

En síntesis: la sistematización de la información es una tarea de todos en cada organización, fundamentada sobre la base sólida del conocimiento específi-

co de las necesidades que brotan de sus objetivos, sus funciones y sus formas de operación.

PROBLEMAS DEL APRENDIZAJE DE LA PROGRAMACION DE COMPUTADORES

FRANKLIN J. VALVERDE DELGADO

Especialista en Sistemas de Información, Universidad EAFIT - ICESI, Ingeniero Electricista, Universidad del Valle. Profesor del Liceo Benalcázar, de la Universidad del Valle y del ICESI. Asesor en Sistemas.

El mundo entero se enfrenta hoy al reto de los computadores. Todos quieren aprender; sin embargo, ¿será cierto que se aprende fácilmente y en poco tiempo? Creo que depende de la persona y de la orientación que reciba, ya sea a través de una institución o por su propia cuenta; muchos dicen que aprendieron a programar solos o que determinada institución es buena o mala porque aprendieron o no a programar.

Cualquier área del conocimiento tiene su grado de dificultad y el esfuerzo que hacen las personas por aprender lo que les gusta será menor que aquel que harán por lo que la necesidad les obliga. La palabra clave del aprendizaje es "necesidad". Para muchos será impositiva mientras que para otros será placentero el esfuerzo. Analicemos el problema desde otro ángulo. Por ejemplo, un médico de aquellos que se formaron sin estar rodeados de computadores. La necesidad de aprender computación de pronto es otra y considero que es la de aprender a usar

la máquina en aplicaciones propias de la medicina y adicionalmente poder comunicarse con el analista de sistemas, especificándole sus necesidades de manejo de información. En el mismo caso, un estudiante de medicina hoy en día deberá estar preparándose como un buen usuario del computador acompañado de una acertada técnica de comunicación para poder participar fácilmente en grupos de investigación médica que utilicen el computador como ente auxiliar, junto con el personal de sistemas.

Todas las facultades de ingeniería tienen en su pènsum oficial cursos de computación que obligan al estudiante a aprender a programar en uno o varios lenguajes de programación de alto nivel como Basic, Fortran Cobol, Pascal, etc. y el gran problema del aprendizaje está en el diseño de los algoritmos. Cuando estos cursos se ven en los dos primeros semestres de la carrera, el bajo rendimiento observado no es por falta de capacidad del estudiante sino que, pien-

so, es por falta de disciplina mental en el manejo de la lógica y un gran respeto hacia los detalles que el estudiante de bachillerato al llegar a la universidad generalmente adolece de este problema. Y también porque muchos profesores creen que la lógica es algo fácil y en verdad lo es, pero después de un buen entrenamiento para muchos. Además, hay que tener en cuenta que todos no nacimos para ser médicos o ingenieros o administradores, etc.

¿COMO RESOLVER EL PROBLEMA?

1.0 Que cada quien se identifique con lo que es y no pretenda saberlo todo. El médico es médico, lo mismo que es el administrador, el abogado o cualquiera que tenga su profesión definida. La cuestión es educar al profesional del área distinta a la ingeniería de sistemas a comunicarse con el ingeniero, analista o programador de sistemas para que éste le diseñe los programas y se los enseñe a manejar facilitándole su labor profesional. En otros términos, ser un buen usuario de los sistemas de información de manera que las instituciones universitarias ofrezcan seminarios con metodologías orientadas al logro de este fin. Claro está que si la persona tiene facilidad para aprender a diseñar sus propios programas, ¡adelante! —serán excepciones a la regla—.

2.0 Todos los estudiantes que empiezan carrera se pondrán en contacto con la máquina y un lenguaje donde estarán resolviendo programas tipo, a la vez que el profesor explicará lo que está ocurriendo en la máquina tales como flujo de datos en la C.P.U. y el seguimiento, paso a paso, de la lógica del programa. Esto porque el estudiante neófito cree que el computador asume determinados pasos o nombres y variables e incluso, lógica. De allí que sea necesario este primer contacto con la máquina, que el estudiante sepa que ella no hace nada si el programador no le da las instrucciones correctamente.

Cuando se comienza realizando los algoritmos y después se conoce el lenguaje, se encuentran muchos casos de estudiantes que no le metieron datos al programa porque como el algoritmo trabaja sin datos físicos ellos creyeron lo

mismo para el computador, como si éste los asumiese. En otros casos se hallan errores de lógica también porque el estudiante piensa que la máquina se los resolverá.

He observado casos últimamente con estudiantes de primer semestre de ingenierías en la Universidad del Valle que reciben el curso tradicional de programación Fortran y que han tenido su primer contacto con la máquina en los colegios de bachillerato, de donde provienen. Normalmente entienden bien los algoritmos planteados en clase y resuelven otros con gran facilidad, llevándoles gran ventaja a sus compañeros que no han recibido estos cursos de programación.

3.0 Una vez que el estudiante ha pasado el primer curso planteado en el punto 2.0, yo propondría de 50 horas/máquina con la orientación del profesor. Viene la tarea más difícil: enseñarle a pensar. Sí, a pensar, porque eso de diseñar un buen algoritmo no es tarea fácil y se requiere de una gran paciencia de parte y parte, profesor y estudiante. ¡Cuántas veces se cometen errores como pretender diseñar algoritmos de situaciones que el estudiante no conoce! Por ejemplo, es frecuente encontrar en un primer curso de programación modelos de series como funciones trigonométricas o trascendentes como logaritmos naturales o análisis de regresión cuando el estudiante no ha recibido un curso de cálculo o de estadística donde se le haya enseñado los problemas de convergencia o de divergencia de una serie, o por qué tal ecuación de una curva se ajusta a los datos que analiza.

Un fenómeno interesante se observa cuando se resuelven algoritmos de problemas cuyo método de solución él conoce. Por ejemplo, resolver completamente la ecuación de segundo grado, normalmente, el estudiante se emociona cuando ve que puede resolver el algoritmo y más aún poder codificarlo y correrlo en un lenguaje en particular.

Así que no debe acelerarse el conocimiento para que el estudiante no caiga en desaliento ante el curso que recibe. Ahora bien, ¿por qué no ofrecer estos cursos, por ejemplo en quinto semestre, cuando

ya el estudiante ha recibido una mejor formación universitaria? Finalmente, retomemos los planteamientos del autor Frank. D. Vickers, en su libro: "Fortran IV: un enfoque moderno"* acerca de las características personales de los programadores de computadores: "por regla general, aparentemente esas características son las siguientes:

1. Una mente lógica
2. Una buena memoria a corto plazo
3. Un gran respeto hacia los detalles

4. Una gran tendencia hacia la perfección.

5. Una constancia incansable para no desanimarse ante los fracasos".

Así que el reto de aprender a programar un computador requiere de mucho estudio combinado con una práctica constante en la máquina y de esta manera se ajercitan la lógica y la memoria, logrando mejorar algoritmos que se creían eran la mejor solución. Se logrará entonces el respeto hacia los detalles y por ende se buscará la perfección e implícitamente el quinto punto se estará cumpliendo.