Experiencias metodológicas para potenciar el desarrollo de Filosofías de Trabajo con las estructuras de control a partir de una Hoja Electrónica de Cálculo.

Autores:
M. Sc. Esmereldo Carbó Salazar
carbo@hlg.rimed.cu
Lic. Armín González Almaguer
arming@hlg.rimed.cu
M. Sc. José Soler Silva
jsoler@hlg.rimed.cu

Resumen

Debido al vertiginoso desarrollo de la informática ha sido necesario en cortos lapsos variar los contenidos de computación que conforman los currículos de las carreras pedagógicas, en particular la de Informática, lo que ha incidido negativamente en la preparación de los estudiantes para enfrentar los nuevos paradigmas impuestos por el desarrollo acelerado de la tecnología, debido, fundamentalmente, a que a estos no se les ha enseñado a que aprendan Filosofías de Trabajo Informáticas (FTI) que les permitan enfrentar los nuevos cambios sobre la base de habilidades y principios de trabajo que trasciendan un Lenguaje de Programación o un Sistema de Aplicación; de ahí la urgencia de que los profesores comprendan la necesidad de enseñar FTI. Este trabajo tiene como objetivo mostrar una vía metodológica para el desarrollo de estas filosofías en las estructuras condicionales, a partir de un Sistema de Aplicación.

Palabras claves: Filosofías de trabajo informáticas, condicionales, estructura alternativa.

Summary

Due to the computer science's vertiginous development it has been necessary in short lapses to vary the calculation contents that conform the curricula of the pedagogic careers, in particular that of Computer science, it has impacted it negatively in the preparation of the students to face the new paradigms imposed by the quick development of the technology, fundamentally to that to these they have not been taught to that learn Computer Philosophies of Work (FTI) that allow them to face the new changes on the base of abilities and work principles that a Language of Programming or a System of Application transcend; of there the

urgency that the professors understand the nonsense of teaching FTI. This work has as objective to show a methodological road for the development of these philosophies in the conditional structures, starting from a System of Application.
Key words: Computer, conditional work philosophies, it structures alternative.

Tradicionalmente se han impartido los Lenguajes de Programación (LP) como asignaturas independientes y se han abordado las estructuras lineal, condicional y cíclica como patrimonio de un lenguaje determinado (Basic, Pascal, Visual Basic o Delphi), y no a partir de una filosofía de trabajo en la lógica de la programación. Esta problemática también se evidencia en el estudio de los Sistemas de Aplicación (SA). Al tomar en cuenta los referidos elementos, en este artículo se aborda cómo enseñar a aprender Filosofías de Trabajo Informáticas en la programación, al tomar como punto de partida una Hoja Electrónica de Cálculo (HEC).

Por el amplio campo que tiene la Informática, al afrontarla como objeto de estudio en los currículos se aborda en dos vertientes: los SA y los LP. Tanto los SA (Procesadores de Textos, Sistemas de Gestión de Bases de Datos –SGBD–, Graficadores, Presentadores Electrónicos, HEC, etc.) y los LP son tratados como unidades o asignaturas independientes; estos últimos tienen incidencia en las aplicaciones, no solo como sustento de estas, sino también resultan de gran utilidad para lograr una mayor explotación de los SA en la solución de problemas y tareas integradoras; tal es el caso del Visual Basic para Aplicaciones (VBA).

Por otra parte, la programación con mayor incidencia en los diseños curriculares de la especialidad de Informática en el contexto pedagógico, ha tenido el propósito de dotar a los estudiantes de un conjunto de técnicas de programación propias del lenguaje objeto de estudio, pero sin prepararlos para que se apropien de una filosofía de trabajo que les permita integrar los conocimientos y habilidades en la solución de tareas y en la adquisición de nuevos conocimientos.

Las Filosofías de Trabajo Informáticas (FTI) constituyen el conjunto de herramientas, procedimientos y principios de trabajo comunes, que desarrollan habilidades sostenibles en los estudiantes, integrando los recursos computacionales de forma tal que permitan obtener respuestas rápidas y eficientes en la integración y solución de tareas.

Cuando aquí se habla de habilidades sostenibles, se hace referencia a un conjunto de destrezas que adquieren los estudiantes al iniciar el estudio de un Sistema Operativo y que

trascienden y se consolidan en los SA, LP y en la utilización de Softwares Educativos. Ejemplo de estas destrezas son:

- Definir criterios de búsqueda y selección de la información.
- Habilidades en la utilización de la ayuda.
- Algoritmo general para las operaciones con la información.

En la Didáctica de la Informática, según Expósito (2001), hay tres regularidades: la formación de conceptos, la elaboración de procedimientos y la resolución de problemas con computadoras. Las dos primeras constituyen las formas predominantes en la adquisición del conocimiento: la formación de conceptos se enmarca en el desarrollo del saber; mientras que la esencia de la segunda radica en el saber hacer, fundamentalmente, en el perfeccionamiento de habilidades mentales y manipulativas. Ambas regularidades favorecen el desarrollo de las FTI empleando los nuevos paradigmas de la programación y los sistemas de aplicación.

El mecanismo más elemental y que se asume en la mayoría de los LP, consiste en que las instrucciones que conforman el código de un programa se ejecutan en el orden en que aparecen físicamente en este. Tal criterio se corresponde con la jerarquización en que se ejecutan los comandos en la mayoría de las computadoras. Este único criterio de orden de ejecución no aprovecha las ventajas que ofrecen los ordenadores en cuanto al tiempo de cálculo y está limitado por la cantidad de instrucciones que pueden ser representadas consecutivamente en la memoria de estos. La mayoría de los LP incluyen instrucciones que permiten romper explícitamente la secuencia de ejecución y desviarla a otro punto del programa.

En este trabajo se toma como punto de referencia la asignatura Programación Visual II de la especialidad de Informática, para la cual se desarrolla un ejemplo donde se muestra cómo potenciar el desarrollo de FTI en la programación, a partir de un SA (Microsoft Excel como representante de las HEC).

Cuando se opera con más de una hoja de trabajo y es necesario borrar una de ellas, al invocar el procedimiento de eliminar, aparece un mensaje de alerta con dos botones “Eliminar y Cancelar”; tradicionalmente se centra la atención en dar respuesta a este mensaje, en dependencia de los intereses del usuario, sin importar en qué se sustenta dicha pregunta. Un usuario común sólo debe saber qué resultado obtendrá al tomar una decisión; sin embargo, los profesionales en formación de la especialidad de Informática, necesitan

conocer la lógica del funcionamiento de la "*caja negra*", es decir, lo que no es visible al que opera con los SA, y no se hace referencia a la forma en que esté programado el procedimiento o la función, sino a mostrar la realización de las acciones en dependencia de una condición (SI), que puede ser simple o compleja.

En el ejemplo anterior se realiza una acción: "Hacer clic sobre el botón eliminar" (para borrar la hoja). En el procedimiento hay algo que debe cumplirse, por lo que la operación no se hará siempre, sino que debe verificarse una condición.

Todo esto permite fomentar en los estudiantes FTI que sean sostenibles y les permitan un aprendizaje continuo y eficiente ante las tareas profesionales.

Retomando el ejemplo del borrado de la hoja de trabajo se puede valorar con los estudiantes, sobre la óptica del desarrollo del pensamiento algorítmico, un fragmento de uno de estos:

SI se cumple la condición, entonces: hago las acciones del paso 1. .... Si se cumple la condición, entonces hago las acciones del paso 2.

Lo cual se traduce en:

Si se hace clic sobre el botón eliminar, entonces se ejecutan las operaciones del procedimiento borrar. Si se hace clic sobre el botón cancelar, entonces se ejecutarán las operaciones para retornar a la hoja de trabajo.

En este caso es obligatorio ejecutar una de las dos acciones para poder continuar las operaciones de trabajo con la HEC. Hay muchos ejemplos en los SA que permiten potenciar el desarrollo de FTI a través de las condicionales, incluyendo el Sistema Operativo; por ejemplo, cuando se solicita la confirmación del borrado de un fichero o una carpeta puede realizarse un análisis similar. No es propósito de este trabajo valorar todas las estructuras.

El siguiente ejercicio se ha utilizado para la introducción a la programación:

- ✓ Un Profesor General Integral de preuniversitario tiene los resultados de Matemática, Historia y Español referidos a la prueba de ingreso a la Educación Superior de 20 de sus alumnos y necesita:

a) Imprimir el nombre de los estudiantes que tienen nota superior a 95 puntos en Matemática.

b) Calcular el promedio de los estudiantes que tienen notas superiores a 90 puntos en todas las asignaturas, de no ser posible, expresarlo a través de un texto.

Aspectos a tener en cuenta en el análisis del ejercicio para fomentar el desarrollo de FTI:

- Orientar a los estudiantes sobre la base del objetivo a lograr (principios de las condiciones).
- Proponer que se elabore un algoritmo general para la solución del ejercicio, sin estar dirigido a un SA o LP en particular.
- Propiciar un análisis con los estudiantes para que identifiquen los elementos esenciales que caracterizan las condiciones.
- Sobre la base del algoritmo general, ¿en qué SA de los que usted conoce, es más viable la solución del ejercicio?
- De los SA estudiados el que mayores potencialidades tiene para resolver este tipo de ejercicio es la HEC y, en particular, Microsoft Excel.

Un algoritmo general pudiera ser el siguiente:

- Recopilar la información de los 20 estudiantes.
- Si la nota de Matemática es mayor que 95, entonces imprimir el nombre (este paso se repite para los 20 estudiantes).
- Si el estudiante tiene notas superiores a 90, entonces calcular el promedio, sino imprimir un mensaje (esto se repetirá para cada estudiante).

Como se puede observar, el algoritmo general necesita de otros refinamientos para poder resolverse en un lenguaje o sistema particular: lo más importante es destacar el funcionamiento de las condiciones que constituye la base para formar una filosofía de trabajo relacionada con la estructura condicional, que los estudiantes reconocerán posteriormente en los fundamentos de la programación, teniendo como elementos distintivos:

- Siempre se hará una acción que depende de algo (condición).
- Definen el orden en que tienen que realizarse los pasos de un determinado algoritmo.
- Seguidamente se propone refinar el algoritmo general para darle solución con Excel. El profesor debe hacer énfasis en que sólo se trabaja con el alumno que se encuentra en el primer registro, pues para el resto sería replicar la fórmula, lo cual no es objetivo del trabajo.

1. Recopilar en forma de tabla la información de los 20 alumnos.

- Si el contenido de la celda B2 es mayor que 95, imprimir el contenido de la celda que almacena el nombre.
- Si el contenido de la celda B1 es mayor que 90 y el contenido de la celda B2 es mayor que 90 y el contenido de la celda D1 es mayor que 90, entonces calcular el promedio, sino imprimir “Tiene asignaturas por debajo de 90”.
- Determinar qué funciones necesita.
- Cargar el Excel.
- Confeccionar la tabla.
- Seleccionar la celda donde obtendrá el resultado.
- Escribir la función con sus parámetros o utilizar el asistente de funciones (generalmente se emplea la segunda variante).

En los pasos 2 y 3 de este algoritmo se aplican las condiciones y se obtendrá un resultado que depende del cumplimiento de una condicional, en este caso representada a través de una función con uno o varios parámetros. Lo más importante es que los estudiantes reconozcan que se sigue funcionando bajo los mismos principios lógicos.

Transcribiendo a Excel la solución del paso dos es =Si(B2>95; A2)

	A	B	C	D
1	Nombres y Apellidos	Matemática	Español	Historia
2	Lidia Pérez Cuba	99	95	89
3	Evelyn González Almira	98	92	100
4	Annia Cedeño Silva	78	98	97
5	Tamara Rodríguez Oro	90	89	91

Como se observa en el ejemplo resuelto con Excel, se utiliza el si estructural como función, el cual se ejecuta en dependencia del cumplimiento de una condición.

Posteriormente se pregunta: ¿Cómo obtener este resultado empleando los LP? El algoritmo pudiera quedar de la forma siguiente:

Inicio

Declarar las variables u objetos para la entrada de información (NotaE, NotaM, NotaH, Nombre, Promedio).

Leer el nombre y las notas.

Si la variable NotaM es superior a 95, entonces imprimir el Nombre.

Si la variable NotaM es mayor que 90 y la variable NotaE es mayor que 90 y la variable NotaH es mayor que 90, entonces calcular e imprimir el promedio, sino imprimir “Hay al menos una asignatura con menos de 90”.
Fin

A continuación se propone valorar las interrogantes siguientes:

1. ¿Cuántos nombres se imprimirán en cada corrida de algoritmo?
2. ¿Qué elemento determina la impresión del nombre?
3. ¿Qué representa ese elemento?
4. ¿En cada corrida del algoritmo se ejecutarán los pasos 3 y 4 en el mismo orden? ¿Por qué?
5. ¿Cómo expresar en forma general el paso 3?

Durante el análisis y caracterización de la estructura alternativa se retoman los elementos valorados en los algoritmos y se arriba a las conclusiones siguientes:

- Se mantiene el principio de funcionamiento de la condición.
- Prevalece la toma de decisión.
- Define el orden en que se ejecutan los pasos de un algoritmo.
- Si la expresión lógica es verdadera, entonces ejecutar sentencia.
- Si la expresión lógica es verdadera, entonces ejecutar sentencia 1, sino ejecutar sentencia 2.

Estructuras alternativas: en estas prevalece una toma de decisión donde el camino a seguir en la solución del problema depende del cumplimiento o no de una condición.

Se propone al profesor que oriente para el estudio independiente la otra variante de la estructura condicional (Select Case Selector), para dar respuesta a las preguntas siguientes:

- ¿Cuándo es conveniente el empleo del selector de casos?
- ¿Qué ventajas ofrece su uso?
- ¿Qué tipos de datos se utilizan para operar con el selector de casos?
- ¿Es aplicable al paso 4? Demuéstrelo a través de un lenguaje de programación.

El profesor debe fundamentar que este algoritmo es aplicable en los lenguajes de programación estructurada y en la programación visual. Se recomienda mostrar fragmentos de programas en Pascal y VB para los pasos 3 y 4, y analizar cómo opera en cada caso para

que los estudiantes lleguen a conclusiones. Sólo cambian las instrucciones del lenguaje y no el funcionamiento de la estructura. Por ejemplo:

En Pascal:

```
If NotaE>95 then Writeln(Nombre)
If (NotaE>90) and (NotaM>90) and (NotaH>90) then
Begin
Promedio:=(NotaE+ NotaH+ NotaM)/3;
Writeln('El promedio es', Promedio:4:2);
        end
Else
Writeln('No se cumple');
READLN;
END
```

En VB:

```
If NotaM>95 then Text1.Text = Nombre
If NotaM>90 and NotaM>90 and NotaM>90 then
Promedio = (NotaM+ NotaH+ NotaE)/3
Text2.Text="El promedio es " & Promedio
Else
Text2.Text = "No se cumple"
End If
```

Propuesta de ejercicios

1.- Dados tres números imprima el mayor.

2.- Dada la fecha actual y la fecha de nacimiento de una persona, decir la edad que cumple.

3.- Dados dos números, si ambos son ceros, imprima un título; si solamente uno de ellos es cero, imprima el cuadrado del otro; si tienen igual signo, imprima su producto; y si uno de ellos es positivo y el otro negativo, imprima el cociente del positivo entre el negativo.

Seguidamente se retoma el ejercicio base y se hacen las preguntas siguientes:

1. ¿Cuántas veces hay que ejecutar el programa propuesto para darle solución al ejercicio?
2. ¿Qué instrucciones se repiten en cada corrida?
3. ¿Qué característica tiene este proceso?

4. ¿Será posible analizar en una corrida del programa todos los estudiantes?
5. ¿Qué modificaciones hay que hacer para obtener el resultado de los 12 estudiantes en una corrida?
6. ¿Cómo se implementa este proceso en los lenguajes de programación?

En estos momentos es oportuno que el profesor oriente la modificación del algoritmo para que incluya a todos los alumnos. Una propuesta de solución pudiera ser:

Declarar las variables u objetos para la entrada de información.
Desde 1 hasta 12 hacer:
Leer el nombre
Leer NotaE, NotaM y NotaH
Si la variable NotaM es superior a 95, entonces imprimir el nombre.
Si la variable NotaM es mayor que 90 y la variable NotaE es mayor que 90 y la variable NotaH es mayor que 90, entonces calcular e imprimir el promedio,
sino mostrar el texto “no se cumple la condición”.
Fin

A partir de las modificaciones realizadas al algoritmo, se presentan las interrogantes:

1. ¿Qué caracteriza este algoritmo?
2. ¿Cómo opera el proceso de repetición?
3. ¿Qué estructura de control responde a este algoritmo?

El profesor valorará con los estudiantes las características de este algoritmo, en el cual se conoce de antemano la cantidad de datos que se deben procesar y la presencia (implícita) de un contador que se incrementa desde el valor inicial hasta el final; además, puede ser transcripto a diferentes LP, aunque la sintaxis no sea exactamente la misma en todos.

Ahora se verá cómo se representa la estructura de control FOR para los lenguajes que operan sobre la base del Basic.

- For <Variable (de control) =<valor inicial> To <valor final> [Step valor de incremento].
- Bloque de instrucciones.
- Next [variable].

Esta clásica estructura de control es muy utilizada en los algoritmos en que se conoce la cantidad de datos que se requiere procesar para la solución de un problema determinado.

Al ejecutarse el For:

- Establece la variable de control al valor inicial.
- Comprueba si la variable es mayor que el valor final. Si lo es, sale del ciclo.
- Ejecuta las instrucciones.
- Incrementa la variable de control en 1 o en el valor especificado (STEP).
- Se repiten los pasos del 2 al 4.
- Se propone para el estudio independiente mostrar, a través de un ejemplo resuelto en VB, cómo opera la estructura For, cuando el valor inicial es mayor que el final.
- Consultar como bibliografía: Rivero (2001). Introducción a la programación Visual.
- Plantear a los estudiantes transcribir el algoritmo que se ha venido trabajando en un procedimiento en VB sin utilizar la máquina. Al concluir, el profesor mostrará el mismo programa en una programación no visual (Pascal) e invitará a los estudiantes a realizar un rastreo con un juego de datos en ambos programas.

En Pascal:

```
Program ejemplo
Var
I: Integer;
Nombre: String;
NotaE, NotaM, NotaH: Byte;
Begin
 For I= 1 to 12 do
 Begin
Promedio:=0;
Writeln('Entre el nombre');
Readln(Nombre);
Writeln('Entre nota de Matemática');
Readln(NotaM);
Writeln('Entre nota de Español');
Readln(NotaE);
Writeln(' Entre nota de Historia');
Readln(NotaH);
If NotaE>95 then Writeln(Nombre);
If (NotaE>90) and (NotaM>90) and (NotaH>90) then
```

```
Begin
Promedio:=( NotaE+ NotaH+ NotaM)/3;
Writeln(‘ El promedio es ´,Promedio:4:2);
End;
Read;
End;
End.
```

En VB:

```
Private Sub Command1_Click()
Dim I As Integer
Dim NotaM, NotaM, NotaM As Byte
Promedio = 0
For I= 1 to 12
Nombre = InputBox("Entre el Nombre")
NotaM = InputBox("Entre Nota de Matemática")
NotaE = InputBox("Entre Nota de Español")
NotaH = InputBox("Entre Nota de Historia")
If NotaM>95 then Text1.Text = Nombre
If (NotaM>90) and (NotaM>90) and (NotaM>90) then
Promedio = (NotaM+ NotaH+ NotaE)/3
Text2.Text=”El promedio es “ & Promedio
Else
Text2.Text = “No se cumple”
End If
MsgBox ("Próximo")
Text1.text = ””
Text2.text = “”
Next
End Sub
```

Durante el rastreo se debe destacar la filosofía de trabajo sobre la base de las estructuras de control y no a otros componentes del lenguaje, destacando la presencia y funcionamiento de

condiciones, tanto en la estructura alternativa, como en la repetitiva, y que su aplicación depende de las características del problema y/o ejercicio que se debe resolver.

En la exposición del artículo se han expuesto experiencias metodológicas para potenciar el desarrollo de filosofías de trabajo en el estudio de la estructura condicional, partiendo de las posibilidades que ofrecen los SA y, en particular, las HEC. Se ha enfatizado en la lógica del funcionamiento de estas, independientemente del contexto en que se apliquen, ya sea para la solución de tareas integradoras en el orden de la docencia, o para la solución de problemas en la investigación, donde se deben tener en cuenta los elementos siguientes:

- Aprovechar todas las posibilidades que brindan los SA para potenciar el desarrollo de FTI, dirigido a la estructura de control.
- Demostrar a los estudiantes el principio de funcionamiento de las condicionales.
- Exigir la elaboración de algoritmos generales y específicos en la solución de ejercicios y problemas que impliquen situaciones condicionales.
- Destacar en los procedimientos de trabajo de los SA las invariantes de las condicionales.

También el desarrollo de filosofías de trabajo favorece:

- Saber y saber hacer; es decir, la asimilación de conceptos, el conocimiento de estrategias de solución de problemas y tareas integradoras.
- Preparar al estudiante para enfrentar un aprendizaje continuo en el dinámico campo de la Informática.

BIBLIOGRAFÍA

EXPÓSITO, C. Algunos elementos de metodología de la enseñanza de la Informática. [s.l., s.e.], 2001. [Edición digital].

GOTTFRIED, B. Programación en Pascal. Ciudad de La Habana, Editorial Pueblo y Educación, 1989.

KATRIB, M. Lenguajes de programación y técnicas de compilación. Ciudad de La Habana, Editorial Pueblo y Educación, 1988.

RIVERO, A. Introducción a la programación Visual. Ciudad de La Habana, Editorial Pueblo y Educación, 2001.

VAQUERO, A. Y G. QUIROZ. Microsoft Visual Basic 6.0 Manual del Programador. Madrid, Editorial Paidós, 1998.