

ABRE No. de archivo borrador: "dispositivo.nombre del archivo", código de lecto-escritura, orden de lecto-escritura, longitud de anotación, acción a efectuar

ESCRIBE No. de archivo borrador, ANOTACIÓN número: lista a escribir

LEE No. de archivo borrador, ANOTACIÓN número: lista a escribir

CIERRA No. de archivo borrador

Como el lenguaje es el gran manipulador de símbolos, resulta evidente que si se quiere vulgarizar el uso de las computadoras es imprescindible desarrollar lenguajes castellanos, pero además es conveniente que las expresiones usadas puedan ser fácilmente retenidas y estructuradas. Una de las formas de facilitar esta estructuración es usar una metáfora apropiada. Un contraejemplo es la versión castellana del TI LOGO,⁷ que por no cumplir estos requisitos resulta muy difícil de memorizar cuando no se hace uso continuo de ella.

Una computadora es un mandadero

Al encarar el aprendizaje de la programación propiamente dicha, la primera dificultad que se encuentra es diferenciar entre lo que está escrito en la hoja pantalla y una línea de programa. Ya en esta primera etapa (y lo será todavía más con posterioridad) resulta útil la metáfora de considerar a la computadora como un mandadero al que se le encomiendan tareas. Esta metáfora tiene, además, el mérito de poner claramente en evidencia que una computadora no es sino un ejecutor versátil de tareas simbólicas, sean estas numéricas, literales o gráficas.

Claro está, no basta escribir una lista de tareas para el mandadero; éste debe enterarse. Es inmediato, entonces, traducir el comando ENTER por ENTÉRATE. Como el mandadero es muy hábil, pero totalmente carente de iniciativa, hay que explicarle detallada y ordenadamente cada uno de los pasos que debe dar. Cada párrafo de la hoja corresponde así a una tarea (línea u oración⁹) y las tareas deben hacerse en un orden que se especifica asignándole a cada una un número. El programa es la secuencia de tareas que el mandadero memoriza para el momento en que se le ordene hacerlas. El comando RUN de comienzo de tareas y el CLEAR de interrupción (FCTN 4) pueden así traducirse por HAZ y PARA, y el mensaje DONDE de pantalla por HECHO. En todo momento debe tenerse especial cuidado con el lenguaje usado diciendo, por ejemplo, "haz la tarea 4" en vez del habitual "ejecutar la línea o sentencia 4"; o, "hacer el programa de tareas" en vez del barbarismo "correr el programa". Se da a continuación una posible lista de equivalencias:

⁹ Además de los frecuentes barbarismos se encuentran también errores de traducción como el de *sentence* por *sentencia* en vez de *oración*.

TERMINOLOGÍA ORIGINAL	REFORMULACIÓN METAFÓRICA
BREAK	INTERRUMPE
BYE	ADIÓS
CALL SOUND	SUENA
CLEAR (FCTN 4)	PARA
DATA	DATOS
DONE	HECHO
ENTER	ENTÉRATE
INPUT	LEE
Línea	Tarea
LINPUT	LEE TODO
LIST	LISTAR
NEW	OLVIDA el PROGRAMA previo de tareas
OLD	MEMORIZA el PROGRAMA de tareas
ON	CUANDO
PEEK	ESPÍA
READ	LEE
REM	COMENTARIO
RESEQUENCE	RENUMERA
RESTORE	RESTABLECE
RUN	HAZ
SAVE	ARCHIVA el presente PROGRAMA

Para poder usar bien los servicios de un mandadero sin iniciativa (aprender a programar) hay dos requisitos. El primero es conocer las habilidades (esquemas o comandos) del mandadero. El segundo es comprender como actúa un mandadero sin iniciativa. De los esquemas provistos por el lenguaje BASIC el más interesante desde el punto de vista de la metáfora es el de las decisiones lógicas tales como

IF A<B THEN C1 ELSE C2

Este esquema lógico se comprende inmediatamente asimilándolo a la siguiente situación familiar. El mandadero, que va de regreso a su casa, llega a un punto donde se bifurca el camino en dos diferentes, C1 y C2. Por ejemplo, C1 es el camino más largo para ir a su casa y C2 el más corto. Si es temprano, el mandadero, a quien le gusta mucho pasear, elegirá el camino más largo C1. Si es tarde, el mandadero, a quien le gusta llegar con puntualidad, elegirá el más corto C2. Si temprano quiere decir antes de las 12 de la mañana y HORA es lo que marca su reloj, enton-

ces $A < B$ corresponde a la comparación $HORA < 12$. En lenguaje común escribiríamos entonces:

SI $HORA < 12$ ENTONCES tomo el camino C1
DE LO CONTRARIO tomo el camino C2

Para aprender a programar es necesario hacer una profunda reflexión sobre la manera en que nosotros mismos hacemos una tarea cualquiera tal como barrer un piso. La metodología así explicitada se aplica luego al caso computacional. Encontraremos que una vez fijada una meta no es en general posible determinar de inmediato un camino bien definido para alcanzarla, y que una vez encontrado este camino, en general, no es único. Para llegar a escribir el programa de tareas el programador debe cumplir en general las siguientes etapas:

1. Elección de una meta computacional.
2. Discusión de los medios computacionales disponibles para llegar a la meta.
3. Formulación tentativa de una secuencia ordenada de tareas que conduzcan a la meta elegida.
4. Realización ordenada y control individual de cada una de las tareas, verificando el grado de acercamiento a la meta.
5. Modificación de las tareas que no cumplen la función deseada.
6. De nuevo 4, etcétera.

La programación se convierte así en una réplica del método científico, donde 3 juega el papel de las hipótesis y 4 el de la experimentación y verificación. En la etapa 3 se puso claramente de manifiesto la ventaja de los sujetos con formación científica (profesores universitarios) que demostraron de manera inmediata su capacidad de explorar ordenadamente un gran número de alternativas.

El valor formativo del aprendizaje de la computación encarado de esta manera es que la experiencia adquirida es metafóricamente aplicable a tareas muy diferentes de las que puede hacer una computadora. En efecto, los esquemas y métodos usados en la tarea no necesitan tener conexión alguna con la lógica o la matemática, y en todas las etapas del proceso pueden jugar un papel decisivo las escalas personales de valores. Piénsen para ello, por ejemplo, en la tarea de consolar a un niño, donde los medios usados son esencialmente intuitivos y hay maneras lícitas y no lícitas de llegar al resultado deseado.

Conclusiones

La experiencia de 24 años de docencia universitaria, con una ocasional incursión en el reciclaje de profesores secundarios, me indicaba que el uso de analogías jugaba un papel importante en el proceso de enseñanza-aprendizaje. Mi reciente descubrimiento de los trabajos de Piaget¹ y otros^{2,3,4,5,6,7,8} sobre los procesos cognoscitivos me alentó a intentar una formulación coherente de este papel. La experiencia hecha con 8 sujetos me permitió comparar la rapidez de aprendizaje de la computación cuando se usa el lenguaje técnico usual y cuando se usan metáforas apropiadas. Pude así detectar en el segundo caso aceleraciones muy marcadas del aprendizaje, que si bien no son estadísticamente significativa, lo son desde el punto de vista cualitativo.

Finalmente, es importante señalar que el proceso inverso también es cierto, ya que la falta de **programas específicos** adecuados puede retardar seriamente el proceso de aprendizaje. Mirelle de Meuron,¹⁰ por ejemplo, señala que los llamados "problemas de conducta" están muchas veces relacionados con la carencia de instrumentos cognoscitivos apropiados para resolver los conflictos humanos. He detectado, por mi parte, que una metáfora inadecuada puede representar una valla insalvable para el aprendizaje de ciertos temas avanzados de la Física. Es imposible, por ejemplo, comprender procesos irreversibles como los acompañados de rozamiento si no se descarta primero la demasiado común metáfora *los procesos físicos son lineales*.

Agradecimientos

Agradezco muy especialmente las valiosas discusiones mantenidas sobre el tema con Eduardo Rabossi, Oscar Nudler y Carlos Mallmann

¹⁰ M. de Meuron, *El uso de información clínica y cognoscitiva en el aula*, en *Piaget en el aula*, libro compilado por M. Schwebel y J. Raph, Editorial Abril, Buenos Aires, 1984, pp. 287-318.