

ANUARIO DE PSICOLOGÍA
Núm. 26 - 1982 (1)

COMPUTADORAS E INVESTIGACION
PSICOLOGICA

FRANCISCO VALLE ARROYO

Universidad de Oviedo

Francisco Valle Arroyo
Edificio Valdés Salas
Universidad de Oviedo

El presente artículo tiene por finalidad señalar algunos de los servicios que las computadoras pueden prestar a la investigación en general y a la psicología en particular, aunque me limitaré a exponer aquellos programas con los que he tenido experiencia directa. Tales servicios pueden ir desde los de carácter puramente secretarial, como la edición de textos, hasta los típicos de un experimentador cualificado, como la presentación de estímulos y la recogida de datos. La parte final —y más larga— del artículo estará dedicada a un caso concreto de programación de un experimento psicolingüístico. El deseo sería que este artículo contribuyera a subsanar dos vicios o errores que, en mi opinión, existen en el ambiente estudiantil e intelectual español. El primero de ellos es el escaso número de personas que conocen una lengua¹ de programación y el aún menor de conocedores que utilizan de manera sistemática sus conocimientos. El segundo se refiere a una especie de identificación o asociación entre computadoras y análisis matemáticos, lo cual corresponde con la realidad sólo parcialmente, ya que estas máquinas pueden realizar otras muchas cosas y ser utilizadas en otros menesteres, como veremos, y sobre todo y como el título indica, como instrumentos útiles en la investigación psicológica.

Comencemos por los servicios de carácter secretarial. RUNOFF es un programa que suele venir incorporado en computadoras tales como Burroughs 7700 y en la DEC-10 y el cual convierte —casi se diría que por arte de magia— un texto desordenado: con títulos y encabezamientos no centrados, sin separación de párrafos, con líneas de distinta longitud, etc., en un texto que sería aceptado por el más exigente consejo de redacción de una prestigiosa casa editora. Para ello basta con colocar en el lugar apropiado las órdenes correspondientes. Así, por ejemplo, un espacio en blanco al comienzo de una línea indica a la computadora que lo que viene a continuación forma un párrafo diferente y en consecuencia aparecerá con un mayor margen a la izquierda y separado del renglón anterior por una línea más en blanco. Existen órdenes que controlan los márgenes de la izquierda y los de la derecha, el tamaño de la página (n.º de líneas y espacios por folio), la paginación, las notas al pie de página, espacios para dibujos; incluso existe un sistema por el cual se puede formar un índice de materias. Aparte de la forma perfecta de presentación, hay otras ventajas que es preciso poner de manifiesto, como son la faci-

(1) En España se suele hablar de lenguajes de programación. La palabra correcta, sin embargo, es lengua. El error proviene de la traducción, al pie de la letra, del original inglés Programming Language.

lidad de corrección de errores, la posibilidad de intercalar o suprimir párrafos, la posibilidad de alterar el orden, etc. Estas ventajas las comprenderán más fácilmente aquellas personas que al pasar a limpio un trabajo se hayan «comido» una línea y se hayan visto obligadas a repetir tal vez 4 ó 5 folios para poder subsanar su error. Hasta hace poco, la única limitación que presentaba este programa era la de no poder hacer dibujos ni gráficas, ni tampoco modificar el tamaño de las letras. Actualmente la mayoría de las computadoras, incluso las micro, como puede ser la HP 85, tienen un PLOTTER incorporado con el cual se puede hacer prácticamente cualquier tipo de gráfico y aún utilizar lápices de distintos colores.

Sin duda el beneficio más conocido en España de las computadoras son sus maravillosas capacidades para realizar análisis matemáticos y estadísticos. Cualquier psicólogo que haya tenido que hacer un análisis de varianza (varios factores y sus respectivas interacciones) con 6.000 tiempos de reacción, por ejemplo, habrá consumido meses de su vida para poder llevar a cabo tan tediosa tarea. Con la utilización de programas como SPSS, BMD, ANOVR, MULTIVARIANCE, etc., el tiempo se habría reducido a días, duración todavía considerable ya que en la medida de datos en la computadora no hay ahorro de tiempo posible. La realización del análisis, una vez que los datos han sido dados a la computadora, puede tardar minutos, escasos minutos. Todos los programas anteriores pueden hacer desde simples recuentos de frecuencias hasta los más complicados análisis de varianza, correlaciones, regresión múltiple, covarianza, análisis factorial. Además de su rapidez, hay que notar que la exactitud es mayor, que la información que proporcionan es superior y que la posibilidad de error es escasa. Estos programas han sido escritos o bien por personas que son a la vez grandes matemáticos y consumados especialistas en programación, o bien en colaboración de sujetos que reúnen las condiciones anteriores. Así por ejemplo, el programa ANOVR que como su nombre indica, realiza análisis de varianza, además de proporcionar las sumas de cuadrados, los g , I , medias cuadráticas y F s de cada uno de los factores e interacciones, informa también del nivel de significación, da las medias de cada una de las celdas y las varianzas de cada grupo, realiza el test de Bartlett sobre la homogeneidad de la varianza, proporciona la matriz varianza-covarianza y avisa de si la homogeneidad de las matrices de covarianza permite los análisis realizados. En cualquiera de estos programas, las tarjetas de control que el experimentador (analista) tiene que introducir suele ser menor de cinco y casi nunca superior a diez. En ellas se especifica el número de variables dependientes e independientes, el número de sujetos por grupo, el número de las variables inter e intragrupo y el formato, es decir, cómo están escritos los datos; tareas todas ellas tan sumamente fáciles que cualquier persona puede ejecutar uno de esos programas en un par de horas.

Aunque los campos de mayor futuro y en los que probablemente la aportación de la informática a la psicología va a ser más decisiva, sean los de la Inteligencia Artificial y Simulación, el entrar en una descripción de los mismos, por somera que fuera, desborda los límites de este artículo.

Hay, con todo, otro aspecto que quisiera destacar antes de pasar al núcleo de

este artículo. Me refiero a la información bibliográfica, siempre tan necesaria en cualquier investigación, a la que se puede acceder a través de terminales de computadoras. Piensese, por ejemplo, en el banco de datos de la Lockheed, en Palo Alto, y al que se puede tener acceso mediante los terminales que el Ministerio de Universidades e Investigación tiene instalados en diferentes capitales de provincia. El mayor inconveniente que este tipo de información tiene —aparte de otros de tipo más técnico— es el elevado costo, lo cual hace cada vez más necesaria la creación de un banco similar aquí en España.

Pasamos ya a la investigación psicológica propiamente tal. Es preciso tener en cuenta que para todos los tipos de experimentos de los que voy a tratar no son necesarias grandes computadoras, con elevado número de K de memoria y consiguientemente de varios millones de presupuesto. Todos ellos pueden realizarse en computadoras pequeñas (mini o micro) y con 32 K de memoria son más que suficientes. Lo único que es absolutamente necesario es que tengan una pantalla, un cronómetro y una impresora (line printer). Repito una vez más que el tipo de experimentos que voy a exponer son aquellos con los que he tenido una experiencia directa y que, por consiguiente, la lista no es exhaustiva, sino tan sólo indicativa.

Probablemente la utilización más elemental de una computadora sea como *tambor de memoria*. Los estímulos pueden aparecer tanto en la pantalla como en la impresora y son realizables todos los tipos de variaciones. Se pueden alterar la longitud de la lista, el tipo de estímulos presentados, el intervalo interestimular, etc. La programación de este tipo de experimentos es elemental. Se pueden utilizar también en *tests de reconocimiento*. La primera parte del experimento sería semejante al anterior, pero la segunda —el test de reconocimiento propiamente tal— exigiría la elaboración de una segunda lista de estímulos que deberían ser presentados mezclados con los primeros y además habría que programar la máquina para que identificara los reconocimientos correctos y los incorrectos. La programación sería un poco más difícil, pero todavía bastante asequible, incluso para programadores incipientes. Por supuesto que en las palabras de relleno se pueden utilizar homófonos, sinónimos, antónimos, palabras neutras, etc., y se puede ordenar a la máquina que realice los análisis pertinentes sobre los datos recogidos. Una tercera utilización que se me ocurre es la de la computadora como *taquistoscopio*. Las limitaciones en este caso vienen dadas por la precisión del cronómetro, aunque aun en el peor de los casos, puede medir sexagésimas de segundo y muchos de ellos tienen precisiones superiores como centésimas y milésimas de segundo. Pueden asimismo utilizarse en tareas de *búsqueda visual* (visual search) y *enmascaramiento*, aunque aquí la parte de programación ya se va haciendo más compleja. En fin que las aplicaciones son casi ilimitadas, siempre que el experimentador tenga ideas y posea un buen conocimiento del software que le permita traducir a una lengua artificial el diseño experimental que quiera llevar a cabo. Yo, por ejemplo, utilicé un programa que consistía en adivinar anagramas —y que los alumnos realizaban con auténtico deleite— para someter a prueba la teoría de la rigidez funcional de Duncker (1945).

Conviene tener presente que, aunque la mayoría de las Universidades españolas ya tienen un centro de cálculo con computadoras nada desdeñables, estos estudios no deben realizarse a través de esos centros por dos razones fundamentales. 1) En ellos no existen las condiciones de aislamiento que en general son recomendables en cualquier experimento. 2) Aún en el supuesto de que existan terminales en el propio departamento suficientemente aisladas, las computadoras a las que están conectadas trabajan con el sistema de tiempo compartido (time-sharing) con lo cual el experimento se puede ver alterado en el ritmo o incluso interrumpido en cualquier momento.

Y ahora ya pasamos al estudio de un caso concreto. La computadora en que se realizó este trabajo fue una PDP-11/03² y la lengua de programación usada fue FOCAL, cuyo desconocimiento aquí en España es totalmente injustificado, dadas algunas de sus propiedades, como su carácter interactivo que permite en todo momento interactuar con la máquina, conocer el valor de las variables, ejecutar bloques o líneas de programas sin necesidad de compilar el programa entero, etc.

Lo que el estudio (Valle Arroyo 1979, 1982) pretendía era determinar la posible influencia del contexto en la comprensión de las frases negativas. En otras palabras, se suponía que parte de la dificultad de las negativas en los experimentos podría deberse a la artificialidad de la situación experimental, y que si las frases negativas cumplieran en los estudios experimentales su función de destacar la excepción frente a la norma (Wason, 1965) se obtendría una reducción significativa en el TR de las negativas que cumplieran esta condición, en comparación con aquellas en que tal función fuera violada. También se pretendía estudiar el tipo de contexto que podía ser más efectivo en producir tal reducción. El contexto que se proporcionaba a los sujetos consistía en un conjunto de cuatro palabras de las cuales tres pertenecían a una categoría común y la cuarta formaba una clase diferente; por ejemplo: Vencejo Golondrina Murciélago Alondra. El sujeto tenía que identificar la palabra discrepante pulsando el n.º correspondiente —cada palabra llevaba debajo un n.º, del 1 al 4. El contraste o distancia semántica entre los elementos semejantes (norma) y el diferente (excepción) podía ser pequeño o grande. La distancia semántica fue determinada mediante consenso entre un jurado de 10 personas de cultura y formación diferentes. Naturalmente había grupos de control que no tenían que solucionar este problema (selección) y que sólo verificaban las frases, como veremos inmediatamente.

Después de esto los sujetos tenían que evaluar —determinar el valor de verdad— una frase sobre una de las palabras del conjunto; por ejemplo: «El murciélago no es un ave». Ocho eran los tipos de frases que se podían presentar al

(2) Las PDP-11 son producidas y distribuidas por la Digital Equipment Corporation de Maynard, Mass., USA., la cual tiene sucursales en Barcelona y Madrid. El precio de una PDP-11/03 con 32k de memoria, pantalla, impresora e intérpretes de Fortran y Focal es de 1.5 millones de Ptas. (Precios de 1980.)

sujeto: Afirmativa Verdadera (AV), Afirmativa Falsa (AF), Negativa Verdadera (NV) y Negativa Falsa (NF), cada una de las cuales podía referirse bien a uno de los elementos semejantes bien al discrepante, o sea, que en total los ocho tipos de frases eran: AVS (semejante), AFS, NVS, NFS, AVD (discrepante), AFD, NVD y NFD. El experimento en total tenía un diseño mixto con dos factores intergrupo—cada uno de ellos con dos niveles: Contexto/Sin Contexto y Contraste Pequeño /C. Grande; y tres intragrupo también con dos niveles cada uno (2^3) que eran los que producían los diferentes tipos de enunciados: Semejanza/Discrepancia, Afirmación/Negación y Verdad/Falsedad. Sobre cada uno de estos tipos el sujeto debía de evaluar ocho frases diferentes. Por tanto el diseño, si incluimos las repeticiones como un factor más, era $2 \times 2 \times 2 \times 2 \times 2 \times 8$.

En este estudio a diferencia de los que sobre este tema se han realizado, cada una de las 64 frases presentadas era diferente, es decir, que tanto los sujetos como los atributos eran nuevos en cada presentación. Compárese, por ejemplo, con el estudio de Clark y Chase (1972) en el que los sujetos (gramaticales) eran siempre: «La estrella» o «El signo más» y los predicados «encima» y «debajo», o con el de Wason (1965) cuyo sujeto siempre era: «El círculo n.º X», y el predicado podía variar entre «rojo» y «azul».

Por lo tanto, en resumen y centrándonos únicamente en los factores intragrupo, lo que se pretendía hacer era presentar un total de 64 enunciados a cada sujeto, se deseaba además que el número de frases sobre cada uno de los factores fuera exactamente igual, es decir, debía de haber 32 frases sobre los elementos semejantes y otras tantas acerca de los discrepantes. Esos 32 enunciados estarían, a su vez, distribuidos homogéneamente en cada uno de los tipos de frases. Si el sujeto cometía algún error existía una rutina en el programa que se encargaba de registrar el TR de dicha respuesta, el intento en el que se había producido y naturalmente almacenaba también el tipo de frase. La máquina se encargaba de presentar una frase semejante, es decir, del mismo tipo, al menos 8 intentos después de haber ocurrido el error. Con esto se garantizaba que al final del experimento se tuviera exactamente el mismo número de observaciones válidas en cada condición experimental y se tenía un registro completo de los errores que se habían cometido, con lo cual se podía comprobar la relación entre errores y TR y si había habido alguna especie de compromiso entre la exactitud de la respuesta y la velocidad de la misma. No es necesario decir que la variable dependiente eran los TR.

Para evitar que el orden de presentación pudiera tener algún efecto *sistemático* en los resultados, el orden de la misma era totalmente aleatorio. ¿Cómo se lograba todo esto en la computadora? ¿Cómo había sido programada para que pudiera realizar la tarea encomendada? Para poder responder estas preguntas de manera inteligible para el lector desconocedor de FOCAL, se hacen necesarias algunas aclaraciones y precisiones sobre dicha lengua de programación. En Focal lo mismo que en Basic cada línea va precedida de un número indicador de posición, pero a diferencia de lo que ocurre en Basic, estos números se componen de una parte entera y de una parte decimal. Conviene recordar también que pueden ir en

la misma línea tantas sentencias como se quiera con tal que no sobrepasen el espacio 72; la separación de sentencias dentro de una misma línea se hace con el punto y coma (;). Todas las líneas con la misma parte entera forman un bloque, que en algunos aspectos —sólo en algunos— es semejante a las «rutinas» y «procedimientos» de otros idiomas artificiales. Y por último, en Focal, todas las órdenes pueden escribirse con el nombre completo o pueden reducirse a su inicial. (Para que sea más fácil la comprensión, de vez en cuando se dará la traducción en Basic y/o Fortran.)

Las palabras estímulos que formando tétradas eran presentadas en el primer problema (Tarea de selección) estaban almacenadas en los bloques 51, 52, 53 y 54. Los tres primeros bloques contenían los elementos semejantes y el 54 los diferentes. Cada uno de los bloques tenía 64 líneas —de la 0.1 a la .64— y cada línea almacenaba una palabra. Sólo aquellas palabras que estaban en líneas con la misma parte decimal podían formar una tétrada. A continuación van tres líneas de cada uno de estos bloques para que sirvan de modelo.

51.01 T «DALLAS»
 51.19 T «FORD»
 51.55 T «BALLENA»
 52.01 T «CHICAGO»
 52.19 T «JOHNSON»
 52.55 T «DELFIN»
 53.01 T «MIAMI»
 53.19 T «NIXON»
 53.55 T «CACHALOTE»
 54.01 T «LONDRES»; S D = I
 54.19 T «KISSINGER»; S D = I
 54.55 T «TIBURÓN»; S D = I

(La T de Focal equivale al Print de Basic y al Write de Fortran y en consecuencia lo que hace es escribir en la pantalla lo que va entre comillas; la S es lo mismo que el LET en Basic, es decir que es el enunciado por el cual se asignan valores a las variables. Así, por ejemplo, S D = I, al ser ejecutado, asigna a la variable D el valor que en ese momento tenga la variable I.)

En los ejemplos presentados —Dallas Chicago Miami Londres— aparecían en la misma tétrada, pues todas ellas tienen una misma parte decimal (.01) y lo mismo ocurría con las otras.

El bloque 55 contenía los atributos asociados a los términos semejantes y el 56 los de los discrepantes, colocados en el mismo orden que los correspondientes sujetos, o sea,

- 55.01 T «UNA CIUDAD ESTADOUNIDENSE»
 55.19 T «UN EX-PRESIDENTE»
 55.55 T «UN MAMÍFERO»
 56.01 T «UNA CIUDAD INGLESA»
 56.19 T «UN EX-SECRETARIO DE ESTADO»
 56.55 T «UN PEZ»

El bloque 57 sólo tenía dos líneas: la 57.01 y la 57.02, la primera de ellas producía sentencias afirmativas (T «ES»), la 57.02 generaba enunciados negativos (T «NO ES»).

Con todos estos datos ya podemos pasar a explicar el programa con más detalle. Después de una serie de instrucciones que aparecían en la pantalla, la máquina preguntaba al sujeto si era diestro o zurdo. Si era zurdo las teclas que había de pulsar para evaluar la frase como Verdadera o como Falsa eran la Z y la X, si era diestro pulsada el punto (.) y el ENTER (que se hallan a la derecha del teclado). El sujeto podía además escoger que la Z representara el valor «verdadero» (V) y la X el «falso» (F) o viceversa, y otro tanto ocurría con los diestros. De esta forma el sujeto podía encontrarse suficientemente cómodo y se disminuirían los TR debidos a causas extrínsecas. En el mismo bloque I (main program) la computadora realizaba otras funciones muy importantes, así por ejemplo, creaba dos vectores, ambos con 64 elementos. El primero de ellos —vector A1— estaba formado por números del 1 al 64, en orden aleatorio; números que representaban la parte decimal de los bloques que contenían los estímulos y que controlaban el orden de presentación de dichos estímulos. El segundo —vector A— también contenía 64 números, pero cuyos valores oscilaban entre 1 y 8; más en concreto, estaba formado de 8 bloques cada uno con 8 números, comprendidos entre 1 y 8, pero el orden de cada bloque era diferente (determinado al azar). Supongamos que los tres primeros valores de A1 sean: 55, 1, 19, y que 2, 1, 8 sean los tres primeros elementos del vector A. Esto querría decir que la primera tétrada presentada sería la 55, es decir, «Ballena Delfín Cachalote Tiburón». (El orden en que aparecían las palabras de cada tétrada era asimismo aleatorio). Si el vector A controlaba, como habrá podido intuirse, el tipo de frase y si su primer valor era un 2, la primera frase presentada sería una frase afirmativa, falsa y referida al término semejante (AFS), es decir el segundo tipo de frases de las presentadas al hablar del diseño experimental, y la frase concreta podría ser: «La ballena es un pez». Una cosa más se hacía en el bloque 1: contar el n.º de intentos o pruebas que se iban presentando. Esto se hacía en las líneas 1.82 y 1.83.

1.82 S Z = Z + 1; D 21

1.83 I (Z-64) 1.82

Traducido al lenguaje normal esto significa: añade una unidad al valor que tenga Z, haz el bloque 21. Después de haber sido ejecutado el bloque 21, la máquina volvía a la línea 1.83 que dice: si el valor de Z es menor de 64, vete a la línea

1.82, con lo cual el proceso se repetía hasta que Z fuera igual a 65, es decir, que se presentaban en total 64 frases, como ya se ha indicado con anterioridad.

Ahora vamos a exponer el bloque 21, que era el que determinaba los valores de las variables intragrupo. He aquí tal bloque en su conjunto³.

| | | | |
|---|---------|-------|-------|
| 21.10 S XO = A1(Z)/100 | | | |
| 21.20 S X1 = A(Z) | | | |
| 21.22 S X8 = FSGN(X1/2-FITR(X1/2)--.25) | | | |
| 21.24 I (X1-5) 21.26; S X3 = 54 + XO; G 21.30 | | | |
| 21.26 S X3 = 51 + FITR(FABS(FRAN()*3)) + XO | | | |
| 21.30 S I6 = ((X1-1)/4-FITR((X-1-1)/4))*4 | | | |
| 21.32 I (I6-1.5) 21.36 | | | |
| 21.34 S X6 = 57.02; G 21.40 | | | |
| 21.36 S X6 = 57.01 | FORTRAN | BASIC | FOCAL |
| 21.40 I (—FABS(X1-1)) 21.41,21.51 | | LET | S |
| 21.41 I (—FABS(X1-4)) 21.42,21.51 | INT | INT | FITR |
| 21.42 I (—FABS(X1-6)) 21.43,21.51 | ABS | ABS | FABS |
| 21.43 I (—FABS(X1-7)) 21.44,21.51 | RAN | RND | FRAN |
| 21.44 S X7 = 56 + XO | GOTO | GOTO | G |
| 21.46 G 21.99 | IF | IF | I |
| 21.51 S X7 = 55 + XO | CALL | GOSUB | D |
| 21.99 D 20 | | | |

Si A1 contenía números cuyo valor oscilaba entre 1 y 64, XO adquirirá valores entre .01 y .64, es decir, que va a determinar los estímulos a presentar, como ya se ha indicado. A, por el contrario, tenía valores de 1 a 8, por tanto X1 controla el tipo de frase que se va a presentar en cada intento (cuyo n.º de orden viene deter-

(3) Para aquellos que no conocen ninguna lengua de programación, siguen unas aclaraciones. El * es el símbolo de la multiplicación.

FSGN = función SIGNO, cuyos valores son -1, si el argumento es negativo, 0 si es igual a cero y +1 si el valor del argumento es positivo.

FITR = función ENTERO, es decir, que sólo tiene en cuenta la parte entera de un número. Así FITR (26.47) = 26.

I = condicional. Si el argumento —lo que va entre paréntesis— es negativo, se ejecuta la línea que sigue al paréntesis; si es cero la que va en segundo lugar y si es positivo se ejecuta la tercera línea que sigue al paréntesis. De no cumplirse la condición el programa sigue el orden normal.

G = vete. Al ser ejecutada, la máquina salta a la línea que va detrás de la G.

FABS = función ABSOLUTO. Proporciona el valor absoluto del argumento, prescindiendo del signo que el mismo pueda tener.

FRAN = función ALEATORIA, que produce números comprendidos entre -1 y +1, con cuatro cifras decimales.

D = haz o ejecuta. Ordena a la máquina que ejecute la línea o bloque que le sigue. Una vez ejecutados tal línea o bloque, la máquina vuelve a la línea u orden siguientes a la D de donde partió.

minado por el subíndice Z). Veamos que ocurría en la línea 21.22. La función FS-GN puede adquirir tres valores: 1, 0 y -1, según que el valor del argumento —lo que va entre paréntesis— sea positivo, cero o negativo. Si se ejecuta (efectúa) el paréntesis se verá que X8 adquiere valores de -1 cuando X1 es par: 2, 4, 6 y 8; y de +1 en los casos en que X1 es impar: 1, 3, 5 y 7. En una palabra X8 controla la respuesta —la tecla que debía pulsar el sujeto; si $X8 = +1$, la frase es V y por tanto el sujeto había de pulsar la tecla correspondiente a V, si $X8 = -1$, entonces la frase era falsa. En la línea 21.24 la computadora tomaba una «decisión» que dependía del valor del paréntesis del condicional. Si X1-5 era negativo, es decir, si X1 valía 1, 2, 3 ó 4, entonces la máquina pasaba a la línea 21.26 y asignaba a X3 un valor de 51, 52 ó 53 (el que fuera uno u otro dependía del n.º aleatorio obtenido en la función FRAN) + la parte decimal de XO. Si X1-5 era positivo entonces la máquina ejecutaba la sentencia que estaba a continuación y en la misma línea, es decir, X3 adquiría el valor de 54 + la parte decimal de XO y a continuación pasaba a la línea 21.30 (G 21.30). Dicho en lenguaje corriente, X1 controla el sujeto de la frase. Si X1 era menor de 5 entonces el sujeto sería un término semejante (de los bloques 51, 52 ó 53); si era 5 o mayor entonces la frase se referiría al término discrepante (bloque 54). I6 determinaba si la frase era afirmativa o negativa. Si se ejecuta la expresión aritmética de la línea 21.30 se verá que I6 adquiere valores de 0, 1, 2, 3, 0, 1, 2 y 3 correspondientes a valores de X1 de 1, 2, 3, 4, 5, 6, 7 y 8. Si el valor de I6 era menor de 1.5 entonces $X6 = 57.01$, es decir, la frase era afirmativa; en caso contrario, $X6 = 57.02$, o sea, se presentaría una frase negativa. Las líneas 21.40 a 21.43 determinaban selectivamente el tipo de predicado; cuando $X1 = 1, 4, 6$ ó 7 , entonces $X7 = 55 + XO$, si $X1 = 2, 3, 5$ ó 8 , entonces X7 era igual a $56 + XO$. En el primer caso el atributo sería el correspondiente al término semejante, en el segundo el propio del discrepante.

Después, es decir, una vez que todas las variables anteriores habían adquirido el valor correspondiente, se ejecutaba el bloque 20 (D 20 de la línea 21.99), que era el encargado de escribir en la pantalla las tétradas y las frases, de registrar los TR y de informar al sujeto de qué tal lo había hecho. Este bloque 20 remitía a otro que se encargaba del almacenamiento de los TR en las matrices correspondientes o si el sujeto había cometido un error en la verificación de la frase, el bloque que se ejecutaba era el correspondiente a errores. Se prescinde de estos bloques porque su explicación alargaría demasiado el artículo y haría su lectura aun más difícil y aburrida.

En resumen, podríamos exponer la correspondencia entre diseño experimental y programa de esta forma:

DISEÑO EXPERIMENTAL

| | | | | | | | | |
|--------------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Condición Experm. | AVS | AFS | NVS | NFS | AVD | AFD | NVD | NFD |
| Sujeto de la Frase | S | S | S | S | D | D | D | D |
| Verbo de la Frase | A | A | N | N | A | A | N | N |
| Predicado de la F. | S | D | D | S | D | S | S | D |
| Respuesta | V | F | V | F | V | F | V | F |

PROGRAMA

| | | | | | | | | |
|----|-------|-------------|-------|-------|-------|-------|-------|-------|
| X1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| X3 | \$ | \$ | \$ | \$ | 54 | 54 | 54 | 54 |
| X6 | 57.01 | 57.01 | 57.02 | 57.02 | 57.01 | 57.01 | 57.02 | 57.02 |
| X7 | 55 | 56 | 56 | 55 | 56 | 55 | 55 | 56 |
| X8 | 1 | -1 | 1 | -1 | 1 | -1 | 1 | -1 |
| | \$ | 51, 52 ó 53 | | | | | | |

En una palabra lo que había que lograr con el programa era que, dada una condición experimental determinada (X1), los valores de las otras variables —que controlan el sujeto, el verbo, el predicado, la respuesta y los estímulos— fueran tales que respondieran a la condición experimental dada.

Espero que el presente artículo sirva para convencer a determinadas personas de la necesidad de incorporar las computadoras en los laboratorios de Psicología, y ayude a otros a utilizar sus conocimientos de software en el diseño de experimentos que, a primera vista, pudieran parecer alejados de lo que aprendieron en los cursos o estudios de programación.

CLARK, H. H. y CHASE, W. G. On the process of comparing sentences against pictures, *Cognitive Psychology*, 3, 472-517, 1972.

DUNCKER, K. On problem solving, *Psychological Monographs*, n.º 270, 1945.

VALLE ARROYO, F. *Negation in objectively plausible contexts*. University Microfilms International, n.º 79-21840.

VALLE ARROYO, F. Negatives in context, *Journal of Verbal Learning and Verbal Behavior*, 21, 118-126, 1982.

WASON, P. C. The contexts of plausible denial, *Journal of Verbal Learning and Verbal Behavior*, 4, 7-11, 1965.