

# ¿Cómo perfeccionar el proceso de enseñanza y aprendizaje de la programación en carreras de ingeniería?

## How to improve the teaching and learning of programming in engineering careers?

Juan C. Fonden Calzadilla<sup>1</sup>, Raúl Rodríguez Lamas<sup>2</sup>, Rolando Serra Toledo<sup>3</sup>, María C. Mayeta García<sup>4</sup>

<sup>1,4</sup>Universidad Tecnológica de La Habana José Antonio Echeverría, Cujae, La Habana, Cuba

Correo electrónico: [fonden@ind.cujae.edu.cu](mailto:fonden@ind.cujae.edu.cu)

Este documento posee una licencia Creative Commons Reconocimiento/No Comercial 4.0 Internacional 

Recibido: 13 de septiembre de 2017 Aprobado: 22 de noviembre de 2017

### Resumen

El objetivo del presente trabajo está relacionado con el perfeccionamiento del proceso de enseñanza-aprendizaje (PEA) de la asignatura de Programación en las carreras de Ingeniería Industrial e Informática que se desarrollan en la Universidad Tecnológica de La Habana José Antonio Echeverría. Los principales resultados se enmarcan en la sistematización de un grupo de reflexiones que enriquecen la didáctica del PEA. La elaboración de sistemas de acciones para la adquisición de conocimientos, desarrollo de habilidades y formación de valores en los estudiantes de Ingeniería Industrial e Informática, a través del proceso de enseñanza-aprendizaje de la Programación constituye en sí misma una labor compleja y necesaria, encaminada a la formación profesional del futuro ingeniero. Estas acciones requieren de la praxis, que garantice los niveles de asimilación del estudiante por lo que deben transitar desde la familiarización con el contenido, la reproducción, aplicación y creación. Además, envuelven contenidos relacionados con el lenguaje algorítmico, elementos del lenguaje de Programación, dominio de la interfaz del lenguaje de programación, solución de problemas mediante el lenguaje de programación y la realización del proyecto por parte de los estudiantes.

Palabras claves: programación, lenguajes, algoritmos, tareas, enseñanza

### Abstract

The objective of this work is related to the improvement of the teaching - learning process (PEA) of the subject of Programming in the Industrial Engineering and Computer Science courses that take place in the Technological University of the Havana José Antonio Echeverría. The main results are framed in the systematization of a group of reflections that enrich the teaching of the PEA. The elaboration of systems of actions for the acquisition of knowledge, development of skills and formation of values in the students of Industrial Engineering and Computer Science, through the teaching-learning process of the Programming constitutes in itself a complex and necessary task, directed to the professional training of the future engineer. These actions require praxis to guarantee

the levels of assimilation of the student so they must move from the familiarization with the content, reproduction, application and creation. In addition, they involve contents related to the algorithmic language, elements of the programming language, domain of the programming language interface, problem solving through the programming language and the realization of the project by the students.

Key words: programming languages, algorithms, tasks, teaching

## **INTRODUCCIÓN**

El contexto socioeconómico actual sustentado en las Tecnologías de la Información y las Comunicaciones (TIC), presentes en todos los ámbitos de la vida, modifica los entornos educativos existentes y los procesos de enseñanza– aprendizajes actuales, lo que exige de un creciente perfeccionamiento de la Universidad cubana tanto en el ámbito instructivo como educativo. Entre sus principales retos está el lograr un incremento del papel activo de los estudiantes en la adquisición del nuevo conocimiento, desarrollar un nivel superior de independencia y crearles las convicciones para transformar el entorno en que se desarrollan, de tal manera, que al concluir sus estudios sean capaces de integrarse al medio productivo o social de forma diligente, participativa, creativa e innovadora. En este contexto se examina el PEA de la Programación, el cual constituye un componente fundamental en la formación informática del Ingeniero. Es un proceso continuo, complejo e intencional, que para lograrlo es necesario proyectar, organizar, ejecutar y controlar sistemas de acciones con una adecuada relación teoría y práctica, donde se integren elementos motivacionales, afectivos, cognitivos y valorativos, en correspondencia con las exigencias y necesidades que plantea el plan de estudio y los objetivos que persigue la asignatura en la formación integral del futuro profesional.

Aprender a programar en algún lenguaje, diseñar programas educativos, iniciar a los alumnos en las técnicas básicas de programación de computadoras, resolver problemas docentes y profesionales, entre ellos: matemáticos, eléctricos, físicos, relacionados con la gestión económica, comercio y otras muchas esferas de la vida moderna, han contribuido, al eficiente uso del ordenador y a la adquisición de conocimientos, desarrollo de habilidades y formación de valores de estudiantes de todos los niveles y especialidades.

En diversas publicaciones [1- 6] se identifican elementos presentes en la formación informática de un estudiante y que los autores consideraron oportuno destacarlos y reelaborarlos en este trabajo, ellos son: Definir conceptos y relacionarlos con otros; explicar y aprovechar procedimientos informáticos; localizar, adquirir, procesar, almacenar, comunicar y evaluar informaciones; mantener una actitud ética y responsable al manipular la información y resolver problemáticas diversos con el uso de la computadora empleando lenguajes de programación y aprovechar las potencialidades que ofrece la informática para la realización de tareas de otras disciplinas.

Para el desarrollo de este trabajo se proyectaron y cumplieron las siguientes tareas:

Búsqueda y sistematización bibliográfica; Diagnóstico del PEA en las asignaturas Programación Estructurada y Programación Orientada a Objetos; Identificación de las particularidades de la enseñanza de la Programación e Identificación de algunas reflexiones, ideas y procedimientos didácticos para la enseñanza de la Programación.

Todo lo antes expresado posibilitó una reconceptualización de los programas de estudio, distribución de clases y orientaciones metodológicas a los profesores para su aplicación en la práctica docente.

Indagaciones empíricas y teóricas permitieron a los autores identificar el estado actual de la situación académica en el proceso de enseñanza – aprendizaje de la Programación en las facultades de Ingeniería Informática e Industrial.

### **Identificación de la situación académica actual**

Con el empleo del método de investigación científica e histórico–lógico se realizó un estudio de los resultados docentes de años anteriores para observar el comportamiento de la asignatura Programación y otras afines, y se pudieron conocer las distintas etapas del desarrollo y evolución del proceso de enseñanza–aprendizaje de la Programación, sus antecedentes y desarrollo hasta el presente en la Cujae.

Mediante el análisis documental, se pudo investigar sobre los contenidos de planes de estudios, programas aplicados, planes de clases, registros de asistencia y evaluación, libros de texto y resoluciones vigentes, para

conocer, el estado actual del proceso de enseñanza-aprendizaje de la Programación en las carreras de informática e industrial.

La aplicación de encuestas a estudiantes y a profesores con categoría docente de auxiliar y asistentes con más de 5 años de experiencia en la impartición de las asignaturas y el análisis de los resultados de controles parciales y exámenes finales, permitió identificar un conjunto de insuficiencias, entre ellas: deficiente interpretación de problemas y su algoritmización, escasa autopreparación al asistir a clases, insuficiente tiempo de estudio y literatura consultada no adecuada.

Toda la investigación realizada junto a la experiencia del colectivo de profesores, posibilitó determinar algunas particularidades de la enseñanza en estas asignaturas, las cuales se expresan a continuación.

### **IDENTIFICACIÓN DE ALGUNAS PARTICULARIDADES DE LA ENSEÑANZA DE LA PROGRAMACIÓN**

- Una lenta evolución en el uso de las plataformas interactivas de trabajo en contraposición con las exigencias de la sociedad de la información y el conocimiento, que requiere de un elevado nivel de autoaprendizaje por parte de los estudiantes.
- Poca preparación individual de los estudiantes, lo que requiere de la maestría de los educadores para enseñar con creatividad, promover la independencia y la ética, auxiliándose de estrategias de aprendizaje.
- La necesidad de una enseñanza apoyada tanto en software educativo como en herramientas de gestión de conocimiento.
- El empleo de un sistema evaluativo que exija del estudio sistemático por parte de los estudiantes y que contribuya al desarrollo de valores.
- La necesidad de formar valores en la formación del ingeniero, a través de las asignaturas de Programación.
- La carencia de métodos productivos de estudio para la solución de problemas a través de la programación, sustentado en la concepción de aprender a aprender, que desarrollen el pensamiento lógico y las habilidades para el diseño.
- Necesidad de una motivación constante por el estudio de la carrera a partir de su perfil profesional.

La identificación de las anteriores particularidades, articuladas con otras indagaciones realizadas durante los cursos escolares 2013 – 2014, 2014 – 2015 y 2015 - 2016, permitieron arribar a algunas reflexiones esenciales para una concepción diferente en el PEA de la Programación.

### **REFLEXIONES SOBRE UNA CONCEPCIÓN TEÓRICO–METODOLÓGICA DE LA ENSEÑANZA Y EL APRENDIZAJE DE LA PROGRAMACIÓN**

La programación, al ser una actividad compleja, requiere de diversas estrategias para su mejor comprensión, y entre sus acciones principales juega un papel primordial la elaboración de algoritmos que resuelvan el problema planteado, al ser una tarea previa y necesaria antes de emplear un lenguaje de programación, sin embargo, es frecuentemente omitida.

Mediante el desarrollo de **algoritmos** se puede modelar, teórica o gráficamente, la ejecución de un conjunto de acciones que solucionan un problema, tanto en la docencia como en la vida profesional del futuro ingeniero.

Se entiende por **algoritmo**, el ordenamiento y ejecución de un número finito de pasos lógicos y precisos que conllevan la realización de acciones concretas, para la entrada, procesamiento y salida de la información, dirigidos a la solución de un problema. Para ello se buscan los elementos informáticos, aplicando recursos heurísticos (reglas, estrategias, principios, etc.), así como recursos algorítmicos (procedimientos básicos ya conocidos) que permitan modelar una solución a través de la descripción algorítmica.

Los autores concuerdan [2,3-7] que el **enfoque algorítmico** y la gestión de proyectos son predominantes en los cursos que tienen como objetivo central enseñar a programar, se caracteriza por una enseñanza que hace el énfasis principal en el desarrollo de métodos para elaborar algoritmos, es decir, en procedimientos algorítmicos y heurísticos para resolver problemas por medios informáticos. Los contenidos referidos a lenguajes o software para usos específicos pasan a ocupar un segundo plano.

Los educandos de Ingeniería Industrial e Informática reciben el **lenguaje algorítmico**, en el que aprenden los elementos básicos y las estructuras fundamentales que les permiten resolver un problema, ya sea mediante la

escritura de pseudocódigos, realizando diagramas de actividad UML, o ambas formas para modelar la solución de un problema.

El proceso de enseñanza–aprendizaje de la Programación, tomando como base el **lenguaje algorítmico**, permite proceder a realizar las analogías correspondientes con las instrucciones estudiadas en el lenguaje de programación, para construir paso a paso el código de la aplicación. A partir de este momento, los estudiantes pueden escribir programas, que posteriormente serán introducidos en la computadora para su ejecución y realizar las pruebas necesarias.

La elaboración de **algoritmos computacionales** va respaldada de acciones que posibilitan el desarrollo del pensamiento lógico del estudiante y la realización eficiente de la solución del problema planteado.

Es recomendable plantearse las siguientes interrogantes antes de iniciar un algoritmo, por simple que este parezca:

¿Qué datos o elementos iniciales se tienen? ¿Qué se desea obtener? ¿Qué otros datos necesitarían? ¿Cómo llegar al resultado? ¿Qué fórmula(s) se deben aplicar? ¿Qué otro procesamiento se necesita? ¿Qué pasos dar?

Es conveniente que, en el PEA, el profesor, durante la fase de **elaboración de algoritmos**, exija al estudiante la explicación del mismo y la corrida con dos o más juegos de datos que permitan validarlo, y además, que pueda responder otras preguntas como las que se presentan a continuación:

¿Qué tipo de estructuras algorítmicas se están empleando? ¿Por qué? ¿Qué variables y constantes intervienen en este?

¿Cuándo y dónde se produce la entrada de datos? ¿Cuándo es que se sale de la repetición? ¿Qué se sucede cuando se sale de la repetición?

**Diseñar e implementar** un programa para luego ser ejecutado en una computadora requiere del estudiante el dominio de un sistema de conocimientos y habilidades. Para ello se recomienda tener en cuenta los siguientes pasos: Lectura y análisis del problema, diseño de su solución mediante un lenguaje algorítmico, implementación en un lenguaje de Programación, puesta a punto de la solución obtenida y validación del resultado.

El proceso de PEA de la Programación es de interacción e interrelación entre educadores y estudiantes, mediante el cual los primeros dirigen, motivan, controlan y participan activamente en el aprendizaje, a través de una adecuada actividad y comunicación, que facilite la adquisición de la experiencia histórico-social y el crecimiento personal de los estudiantes en un proceso de construcción y reconstrucción individual y colectiva.

Un **aprendizaje desarrollador** de la Programación se logra cuando el estudiante garantiza la apropiación activa y creadora de la cultura, propiciando el desarrollo de su auto-perfeccionamiento constante, de su autonomía y autodeterminación, en íntima conexión con los necesarios procesos de socialización, compromiso y responsabilidad social [3].

Lo esencial en el PEA y en particular de la Programación no es transmitir información, sino que los estudiantes “aprendan a aprender” y aprendan a hacer.

Así como sucede en otras disciplinas, el contenido de la **enseñanza de la Programación**, incluye los conocimientos expresados en conceptos, teorías, leyes; la actividad creadora del estudiante; las normas de relación con el mundo donde vive; los valores, las habilidades y hábitos.

La **calidad del PEA** de la Programación se logra cuando se satisfacen las necesidades de los estudiantes, de los profesores, la escuela, familia y sociedad, por lo que todo cambio a realizar en el sistema educativo, deberá orientarse a la formación de un profesional capaz de vivir y desarrollarse en el contexto actual [8].

Debe existir una correspondencia directa al concebir el sistema de objetivos en el programa de estudio, con los objetivos definidos para el **perfil del profesional** y los principales problemas científicos de la especialidad a resolver por el estudiante, en interrelación estrecha con las demás asignaturas.

La dirección del trabajo independiente y en particular el **estudio individual** conllevan al desarrollo de acciones que permitan el cumplimiento de los objetivos de la asignatura y constituyen un momento muy especial del PEA de la Programación, en el que profesores y estudiantes pueden comprobar hasta dónde se han cumplido los objetivos propuestos: los primeros en la orientación y dirección y los segundos en la adquisición de los niveles de independencia necesarios y suficientes para aprender, para autoevaluarse y participar activamente en el proceso de la evaluación [9].

La orientación de la actividad de aprendizaje de esta asignatura debe trascender los marcos teóricos: es necesario hacer énfasis en que lo más importante es **educar enseñando** y para ello se necesita concebir un **sistema de tareas docentes** que propicien la adquisición de conocimientos, desarrollo de habilidades y

formación de valores, por lo que es necesario descubrir modelos más abiertos y flexibles, donde el estudiante pueda hacerse responsable de la construcción de su propio aprendizaje [10].

Un enorme cúmulo de **tareas docentes** [1,11] se pueden realizar con el empleo de las computadoras en función de la enseñanza – aprendizaje de los estudiantes, entre ellas: “explicar un concepto; elaborar un algoritmo; resolver un problema; observar, describir, analizar y reflexionar sobre objetos y procesos; navegar dentro de redes, interactuar con softwares, la búsqueda e intercambio de información y concebir, planificar, realizar y validar un proyecto informático. Estas contribuyen a una educación con un incremento demostrado en los conocimientos y habilidades, lo que convierten estas tecnologías en poderosos medios de enseñanza, aprendizaje y autoaprendizaje” [9].

La **tarea docente** permite concretar las acciones y operaciones a realizar por el estudiante, tanto en la clase, como fuera de esta, por lo que han de diseñarse tareas orientadas al estudio independiente de los estudiantes.

Para la **enseñanza–aprendizaje de la programación** se sugieren sistemas de **tareas docentes** cuidadosamente concebidas, que garanticen el desarrollo de los niveles de asimilación de los estudiantes, las cuales deben transitar por la familiarización con el contenido; la reproducción; aplicación hasta la creación, y desde la solución del ejercicio más sencillo, hasta problemas complejos como plantearse hipótesis, concebir y ejecutar proyectos y redactar problemas, entre muchas otras posibles [8].

Las **tareas docentes** deben ser suficientes, variadas, diferenciadas y asequibles, de modo que aseguren el progreso personal del educando y del grupo. Estas deben estar al alcance de todos, en función de las diferencias individuales de cada estudiante, para los más rezagados y los que asimilan más rápido el contenido y así promover el desarrollo de los alumnos más avanzados y del grupo. Se sugiere diseñar y poner en práctica sistemas de tareas para adquirir conocimientos y desarrollar habilidades en el lenguaje algorítmico, elementos del lenguaje de programación, dominio de la interfaz del lenguaje de programación, solución de problemas (análisis de casos de estudio), realización del proyecto por parte de los estudiantes y evaluación periódica de la marcha del proyecto.

Para ello el profesor puede conducir el PEA empleando diversas técnicas y procedimientos didácticos, entre estas:

- La explicación de un concepto, recordarlo tantas veces como sea necesario y expresar dónde o cómo el mismo se aplica. Ejemplo, los conceptos de algoritmo, tipos de datos, estructuras algorítmicas, programación orientada a objeto, clases, objetos, interfaz.
- La creación de mapas conceptuales para relacionar conceptos, ideas, teorías, leyes.
- La exigencia a los estudiantes para que respondan preguntas o sistemas de preguntas de tipo:  
¿Qué? ¿Por qué? ¿Cómo? ¿De qué otra forma? ¿Para qué? ¿Dónde? ¿Quiere decir con esta afirmación que.....? ¿Se puede demostrar lo que dice? ¿Qué razones hay para afirmar eso? ¿En qué se diferencia de...? ¿Qué semejanzas se observan entre estos.....? ¿Y si lo resuelve de esta manera que sucederá? Y sean capaces de argumentar las respuestas o soluciones que se pidan.
- Se presentan situaciones a los estudiantes para que elaboren ejemplos similares.
- La comparación de conceptos, objetos, algoritmos, interfaces, códigos de programas y formas de pensar que permitan la identificación de invariantes y en particular la relación de modelado que se presenta en la programación.
- Los contraejemplos para explicar lo eficiente o no en un algoritmo o programa y las ayudas o impulso motivacionales, centrando la atención en aspectos críticos o más dificultosos en su aprendizaje y afirmando estos con el empleo de ciertos recursos como: mapas conceptuales, gráficos, fórmulas, símbolos, diapositivas, videos, entre otras.
- La solución de problemas relacionados con su futura actuación profesional y el desarrollo de valores.
- La realización de presentaciones electrónicas para exponer las ideas fundamentales de un proyecto informático.

A lo que se aspira en la enseñanza de la Programación es que se desarrollen **tareas docentes**, que posibiliten un aprendizaje productivo y el desarrollo de las potencialidades de los futuros ingenieros, no solo desde lo reproductivo sino, aprendiendo a buscar relaciones entre la cosas en sí, características, ejemplos y contraejemplos, valorando y argumentando sus respuestas y favoreciendo la comunicación interpersonal, la búsqueda y el intercambio de información, proporcionando el aprender a escuchar y a criticar puntos de vistas.

Se sugiere el empleo del **software educativo** como medio de enseñanza–aprendizaje siempre que esté pedagógicamente fundamentado y dentro de un dominio de conocimientos asequible a los educandos. El diseño y empleo de un software educativo conlleva un serio trabajo de mesa.

La **evaluación** debe estar en correspondencia con los objetivos del programa, asignatura y el modelo del profesional. Existe el consenso por los autores de realizarla a partir de acciones sistemáticas, trabajos de control, tareas de curso, seminarios y examen final, para valorar el cumplimiento de los objetivos propuestos.

El **enfoque de proyecto** [2 -13] es un enfoque didáctico general que tiene como objetivo central motivar la enseñanza de los contenidos informáticos contemplados en un curso a través del planteamiento de un proyecto a realizar en el curso. Se caracteriza por la subdivisión del proyecto en problemas parciales, necesarios y que motiven a la vez, la enseñanza del nuevo conocimiento.

Este es un enfoque que se aplica actualmente en varios países como el enfoque predominante de la enseñanza de la Programación. Facilita su aplicación combinándolo con otros enfoques. Crea altos niveles de motivación vocacional en el alumnado, ya que lo ubica en actividades productivas o de servicios.

Se puede proceder según las orientaciones dadas a continuación:

- Se parte de un proyecto a realizar en el curso, que se va ejecutando en la medida que se asimilan los conocimientos informáticos necesarios. Cada fase del proyecto debe motivar la obtención del nuevo conocimiento.
- Se obtiene el nuevo conocimiento informático (conceptos y/o procedimientos), según la vía lógica elegida.
- Se realizan acciones de fijación inmediata, teniendo en cuenta las características esenciales del concepto o pasos esenciales del procedimiento.
- Se aplica el nuevo conocimiento o parte del mismo en la solución de la fase correspondiente del proyecto.

La aplicación de este enfoque en la asignatura Programación, junto con el enfoque algorítmico, es posible si los temas de proyectos son transmitidos (facilitados) a los estudiantes con el tiempo suficiente, para que a partir de intercambios colectivos y la autopreparación personal, consigan realizar el análisis, interpretación y búsqueda parcial de soluciones.

Un **proyecto** informático lo componen un conjunto de tareas cuyo objeto es la realización de un software que automatice el sistema de información requerido por el usuario [2, 14, 15].

A continuación, se ejemplifica un **sistema de tareas** docentes para adquirir conocimientos y desarrollar habilidades en la realización del proyecto por parte de los estudiantes.

- a) Formación de los equipos de trabajo.
- b) Distribución de los temas por equipos de trabajo.
- c) Elaboración del plan de trabajo por equipo, esta tarea implica la realización de acciones como:  
La gestión de la bibliografía. Identificar fechas en que se reunirán en la semana. Gestión de los tiempos de máquina dentro y fuera del centro. Evaluación de la marcha del proyecto.
- d) Ejecución parcial y sistemática del plan de trabajo según se va introduciendo el nuevo contenido -  
Elaboración de los algoritmos de cada método (función) que interviene.
  - Identificación, diseño y representación de las clases que intervienen en el problema.
  - Escritura del código de las clases.
  - Implementación y puesta a punto del código de las clases.
  - Diseño de la interfaz de la aplicación en el trabajo de mesa e implementación de esta.
  - Elaboración de los códigos de cada método (botones u otros controles).
  - Confección del informe del proyecto.
  - Realización de una presentación electrónica para la exposición.
- e) Comunicación en los cortes del proyecto de lo investigado, aun cuando todavía se trabaja en el lenguaje algorítmico.
- f) Evaluación de cada miembro del equipo y del cumplimiento del plan de acciones que se planificó.
- g) Rediseño de las acciones para la elaboración del proyecto de cada equipo.

Las **estrategias de aprendizaje** son esenciales para el PEA de la Programación. Se expresan como el conjunto de actividades, técnicas y medios que se planifican de acuerdo con las necesidades de los estudiantes, los objetivos que se persiguen y el contenido de la enseñanza y tienen como finalidad la eficiencia y efectividad

del proceso de aprendizaje. Es necesario que los estudiantes empleen estrategias de aprendizaje que propicien la reflexión, autovaloración, exposición y la activación de conocimientos previos, entre otros [16-18]. Los autores asumen que las estrategias promueven un aprendizaje autónomo, independiente, de manera que las riendas y el control del aprendizaje pasan de las manos del profesor a las manos de los educandos. Esto es especialmente provechoso cuando el estudiante es ya capaz de planificar, regular y evaluar su propio aprendizaje, es decir, cuando posee y domina las estrategias de aprendizaje llamadas metacognitivas [17].

En la asignatura Programación, en ambas carreras (Industrial e Informática) se han incorporado las siguientes estrategias:

Leer de manera comprensiva un texto tantas veces como se requiera según la complejidad de este, realizar resúmenes parciales y generales, exposición de ideas a otros estudiantes, resolver problemas, esquematizar ideas mediante UML, la realización de mapas conceptuales, elaboración de algoritmos y la autovaloración crítica de lo aprendido [13, 19,20].

### RESULTADOS OBTENIDOS

La constatación de la viabilidad de la concepción empleada queda expresada en el conjunto de reflexiones teóricas, aplicadas en las carreras de Ingeniería Informática e Ingeniería Industrial durante cuatro cursos escolares, en las asignaturas de Introducción a la Programación (IP) y en un curso en la asignatura de Diseño y Programación Orientada a Objetos (DPOO), en contraposición con el modelo anterior. Los análisis docentes de los resultados, mostraron una tendencia a la recuperación, tanto desde lo cuantitativo (un porcentaje superior de aprobados en los exámenes) como en lo cualitativo por la superior calidad de los proyectos presentados y sus defensas en los tribunales, lo cual puede ser observado en la figura 1.

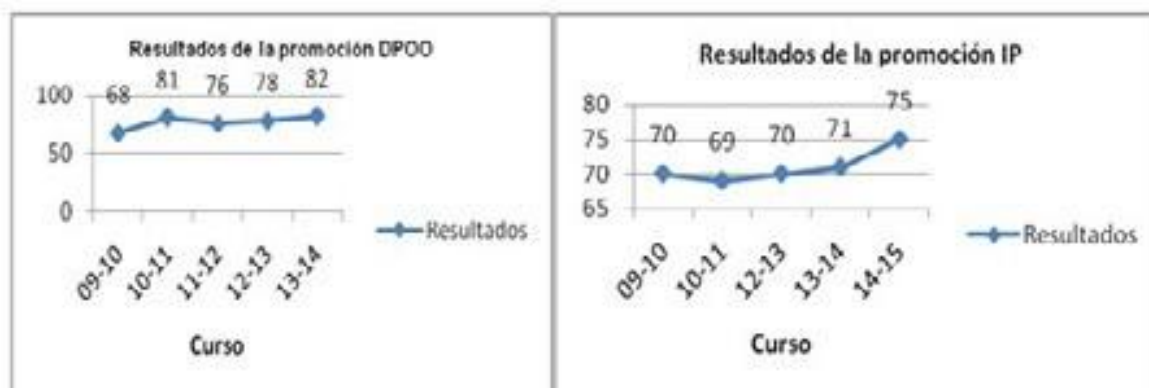


Fig. 1. Resultado de la promoción histórica hasta el momento en IP y DPOO

En entrevistas y análisis realizados con estudiantes y profesores de las facultades de Ingeniería Industrial e Informática durante los cursos escolares 2014 – 2015 y 2015 – 2016 con el objetivo de conocer su opinión y resultados docentes a partir de una nueva concepción teórico-metodológica empleada en el proceso de enseñanza-aprendizaje, se pudo constatar un alto nivel de aceptación, por el nivel metodológico de las clases y las preparaciones metodológicas; el trabajo de análisis y diseño que han realizado al solucionar los proyectos y el método más participativo para abordar la solución de problemas. Los resultados de los trabajos de curso, como parte de la evaluación, se consideraron satisfactorios. No obstante, aunque se apreciaron clases con un mayor nivel metodológico y de contenido y una tendencia a mejorar el método de estudio al resolver problemas, aún se observan insuficiencias en la dedicación al estudio y el desarrollo de las tareas extraclases.

En la figura 2 se puede observar el comportamiento de algunas métricas que se evaluaron antes y después de la aplicación de la propuesta.

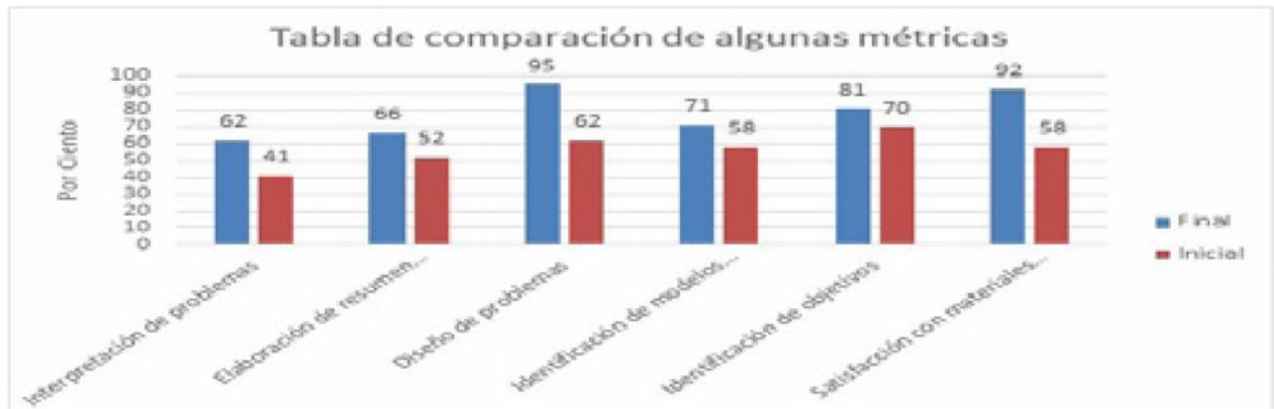


Fig. 2. Otros resultados

Los autores después de exponer las reflexiones teórico – metodológicas y valorar los resultados obtenidos y reflejados en los gráficos 1 y 2 infieren que la aplicación de esta concepción contribuyó al perfeccionamiento del proceso de enseñanza – aprendizaje de la Programación en las carreras de Ingeniería Informática e Industrial y como consecuencia el logro de una la formación informática superior. Además, se elaboraron recomendaciones metodológicas para los profesores, que contribuyen al mejoramiento de los programas de estudios en ambas carreras.

## CONCLUSIONES

1. El PEA de la Programación constituye un componente fundamental en la formación informática del Ingeniero Industrial e Informático, la cual es un proceso continuo, complejo e intencional, que para lograrlo es necesario proyectar, organizar, ejecutar y controlar sistemas de acciones con una adecuada relación teórico-práctica.
2. La aplicación de las reflexiones metodológicas identificadas contribuyeron al desarrollo del pensamiento lógico de los estudiantes, la superación pedagógica de profesores y a alcanzar resultados docentes superiores en la asignatura Programación, en las facultades de Ingeniería Industrial e Informática.
3. El análisis del problema y la posterior escritura del algoritmo constituye el primer paso para la solución de problemas en la computadora mediante un lenguaje de programación, omitir este puede conducir a errores lógicos, pérdida de tiempo y al uso ineficiente de este medio de enseñanza – aprendizaje y su programación en cualquier lenguaje, constituye una actividad intelectual que requiere de conocimientos, habilidades, motivación, práctica y estrategias de aprendizaje para una correcta modelación de la solución.
4. En la enseñanza–aprendizaje de la programación, la dirección del trabajo independiente constituye una actividad metodológica de suma importancia y dentro de ella se sugieren sistemas de tareas docentes que garanticen el desarrollo de los niveles de asimilación de los estudiantes: familiarización con el contenido; de reproducción; de aplicación y de creación que posibiliten un aprendizaje productivo, el desarrollo de las potencialidades de los futuros ingenieros.
5. El enfoque de proyecto, como parte de la evaluación de los conocimientos y habilidades, crea altos niveles de independencia, motivación profesional, colaboración entre los estudiantes y desarrollo de valores.
6. La concepción teórico – metodológica propuesta para el perfeccionamiento del proceso de enseñanza – aprendizaje de la Programación en las carreras de Ingeniería Informática es factible y su aplicación, tal como se corrobora en las figuras 1 y 2 propicia la obtención de resultados cualitativos y cuantitativos superiores y en ascenso.



## RECOMENDACIONES

1. Incorporar los resultados presentados en este trabajo a los departamentos docentes para la superación metodológica de los profesores de Programación en las carreras de Informática e Industrial y su continua validación.
2. Que las reflexiones teórico–metodológicas presentada en este trabajo sean enriquecidas a partir de nuevas investigaciones realizadas en el PEA de la Programación, para el mejoramiento de la didáctica de la enseñanza de la computación en las especialidades de Informática e Industrial.
3. Elaborar, aplicar y analizar instrumentos que evalúen el impacto que tienen las reflexiones propuestas en el PEA de la Programación para la contribución a una formación informática superior en el Ingeniero Industrial e Informático.

## REFERENCIAS

1. García Meneses M, et al. La formación informática en el Licenciado en Educación Primaria. Cuba: Universidad de Ciencias Pedagógicas José de la Luz y Caballero; 2013. [Citado Disponible en: <http://atlante.eumed.net/formacion-informatica>.
2. Rodolfo Sáenz A. El Éxito de la Gestión de Proyectos. Un nuevo enfoque entre lo tradicional y lo dinámico. Publicación electrónica en PDF. 2013:6-10.
3. López García JC. Algoritmos y Programación. Guía para docentes. Disponible en Fundación Gabriel Piedrahita Uribe 2009. ISBN. DOI Disponible en: <http://www.eduteka.org>.
4. Carrion Mustelier Y. SIMPOSIO 5: Enseñanza aprendizaje de las Ciencias en las edades tempranas - Enseñanza aprendizaje de la informática. In: IX CONGRESO INTERNACIONAL DIDÁCTICAS DE LAS CIENCIAS; La Habana Cuba: 2016. p. ISBN DOI [Citado Disponible en: <http://www.reddolac.org/events/ix-congreso-internacional-did-cticasde-las-ciencias-y-en-el-xivwww.conacytprensa.mx/index.php/sociedad/eventos/evento/514-ix-congreso-internacional-didacticas-de->
5. Gabbarini P. Enseñar a enseñar en las aulas universitarias. Experiencia y saber en dos profesoras de formación del profesorado. España; 2017.
6. Aguirre JF, et al. Una experiencia para fortalecer los procesos de enseñanza de la programación mediante el uso de entornos virtuales de aprendizaje. 2015 (5). ISSN 2173-8688.
7. Álvarez de Zayas C. La Pedagogía como ciencia. Epistemología de la educación. Publicación electrónica en CDROM. Cuba: Instituto Superior Pedagógico Enrique José Varona; 1995.
8. García AG. Tareas docentes integradoras en la Física escolar. Atenas Revista científico pedagógica. 2015;2(30):4-7. ISSN 1682 - 2749.
9. Niño Vega JA, Martínez Díaz LY, et al. Entorno de aprendizaje para la enseñanza de programación en Arduino mediado por una mano robótica didáctica. Revista Espacios. 2017;38(60).
10. Pérez L. Estrategias de aprendizaje. Función y diagnóstico del aprendizaje adolescente. Padres y maestros. 2014 (358). ISSN 2255-1042.
11. Sanjur D. Una Metodología para el aprendizaje de la Programación Orientada a Objetos basada en Programación Extrema XP. Universidad Tecnológica de Panamá; 2011.
12. García AG. Tareas docentes integradoras en la Física escolar. Atenas Revista científico pedagógica. 2015;2(30):4-7. ISSN 1682 - 2749.
13. Blanco JM, et al. La espiral de proyectos como eje conductor de asignaturas de Gestión de Proyectos Informáticos. In: Actas de las XX JENUI Oviedo España; 2014. p. 3-10. Disponible en: <http://upcommons.upc.edu>.
14. Niño Vega JA, Martínez Díaz LY, et al. Entorno de aprendizaje para la enseñanza de programación en Arduino mediado por una mano robótica didáctica. Revista Espacios. 2017;38(60).
15. Rodríguez Lamas R. Introducción a la Informática Educativa. La Habana Cuba: Editorial Pueblo y Educación; 2001.
16. Pérez L. Estrategias de aprendizaje. Función y diagnóstico del aprendizaje adolescente. Padres y maestros; 2014 (358). ISSN 2255-1042.

17. Sanjur D. Una Metodología para el aprendizaje de la Programación Orientada a Objetos basada en Programación Extrema XP. Universidad Tecnológica de Panamá, 2011.
18. Zabalza MA. El estudio de las buenas prácticas docentes en la enseñanza universitaria. Revista de Docencia Universitaria, número especial dedicado a Engineering Education; 2012;10(1):17-42. ISSN 1887-4592.
19. Salgado A, et al. Lógica algorítmica para la resolución de problemas de programación computacional: una propuesta didáctica. Didáctica y Educación. 2012;4(1). ISSN 2224-2643.
20. Rodríguez Lamas R. Uso de los Mapas Conceptuales como estrategia de aprendizaje en la Educación Superior. La Habana Cuba: Instituto Superior Politécnico José Antonio Echevarría; 2010.