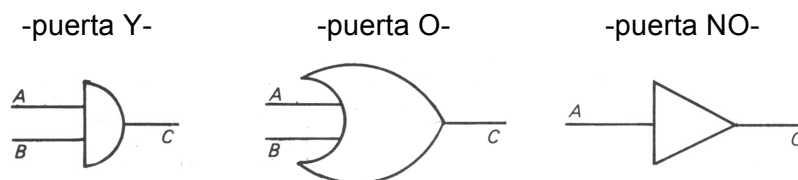


La informática básica

Un ordenador es una estructura física, una máquina que procesa información. Por tanto, en la arquitectura física de un ordenador encontramos: la Unidad de Entrada, la Unidad Central de Proceso y la Unidad de Salida. Y, por supuesto, una red de comunicación entre todos y cada uno de los componentes.

A su vez el **procesador** (Unidad Central de Proceso) está subdividido en: Memoria, Unidad de Control y Unidad Aritmético-Lógica u Operador.

Si atomizamos un ordenador nos encontramos con que éste se reduce a una combinación extraordinariamente compleja de puertas lógicas, que se simbolizan de esta manera:



Cada ordenador tiene una estructura física determinada a causa de la combinación de estas puertas lógicas: en el caso de calculadoras de naturaleza electrónica, estas puertas se materializan en semiconductores sobre una oblea de silicio.

Esta estructura física está diseñada para que al ordenador se le introduzcan programas. Un **programa** es una serie de **instrucciones secuenciales** que la Unidad de Control mete en la Memoria y luego va extrayendo, interpretando, analizando y enviando paulatinamente a la Unidad Aritmético-Lógica para que las procese.

Las estructuras básicas de programación son: la secuencial, la repetitiva (bucle) y el salto de control (condicional e incondicional).

Para elaborar estos programas secuenciales se han creado unos lenguajes de programación adaptados a facilitar los problemas que se quieren resolver: FORTRAN, COBOL, BASIC, PASCAL, LOGO, etc.

Hasta aquí estamos describiendo la calculadora universal numérica secuencial. **Universal** porque en principio puede hacer cualquier tarea, siempre que se le dé el programa **secuencial** adecuado en un lenguaje que la máquina "entienda". Estas son las características más comunes de la mayoría de los ordenadores (o computadores) que conocemos.

Las dificultades aparecen cuando un ordenador trabaja en un problema concreto: tratamiento de textos, contabilidad, EAO (Enseñanza Asistida por Ordenador), etc., y se exige unas condiciones de procesamiento específicas como pueden ser mayor rapidez de cálculo, adaptación al problema concreto, programas más sencillos de confeccionar, etc. En algunos casos es más costoso un programa complejo que la propia máquina con muchos de sus periféricos, incluida la Unidad de Discos y la impresora. En programación siempre queda la posibilidad de mejorar un poco más lo que hasta ese momento está hecho.

Todo esto nos lleva al **meollo del problema**: utilizar algoritmos que permitan una mejora de los programas para los ordenadores, es decir, algoritmos que se adapten de antemano al problema que vamos a tratar.

Algoritmo, en sentido no riguroso pero sí suficientemente preciso para los niveles en que nos movemos, es definido por TRAJTENBROT de esta manera:

"Entendemos por algoritmo la prescripción exacta sobre el cumplimiento de cierto sistema de operaciones en un orden determinado para la resolución de todos los problemas de algún tipo dado" (1).

Pero no basta con esto; una vez hallado un algoritmo ideal hace falta procesarlo, cosa que puede afrontarse de muchas maneras. Así, por ejemplo, tomemos el algoritmo de ordenación alfabética. Podemos procesarlo:

- a mano con lápiz y papel si se trata de una lista; con simple mano si se trata de tarjetas cada una con una palabra o nombre para ordenar,
- máquina de escribir o máquina diseñada adecuadamente para ordenar las tarjetas,
- con un procesador electrónico universal,
- con un procesador diseñado explícitamente para hacer más rápido y sencillo el tratamiento de este algoritmo.

Es decir, que la arquitectura del ordenador (su estructura física) está ligada íntimamente con el algoritmo a tratar. Por ejemplo, para el tratamiento de imágenes lo ideal será un algoritmo que tome como base la necesidad de reticular la imagen y de relacionar cada punto con todos sus vecinos según unos límites de gradación que van del negro al blanco, pasando por toda la gama de grises, si es blanco y negro.

El principio para construir un procesador de este tipo sería: aplicar un tratamiento individual a cada retícula de la imagen. La arquitectura que mejor se adapta a este tratamiento sería una del tipo SIMD en tabla; un multiprocesador en paralelo, es decir, una red de procesadores calcada bajo la retícula de la imagen.

¿Quedarán las actuales CUNES (Calculadoras Universales Numérico-Electrónicas Secuenciales) totalmente obsoletas en poco tiempo? ¿O más bien se emplearán para diseños de arquitecturas específicas de calculadoras concretas? Se habla de máquinas dirigidas por los datos, de lenguajes que supriman variables, de cambiar la mentalidad secuencial de los actuales programas, etc.

Se está investigando activamente en la actualidad con "el principal objetivo de encontrar métodos automáticos que permitan diseñar arquitecturas de este tipo a partir de la descripción del algoritmo que se desea efectuar" (2).

Un problema similar e incluso quizás más grave ocurre con la introducción de los ordenadores en el aula; en especial con la EAO. ¿Están adaptadas las máquinas actuales al problema del diseño multivariante de una programación escolar? Los **lenguajes secuenciales** existentes (BASIC entre ellos) no; y por eso fueron creándose **lenguajes de autor** como PLANIT TUTOR, PILOT, etc., o los **sistemas, de autor** como ARLEQUIN, PLATO, que es un programa o editor pedagógico "que suministra la estructura para hacer el programa definitivo" (3).

El LOGO, a pesar de tener sus raíces en la Inteligencia Artificial (4), supone una adaptación de un lenguaje artificial al pensamiento del niño programador. Su recursividad, o las primitivas y definiciones de tipo diccionario, permiten emplearlo en una serie de utilidades más próximas al escolar que otros lenguajes secuenciales/es. Pero, aun así, tiene muchos límites como, por ejemplo, la complejidad de los niveles de definición que se necesita utilizar para programar un algoritmo medianamente complicado en básica como el cálculo del máximo común divisor y del mínimo común múltiplo.

Ateniéndonos a estas razones, aunque ciertamente ni son las únicas ni quizás las principales razones, pensamos que no se puede exigir hoy día al maestro que utilice el ordenador en el aula como si de la mismísima tiza se tratase. Las horas y la práctica necesaria para elaborar un buen programa secuencial/ de EAO son tan grandes que en la estructura organizativa actual de los centros de EGB españoles hacen inviable que se alcance siquiera un nivel correspondiente al de un buen libro de ejercicios electrónicos.

Y los paquetes de programas confeccionados por informáticos y educadores siempre tienen el riesgo de la imposibilidad de adaptarse a las necesidades concretas de la situación educativa que lo demanda.

Con esto no queremos dar una visión negativa de los grandiosos proyectos de, introducción de la Informática en la Escuela, sino simple y llanamente **prevenir**. No podemos comprar un ordenador simplemente porque nos lo venden y nos cantan cien maravillas del mismo, ni porque es un buen gancho publicitario para el centro en el que trabajamos, ni porque lo tiene el vecino y nosotros no queremos pasar ni por pobres ni por anticuados, ni siquiera para ayudar a la industria nacional o regional. No debemos gastar un dineral en máquinas y programas cuya utilización no está clara y sus ventajas y -utilidades a largo plazo están menos claras aún.

Sin embargo, tampoco hemos de esperar a que el desarrollo de las líneas de investigación actuales nos fabrique un multiprocesador (hiperordenador) que nos resuelva con modos muy simples y directos todos los problemas de la programación.

Desde el punto de vista de la educación básica y general se debe y se puede estar al día en los temas de Informática, y de una manera audaz seguir paralelamente el desarrollo apasionante de esta nueva rama tecnológica. Por esto pensamos que los **Talleres de Informática** con o sin ordenador, con ordenador de papel o juegos lógico-matemáticos son hoy por hoy algo que debería estar en la mayoría de las aulas de EGB.

Un **Taller de Informática** puede desarrollarse construyendo máquinas:

1. Construyendo máquinas.

1.1 Ábacos.

1.2. Minicomputador de Papy.

1.3. Puertas lógicas.

1.4. Artefactos lógicos.

1.5. Cuadrado lógico de Lewis Carroll.

1.6. Tarjetas perforadas.

2. Construyendo programas (o fundamentos de la programación) en este caso secuenciales.

2. 1. Rudimentos de programación: recetas de cocina, juguetes programables, juegos de orientación, robot en clase, geometría de la tortuga.

2.2 Máquina de Post.

2.3. Recursos básicos de programación: contador, ordenación.

3. Construyendo algoritmos.

3. 1. Juegos deterministas (secuenciales): estrategias, diagramas de árbol.

3.2. Juegos no secuenciales: juego de la vida.

3.3. Juegos de búsqueda heurística.

En todo caso remitimos a los interesados a una abundante bibliografía en la que podrán obtener múltiples ideas para lo que estamos proponiendo.

Por otra parte, el propio Taller de Informática pudiera servir para producir materiales o recursos pedagógicos de utilización en el aula, y adaptados a los problemas concretos del propio enseñante y grupo o situación pedagógica en que se utiliza.

NOTAS BIBLIOGRAFICAS

(1) **Trajtenbrot, B.A:** Los algoritmos y la resolución automática de problemas. Mir, Moscú, 1977.

(2) **Quinton, Patrice:** Mundo Científico, 50, septiembre 1985, pág. 856.

(3) **Fernández González, Manuel:** Enseñanza Asistida por Ordenador. Anaya, Madrid, 1983.

(4) **Mullán, A. P.:** El ordenador en la educación básica. Gustavo Gili, Barcelona, 1985.

BIBLIOGRAFIA

Aguado-Muñoz, R.; Blanco, A., y Zamarreño, R.: Las calculadoras en el aula. Anaya, Madrid, 1982.

A.-Iken, H., y otros: Perspectivas de la revolución de las computadoras. Alianza. Madrid, 1975.

Azarquiel (grupo) y Cólera, J.: La calculadora de bolsillo como instrumento pedagógico. ICE de la UA-M, 1983.

Carrol, L.: El juego de la Lógica. Alianza, Madrid, 1972.

Colomb, J., y otros: Fichas perforadas. Teide. Barcelona, 1973.

Gariner, M.: Carnaval matemático. Alianza, Madrid, 1981.

Garimer, M.: Nuevos pasatiempos matemáticos. Alianza, Madrid, 1972.

Garimer, M.: Circo matemático. Alianza, Madrid, 1983.

Grunfeld, Frederic: Juegos de todo el mundo. Edilán, Madrid, 1978.

Sierra Vázquez, M.: El minicomputador de Papy en el cielo inicial de EGB. ICE. Universidad de Salamanca, 1983.

Trajtenbrot, B.: Los algoritmos y la resolución automática de problemas. Mir, Moscú, 1977.

Uspenski, V.: La máquina de Post. Mir, Moscú, 1983.

Diego Justicia Hernández-Prieta
Desiderlo Fernández Malón