

Sin duda, la cuestión más problemática que surge al tratar de acercar un instrumento tan poderoso como es la Informática a los primeros niveles educativos es —aparte del económico— la que plantean los lenguajes que nos permiten establecer el “diálogo” con la computadora.

A pesar de los grandes pasos que se han dado en este terreno (no olvidemos que durante la primera generación de ordenadores, había que comunicarse con ellos en lenguaje de unos y ceros) con el desarrollo de los lenguajes de alto nivel, léase FORTRAN, COBOL, BASIC, PASCAL, etc., la necesidad de hacer accesible la Informática a nivel popular, así como el necesario rigor lógico con el que han de programarse los ordenadores, motivaron la aparición de lenguajes (caso del BASIC) bastante fáciles de comprender, al menos en una primera impresión, lo que motivó, sin duda, su amplia popularidad actual. Cuando abordamos un proyecto simple, el BASIC nos proporciona una forma sencilla de transmitir nuestras órdenes a la máquina para que sea ella la que ejecute el trabajo de resolución. Sin embargo, si el proyecto es complejo, la deficiente estructuración de dicho lenguaje, hace que la elaboración del programa que lo resuelve, se convierta en una ardua tarea, en ocasiones laberíntica y casi nunca standar, lo que origina que dicho programa sea ininteligible para aquella persona que no intervino en su diseño. Y esto no sólo ocurre con el BASIC, aunque sea el ejemplo más extendido.

Por otra parte, los lenguajes de programación existentes se elaboran en inglés en su mayoría, estableciendo una dificultad adicional nada desdeñable en su aprendizaje que, si ya ocasiona problemas a los adultos, se incrementará sin duda en los niveles más tempranos de la enseñanza.

Desde el plano de la educación, es necesario disponer de unos lenguajes de comunicación con el ordenador más naturales; más estructurados y flexibles, de tal manera que la labor de programación se asemeje cada vez más a nuestra forma de pensar.

Uno de los lenguajes que más se acerca a este criterio es LOGO, desarrollado por SEYMOUR PAPERT —discípulo de J. Piaget— en el laboratorio de Inteligencia Artificial del M. I. T. La influencia piagetiana hace concebir a Papert al ordenador como el medio adecuado para que el niño desarrolle sobre él la concepción epistemológica del aprendizaje: El aprendizaje por el descubrimiento natural de las ideas. El aprendizaje sin programas rígidos y preestablecidos.

Esto representa una revolución en la metodología de uso de la Informática en la Educación. Hasta LOGO, los intentos realizados en este campo, habían

concebido al niño como al sujeto “programado por el ordenador”. El alumno está dirigido en todo momento por la máquina quien le facilita información, le examina y le evalúa, y, en todo caso, le aconseja acerca de su mucho o poco aprovechamiento, aunque debamos reconocer que en ciertos aspectos, esta concepción representa una importante ayuda al aprendizaje tradicional. Esta es la llamada filosofía C. A. I. (Enseñanza asistida por ordenador): El *ordenador* es un *potente medio de ayuda al aprendizaje tradicional*.

La filosofía de Piaget plantea un problema más arduo. Debemos encontrar un medio en el cual el alumno pueda apoyarse para explorar y descubrir nuevas ideas. El mundo que le rodea es el primer soporte de esta exploración, pero la mera captación sensorial no le sirve, en muchas ocasiones, para reflexionar acerca de las conexiones y dependencias entre distintas ideas. El paso de la simple exploración y reconocimiento a la abstracción representa el “lo sé hacer pero no se como enseñarle a otro a hacerlo”.

Papert, desarrolla ampliamente esta idea con el lenguaje de ordenadores LOGO. Una de sus “poderosas ideas”, como él denomina, es una nueva Geometría: la “Geometría de la Tortuga”. El niño explora el mundo icónico sobre la pantalla del ordenador. Sencillas pero poderosas órdenes hacen que el ordenador, simbolizado por una “tortuga de pantalla”, se convierta en un aprendiz, al que el niño debe enseñar a diseñar la figura que quiere realizar. No es, en absoluto, este proceso comparable al acto de efectuar el dibujo con un lápiz y un papel. Significa el haber aprendido no sólo “a hacer” sino a “saber enseñar a hacer”.

De esta manera el niño se convierte en protagonista. Es él quien programa y domina al ordenador. El ordenador se comporta, a su vez, como un aprendiz muy riguroso, pues sólo entiende aquello que se le enseña bajo la forma de *algoritmos*. Esto requiere que el lenguaje en el que se comuniquen sea a la vez sencillo de aprender (similar al lenguaje natural), y potente en sus posibilidades.

El LOGO ha pretendido ambas cosas, y aunque las versiones disponibles en la actualidad, todavía deben de mejorarse, ya suponen un gran avance en esta línea. Desde el punto de vista de su uso en un país no anglosajón, quizás radica en que sus traducciones no se ajustan lo debido al lenguaje natural. Otro de sus problemas es la pantalla: es plana y está en posición vertical. El niño vive en 3 dimensiones. Sin embargo, se pueden corregir pequeños detalles, que, sin duda, contribuirán a hacer este lenguaje uno de los más acertados ejemplos de lo que debe ser un lenguaje informático adecuado a la educación.

La más “poderosa idea” de este lenguaje es la de *estructuración*. Quizás más que ningún otro, LOGO ha desarrollado la posibilidad de desmenuzar proyectos complejos en un amplio abanico de proyectos muy simples (procedimientos y subprocedimientos) que, conjuntados armoniosamente, originan el proyecto final.

Usamos una de las partes más importantes de LOGO: La Geometría de la Tortuga para construir un ejemplo. Imaginemos el diseño de un *AVION*.

Procedimiento *avión*:

1. COLA
2. FUSELAJE
3. MORRO
4. HELICE
5. ALA DERECHA
6. ALA IZQUIERDA (ver figura 1).

Nuestro AVION se compone de estos 6 “subprocedimientos”. Por separado (ver figura 2) se irán diseñando estas partes. La “tortuga de pantalla” obedece a las palabras primitivas ADELANTE (abreviado AD) *n*; ATRAS (AT) *m*, GIRAR DERECHA (GD) *r*, y GIRAR IZQUIERDA (GI) *s*, en donde *n*, *m*, representan pasos de tortuga y *r*, *s* representan grados de giro. La tortuga de pantalla es simbolizada por un triangulito (ver figura 3), y conoce estas palabras, siendo también capaz de aprender aquellas nuevas palabras que nosotros le “enseñemos”. Para enseñarle una palabra inédita debemos proceder así:

PARA COLA

GI 30 AD 30 GI 60 AD 10 GI 90 AD 60
FIN

De la misma manera que, cuando dibujamos sobre un papel, levantamos el lápiz cuando nos es necesario, para no producir líneas no deseadas, debemos indicarle a la “tortuga” que se sitúe en un lugar determinado sin haber dejado ninguna estela.

Ahora mismo, la tortuga conoce la palabra nueva COLA. En cuanto le pidamos *cola*, dibujará la cola del avión. Ya forma parte de su vocabulario. Construyamos ahora las restantes partes del AVION.

PARA FUSELAJE

PL GD 9 ϕ AD 1 $\phi\phi$ GI 9 ϕ AD 2 ϕ PB GI 8 ϕ AD 1 $\phi\phi$ GI 2 ϕ AD 1 $\phi\phi$
PL AD 2 ϕ GI 8 ϕ AD 4 ϕ PB GI 8 ϕ AD 1 $\phi\phi$ GI 1 ϕ PL AD 4 ϕ GI 1 ϕ PB
AD 8 ϕ .
FIN

PARA MORRO

REPETIR 2 ϕ veces (AD 2 GD 9) – Semicircunferencia de diametro 4 ϕ –
FIN

PARA HELICE

CUADRADO

AD 1φ GD 1φ

TRIANGULO

GI 9φ AD 1φ GI 9φ AD 1φ GD 1φφ

TRIANGULO

FIN

PARA ALA DERECHA

GI 135 AD 7φ GI 135 AD 15 GI 3φ AD 75

FIN

PARA ALA IZQUIERDA

GI 45 AD 4φ GD 135 AD 15 GD 3φ AD 44

FIN

Como en hélice se utilizan CUADRADO y TRIANGULO, hay que definirlos:

PARA CUADRADO

REPETIR 4 VECES (AD 1φ GD 9φ)

FIN

PARA TRIANGULO

AD 2φ GD 16φ AD 2φ GD 1φφ AD 1φ

FIN

Las piezas, por separado, están construidas en LOGO. Ahora solo falta acoplarlas.

Dos consideraciones para ello. La primera es relativa al orden de colocación. Podemos empezar por la cabeza o por la cola. O, mejor, por el fuselaje, acoplándole todas las demás piezas. La segunda consideración es relativa a la posición de la tortuga cuando empieza a diseñar una de las partes.

Podría ser:

PARA AVION

FUSELAJE PL

GI 8φ AD 4φ GD 9φ PB

MORRO PL

GD 18φ AD 2φ GI 9φ AD 2φ GI 9φ

PB AD 5 GI 9φ AD 5 GI 18φ

HELICE PL

GD 18φ AD 115 GI 9φ AD 2φ PB

ALA IZQUIERDA PL
GD 6¢ AD 4¢ GD 8¢ AD 4¢
PB ALA DERECHA PL
GI 15¢ AD 8¢ GD 9¢ AD 4¢ PB
COLA
FIN

Todas las órdenes primitivas que acompañan a los subprocedimientos, se intercalan para colocar a la tortuga en el sitio oportuno para acoplar las piezas al fuselaje.

Este ejemplo de un proyecto complejo, pone de manifiesto la descomposición en proyectos simples que propicia LOGO. Si se recapacita sobre esta forma de “enseñar al ordenador” a realizar proyectos, se observa que, la forma de decirle las cosas, importa tanto como el orden en el que se le suministran las órdenes simples. En el caso del procedimiento para HELICE, se han dado los procedimientos CUADRADO y TRIANGULO, que si no se acoplan adecuadamente producirían resultados no sospechados por el niño.

Estas sorpresas son las que —en lenguaje de PAPER— nos enseñan a “pensar sobre nuestro propio pensamiento” y, el error tan denostado en el aprendizaje tradicional, puede convertirse en un importante aliado en el desarrollo de nuestra capacidad de razonamiento.

Es, de momento, un poco de ciencia ficción el conjeturar sobre la transformación que todo ello podría ocasionar en el aprendizaje. Sin embargo, la tecnología ya está disponible y las experiencias, aunque en pequeña escala, deben comenzar.

BIBLIOGRAFIA

1. ABELSON, H. “Turtle Geometry. The Computer as a medium for Exploring Mathematics”. M. I. T. Press. 1982.
2. ALVAREZ, M; CAJARAVILLE, J. A.; OTERO, M.; PUEYO, M. “Informática para docentes” Ed. Anaya, 1984.
3. CAJARAVILLE, J. A. “O papel da Informática no ensino”. Revista O ENSINO, núm. 6, pp. 19-24. Abril, 1983.
4. CAJARAVILLE, J. A. “Simulación educativa con microordenador”. Actas de la I semana de Informática aplicada a la Industria y la Educación. E. T. S. de Ingenieros de Minas. Madrid, 1984.
5. PAPER S. “Desafío a la mente”. Ed. Galápagó. Buenos Aires, 1982.
6. REGGINI, H. “Alas para la mente”. Ed. Galápagó. Buenos Aires 1983.

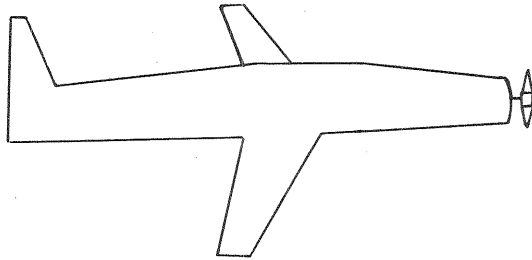


Figura 1.

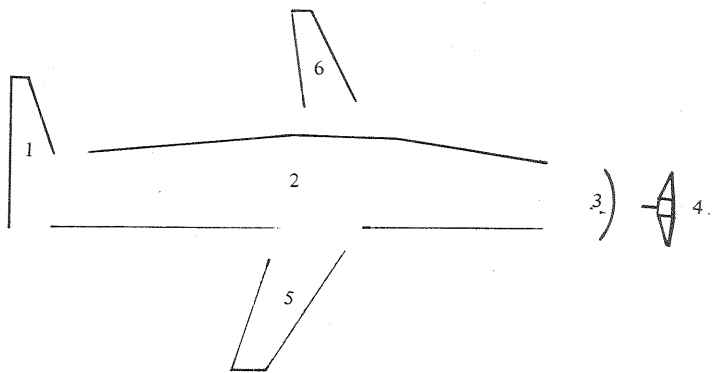


Figura 2.

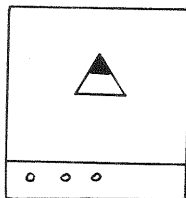


Figura 3.