

Revista Facultad de Ciencias Básicas ISSN 1900-4699

homepage: http://revistas.unimilitar.edu.co/index.php/rfcb

Vol. 13 (2) 2017, 83-89

PROPUESTA METODOLÓGICA PARA CONSTRUIR UN ALGORITMO QUE DETERMINE SI UN NÚMERO ES PERFECTO USANDO PROGRAMACIÓN IMPERATIVA

A METHODOLOGICAL PROPOSAL TO BUILD AN ALGORITHM TO DETERMINE WHETHER A NUMBER IS PERFECT USING IMPERATIVE PROGRAMMING

Omar Iván TREJOS BURITICÁ¹

¹ Ingeniería de Sistemas y Computación, Facultad de Ingeniería, Universidad Tecnológica de Pereira, Colombia.

Historia del Artículo

Recibido: Junio 30, 2016 Evaluado: Diciembre 28, 2016 Aceptado: Enero 20, 2017 Disponible: Febrero 09, 2017

Resumen

El presente artículo da cuenta de una propuesta para enseñar programación de computadores a través de la solución de un problema concreto basándose en el paradigma de programación funcional relacionando un tema problema propio de las matemáticas y capitalizando la tecnología computacional a través de lenguaje de programación DrScheme para resolverlo. En esta investigación se utilizó el método de estudio y resolución de un caso determinado desde el enfoque de su formalización matemática y su implementación computacional en el primer curso de programación de computadores de un programa de Ingeniería de Sistemas. Los resultados obtenidos muestran un proceso de apropiación del conocimiento y retroalimentación del ejercicio puntual en el cual se ha podido aplicar, por los alumnos, dicho conocimiento en ejercicios similares fomentando el aprendizaje autónomo y el active learning sobre estos procesos de formación. Se concluye que resolver problemas matemáticos, desde la programación de computadores, es un área que puede explotarse mucho más toda vez que, desde lo didáctico, el docente se capacite para establecer nexos transversales entre matemáticas y programación en favor de los objetivos de aprendizaje trazados.

Palabras Clave: Algoritmo, número perfecto, matemáticas, programación de computadores, programación funcional.

Abstract |

This article presents a proposal to teach computer programming through solving a particular problem using functional programming paradigm and establishing a link between Maths and computer programming capitalizing the computer technology using DrScheme language to solve it. The method to solve a particular case was used in this research from the perspective of its mathematical formalization and computational implementation in the first semester of a computer programming course in a Engineering Systems degree. The results show a process of appropriation of knowledge and timely feedback exercise in which it has been applied, for students, such knowledge in similar exercises promoting independent learning and active learning training on these processes. It is concluded that solve mathematical problems, from computer programming, is an area that can be exploited much more if the teachers are trained to establish links between mathematics and programming for the objectives outlined learning.

Keywords: Algorithm, computer programming, functional programming, mathematics, perfect number.

INTRODUCCIÓN

En tiempos modernos, una de las más necesarias competencias que han de fortalecerse en los alumnos de un programa de ingeniería radica en poder tener la capacidad de formular problemas a partir del formalismo matemático, es decir, con el rigor que la ciencia exige (Attard et al, 2010),

de aprovechar la lógica como la base para la formulación de posibles soluciones (Brassard y Bratley, 2006) y de ser capaces de implementar soluciones a partir de un lenguaje de programación (Trejos Buriticá, 2009), siempre y cuando la solución sea implementable sobre el marco computacional. En los programas de ingeniería es normal que se planteen situaciones problema que constituyen el corpus esencial de

^{*}Autor Corresponsal. E-mail: omartrejos@utp.edu.co

su esencia y por lo tanto propiciar los espacios para que los estudiantes se motiven a encontrar soluciones dentro de las posibilidades que les ofrece su currículo y a buscar las soluciones en medio del abanico de temáticas que ofrece su formación profesional (Fríes et al. 2014).

Se convierte en un requerimiento que, desde los primeros semestres, el estudiante pueda palpar la relación existente entre las matemáticas, la lógica y la programación como aristas de una misma área que bien podría llamarse Resolución de Problemas. Para ello es necesario que los docentes de cada una de estas áreas sean conscientes de la relación tan estrecha que tienen entre ellas y, al tiempo, adopten estrategias que remarquen esa interrelación en medio de teorías de aprendizaje que refuercen los logros a alcanzar simplificando los esfuerzos dentro de un proceso formativo (Diaz Barriga, 2005).

En este artículo se presenta de manera completa todo el conjunto de acciones, estrategias y actividades que conforman una metodología aplicándola a la implementación de un problema heredado de las de una solución matemáticas apoyándose en programación imperativa y usando lenguaje C como medio para cristalizar su solución. Se muestra en este artículo una forma de aprovechar las potencialidades de los formalismos matemáticos, el planteamiento de soluciones lógicas y la implementación algorítmica que la programación de computadores permite en un esquema de interrelación absolutamente transversal. Igualmente se presentan los resultados obtenidos por parte de los estudiantes. En lo cuantitativo se discute el promedio de cada uno de los grupos en donde la metodología ha sido utilizada y en lo cualitativo se destacan las características de las opiniones de los estudiantes de forma que sirvan como base para retroalimentar la investigación.

Si bien la enseñanza de la programación de computadores ha tenido diferentes enfoques didácticos, el planteamiento de acudir a la resolución de problemas, como lo sugiere (Kaasboll, 1999), se ha convertido en una estrategia muy apropiada para que los estudiantes apropien, asimilen, apliquen, retroalimente y evalúen los conceptos que la lógica computacional exige cuando se trata no solo de usarse sino de capitalizarse como fundamento para la implementación de soluciones (Romero Chaves y Rosero Sosa, 2014). De otra parte, las matemáticas han sido siempre una buena fuente para formular problemas, plantear soluciones y resolverlos bien por la vía del planteamiento puramente matemático o por la vía de la implementación computacional dentro del marco de soluciones sistemáticas (Polya, 1989; Scott Fogler et al, 2014).

Enfrentar, pues, al estudiante a situaciones problema, a la formalización de la misma, a la formulación de una solución y a la implementación algorítmica de la misma usando computadores implica, para el mismo, que el conocimiento adquirido, evaluado y aplicado cobre sentido (por su utilidad y particularmente por su

interdisciplinariedad) y significado (por lo que representa para situaciones problemas posteriores en las cuales dicho conocimiento pueda aplicarse) (Ausubel, 1963) lo cual favorece la autonomía en el aprendizaje y el active learning que propende porque el mismo estudiante se involucre y defina sus propias metas de aprendizaje (Brown Wright, 2011).

Todo esto apunta a que el conocimiento tecnológico no se limita a la comprensión de una sintaxis o de unos requerimientos tecnológicos (Jones, 2007) sino que se necesita un fundamento conceptual que posibilite el aprovechamiento al máximo de dicha tecnología computacional (Cooper, 2004). Esto implica la necesidad de orientar al estudiante para que el conocimiento tecnológico científico no solo se aproveche en las situaciones que se plantean en el aula sino que, además, se se pueda aplicar en situaciones similares o disímiles en las cuales dicho conocimiento sea la base para resolver posteriores situaciones problema (Trejos Buriticá, Fundamentos de Programación, 2009) independiente del área de aplicación o la asignatura en la cual se utilice.

Este tipo de estrategias destacan la ingente necesidad de motivar a los estudiantes a buscar y encontrar soluciones partiendo de modelos que ellos mismos conozcan y atreviéndose, también, a proponer nuevos modelos conceptuales. El presente artículo es un producto asociado al proyecto de investigación "Desarrollo de un modelo metodológico para el aprendizaje de la programación imperativa en Ingeniería de Sistemas basado en aprendizaje significativo, aprendizaje por descubrimiento y el modelo 4Q de preferencias de pensamiento" aprobado por la Vicerrectoría de Investigaciones, Innovación y Extensión de la Universidad Tecnológica de Pereira bajo el código interno 6-16-13.

MARCO CONCEPTUAL

Se define un algoritmo como un conjunto ordenado y secuencial de pasos que permiten lograr un objetivo. El algoritmo se plantea como el mecanismo estándar aceptado por una determinada comunidad de programadores para poder alcanzar un objetivo que sea claro, concreto y cuantificable (Brassard y Bratley, 2006). El concepto de algoritmo comienza a tomar forma en la medida en que se van encontrando, en las matemáticas, procedimientos que manteniendo su estructura- permiten lograr objetivos en áreas, temas y tópicos diferentes.

Los números naturales son todos aquellos que, siendo positivos, permiten cuantificar los elementos de un conjunto y permiten también establecer un orden en su composición. Son los números que originalmente el ser humano ha utilizado para contar y se ubican al lado derecho de la recta de números (Kline, 2012). Los números naturales siempre son enteros y no tienen decimales que los acompañen. Entre los números naturales normalmente no se cuenta el número

cero debido a que si estos números se usan para contar, el número cero no cuenta nada y por tanto existe una confrontación conceptual entre la teoría de números y la matemática para definir la inclusión o no del número cero.

Se definen como números perfectos todos aquellos números en los cuales el resultado de sumar todos su divisores exactos (sin incluirlos) es igual a ellos mismos (Jiménez Murillo, 2014). Tal es el caso del número 6 cuyos divisores exactos son 1, 2 y 3. La suma de 1+2+3 es igual a 6 permitiendo concluir que este es un número perfecto. Lo mismo sucede con el número 28 cuyos divisores exactos son 1, 2, 4, 7 y 14 y cuya suma da como resultado el mismo 28.

Un programa es la expresión tecnológica computacional de un planteamiento solución que, normalmente, es heredado de las matemáticas o que se ha podido escribir aprovechando los formalismos que las matemáticas proveen (Trejos Buriticá, Fundamentos de Programación, 2009). Un programa también se puede definir como un conjunto de instrucciones que cristalizan un algoritmo y que están escritos a la luz de las reglas sintácticas y semánticas de un lenguaje de programación permitiendo que el computador logre un resultado que satisface la resolución de un problema y apoyado en la aplicación de un determinado paradigma de programación (Schildt, 2010). También puede considerarse a un lenguaje de programación como un conjunto de instrucciones que son entendibles y ejecutables por un computador y cuya combinación permite que los recursos de un computador se aprovechen para lograr alcanzar un determinado resultado (Trejos Buriticá, Algoritmos Problemas Básicos, 2008). Normalmente este resultado satisface la solución de un objetivo específico.

Un paradigma de programación es una forma de entender, concebir y aplicar determinados criterios en la construcción de una posible solución a un problema (Schildt, C++ Programming, 2010). El determinado paradigma es aceptado por una comunidad de programadores y, sobre él, se han desarrollado lenguajes de programación que posibilitan su aplicación. Normalmente un paradigma de programación tiene un serie de formalismos matemáticos que le subyacen y que le permiten que, desde lo puramente formal, se pueda expresar el problema en condiciones que sea irrebatible en su descripción así su solución no siempre sea alcanzable bien por el lado matemático o bien por el lado computacional (Van Roy, 2008).

La programación imperativa es un paradigma de programación que se basa en el concepto de estados y cuya unidad de trabajo principal es la variable que puede definirse como el espacio de memoria en donde se almacenan datos (uno a la vez) y cuya interacción se logra a partir de la ejecución de unas instrucciones de un lenguaje de programación alcanzando determinados valores que, al final, son los que resuelven determinado problema (Hartell y

Muller, 2006). La programación imperativa es la más antigua que ha existido en el mundo de la programación y ha servido como base para encontrar diversas soluciones a problemas del mundo matemático.

Como aprendizaje significativo se conoce una teoría que sostiene que el aprendizaje se basa en lo que el estudiante ya sabe y que le confiere al sentido y al significado del conocimiento toda la relevancia necesaria para que el estudiante aprenda dentro de un proceso de formación (Ausubel, Psychology of Meaningful Verbal Learning: An Introduction to School Learning, 1963). La teoría de aprendizaje significativo basa sus conceptos en el conocimiento previo, el nuevo conocimiento y la actitud del estudiante. El conocimiento previo es todo aquello que el alumno ya sabe y que ha ido adquiriendo en los procesos de aprendizajes conscientes e inconscientes que suceden en los tres contextos de su vida: el escolar, el institucional y el extraescolar. El nuevo conocimiento es todo aquello que el estudiante debe o quiere aprender y que aún no forma parte de sus estructuras cognitivas (Trejos Buriticá, Aprendizaje en Ingeniería: un problema de incomunicación, 2012). La actitud del estudiante tiene dos componentes importantes: la motivación y la capacidad de relaciones conocimiento previo con nuevo conocimiento. En ambas componentes tiene gran incidencia el papel del docente, acompañante o guía que conduzca el proceso de formación en el cual esté inmerso el aprendiz.

El aprendizaje activo es una estrategia según la cual se adoptan acciones para que el estudiante se motive a involucrarse en sus propios procesos de aprendizaje, exigiéndose a él mismo y estableciendo sus propias metas de forma que puedan ser cuantificables, calificables y comparables desde una óptica motivante y provocadora (Bruner, 1969). El estudiante bajo el active learning es consciente de lo que debe aprender, de cómo debe aprenderlo y de la manera como debe abordar las dificultades, todo ello sin desconocer la labor del docente como un acompañante en dicho proceso.

METODOLOGÍA

En primera instancia, la tabla 1 presenta la metodología utilizada en el desarrollo de la propuesta metodológica dejando claro que cuando se hace referencia a la metodología tradicional se hace hincapié en una metodología en donde la exposición magistral es la protagonista pero sin ningún tipo de estructura prevista ni tampoco ninguna estrategia que involucre al estudiante en el proceso de aprendizaje. La metodología propuesta precisamente acude a una exposición magistral acompañada de una estructura prevista y busca involucrar el estudiante de manera consciente en todo el proceso de aprendizaje. Esta investigación se realizó en la misma asignatura durante los dos semestres del año 2015.

Tabla 1. Metodología utilizada. Fuente: El autor

Fase	Descripción
1	Se comparte con los estudiantes la metodología que se va a utilizar y los objetivos de la misma. Se distribuye un instrumento para categorizarlos y poder hacer una distribución equilibrada de acuerdo a dicha categorización (AT Altamente Talentosos, MT Medianamente Talentosos, BT Bajamente Talentosos). Se organizan dos subgrupos en las mismas condiciones y casí en los mismos horarios
2	Se establece que el SubGrupo 1 recibirá la exposición magistral, la aplicación y la práctica por el método tradicional y el SubGrupo 2 recibirá la exposición magistral, la aplicación y la práctica usando la metodología sugerida en este artículo. Se inicia el proceso de exposición magistral.
3	Se plantea el problema a resolver y se posibilita para que los estudiantes puedar proponer, desde lo imperativo, soluciones posibles al problema. Se escoge una solución específica y se implementa en lenguaje C++ (arista imperativa). Se realizar pruebas y el programa se pone a punto. Una vez terminado se atiender sugerencias, observaciones y mejoras sugeridas con los estudiantes
4	Se posibilita un espacio para que los estudiantes opinen al respecto de cada una de las metodologías utilizadas. Finalmente se realiza una prueba evaluativa calificable cuyo enunciado es heredado (y tiene relación directa) con el problema resuelto

La metodología propuesta busca que el estudiante se involucre y capitalice al máximo los logros que se alcancen en cada fase. Igualmente se buscaba tener una perspectiva tanto cualitativa como cuantitativa, la primera desde la observación y la segunda desde los resultados que originen la evaluación escrita. La metodología propuesta busca rescatar la capacidad de formular los problemas a partir de formalismos que la matemática provee y resolverlos a partir de las herramientas conceptuales que proporciona la lógica y las herramientas tecnológicas que proporciona la programación de computadores. Los resultados de la categorización de los estudiantes se presentan en la tabla 2.

Tabla 2. Categorización y distribución de estudiantes. Fuente: El autor.

Sem	Grp	Categoría de Estudiantes		SubG		ib de diantes	Total Estuds		
		ΑT	MT	ВТ		AT	MT	ВТ	LStuus
I 2015	1	6	5	9	1 2	3	3	4	20
					1	3	2	5 4	
II 2015	1	5	11	7	2	2	6	3	23

SubG = SubGrupo (1 → Metodología Tradicional, 2 → Metodología propuesta)

Algunas opiniones de estudiantes y el concepto del docente autor de este artículo se presentan en la tabla 3. El enunciado a resolver era "Construir un programa en programación imperativa que reciba un número entero y determine si es un número perfecto o no". Tanto el algoritmo como su implementación en lenguaje C++ (arista imperativa) se presentan a continuación.

```
// PROGRAMA PARA DETERMINAR SI UN NÚMERO ES PERFECTO
```

#include <iostream> // Inclusión de librerías using namespace std;

// FUNCION QUE DETERMINA SI UN NÚMERO ES PERFECTO O NO

```
// Definición de función
int es perfecto (int n)
                             // Inicio de función
{
    int i, sumadivs=o;
                             // Declaración variables locales
    for(i=1; i<=n/2; ++i)
                             // Ciclo desde 1 hasta la mitad de n
    if(n\%i == 0)
                            // Si se encuentra un divisor exacto
            sumadivs=sumadivs+i; // sumarlo
    if(sumadivs==n) // si la suma de los divisores es igual a n
         return(sumadivs); // retorne el valor de la suma
```

Obietivo

Se pretende que el estudiante se involucre en el proceso que se va a adoptar y saque el máximo partido de los logros que se vayan alcanzando en cada fase. La distribución en subgrupos busca obtener unos resultados más objetivos a partir de una distribución objetivo de los estudiantes. La categorización y la distribución hace más confiables los resultados que se obtengan

Se pretende recoger resultados de la aplicación de las dos metodologías en paralelo, la aplicación del conocimiento técnico y tecnológico y la forma como los estudiantes van siendo impactados desde las dos metodologías intentando establecer un paralelo comparativo que nos permitan analizar los resultados

Poder tener un escenario desde donde se pueda observar la participación de los alumnos y, de esta forma, poder inferir el impacto posible que tengan la metodología adoptada con cada uno de los subgrupos, de forma que pueda tenerse una percepción a partir de la participación activa o pasiva de los mismos

Abrir un espacio para escuchar la opinión de los alumnos y, de esa forma, documentar la experiencia. Se realiza una prueba para intentar aproximarse al posible impacto de cada una de las metodologías y para poder hacer los análisis pertinentes basado en los datos recolectados

```
return(o);
                              // sino, retorne o (falso)
     }
                              // fin función
// Función que muestra la prueba de que un número es
PERFECTO
    int prueba (int n)
                              // Definición de función
    {
                              // Inicio de función
                              // Variable local
          int i;
          cout<<"\n\nPrueba:\n";
                                        // Título
          if(es perfecto(n)) // Si n es un número perfecto
           for(i=1; i<=n/2; ++i) // Ciclo que va de 1 a n/2
             if(n\%i == 0)
                              // Si i es un divisor exacto de n
           if (i==n/2)
                              // Si llegó a n/2
           cout<<i;// muestre el valor n/2
                              // sino
          cout<<i<" + "; // muestre el valor de i y el signo +
          cout<<" = "<<es_perfecto(n); // muestre el resultado</pre>
de la suma
                                        // Fin condicional
                                        // Fin función
// FUNCIÓN PRINCIPAL
 int main ()
                                        // Inicio función
    int num;
                              // Declaración variable local
    cout<<"Digite un entero: "; // Título
    cin>>num;
                              // Lea un número
    if(es perfecto(num))
                             // Si es un número perfecto
    cout<<"\n"<<num<<" es un numero perfecto"; // avise
          prueba(num);
                                        // y demuéstrelo
          else
                                        // sino
    cout<<"\n"<<num<<" NO es un numero perfecto"; // avise
                                        // Fin función
```

Tabla 3. Opiniones de estudiantes y concepto del docente. Fuente: El autor

	Algunas opiniones o	le Estudiantes	Opinión del Profesor			
Sem	SubGrupo 1 Metodología Tradicional	SubGrupo 2 Metodología Sugerida	SubGrupo 1 Metodología Tradicional	SubGrupo 2 Metodología Sugerida		
l 2015	 El tema es bastante claro Se entiende La explicación podría ser mejor Programar es bastante complejo El profe explica muy bien No entendí ni pio (sic) Yo ya sabía eso 	Se entiende bastante claro Programación y matemáticas son casi lo mismo Aprender algo por fases es muy bueno La metodología es bastante clara Estoy motivado a seguir programando	Los estudiantes prestan atención pero por momentos parecen dispersos, son poco participativos en clase y solo preguntan cuándo la clase ha terminado, por momentos se distraen	La actividad participativa de los alumnos es sorprendente, preguntan, proponen y se atreven, si no entienden algo lo preguntan ahí mismo, sugieren cosas nuevas y critican, proponen formas de comprobación		
II 2015	No entiendo, no pregunto No me gusta programar Me siento bien con la explicación Programar es muy complejo Cada vez me enredo más	Motivadora la metodología Bueno programar cuando se sabe de matemáticas Con las matemáticas, programar es muy fácil El profe es bueno La metodología es muy organizada El profe es muy organizado	Los estudiantes son muy callados, temen preguntar, pareciera que prestan atención pero solo preguntan al final. Por comentarios varios han sugerido estar en el otro grupo. Entre compañeros no son tan cercanos.	Mucha actividad durante la experiencia, algo de desorden pero todo producto de la participación de los alumnos, hacen observaciones y críticas no solo a la propuesta de solución del enunciado sino también a la metodología y a la forma de exponerla. Sugieren nuevas cosas.		

RESULTADOS Y DISCUSIÓN |

Cuando se ejecuta este programa con diferentes valores, los resultados que arroja se muestran en la tabla 4.

Tabla 4. Resultados obtenidos con el programa. Fuente: El autor.

Con números no perfectos	Con números perfectos
Digite un número: 6	Digite un número: 25
6 es un numero perfecto	25 NO es un número perfecto
Prueba:	
1+2+3=6	
Digite un número: 28	Digite un número: 12
28 es un numero perfecto	12 NO es un número perfecto
Prueba:	
1 + 2 + 4 + 7 + 14 = 28	
Digite un número: 496	Digite un número: 100
496 es un numero perfecto	100 NO es un número perfecto
Prueba:	·
1 + 2 + 4 + 8 + 16 + 31 + 62 + 124 = 496	

La prueba evaluativa tenía el siguiente enunciado "Construir un programa que permita determinar cuántos números impares existen en el conjunto de divisores exactos de un número perfecto en el caso de que éste sea un número perfecto". Con este enunciado se buscaba que el estudiante aprovechara lo aprendido y, a partir de allí, avanzara un poco más. Los resultados cuantitativos y cualitativos se presentan en la tabla 5.

Se ha acudido a una metodología que socializa con el estudiante los propósitos de la investigación que se va a realizar permitiendo que él se articule con los objetivos y se apropie de los beneficios de ella. Este tipo de investigaciones tienen todo el sentido si de ellas se derivan nuevos caminos o se refinan caminos existentes para que el aprendizaje sea un proceso exitoso y se llegue a él maximizando los logros y, al tiempo, minimizando los esfuerzos.

Los estudiantes se han distribuido a partir de la aplicación de un instrumento que permita balancear los subgrupos. Para ello se han tenido en cuenta dos consideraciones: se deben mantener las mismas características y factores en ambos subgrupos para que alguno de ellos, por ejemplo el cambio de temperatura en el medio ambiente, no vaya a malograr los resultados y se obtengan resultados sesgados sin querer. Por ejemplo, si un subgrupo recibiera la exposición magistral en la mañana con temperatura ambiente fría y otro subgrupo recibiera su exposición magistral en la tarde con temperatura ambiente alta, los resultados podrían ser significativamente diferentes y no por las características de la metodología sino por factores externos que, como el clima, influyen poderosamente en el desarrollo de un proceso de aprendizaje.

Lo segundo que se tomó en cuenta es que si se hiciera una distribución aleatoria en los grupos podrían arriesgarse los resultados de la investigación pues no es lo mismo que los grupos estén balanceados en cuanto a la capacidad académica de los estudiantes a que todos los estudiantes categorizados como altamente talentosos estén en un subgrupo y los estudiantes categorizados como bajamente talentosos estén en otro subgrupo. Por eso se ha acudido a un instrumento para apoyar dicha distribución, adicional a la observación directa del docente en relación con el comportamiento, rendimiento y capacidad académica de cada uno de los alumnos.

La opinión del docente es producto de la observación de cada subgrupo y esta observación es determinante para validar, en la práctica, los resultados que se obtienen y, de paso, la metodología que se ha aplicado. Esta investigación se ha realizado en la 7ª semana de clases (en un semestre de 16 semanas académicas) tiempo apropiado para que el profesor pueda tener un concepto claro de cada uno de los alumnos y pueda validar, con su observación, los resultados que se obtienen para hacer los correctivos que sean pertinentes.

	SubGrupo 1 Metodología Tradicional						SugGrupo 2 Metodología Sugerida					
Sem	Nota Mas Baja	Nota Mas Alta	Prom Del Grupo	Hicieron La Eval.	No Hicieron La Eval.	Nota Mas Baja	Nota Mas Alta	Prom Del Grupo	Hicieron La Eval,	No hicieron La Eval.		
I 2015	3.4	3.9	3.73	4/10	6/10	4.2	4.6	4.44	9/10	1/10		
II 2015	3.1	3.5	3.39	6/12	6/12	4.3	4.8	4.61	9/11	2/11		

El código del programa se ha implementado en lenguaje C++ aprovechando su arista imperativa dadas las características que tiene el paradigma imperativo de aproximarse fácilmente a una solución inmediata y simple basada en la aplicación de sus tres estructuras (secuencia de instrucciones, condicionales y ciclos). Es importante tener en cuenta que la este tipo de soluciones tienen su dificultad en la capacidad de los tipos de datos pues, en este caso, se han utilizado variables de tipo entero que almacenan sus datos en dos bytes (de los cuales solo 15 bits son para información real ya que el 16° bit se reserva para el signo). Las fronteras superiores de los números enteros representados en variables de tipo entero en lenguaje C++ implica que ningún número supere el valor 32767 con lo cual, es posible, que esta cota sea muy baja cuando se requiera verificar si un número mayor es perfecto.

Por lo pronto, la solución sería cambiar el tipo de dato int a tipo de dato long (que son enteros que se almacenan en 4 bytes) y de esa forma se amplía el rango de números a evaluar. De todas formas, a menos que se utilicen simulaciones a partir de estructuras de datos y mientras este tipo de ejercicios se fundamenten en los tipos de datos primitivos, siempre existirá una cota superior que limita el número a evaluar. Los resultados obtenidos a través del programa son suficientes no solo para verificar si un número es perfecto o no sino también para desplegar en pantalla la prueba calculando el resultado de sumar los divisores exactos menores al número dado.

Los resultados cuantitativos obtenidos con la prueba evaluativa evidencian una diferencia significativa al momento de realizarla y, según estos datos, es claro que favorecen mucho más a los estudiantes a quienes se les aplicó la metodología sugerida que aquellos a los que no. La capacidad de haber podido realizar un ejercicio completo – basado en lo visto- es otro termómetro para verificar el impacto y efectividad de la metodología utilizada que pareciera denotar una favorable diferencia en el grupo en el cual se aplicó. Es aquí en donde la categorización de estudiantes cobra su mayor importancia.

CONCLUSIONES

Los resultados invitan a concluir que es posible lograr muy buenos resultados cuando se aplican metodologías de enseñanza y aprendizaje innovadoras, en la medida en que sean socializadas con los alumnos tanto lo que se quiere lograr como la manera como se quiere avanzar en un contexto asignaturista de formación profesional. De allí que

pareciera ser útil tener en cuenta que, si se quieren categorizar a los estudiantes, se deben mantener las condiciones para cualquier subgrupo que se origine y que siempre será conveniente poder contar con instrumentos técnicos que sean confiables sin desconocer la necesidad de la observación directa por parte del profesor.

Todo parece indicar que conviene siempre dejar que los datos hablen por ellos mismos y que, si se usa la estrategia de subgrupos, cada uno deberá recibir una formación que no le afecte sus procesos. En el caso de la programación de computadores conviene que los alumnos apropien la íntima relación que existe entre las matemáticas y la programación, para capitalizar los formalismos de las matemáticas como la implementación de la programación mediados por la lógica.

Es muy valioso que estos procesos investigativos se retroalimenten con la opinión de los estudiantes propiciándoles espacios de opinión libres y sin restricciones dado que ello son la razón de ser de estas investigaciones. De otra parte, la programación imperativa es un buen recurso para resolver fácilmente problemas cuya solución sea computable dada la facilidad de poder establecer relaciones entre el planteamiento matemático y la solución algorítmica.

En consecuencia debe tenerse en cuenta que la evaluación escrita es un buen recurso para validar lo que se investiga, la metodología utilizada, los resultados obtenidos y el impacto de todo el proceso sobre el aprendizaje de los estudiantes. Debe ser especialmente cuidadoso el docente en el diseño de la prueba evaluativa para que ésta (como nuevo conocimiento) tenga una relación directa con el conocimiento adquirido.

REFERENCIAS

- Attard, A., Di Ioio, E., & Geven, K. 2010. Student Centered Learning. An insight into theory and practice. Bucarest: Lifelong learning programme -European Community. Bucarest. 47p
- Ausubel, D. 1963. Psychology of meaningful verbal learning: an introduction to school learning. New York: Grune & Straton, 272p
- Brassard, G., & Bratley, P. 2006. Fundamentos de Algoritmia. Madrid: Prentice Hall, 581p
- Brown Wright, G. 2011. Student centered learning in Higher Education. International Journal of Teaching and Learning in Higher Education, 23(3), p. 02-07.
- Bruner, J. S. 1969. Hacia un teoría de la instrucción. Ciudad de México: Hispanoamericana. 1850
- Cooper, D. 2004. Locos por la tecnología. México: Editorial Limusa, 210p
- Diaz Barriga, F. 2005. Estrategias docentes para un aprendizaje significativo. México: McGraw Hill, 476p
- Felleisen, M. e. 2006. How to design Programs. Boston: MIT Press, 720p

- Fríes, E., Monzón, G., & Di Paolo, J. Junio de 2014. Resolución de una situación problemática mediante la utilización de TIC. (A. C. ACOFI, Ed.) Revista Educación en Ingeniería, 9(17), p 45-52.
- Hartell, P., & Muller, H. 2006. Functional C. Helsinki: Peter Hartel & Henk Muller Press, 429p
- Jiménez Murillo, J. 2014. Matemáticas para la computación. Ciudad de México: Alfaomega, 551p
- Jones, L. 2007. Student Centered Learning. Cambridge: Cambridge University Press. 272p
- Kaasboll, J. 1999. Exploring didacti models for programming. Oslo: Universidad de Oslo
- Kline, M. 2012. El pensamiento matemático de la antiguedad a nuestros días. Madrid: Alianza Editorial, 1634p
- Polya, G. 1989. Cómo plantear y resolver problema. México D. F. : Editorial Trillas, 109p
- Romero Chaves, C., & Rosero Sosa, M. Junio de 2014. Modelo de Enseñanza y su relación con los procesos metacognitivos en programación de sistemas. (A. C. ACOFI, Ed.) Revista Educación en Ingeniería, Vol 9, No 17, pp 1-12

- Schildt, H. 2010. C Programming. México: McGraw Hill, 810p
- Schildt, H. 2010. C++ Programming. Vancouver: McGraw Hill, 805p
- Scott Fogler, H., Le Blanc, S., & Rizzo, B. 2014. Strategies for creative solving problem. Boston: Prentice Hall. 343p
- Trejos Buriticá, O. 2000. La Esencia de la Lógica de Programación. Pereira: Papiro, 368p
- Trejos Buriticá, O. 2008. Algoritmos Problemas Básicos. Pereira (Colombia): Papiro, 120p
- Trejos Buriticá, O. 2009. Fundamentos de Programación. Pereira: Editorial
- Trejos Buriticá, O. 2012. Aprendizaje en Ingeniería: un problema de incomunicación. Pereira: Tesis Doctoral - Doctorado en Ciencias de la Educación - RudeColombia Cade Pereira.
- Trejos Buriticá, O. 2013. Significado y Competencias. Pereira: Papiro, 145p
- Van Roy, P. 2008. Concepts, Techniques and Models of Computer Programming. Estocolmo: Université catholique de Louvain, 939p