

# Metodología y algoritmo para encontrar números perfectos en un rango determinado de números naturales usando Programación Funcional<sup>1</sup>

Methodology and algorithm to find perfect numbers in a given range of natural numbers using Functional Programming

Metodologia e algoritmo para encontrar números perfeitos numa determinada faixa de números naturais usando Programação Funcional.

O.I. Trejos

Recibido: junio 27 de 2016 - Aceptado: enero 13 de 2017

**Resumen**— Este artículo pretende dar cuenta de un modelo de enseñanza de la programación de computadores aplicado a la solución de un problema concreto a partir del paradigma de programación funcional estableciendo una relación tangible, para el estudiante, en lo académico entre los temas vistos en matemáticas y la posibilidad de capitalizar la tecnología computacional para resolverlo. En esta investigación se ha utilizado el método de estudio y solución de un caso específico desde la perspectiva de su implementación lógica matemática desarrollada en el primer curso de programación de computadores de un programa de Ingeniería de Sistemas. Los resultados muestran una asimilación, apropiación, aplicación y retroalimentación del ejercicio puntual realizado de forma que se ha podido aplicar, por parte de los estudiantes, en ejercicios similares desde la óptica del aprendizaje autónomo y la participación activa en estos procesos de formación. Se concluye que la solución de problemas matemáticos a partir de la programación es un área que se pudiera explotar mucho más, desde lo didáctico, en bien del logro de los objetivos de aprendizaje de los estudiantes y que ésta estrategia posibilitaría el autoaprendizaje y el *active learning*.

**Palabras clave**— algoritmo, número perfecto, número

natural, programación de computadores, programación funcional.

**Abstract**— This article explains a model of teaching computer programming applied to the solution of a specific problem from functional programming paradigm establishing a relationship to the student, academically among the topics covered in mathematics and possibility of capitalizing computer technology to solve it. This research used the method of study and solution of a specific case from the perspective of mathematical logic implementation developed in the first course of computer programming in Systems Engineering. The results show an assimilation, appropriation, application and feedback conducted so that it has been applied, by students in similar exercises from the perspective of autonomous learning and active participation. The conclusion is that the solution of mathematical problems from programming is an area that could be exploited much more, from the teaching, to reach learning goals of student and that this strategy enable self-learning and active learning.

**Key words**— algorithm, perfect number, natural number, computer programming, functional programming.

**Resumo**: Este artigo pretende mostrar um modelo de ensino da programação de computadores aplicado à solução de um problema concreto a partir do paradigma de programação funcional estabelecendo uma relação tangível, para o estudante, com relação ao acadêmico entre os temas vistos em matemática e a possibilidade de capitalizar a tecnologia computacional para resolve-lo. Nesta pesquisa foi utilizado o método de estudo e solução de um caso específico desde a perspectiva de sua implementação lógica matemática desenvolvida no primeiro curso de programação de computadores de um programa

<sup>1</sup>Producto derivado del proyecto de investigación “Desarrollo de un modelo metodológico para el aprendizaje de la programación 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 bajo el código 6-16-13”. Presentado por el Grupo de Investigación en Informática, de la Universidad Tecnológica de Pereira.

O.I. Trejos, Docente de Planta, Programa Ingeniería de Sistemas y Computación, Facultad de Ingenierías, Pereira (Risaralda); email: omartrejos@utp.edu.co .

de Engenharia de Sistemas. Os resultados mostram uma assimilação, apropriação, aplicação e retroalimentação do exercício pontual realizado de forma possível de aplicar, por parte dos estudantes, em exercícios similares desde a visão da aprendizagem autônoma e a participação ativa nestes processos de formação. Se conclui que a solução de problemas matemáticos a partir da programação é uma área que pode ser muito mais explorada, em contexto didático, com a finalidade de objetivos de aprendizagem dos estudantes e que esta estratégia possibilitaria a autoaprendizagem e o active learnig.

**Palavras chave**— algoritmo, número perfeito, número natural, programação de computadores, programação funcional.

## I. INTRODUCCIÓN

LA labor cognitiva que implica la participación de los estudiantes en la resolución de problemas es profundamente significativa debido a que es la búsqueda de soluciones la que ha de permitir que se puedan articular profesionalmente en la sociedad que los requiere [01]. Resulta ser de gran motivación para los estudiantes que optan por estudiar una carrera de Ingeniería el hecho de tener que enfrentarse a enunciados que se constituyen en problemas a resolver [02] y esa motivación aumenta cuando se presentan tres factores adicionales: a) encuentran que dichos enunciados tienen relación directa con -o son heredados de- otras áreas de conocimiento, b) cuando encuentran que esas áreas de conocimiento están íntimamente ligadas a su formación disciplinar y c) cuando las soluciones las pueden encontrar a través del punto en donde se encuentran esas dos o más áreas de conocimiento de donde se derivan los enunciados en cuestión.

Esta debe ser una situación que con gran frecuencia el estudiante de programación en un curso de ingeniería de sistemas debería encontrar, puesto que la relación entre los conocimientos, herramientas y recursos que provee la programación y las otras áreas de conocimiento, sean disciplinares o no, le confiere significado y sentido a su formación profesional [03] aproximándolo tanto a la aplicación del conocimiento tecnológico (como posible camino de solución a problemas definidos) como a la notación y descripción formal de los problemas que ha de resolver haciendo uso de los recursos que la matemática provee.

Teniendo en cuenta esto se ha planteado como estrategia de aprendizaje, haciendo uso de la teoría del aprendizaje significativo, teoría de aprendizaje por descubrimiento y active learning, la resolución de problemas heredados de las matemáticas y en las cuales la programación de computadores a través del paradigma funcional proporciona caminos lógicos y alcanzables [04] por los estudiantes de primer semestre como mecanismo para resolverlos.

En este artículo se expone la metodología usada de acuerdo a los principios establecidos por las teorías de aprendizaje enunciadas en un problema específico, el camino lógico algorítmico al cual se ha acudido para resolverlo acudiendo al paradigma de programación funcional, el impacto en los estudiantes y los resultados de realizar una evaluación en consonancia con lo planteado. Todo el

proceso se ha realizado con estudiantes de programación de 1º semestre de Ingeniería de Sistemas y Computación de la Universidad Tecnológica de Pereira.

Si bien es claro que existen diversos enfoques para la enseñanza de la programación de computadores, uno de los modelos didácticos que más sintonía encuentra con los objetivos de aprendizaje corresponde a la resolución de problemas [05] que puede ser considerado más un modelo de aprendizaje que una estrategia de enseñanza [06]. De una forma muy concreta, el presente artículo presenta una situación de aprendizaje que se basa en la resolución de un problema -que no está rigurosamente definido como lo hace la metodología de aprendizaje basado en problemas- pero si toma los elementos de interdisciplinariedad y transdisciplinariedad que posibilitan la solución requerida. No ha de desconocerse de todas formas que la programación de computadores puede considerarse la expresión tecnológica de la notación matemática de forma que ésta provee enunciados en donde aquella provee posibles soluciones [07].

Resulta interesante saber que esta tarea es más motivante cuando, según lo escrito por los mismos estudiantes, se realiza en un curso de 1º semestre de ingeniería de sistemas puesto que los estudiantes desde sus primeros semestres a) comienzan a ver la íntima relación entre sus diferentes áreas de formación, b) comienzan a buscar en la programación solución a problemas de otras áreas y c) comienzan a aplicar los conocimientos disciplinares de su futura profesión dentro de las necesidades que, de manera instantánea, se le presentan.

En lo práctico el propósito fundamental de esta investigación gira en torno a la búsqueda de caminos que, desde las primeras instancias de formación profesional, le permitan al estudiante concebir la aplicación práctica que tiene la ingeniería de sistemas -desde la óptica de la programación de computadores, haciendo uso de un paradigma de programación específico [08] y en relación con un enunciado concreto heredado de las matemáticas- y evidenciar la relación íntima entre las diferentes situaciones problemáticas que pueden ser computables y las posibilidades que abren las tecnologías computacionales modernas.

A todo esto debe adicionarse el hecho de que, en tiempos modernos, y mucho más, en un programa de ingeniería de sistemas y computación, la integración de las TIC en el aula dependerá de la capacidad de los maestros para estructurar el ambiente de aprendizaje de manera no tradicional, fusionar las TIC con nuevas tecnologías y fomentar clases dinámicas en el plano social, estimulando la interacción cooperativa, el aprendizaje colaborativo y el trabajo en grupo lo cual exige adquirir un conjunto diferente de competencias para manejar la clase [09].

## II. MARCO TEÓRICO

En primera instancia se debe aclarar que un algoritmo se define como un conjunto de reglas para efectuar algún cálculo bien sea a mano o, más frecuentemente, con una máquina [10] que establece una relación muy íntima entre lo que significa un programa como expresión tecnológica

de las matemáticas y como aplicación sistematizada de la algoritmia. En el área de las matemáticas, en la lógica y en las ciencias de la computación un algoritmo se define como un conjunto establecido de instrucciones o reglas claramente definidas, que tienen un orden determinado y que son finitas de manera que, en conjunto, permiten lograr un determinado objetivo a través de la formulación y ejecución de pasos sucesivos llevando la lógica de solución desde un estado inicial y unas entradas definidas hasta un estado final y unos resultados esperados [11].

La algoritmia es la ciencia que estudia los algoritmos, sus características, sus funcionalidades y la manera como éstos pueden buscar caminos óptimos para resolver determinados problemas bien sean éstos computables (que pueden resolverse con la ayuda de un computador) o no computables (que solo se pueden resolver de manera formal y para los cuales la tecnología computacional todavía no ha proveído los mecanismos para resolverlos).

Un número perfecto se define como un número natural que igual al resultado de sumar sus propios divisores siendo éstos positivos [12]. En otras palabras podría decirse que un número perfecto es lo que se llamaría un número amigo de sí mismo. Se define un número  $a$  amigo de otro número  $b$  cuando los números divisores del número  $a$  sumados dan como resultado el número  $b$ . Dos números son mutuamente amigos cuando sus divisores mutuamente dan como resultado cada uno de los otros números. El número 6, por ejemplo, es un número perfecto pues sus divisores (que son 1, 2 y 3) sumados dan como resultado 6. Lo mismo se puede decir del número 28 cuyos divisores (1, 2, 4, 7 y 14) sumados dan como resultado el mismo número 28.

Hasta el momento aún no se ha podido demostrar la existencia de infinitos números perfectos por el camino de los formalismos matemáticos así como tampoco se ha demostrado la imposibilidad de que exista un número perfecto impar aunque de los 49 números perfectos que se han encontrado hasta Enero de 2016 ninguno es impar, pero dado que ninguna de las dos demostraciones se ha logrado solo se toman como ejemplos mas no como demostraciones formales. Los valores encontrados hasta ahora como números perfectos corresponden al rango  $(1, \infty)$ . Un rango se define como el conjunto de valores que puede llegar a tomar una función, sea ésta continua o discreta. También puede definirse como el conjunto de valores enteros secuenciales que, teniendo un inicio y un final y estando inscritos en la recta de los números naturales parte positiva, satisfacen o posibilitan la caracterización de determinados procesos matemáticos [13].

Por su parte, en matemáticas se define un número natural como cualquiera de los números que se utilizan para realizar procesos de conteo en relación con un conjunto, sea éste finito o infinito, y que posibilitan operaciones de cálculo elementales y complejas [12]. Los números naturales heredan su nombre de los números que el ser humano, en sus primeras expresiones matemáticas, utilizó para contar y que posteriormente se caracterizaron como un conjunto infinito dado que para cualquier número natural  $k$  siempre existirá un número natural  $k+1$  mayor que él. Es de anotar que el

número cero no es considerado un número natural dado que estos números eran utilizados para describir el mundo real y, de alguna u otra forma, la humanidad tardó demasiado tiempo en aceptar el cero como descripción de la nada y, además, como descripción de un mundo inexistente.

La programación funcional se define como un paradigma heredado de la programación declarativa cuyo fundamento es la definición y uso de funciones matemáticas a diferencia (no necesariamente antagónica) de la programación imperativa. En la programación funcional las funciones se destacan como núcleos fundamentales de trabajo que permiten obtener resultados en determinados procesos, operaciones y estructuras [14]. En la programación imperativa su base es el concepto de estados ligado a la utilización de variables que son las que guardan los datos y cuya interacción se logra usando instrucciones de lenguajes de programación que alteran los contenidos de las variables, interactúan con ellos y los operan de forma que puedan obtenerse resultados esperados [08].

Debe tenerse en cuenta que el uso de variables (y de “estados”) implica efectos adicionales en el desarrollo de un algoritmo y su implementación en un lenguaje de programación [15]. En la programación funcional, éstas se alimentan de argumentos que hacen que función funcione – válida esta redundancia. Eliminar los efectos secundarios que involucra la programación imperativa permite comprender y hasta llegar a predecir el comportamiento de un programa específico.

Los lenguajes de programación funcionales han tenido un gran uso académico ya que permiten apropiarse el concepto de función, sus características, fomentar la expresión formal y establecer un enlace muy cercano entre las matemáticas y la programación de computadores [04]. La programación funcional también ha tenido aplicaciones en la estadística, las matemáticas y el análisis financiero. Es posible apropiarse los conceptos de la programación funcional (desarrollo de una solución algorítmica apoyado en funciones, diseño de funciones basadas en argumento, capitalización de las funciones –llamados, recursividad, operatividad). Incluso, a partir de los métodos que se involucran en la Object Oriented Programming son útiles en este paradigma [16].

### III. METODOLOGÍA Y RESULTADOS

La metodología utilizada se dividió en varias fases a lo largo de dos sesiones (cada una de 2 horas). La distribución del tiempo se presenta en la Tabla I.

TABLA I  
FASES DE LA METODOLOGÍA - FUENTE: EL AUTOR

Fase	Descripción	Objetivo
1	Se presenta a los estudiantes la metodología que se va a utilizar en el ejercicio, se presentan los objetivos de la investigación	Involucrar a los estudiantes de manera activa en el proceso de aprendizaje, revisión y retroalimentación de la investigación
2	Se explica matemáticamente lo que es un número perfecto	Permitir que los estudiantes conozcan una descripción formal de un problema desde el punto de vista matemático
3	Se plantea el enunciado del problema “Desarrollar un algoritmo funcional que permita implementar el hallazgo de números perfectos en un rango determinado”	Permitir establecer una relación entre los problemas formales que se pueden describir desde las matemáticas y las posibilidades de solución que se pueden implementar desde la programación
4	Se abre un espacio para que los estudiantes, en un ejercicio de lluvia de ideas, planteen posibles soluciones al problema	Permitir que los estudiantes comiencen a buscar posibles soluciones al problema planteado
5	Se organiza la solución escogida basándose en el paradigma funcional y planteándola a partir de funciones	Facilitar a los estudiantes la puesta en práctica del paradigma funcional y el diseño algorítmico a partir de funciones
6	Se comienza a implementar la solución a partir de la práctica Constructivismo In Situ utilizando los recursos tecnológicos y mediáticos necesarios	Presentar a los estudiantes una implementación real a partir de sus sugerencias e ideas debidamente organizadas
7	Se revisan los errores lógicos y sintácticos, se pone a punto el programa y se ejecuta obteniendo los resultados	Permitir que los estudiantes verifiquen que los resultados obtenidos efectivamente satisfacen el objetivo inicial planteado
8	Se documenta el programa	Facilitar a los estudiantes la aplicación de los fundamentos de documentación como un mecanismo de describir lógicamente el programa
9	Se realiza una prueba de evaluación basada en un enunciado que se fundamenta en el concepto de número perfecto	Permitir que los estudiantes validen lo aprendido con el enunciado “Desarrollar un programa que permita detectar números amigos para dos números naturales dados”. Un número $a$ es amigo de otro número $b$ si la suma de los divisores exactos de $a$ es igual a $b$
10	Se abre un espacio para aportes, críticas y mejoras al programa	Permitir que los estudiantes manifiesten sus inquietudes y dudas al respecto de lo realizado
11	Se documenta la experiencia con las opiniones de los mismos estudiantes	Facilitar que se manifiesten libremente en relación con el ejercicio realizado

Luego de realizadas las fases de la 1 a la 5, se concertó

el algoritmo, el código y los resultados que se muestran a continuación. Es de anotar que el algoritmo se muestra después de los signos punto y coma (;), es decir, en la parte donde se documenta el programa. El código se encuentra antes de dichos signos de puntuación.

*; CÓDIGO Y ALGORITMO*

*; Función que determina si un valor es múltiplo de otro*  
*(define (multiplo a b) ; Definición de la función*  
*(if (= (remainder a b) 0) ; Si a es múltiplo de b*  
*b ; Entonces retorne el valor b*  
*0 ; Si no, retorne 0*  
*)) ; Fin función*

*; Función que calcula la suma de los divisores de un valor n*

*; (menores a n)*

*(define (sumadiv m n) ; Definición de la función*

*(if (> m (quotient n 2)) ; Si se superó la mitad*  
*0 ; retorne 0*

*(+ (multiplo n m) ; Si no sume retorno función*

*multiplo*  
*(sumadiv (+ m 1) n)); con retorno de llamado*

*recursivo*  
*)) ; Fin de la función*

*; Función que muestra los divisores de un número n*

*(define (muestradiv m n) ; Definición de la función*

*(if (> m (quotient n 2)) ; Si se superó la mitad*  
*del número n*

*(display “”) ; Muestre un espacio en blanco*

*(begin ; Si no*

*(if (> (multiplo n m) 0) ; Si el valor n es múltiplo de m*

*(begin ; Inicio*

*(display “ “) ; Muestre espacio en blanco*

*(display m) ; Muestre el divisor exacto*

*)) ; Fin condicional*

*(muestradiv (+ m 1) n) ; Llamado recursivo*

*))) ; Fin función*

*; Función que encuentra los números perfectos en un rango*

*; específico*

*(define (rango li ls) ; Definición de la función*

*(if (> li ls) ; Si se recorrió todo el rango*

*(begin ; Muestre título “F i n ...”*

*(newline)*

*(display “F i n ...”)*

*)*

*(begin ; Si no*

*(if (= (sumadiv 1 li) li) ; Si se encontró número perfecto*

*(begin*

*(newline)*

*(display “Número Perfecto -->”)*

*(display li) ; Mostrarlo en pantalla*

*(display “ ... Divisores --> (“*

*(muestradiv 1 li) ; Mostrar sus divisores*

*(display “)”)*

))  
 (rango (+ li 1) ls) ; Evaluar el sigte núm del  
 rango )  
 )) ; Fin de la función

Los resultados obtenidos se muestran a continuación. Para ello se han establecido un llamado para encontrar los números perfectos comprendidos en el rango de 1 a 1000.

> (rango 1 1000)

Número Perfecto -->6 ... Divisores --> ( 1 2 3 )

Número Perfecto -->28 ... Divisores --> ( 1 2 4 7 14 )

Número Perfecto -->496 ... Divisores --> ( 1 2 4 8 16 31 62 124 248 )

Fin ...

Como se puede observar, el programa ha encontrado tres números perfectos en el rango de 1 a 1000 que corresponden a los números 6, 28 y 496 y que, luego de confirmados, efectivamente verifican que son números perfectos en esos rangos. Si se buscan los números perfectos en el rango de 300 a 800, se encuentra solo un número tal como se presenta a continuación.

> (rango 300 800)

Número Perfecto -->496 ... Divisores --> ( 1 2 4 8 16 31 62 124 248 )

Fin ...

Finalmente si se buscan números perfectos en el rango de 1000 a 2000 no se encuentran y así lo reflejan los resultados del programa.

> (rango 1000 2000)

Fin ...

Algunas de las opiniones de los estudiantes, al finalizar la última fase de la experiencia, se muestran en la tabla II.

TABLA II  
OPINIONES DE ESTUDIANTES

Estud	Opinión
1	Me ha parecido muy bueno que le expliquen a uno los procesos metodológicos a través de los cuales quieren que uno aprenda un determinado concepto
2	Me llama mucho la atención poder encontrar claramente un punto en donde se encuentran las matemáticas y la programación ya que es la primera vez que lo veo pues las materias van cada una por su lado (sic)
3	Me gusta mucho la metodología que se utilizó en este tema porque pude ver todo el proceso completo desde el principio hasta el final. Todo el tiempo estuve muy atenta y me pareció maravillosa la metodología
4	Creo que todos los temas de las asignaturas de programación deberían ser presentados como se hizo en este tema específico ya que, desde el principio, uno se da cuenta para donde va y qué se quiere lograr
5	Por primera vez una clase capturó toda mi atención y por primera vez, gracias a la forma metodológica que adoptó el profesor, pude entender todo

La tabla III resume los resultados cualitativos de las opiniones y la valoración cualitativa de un ejercicio que se realizó al finalizar las dos sesiones previstas para el

tema y que se basaba en el concepto del número perfecto (desarrollo de un programa para hallar números amigos) que corresponden al tema de otro artículo.

TABLA III  
RESULTADOS CUALITATIVOS Y CUANTITATIVOS

Periodo	Result Cualitat		Result Cuantitat		Total Estud
	Op Fav	Op Desfav	Realizaron el Ejercicio	No realizaron el ejercicio	
I Sem 2015	19	1	18	2	20
II Sem 2015	20	2	21	1	22

#### IV. DISCUSIÓN

En referencia a la metodología utilizada, se buscan varios objetivos con la ejecución de las 10 fases que se plantean. En primer lugar se pretende que los estudiantes conozcan lo que se quiere investigar para que participen activamente y se articulen con los objetivos a lograr y aprovechen cada fase de la metodología. De la misma manera se quiere que los estudiantes pongan en práctica una metodología que permita solucionar un problema pasando del planteamiento del mismo a través del formalismo matemático y llegando a la implementación de la solución aprovechando los recursos que la programación de computadores provee. Se busca también que los estudiantes tengan un espacio –como se indica en la fase 4 de la tabla 1- en el cual puedan proponer todo lo que se les venga a la cabeza como parte de esa lluvia de ideas entre las cuales pueden estar posibles soluciones al problema planteado.

De igual forma con la metodología se pretende que los estudiantes puedan plantear soluciones a partir del paradigma funcional y que dichas soluciones se basen en el planteamiento formal de un problema, es decir, que puedan aprovechar matemáticas y programación en la resolución del problema. En este mismo sentido se busca que los estudiantes implementen la solución aprovechando las facilidades que provee un lenguaje de programación como DrScheme que es un lenguaje mucho más orientado al paradigma funcional que a cualquier otro paradigma. Para ello se aprovecha la estrategia de constructivismo in situ para que el estudiante vea directamente la forma como se implementa la solución a partir de su presentación formal.

Se pretendió también con la metodología, que los estudiantes conocieran, apropiaran y practicaran conceptos como la detección de errores a partir de la revisión del diseño funcional de un programa y la documentación (y autodocumentación) del código fuente. En la fase 8 se buscaba tener una aproximación a lo que realmente aprendieron los estudiantes a partir de una prueba que les permitiera aprovechar lo que se había visto (y posiblemente aprendido) en las tres sesiones y avanzar un poco más, como es el caso de construir un algoritmo que permita encontrar números amigos a partir del concepto de número perfecto que

se implementó en esta investigación. Finalmente se buscaba, en las últimas fases, abrir espacios de opinión no solo para conocer de viva voz el impacto de la metodología utilizada en la construcción de la solución al problema planteado sino también para que los mismos estudiantes manifestaran aportes, críticas, observaciones e, incluso, correcciones que consideraran pertinentes y que enriquecieran la experiencia investigativa en la cual ellos estaban inmersos de manera activa.

En relación con el código pueden destacarse tres factores que constituyen el corpus lógico de la solución implementada: en primer lugar es de destacar el diseño a partir de la construcción e interacción entre funciones manteniendo las características de nutrirse de argumentos que permitan el correcto funcionamiento de las funciones –valga esta redundancia- y además aprovechando los valores retornados sin almacenamiento adicional que posibilita una lógica de ejecución más simple y efectiva. En segundo lugar cabe destacar el aprovechamiento del concepto de recursión en la construcción de las funciones centrales tanto para recorrer el rango definido como para determinar si un número es perfecto, así como para sumar los divisores del número y desplegarlos en pantalla. Finalmente se destaca la normativa técnica de mantener una documentación simple, sencilla y entendible apoyada en una autodocumentación en relación con el nombre de las funciones y los argumentos a la luz de los estándares matemáticos.

Las opiniones, de los estudiantes, presentadas en la tabla 2 dejan entrever un panorama favorable a la luz de la visión que tiene cada alumno del proceso investigativo, de la metodología utilizada y del trabajo en el aula. Se seleccionaron solo 5 de ellas pero es de anotar que esta metodología se aplicó en los dos semestres del año 2015 en asignaturas equivalentes en contenido, semestre y nombre. Se nota que impacta mucho en los estudiantes el hecho de conocer previamente las estrategias que se van a utilizar, la metodología en la cual se enmarcan dichas estrategias, los temas a revisar y todos los demás elementos que forman parte de unidades completas de estudio. La tabla 3 muestra resultados tanto cualitativos como cuantitativos y éstos parecieran ser contundentes toda vez que una abrumadora mayoría de los estudiantes tuvieron opiniones favorables en lo cualitativo y pudieron realizar los ejercicios propuestos en lo cuantitativo.

Es de suponer que los resultados de esta investigación invitan a realizar experiencias similares con otros temas de la misma asignatura y con otras asignaturas de manera que se pueda lograr darle significado y sentido en el aula al conocimiento que se quiere impartir a los estudiantes.

## V. CONCLUSIONES

Hacer partícipes a los estudiantes de una metodología determinada cuando se va a iniciar un proceso investigativo permite que ellos, de manera activa, participen de tal forma que apropien como suyos los objetivos a alcanzar y los logros de aprendizaje que se quieren cubrir. De la misma manera es muy importante dotar a los estudiantes de la capacidad de describir formalmente un problema tal como lo provee la matemática y, de la misma forma, desarrollar

de manera sistemática una solución algorítmica que, en este caso apoyado en programación funcional, facilite la implementación en un lenguaje –como DrScheme- que permita la codificación de la idea solución.

Formular enunciados que permitan al estudiante avanzar, por su propia cuenta, un poco más allá de lo que el docente les ha entregado de manera que puedan conferirle sentido y significado a los temas vistos. En cuestión de resolver problemas cuya solución puede ser implementada en un computador, es de anotar que siempre será conveniente abrir espacios donde los estudiantes dejen volar la imaginación en la búsqueda de las soluciones pues a partir de este tipo de dinámicas se llegan a encontrar, incluso, muchas veces las soluciones óptimas. Ahora bien, en el desarrollo de un curso de programación funcional vale la pena articular muy bien las posibilidades de los formalismos matemáticos con las opciones que proporciona un lenguaje de programación como DrScheme aprovechando elementos como el diseño de funciones, la recursividad y los valores de retorno de las funciones como base para la construcción de soluciones funcionales.

Se hace muy útil en estos procesos investigativos que los estudiantes puedan verificar que la implementación de los algoritmos y su posterior codificación en un lenguaje de programación se ajustan a los planteamientos teóricos y formales que se hicieron desde el momento en que se escogió determinada solución para un problema definido. La evaluación del conocimiento, como recurso de aprendizaje, siempre será una opción muy valiosa que el docente debe tener muy en cuenta toda vez que la diseñe tan objetivamente que el mismo estudiante reconozca la relación entre lo expuesto, lo aplicado, lo implementado, lo codificado y lo evaluado. La evaluación como mecanismo de validación del impacto de estas metodologías siempre será necesaria. Finalmente abrir espacios de opinión a los estudiantes en relación con este tipo de investigaciones posibilita la retroalimentación por parte de los protagonistas de estos procesos. Conviene recordar que este tipo de investigaciones hacen su gran aporte toda vez que se publiquen y que se encuentren completamente documentados para posteriores investigaciones.

## REFERENCIAS

- [1] Fries, E., Monzón, G., & Di Paolo, J. Resolución de una situación problemática mediante la utilización de TIC. (A. C. ACOFI, Ed.) *Revista Educación en Ingeniería*, 9(17), pp. 45-52. Junio 2014.
- [2] Attard, A., Di Ioio, E., & Geven, K.. *Student Centered Learning. An insight into theory and practice*. Bucarest: Lifelong learning programme - European Community. 2010, p. 52.
- [3] Trejos Buritica, O. *Significado y Competencias*. Pereira: Papiro. 2013, p. 39.
- [4] Van Roy, P. *Concepts, Techniques and Models of Computer Programming*. Estocolmo: Université catholique de Louvain. 2008, p. 112.
- [5] Kaasboll, J. *Exploring didactic models for programming*. Oslo: Universidad de Oslo. 1999, p. 12.
- [6] Romero Chaves, C., & Rosero Sosa, M. Modelo de Enseñanza y su relación con los procesos metacognitivos en programación de sistemas. (A.C.ACOFI, Ed.) *Revista Educación en Ingeniería*, 3. Junio 2014.
- [7] Schildt, H. *C++ Programming*. Vancouver: McGraw Hill. 2010, p. 354.

- [8] Deitel & Deitel. *C++ Programming*. New York: Prentice Hall. 2013, p. 220.
- [9] Unesco. <http://portal.unesco.org>. Recuperado 15 jun 2016, de [www.eduteka.org](http://www.eduteka.org): <file:///C:/Users/EQUIPO/Desktop/Unesco%20Estandares%20TIC%20Docentes%2028p.pdf>. 8 de enero de 2008.
- [10] Brassard, G., & Bratley, P. *Fundamentos de Algoritmia*. Madrid: Prentice Hall. 2006, p. 97.
- [11] Skiena, S. *The Algorith design manual* (2a Edición ed.). New York: Springer. 2008, p. 45.
- [12] Boyer, C. *Historia de la Matemática*. Madrid (España): Alianza Editorial. 2010, p. 611.
- [13] Rey Pastor, J., & Babini, J. *Historia de la Matemática*. Barcelona (España): Editorial Gedisa. 2005, p. 212.
- [14] Felleisen, M., Findler, R., Flat, M., & Krishnamurthi, S. *How to design programas* (2a Ed. ed.). Boston: MIT Press. 2013, p. 39.
- [15] Schildt, H. *C Programming*. México: McGraw Hill. 2010, p. 552.
- [16] Van Roy, P. (2008). *Techniques and methods in programming computer*. Louvaine: University Press. 2008, pp. 342 – 345.



**Omar I. Trejos B.** nació en Buenaventura, (Valle del Cauca) el 16 de Septiembre de 1965. Ha recorrido todos los grados de formación universitaria desde la Ingeniería de Sistemas, la Especialización en Instrumentación Física, MsC en comunicación Educativa y PhD en Ciencias de la Educación. Ha publicado más de 14 libros y ha escrito en la mayoría de revistas de divulgación científica y tecnológica colombiana.

Durante más de veinte años ha sido docente de programación de computadores lo cual le ha permitido, junto con sus estudios, refinar técnicas, metodologías, estrategias y actividades, todas ellas tendientes a simplificar la comprensión de la lógica de programación para resolver problemas computables, especialmente los que proveen las matemáticas. Actualmente es profesor de planta de la Universidad Tecnológica de Pereira. Su correo electrónico es [omartrejos@utp.edu.co](mailto:omartrejos@utp.edu.co)