);write(' ');
   if con then
    begin
     sacar_texto;
     respuesta;
     if resp<>palabra then
        aux.mal:=aux.mal+1
      else
        aux.bien:=aux.bien+1;
     coger_hora(h4,m4,s4,d4);
     medir_tiempos(h3,m3,s3,d3,h4,m4,s4,d4,h,m,s,d);
     aux.med3[i].seg:=s;
     aux.med3[i].cent:=d;
     gotoxy(p+16,8);textcolor(15);write('*');textco-
     lor(9);
     pulsar_tecla(p+17,8);
     coger_hora(h5,m5,s5,d5);
     gotoxy(p+16,8);write(' ');
    end;
   gotoxy(p+16,8);
   write(T3[x,y]);
   p:=p+length(T3[x,y]);
   gotoxy(p+17,8);textcolor(15);write('*');textcolor(9);
   pulsar_tecla(p+18,8);
   coger_hora(h6,m6,s6,d6);
   if con then
     medir_tiempos(h5,m5,s5,d5,h6,m6,s6,d6,h,m,s,d)
     else
     medir_tiempos(h3,m3,s3,d3,h6,m6,s6,d6,h,m,s,d);
```

```
    aux.med4[i].seg:=s;
    aux.med4[i].cent:=d;
   end;
   clrscr;
end;
```

1.3.–RUTINAS SECUNDARIAS UTILIZADAS POR LOS PROCEDIMIENTOS PRINCIPALES.

1.3.1.–*Sorteo aleatorio del número de frase y la terminación.*

El procedimiento que se presenta a continuación sortea un número de frase y de terminación teniendo en cuenta si la combinación elegida ha sido ya presentada.

```
Procedure coge_aleatorio;
var x1,y1:real;
begin
 x1:=int(random(24))+1;
 y1:=int(random(3))+1;
 x:=round(x1);
 y:=round(y1);
 if SUCESO[x,y]=1 then correcto:='n'
  else SUCESO[x,y]:=1;
end;
```

1.3.2.–*Definición de la tecla a utilizar para la autoadministración de los fragmentos de las frases.*

La tecla que sirve para la auto-administración de los fragmentos de las frases es *cursor-hacia-abajo.*

```
Procedure pulsar_tecla(x,y:integer);
var tecla:char;

begin
 repeat
  tecla:='x';
  gotoxy(x,y);read(kbd,tecla);
  if keypressed then
   begin
    read(kbd,tecla);
    if (tecla=#27) and keypressed then read(kbd,tecla);
   end;
  if (tecla<>#80) then
   begin
    sound(2000);delay(50);nosound;
   end;
 until (tecla=#80);
end;
```

1.3.3.–*Procedimiento para coger la hora del reloj interno.*

Mediante esta rutina se definen una serie de variables que se utilizan para guardar el tiempo. El registro AX se carga con valor de interrupción 2C, y esto hace que el sistema devuelva las variables CX y DX con la hora del reloj interno.

```
Procedure coger_hora(var hh,mm,ss,dd:integer);

Type
 regt = record
       AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS:integer;
      end;
Var
 tregistro:regt;

begin
 with tregistro do
 begin
  AX:=$2C00;
  MSDOS(tregistro);
  hh:=HI(CX);
  mm:=LO(CX);
  ss:=HI(DX);
  dd:=LO(DX);
 end;
end;
```

1.3.4.–*Procedimiento para medir los tiempos.*

Este procedimiento es llamado también por *proceso*, como el anterior, y básicamente se ocupa de calcular las diferencias entre dos tiempos (t_2-t_1) recogidos del reloj interno con el fin de determinar los segundos y las centésimas de segundo que un sujeto tarda en leer un segmento de las frases. Los condicionales que aparecen en esta rutina se aplican en los casos donde esta diferencia sería negativa o irreal (por ejemplo, cuando hh2=3 y mm2=00, y hh1=2 y mm1=59, el cálculo de una diferencia sin restricciones establecería que hay una hora entre ambas medidas y que existe una diferencia negativa respecto a los minutos).

```
Procedure
medir_tiempos(hh1,mm1,ss1,dd1,hh2,mm2,ss2,dd2:
```

```
integer;var h,m,s,d:integer);
begin
   if mm2<mm1 then begin
                   hh1:=hh1+1;
                   mm2:=mm2+60;
                  end;
   if ss2<ss1 then begin
                   mm1:=mm1+1;
                   ss2:=ss2+60;
                  end;
   if dd2<dd1 then begin
                   ss1:=ss1+1;
                   dd2:=dd2+100;
                  end;
  h:=hh2-hh1;m:=mm2-mm1;s:=ss2-ss1;d:=dd2-dd1;
end;
```

1.3.5.–*Presentación de las palabras en la tarea de decisión léxica.*

La rutina que se incluye en este apartado sirve para presentar los estimulos de la tarea de decisión léxica y para llevar un contador de las respuestas correctas e incorrectas de los sujetos en la misma sobre la base de la regla siguiente: si i+j es par, el estimulo es palabra, por el contrario si es impar, el estimulo es una no-palabra. Esta regla permite posteriormente saber si la respuesta es correcta o incorrecta (Véase apartado 1.2.7).

```
Procedure sacar_texto;
var z:integer;
begin
 gotoxy(35,12);sound(2000);delay(50);nosound; delay(50);
 textcolor(15);
 write(PALABRAS[x,y]);
 delay(100);gotoxy(35,12);textcolor(9);write('        ');
 if ((x+y) mod 2) = 0 then palabra :='s'
                      else palabra :='n';
end;
```

1.3.6.–*Recogida de la respuesta en la tarea de decisión léxica.*

Si se ha elegido la opción 1 (con tarea de decisión lexica) aparecerá en el momento de la decisión un mensaje como el siguiente:

NO <—— ——> SI

Las teclas definidas para dar la respuesta a la tarea de decisión léxica son *cursor-hacia-la-izquierda* en caso de respuesta negativa, y el *cursor-hacia-la-derecha* en caso de respuesta afirmativa.

```
Procedure respuesta;
var tecla:char;
begin
 repeat
  textcolor(15);
  gotoxy(30,23);write('NO <——  ——> SI');
  textcolor(9);
  tecla:='x';
  gotoxy(40,23);read(kbd,tecla);
  if keypressed then
   begin
    read(kbd,tecla);
    if (tecla=#27) and keypressed then read(kbd,tecla);
   end;
   if (tecla<>#75) and (tecla<>#77) then
    begin
     sound(2000);delay(50);nosound;
    end;
  until (tecla=#75) or (tecla=#77);
  if tecla=#75 then resp:='n' else resp:='s';
  gotoxy(30,23);write('                ');
end;
```

II. IMPRESION DEL FICHERO DE RESULTADOS O REDIRECCIONAMIENTO DEL MISMO A UN FICHERO ASCII.

Los datos correspondientes a cada sujeto se van almacenando a medida que se recogen en un único fichero (datos.dat). Los datos de todos los sujetos de un experimento pueden imprimirse llamando (fuera ya de FRA3) a un programa ejecutable que debe estar en la misma unidad donde se encuentra el fichero datos.dat. El código fuente de este programa que se ha denominado RECUENTO se presenta en el Apéndice I. Un ejemplo de una parte de un fichero de datos obtenido con FRA3 aparece en el Apéndice II.

Otra posibilidad es redireccionar la impresión del fichero de datos a un fichero ASCII mediante la utilidad PRN2FILE, que requiere asignar un nuevo nombre al fichero resultante. Esta posibilidad facilita el tratamiento de los datos con cualquier paquete de análisis estadístico.

REFERENCIAS BIBLIOGRAFICAS

Clifton, C. E., Frazier, L., y Connine, C. M. (1984). Lexical expectations in sentence comprehension. *Journal of Verbal Learning and Verbal Behavior* 23, 696-708.

Ferreira, F. y Clifton, C. (1986). The independence of syntactic processing. *Journal Of Memory And Language,* 25, 348-368.

Ferreira, F. y Henderson, J. (1990). The use of verb information in syntactic parsing: evidence from eye movements and word-by-word self-paced reading. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 16, 555-568.

Fodor, J. D. (1983). *The modularity of mind.* Cambridge: MIT Press.

Forster, K. (1979). Levels of processing and the structure of the language processor. En W. E. Cooper and E. C. T. Walker (Eds.), *Sentence Processing: Psycholinguistic studies presented to Merrill Garrett.* London: Lawrence Erlbaum Associates Ltd.

Frazier, L. (1987). Theories of sentence processing. En J. Garfield (Ed.), *Modularity in Knowledge Representation and Natural Language Processing*. Cambridge, Mass.: MIT Press.

Johnson, E., Payne, J. W., Schkade, D. y Bettman, J. (1989). Monitoring information processing and decision: the Mouselab System 4.2.

Just, M. A., Carpenter, P. A. y Woolley, J. D. (1982). Paradigms and processes in reading comprehension. *Journal of Experimental Psychology: General,* 111, 228-238.

Marslen-Wilson, W. D., Brown, C., y Tyler, L. K. (1988). Lexical representations in spoken language comprehension. *Language and Cognitive Processes*, 3, 1-16.

Marslen-Wilson, W. D. y Tyler, L. (1980). The temporal structure of spoken language understanding. *Cognition* 8, 1-71.

Marslen-Wilson, W. y Tyler, L. (1987). Against modularity. En J. Garfield (Ed.), *Modularity in knowledge representation and natural-language understanding.* Cambridge, Mass.: MIT Press.

Mayor, J. (1979). La ambigüedad: ¿un nuevo paradigma?. *Informes del Departamento de Psicologia General*, 2/5, 121-126.

Mayor, J. (1980). La comprensión del lenguaje desde un punto de vista experimental. *Revista española de Lingüistica*, 10, 1, 59-111.

Mayor, J. y Moya, J. (1991). La ambigüedad. En J. Mayor and J. L. Pinillos (Eds.), *Tratado de Psicología General: Comunicación y Lenguaje.* Madrid: Alhambra Universidad.

McClelland, J. L.(1987). The case for interactionism in language processing. En M. Coltheart (ed.), *Attention and Performance XII: The Psychology of reading* (pp. 3-36). London: Lawrence Erlbaum Associates.

Mitchell, D. C. y Green, D. W. (1978). The effects of context and content on immediate processing in reading. *Quarterly Journal of Experimental Psychology*, 30, 609-636.

Rayner, K., Sereno, S. C., Morris, R., Schmauder, y Clifton, C. (1989). Eye movements and on-line language comprehension processes. *Language and Cognitive processes, 4,* (3/4) SI 21-29.

López Sánchez, R. (1992). Resolución de la ambigüedad sintáctica basada en las expectativas léxico-semánticas. Tesis doctoral remitida para su publicación. Universidad Complutense de Madrid.

Steedman, M. y Altmann, G. (1989). Ambiguity in context: a reply. *Language and Cognitive Processes*, 4, (3/4), SI 105-122.

Taraban, R. y McClelland, J. L. (1988). Constituent attachment and thematic role assignment in sentence processing: Influence of content-based expectations. *Journal of Memory and Language*, 27, 597-632.

Tyler, L., y Marslen-Wilson, W. (1977). The on-line effects of semantic context on syntactic processing. *Journal of Verbal Learning and Verbal Behavior* 16, 683-692.

Aceptado, 29 de octubre de 1993

APENDICE I

Listado completo del código fuente de RECUENTO.

```
Program Frases;
Type
  tiempo = record
          seg : integer;(* Número de segundos *)
          cent: integer;(* Número de centésimas de segundo *)
         end;
  numero = record
          fra : integer;(* Número de frase (1-24) *)
          term: integer;(* Número de terminacion (1,2,3) *)
         end;
  reg    = record
          c_s  : char;(* 1-Con palabras, 2-Sin palabras *)
          num  : array [1..72] of numero;(* Número de frase y terminac. *)
          med1 : array [1..72] of tiempo;
          med2 : array [1..72] of tiempo;
          med3 : array [1..72] of tiempo;
          med4 : array [1..72] of tiempo;
          bien : integer;
          mal  : integer;
         end;

var
datos:file of reg;
aux:reg;
tecla:char;
ok:boolean;

(************************ Fichero Inexistente ****************************)

Procedure fichero_inexistente;
var ok:boolean;

begin
  clrscr;
  sound(2000);delay(60);nosound;
  gotoxy(26,10);write('El Fichero de Datos no existe');
  gotoxy(26,11);write('_____');
end;

(************************** Seguir ********************************)

Procedure seguir;
var tipo:string[14];
i:integer;

begin;
  clrscr;
  gotoxy(20,10);write('Prepare la impresora y pulse una tecla...');
  read(kbd,tecla);
```

APENDICE I (Continuación)

```
  gotoxy(20,10);write('                                    ');
  gotoxy(30,10);write('Imprimiendo Resultados...');
  while not(eof(datos)) do
   begin
    read(datos,aux);
    writeln(lst,' ');
    writeln(lst,' ');
    writeln(lst,' ');
    if aux.c_s='1' then tipo:='Con palabras'
                   else tipo:='Sin palabras';
    write(lst, '   Caso  : ',tipo);
    if aux.c_s='1' then
     begin
      write(lst,'    Bien : ',aux.bien:2);
      writeln(lst,'    Mal : ',aux.mal:2);
     end
     else writeln(lst,' ');
    writeln(lst,' ');
    writeln(lst,' ');
    writeln(lst,' Fra/Term      Tiempo 1       Tiempo 2       Tiempo 3       Tiempo 4');
    writeln(lst,'------------------------------------------------------------------------');
    writeln(lst,' ');
    for i:=1 to 72 do
     begin
      write(lst,'   ',aux.num[i].fra:2,'-',aux.num[i].term:1);
      write(lst,'          ',aux.med1[i].seg:2,' : ',aux.med1[i].cent:2,'          ');
      write(lst,aux.med2[i].seg:2,' : ',aux.med2[i].cent:2,'          ');
      write(lst,aux.med3[i].seg:2,' : ',aux.med3[i].cent:2,'          ');
      writeln(lst,aux.med4[i].seg:2,' : ',aux.med4[i].cent:2);
     end;
  end;
  clrscr;
  gotoxy(28,10);write('Fin del listado de resultados');
end;

(************************** Programa Principal **************************)

Begin

  assign(datos,'datos.dat');
  (*$I-*)reset(datos)(*$I+*);
  ok:=(ioresult=0);
  if not ok then fichero_inexistente
    else seguir;
end.
```

APENDICE II

Ejemplo de un fichero de datos creado tras la aplicación de FRA3. En la cabecera aparece la opción elegida y el número de respuestas correctas e incorrectas en la tarea de decisión léxica. Debajo de la cabecera aparecen las etiquetas de cinco columnas: (1) Fra/Term: indica el número de frase y la terminación, (2) Tiempo 1: hace referencia al tiempo de lectura en segundos y centésimas del primer segmento de las frases, (3) Tiempo 2: indica el tiempo de lectura para el segundo fragmento, (4) Tiempo 3: sería el tiempo que el sujeto tarda en tomar la decisión léxica (este tiempo sería cero si la opción elegida fuera la 2), por último, (5) Tiempo 4: sería el tiempo de lectura del tercer segmento de las frases (terminación).

Caso: Con palabras Bien : 54 Mal : 18

Fra/Term	Tiempo 1	Tiempo 2	Tiempo 3	Tiempo 4
3-2	2 : 36	1 : 48	2 : 47	3 : 2
22-3	1 : 65	1 : 98	1 : 54	1 : 64
4-2	2 : 85	1 : 32	2 : 3	1 : 21
21-3	2 : 37	1 : 31	1 : 65	0 : 98
12-2	1 : 15	1 : 26	1 : 27	1 : 38
16-1	1 : 82	1 : 26	1 : 21	1 : 81
20-1	2 : 31	1 : 32	2 : 63	2 : 9
15-1	2 : 42	1 : 65	2 : 47	0 : 99
19-3	1 : 86	1 : 65	1 : 10	0 : 99
9-3	1 : 48	1 : 82	1 : 54	0 : 50
2-3	2 : 14	2 : 64	1 : 42	1 : 32
23-1	1 : 65	1 : 10	1 : 92	1 : 98
21-1	1 : 60	1 : 75	1 : 71	1 : 10
22-2	1 : 71	1 : 59	1 : 43	0 : 94
18-1	1 : 87	2 : 14	1 : 49	0 : 93
18-2	2 : 4	1 : 86	1 : 5	0 : 77
7-3	2 : 20	1 : 48	1 : 21	1 : 75
18-3	1 : 92	1 : 48	2 : 31	0 : 99
14-2	1 : 75	1 : 10	1 : 10	0 : 44
13-2	2 : 47	1 : 76	1 : 15	2 : 3
13-3	1 : 75	1 : 32	1 : 5	1 : 53
24-2	1 : 81	1 : 10	1 : 21	1 : 15
1-1	1 : 38	1 : 15	0 : 93	0 : 88
4-1	1 : 65	1 : 38	1 : 48	0 : 94
17-1	1 : 21	1 : 10	1 : 48	1 : 98
23-2	1 : 59	1 : 43	1 : 32	0 : 77
24-1	1 : 48	1 : 15	1 : 60	0 : 76
12-1	1 : 49	1 : 42	1 : 98	0 : 76
7-1	1 : 21	1 : 31	1 : 21	0 : 83
19-2	1 : 70	1 : 43	1 : 76	1 : 15
9-1	1 : 32	1 : 15	0 : 99	0 : 61
11-3	1 : 37	1 : 21	1 : 59	0 : 99
10-1	1 : 43	1 : 43	1 : 32	1 : 10
10-2	1 : 21	1 : 32	1 : 26	1 : 5
2-2	1 : 4	1 : 43	1 : 32	0 : 99
15-3	1 : 15	1 : 21	1 : 32	0 : 33