

Aplicación de una herramienta basada en Software Libre para la enseñanza de Algoritmos y Lógica de Programación

Carrizo Blanca Rosa ⁽¹⁾, Corso Cynthia Lorena ⁽²⁾, Olmedo Adriana ⁽³⁾

⁽¹⁾ *Universidad Tecnológica Nacional, Facultad Regional Córdoba*
Dpto. Ing. Industrial/Dpto. Ing. Mecánica/Dpto. Ing. Sistemas de Información
[Correo Electrónico: bcarrizo@tecnicatura.frc.utn.edu.ar](mailto:bcarrizo@tecnicatura.frc.utn.edu.ar)

⁽²⁾ *Universidad Tecnológica Nacional, Facultad Regional Córdoba*
Dpto. Ing. Sistemas de Información
[Correo Electrónico: cynthia@bbs.frc.utn.edu.ar](mailto:cynthia@bbs.frc.utn.edu.ar)

⁽³⁾ *Universidad Tecnológica Nacional, Facultad Regional Córdoba*
Dpto. Ing. Industrial/Dpto. Ing. Mecánica/Dpto. Ing. Sistemas de Información
[Correo Electrónico: aolmedo@tecnicatura.frc.utn.edu.ar](mailto:aolmedo@tecnicatura.frc.utn.edu.ar)

Maestro M. López esq. Cruz Roja Argentina. Ciudad Universitaria S/N

RESUMEN

Este trabajo tiene como objetivo exponer una propuesta pedagógica para la cátedra “Informática I” de la carrera Ingeniería Industrial, que facilite el aprendizaje de los alumnos en esta temática. La “Informática”, como una ciencia interdisciplinaria que surge de la interacción sinérgica de varias ciencias, como la computación, la electrónica, la cibernética, las telecomunicaciones, la matemática, la lógica, la ingeniería, la robótica, la biología, las ciencias de la información, cognitivas, organizacionales, entre otras; es aplicada al estudio y desarrollo de los productos, servicios, sistemas e infraestructuras para la sociedad.

Por ello, uno de los objetivos de la cátedra es “aplicar la Informática, como disciplina científica y tecnológica, a través de sus aplicaciones prácticas sustentadas por las teorías específicas que hacen uso de sus métodos y procedimientos; los cuales permiten captar los fenómenos o resolver los problemas relacionados con ellos” así como “desarrollar habilidades de lógica de programación para la resolución de problemas específicos de la carrera mediante la construcción y prueba de algoritmos.”

Cabe aclarar que, en el diseño curricular de la carrera, la temática está incluida en los contenidos mínimos y que al ser incorporada al Programa Analítico de la asignatura, se analizó qué orden debería tener en el mismo así como qué teóricos significativos incluir de tal forma de potenciar la resolución de problemas en forma gradual aplicando el “Enfoque de Sistemas”.

Fruto de este análisis, se decidió reflejarlo en forma temprana en la Unidad 2 con la denominación de “Introducción al Diseño de Algoritmos y Lógica de Programación”; con la finalidad de sentar bases sólidas en lo referente a la capacidad de resolución de problemas y potenciarla incorporando la herramienta denominada PSeInt apoyada por una estrategia heurística para la resolución de problemas como lo es la de Polya.

Mediante la aplicación de esta metodología de trabajo en laboratorio, basada en la aplicación de un software libre, se pretende facilitar la diagramación de estrategias de resolución de problemas a través de Pseudocódigo y Diagramas de Flujo.

Palabras Claves: Resolución de Problemas. Algoritmos. Lógica de Programación. PSeInt. Heurística de Polya.

ABSTRACT

This paper has the objective of exposing a teaching proposal for "I.T. 1 (Information Technology)", a subject that is part of the Industrial Engineering major. This will enable students to study and learn easily in this area.

I.T. is an interdisciplinary science that emerges from the synergic interaction of various scientific areas, such as: computing, electronics, cybernetics, telecommunications, Maths, logic, engineering, robotics, biology, and information, cognitive and organizational sciences, among others, which is applied to the study and development of the products, the services, the systems and the infrastructure necessary for society.

Therefore, one of the objectives of this academic area is "to apply I.T. as a scientific and technological discipline through applications sustained by specific theories that make use of methods and procedures, which allow to acquire phenomena or solve problems related to them", and also "to develop skills in programming for the resolution of specific problems related to their area of specialty by means of the build up and trial of algorithms".

It is important to note that, in the syllabus, this topic is included as a secondary issue and so as to be included as an important issue of the subject, which importance should be given to it and in which order topics should be taught was analyzed. The main idea behind this was to strengthen gradually the resolution of problems by using the "Systems Approach".

From this analysis, it was decided to reflect this topic in Unit 2 under the name of "Introduction to Algorithms Design and Programming Logic"; with the goal of laying the foundations as regards the capability to solve problems and to potentiate this competence by using the tool called PSeINT supported by heuristic strategy to solve problems such as the Polya.

By applying this working procedure in the lab, based on the application of a freeware, the objective is to facilitate the design of problems resolution strategies by means of a Pseudo code and diagrams flow.

Key words: Problems resolution. Algorithms. Logic in Programming. PSeInt. Polya heuristic.

1. INTRODUCCIÓN

En la actualidad los alumnos que cursan carreras de Ingeniería, necesitan incorporar y adquirir destrezas en la resolución de problemas que harán a su quehacer profesional en un futuro no muy lejano, utilizando enfoques con diferentes herramientas y recursos que facilitan la resolución de problemas.

En los diseños curriculares de todas las carreras de la U.T.N. se ha incorporado, como exigencia de la acreditación de las carreras de grado, el tema resolución de problemas a través del diseño de algoritmos.

Esta temática, que parece específica de una Ingeniería en Sistemas, y que hoy es aplicada en forma pertinente a cada perfil a formar, en este caso nuestro Ingeniero Industrial Tecnológico.

Esta temática, facilita el proceso de resolución de problemas mediante etapas o fases referidas a:

- Análisis del Problema: aplicando el “Enfoque de Sistemas”.
- Diseño del Algoritmo: aplicando Diseño Top Down, Algoritmos y Pseudocódigo.
- Programación del Algoritmo: aplicando software que automatice el proceso.

En esta última etapa, y gracias a la creación de herramientas informáticas, como una alternativa interesante en el acompañamiento del proceso de Enseñanza-Aprendizaje, se ha facilitado el diseño de soluciones a problemas concretos en el ámbito de la ingeniería.

Algunas de estas herramientas, basadas en la utilización de software, son: DFD, Raptor (acrónimo del inglés “Rapid Algorithmic Prototyping Tool for Ordered Reasoning”) y PSeInt (abreviatura de pseudo intérprete), entre otras. [1]

Como se ha mencionado existen una variedad de herramientas de distribución libre tanto como comercial, pero la gran mayoría de ellas asume que el alumno tiene un conocimiento minucioso en lo que respecta a estructurar una solución, en función de los datos de entrada y la/s salida/s que deben generar.

En este trabajo se ha considerado como alternativa de acompañamiento en el proceso de enseñanza-aprendizaje la herramienta PSeInt, tomando como criterios de selección la finalidad por el cual ha sido desarrollada es decir como recurso didáctico para la enseñanza de algoritmos y los recursos que utiliza para la diagramación de algoritmos.

PSeInt está pensado para asistir a los estudiantes que se inician en la construcción de programas o en el diseño de algoritmos computacionales. Se trata de un intérprete de pseudocódigo basado en los contenidos de la cátedra de Fundamentos de Programación [2].

Esta herramienta se diseñó para que los estudiantes sin experiencia en programación aprendieran la lógica y conceptos básicos de un algoritmo computacional mediante la utilización de un simple pseudo-lenguaje intuitivo y en español.

El pseudocódigo se suele utilizar como primer contacto para introducir conceptos básicos como el uso de estructuras de control, expresiones, variables, etc. Esta herramienta informática pretende facilitar a los estudiantes la tarea de escritura de algoritmos en este pseudolenguaje ofreciendo un conjunto de ayudas y asistencias que permite a los alumnos a encontrar errores de diseño y comprender la lógica de los algoritmos.

Para facilitar la diagramación de la solución propuesta (Algoritmo), como así también la automatización del test (Prueba de Escritorio) que permite determinar si las salidas generadas son coherentes en función a los requerimientos solicitados.

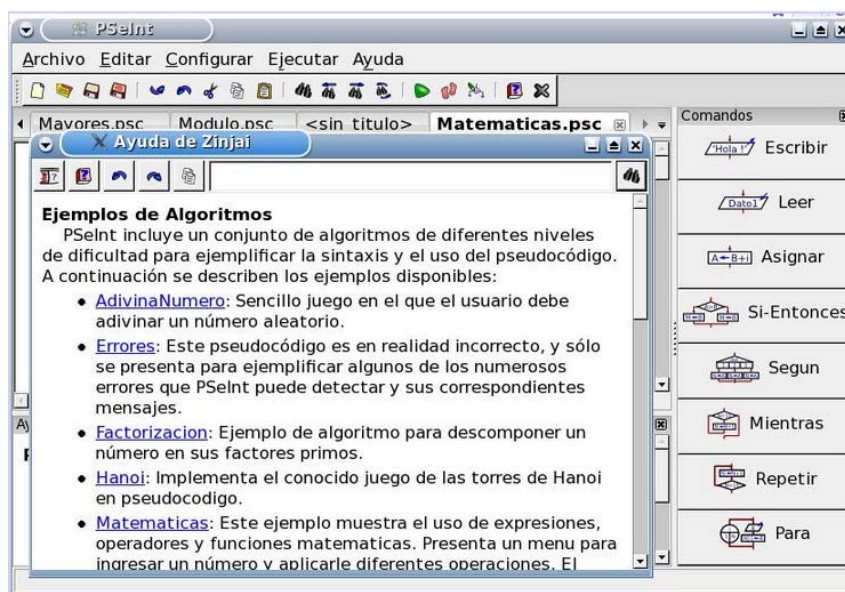


Figura 1. Interfaz gráfica de PseInt.

La gran mayoría de las herramientas informáticas, incluso PSEInt no están vinculadas con alguna estrategia heurística de resolución de problemas.

Una estrategia heurística es una regla general que facilita transformar un determinado problema en una situación más sencilla, facilitando en definitiva el proceso de resolución. Una de las heurísticas más difundidas en el contexto de resolución de problemas es la propuesta por G. Polya. [3]

Esta heurística para la resolución de problemas propone las siguientes etapas:

1. Entender el problema: esta primera etapa tiene como objetivo principal la comprensión del problema, a través de una serie de cuestionamientos sobre el enunciado original.
2. Obtener un plan de solución: el objetivo es idear una estrategia que permita llegar de los datos disponibles hasta la resolución de la incógnita.
3. Aplicar un plan de solución: se pretende en esta etapa llevar a cabo la estrategia sugerida verificando y chequeando cada uno de los pasos propuestos.
4. Revisar la solución: esta etapa indica que se debe analizar la solución para tener la seguridad que los resultados generados son los que se esperaban.

La propuesta de este trabajo consiste en optimizar el proceso de Enseñanza-Aprendizaje correspondiente a la Unidad Nro. 2: "Introducción al Diseño de Algoritmos y Lógica de Programación", combinando el enfoque heurístico para la resolución de problemas propuesta por Polya y como recurso didáctico de acompañamiento la herramienta informática PSEInt.

2. METODOLOGÍA

La metodología de enseñanza propuesta se basa en la heurística de G. Polya usando como herramienta de implementación la herramienta PSEInt.

En la Figura 2 y 3 se puede evidenciar la concordancia que existe entre la heurística de resolución de problemas propuesta por Polya y la alternativa de metodología de enseñanza incorporando como acompañamiento el recurso didáctico la herramienta PSEInt.

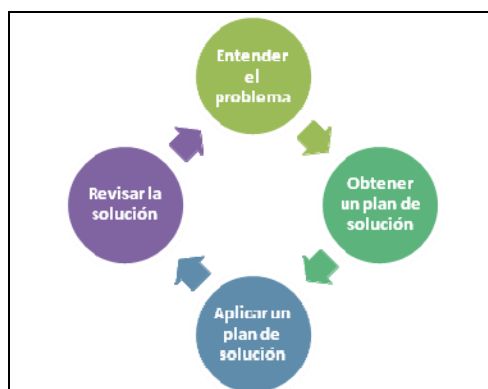


Figura 2. Etapas de la heurística de Polya



Figura 3. Proceso de Enseñanza-Aprendizaje propuesto

"Entender el problema" concuerda con la etapa "Identificación de componentes principales del problema". En esta etapa se pretende que el alumno pueda lograr una "representación del problema".

Esta instancia consiste básicamente saber identificar los objetivos (o resultados) a cumplir, junto a los datos de los cuales se parte, y ser capaz de plantear una estrategia general de resolución en la

que se indiquen los procesos básicos a realizar para llegar a los resultados partiendo de los datos. Si bien puede parecer una etapa simple y obvia, los hechos muestran que en realidad se trata del punto más complicado y difuso en el proceso de resolver un problema. Si el problema no es de solución trivial, quien plantea el problema debe ser capaz de intuir una línea de acción que lo conduzca, eventualmente, a una solución correcta. [4]

La etapa "Obtener plan de solución" concuerda con la "Especificación del algoritmo". Esta fase se espera por parte del alumno el diseño de la estrategia que ha seleccionado.

La herramienta propuesta a los alumnos para facilitar la especificación del algoritmo es el Pseudocódigo. Se trata de una herramienta por medio de la cual se plantea el algoritmo sin usar gráficas, escribiendo cada acción o paso en lenguaje natural o cotidiano. El pseudocódigo está pensado para ser leído y entendido por una persona, y no por una computadora. El alumno que plantea el pseudocódigo dispone de bastante libertad a la hora para expresar cada paso. No hay un estándar general y las convenciones y reglas de trabajo pueden variar. Un ejemplo de cómo especificar un algoritmo con esta herramienta se visualiza en la Figura 4.

```

Algoritmo:
1. Paso 1...
2. Paso 2...
   2.1. Subpaso
   2.2. Subpaso
3. Paso 3...
    
```

Figura 4. Estructura genérica del Pseudocódigo

En esta etapa se propone al alumno que utilice como recurso la herramienta PSeInt. En la misma podrá realizar una especificación más precisa del algoritmo en pseudolenguaje propuesto por la herramienta. La misma propone que el alumno pueda seleccionar de manera muy simple y gráfica cada paso (secuencia, condición, proceso o repetición) para el diseño del algoritmo. En la medida que el alumno selecciona cada paso y lo incorpora al espacio de trabajo, se genera de manera automática el pseudolenguaje (similar al pseudocódigo).

La etapa "Aplicar un plan de solución" concuerda con la fase "Diagramación del Algoritmo". Finalizada la especificación exhaustiva del algoritmo por parte del alumno, se recomienda al alumno la utilización de una herramienta de diagramación del mismo.

Con el propósito de que el alumno sea capaz de detallar al máximo y de manera exhaustiva cada paso, cada condición, cada rama lógica del proceso. Toda la secuencia de acciones a cumplir para llegar a los objetivos planteados debe quedar evidenciada en forma clara y terminante. Normalmente se sugiere a los alumnos en esta etapa, la utilización de una serie de símbolos para facilitar la tarea de visualizar el algoritmo formando lo que se designa como un Diagrama de Flujo. Los símbolos de un Diagrama de Flujo pueden clasificarse en dos grupos como se muestra en la Figura 5.









Símbolos de Proceso	Símbolos de Entrada / Salida
 inicio / fin de diagrama	 salida por impresora (o también pantalla)
 asignación	 carga por teclado
 condición	 acceso a disco
 ciclo (instrucción de repetición)	
 subrutina	

Figura 5. Símbolos principales utilizados en el Diagrama de Flujo

Con la incorporación de la herramienta PSeInt esta diagramación se realiza de manera automática. En la medida que el alumno hace una especificación del algoritmo mediante el pseudolenguaje propuesto por la herramienta, el Diagrama de Flujo se genera de manera simultánea. La incorporación de esta herramienta en esta etapa agiliza y dinamiza de manera significativa la construcción del algoritmo por parte de los alumnos.

La última etapa "Revisar solución" concuerda con el "Testeo del Algoritmo". En esta fase es fundamental conocer con certeza si las salidas generadas por el algoritmo diseñado son apropiadas en función de los requerimientos del problema original. Una herramienta apropiada que se sugiere a los alumnos utilizarla es la Prueba de escritorio.

La prueba de escritorio es una herramienta útil fundamentalmente para verificar que un algoritmo cumple con la especificación o especificaciones del problema original. Básicamente, es una ejecución "a mano" del algoritmo, por lo tanto se debe llevar registro de los valores que va tomando cada una de las variables involucradas en el mismo. Generalmente la confección de una prueba escritorio implica la realización de una matriz o tabla.

En cada columna se especifican todas las variables involucradas en el algoritmo y el alumno puede simular con diferentes datos de entrada para chequear que las variables de salidas se comporten de manera coherente en función de los datos de entrada.

Para clarificar las etapas propuestas en el proceso de enseñanza-aprendizaje se ejemplifica con un caso de estudio.

2.1. Caso de Estudio

El problema planteado, es un caso práctico cuyo objetivo es la enseñanza de estructuras de control combinando con estrategia de búsqueda de mayor o menor de elementos de una secuencia o serie de datos.

En la siguiente sección del trabajo se hará explícita cada etapa de la metodología de enseñanza propuesta en el contexto de una problemática concreta.

2.1.1. Enunciado del Caso

Una empresa necesita conocer información relacionada con el nivel de producción alcanzada por sus operarios.

Se pide generar las siguientes "Salidas" por Impresora:

- Mencione el operario que más produjo en esa jornada.

Para ello Ud. dispone de las siguientes "Entradas":

- Legajo: representa el número del legajo que identifica al operario.
- Nombre: representa el nombre del operario.
- Cantidad: representa cantidad de unidades producidas por el operario en esa jornada. Está expresada en unidades.

La carga de operarios se realizará hasta que el usuario ingrese un valor de legajo negativo.

2.1.2. Propuesta de Resolución

a. Identificación de componentes principales del problema:

En esta instancia de resolución se propone al alumno que complete esta etapa en papel.

- **Resultados:** Nombre del operario que más produjo en una jornada.

"Nombre. " + Nombre del operario (*mayorNom*: cadena de caracteres)

- **Entradas:** Legajo (*leg*: número entero)
Nombre (*nom*: cadena de caracteres)
Cantidad (*cant*: número entero)

- **Procesos:** Problema de búsqueda de mayor en una secuencia de datos. (contexto)
La idea básica es tomar de la serie de entrada el primer elemento de la misma que será considerado como elemento "pivot" o de comparación, con el propósito de determinar el legajo del operario con mayor producción.
En esta instancia se almacena en la variable "mayor" la producción y "mayorNom" el nombre del primer operario.
Luego se compara de los demás operarios de la secuencia la producción que produjo. Si la producción del operario que se consideró como elemento pivot o comparación es menor que la producción del operario que en ese momento se está procesando, se actualiza el valor asignado a la variable "mayor" y "mayorNom".
Finalmente cuando se procesa toda la secuencia en las variables mencionadas anteriormente queda almacenado el mayor valor de producción y el nombre del operario.

b. Especificación del Algoritmo:

Diseño del algoritmo usando el software PSeInt, como modo de representación del algoritmo a través del Pseudocódigo.

Después de un análisis detallado del caso a resolver, el alumno se apoya con el uso de PSeInt. Para generar el Pseudocódigo debe usar el panel izquierdo de la pantalla principal, en él se indican las diferentes operaciones (Comandos) que pueden ser aplicadas en el diseño del algoritmo.

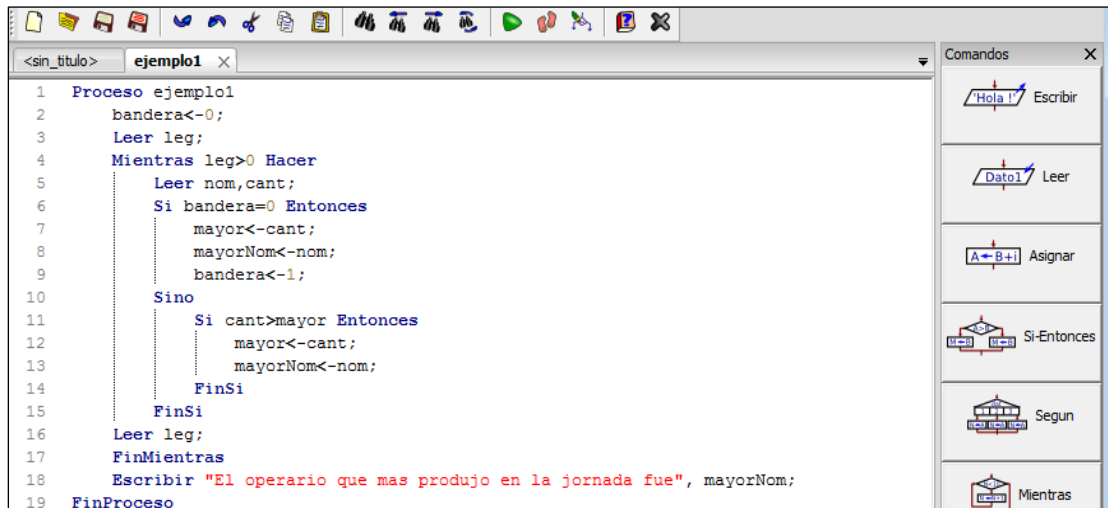


Figura 6. PSeInt: Especificación del algoritmo usando pseudolenguaje

c. Diagramación del algoritmo:

Generación del Diagrama de Flujo a partir del pseudocódigo propuesto. Una vez generado el pseudocódigo y chequeado que la sintaxis del mismo sea correcto, es posible generar el Diagrama de Flujo correspondiente al pseudocódigo generado, en el caso de que no exista ningún error de sintaxis.

Dentro de la opción Ejecutar hay una alternativa que es Dibujar Diagrama de Flujo (Figura 7). A continuación se muestra el Diagrama de Flujo generado por la herramienta.

d. Testeo del algoritmo:

en esta etapa se efectúa la verificación si el diseño del algoritmo planteado por medio de Pseudocódigo genera las salidas esperadas en función de los datos de entrada, es similar a la prueba de escritorio, en este caso la herramienta nos permite realizar esta función de manera automatizada como se muestra a continuación:

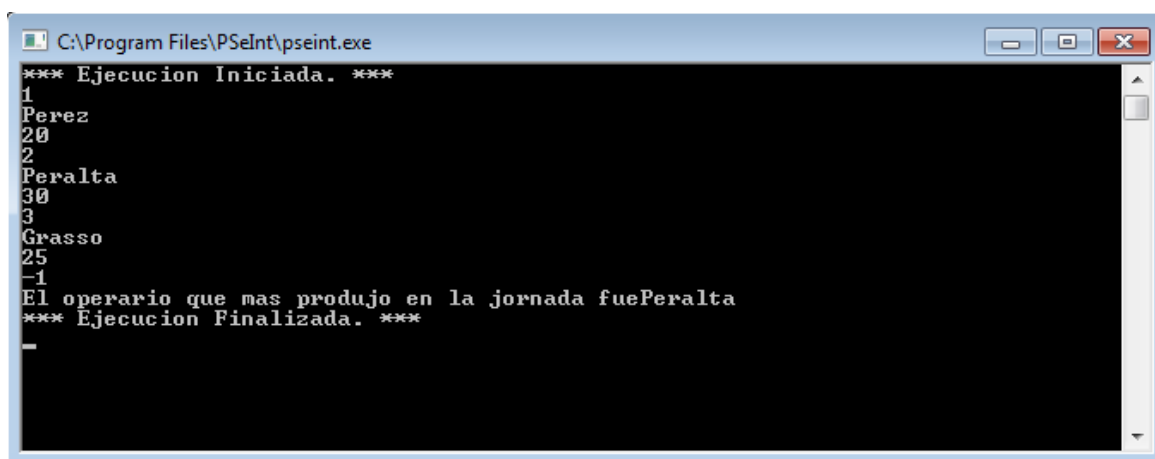


Figura 7. PSeInt: Pantalla de test del algoritmo diseñado

En el caso de que en el proceso de ejecución se obtiene una salida o más salidas que no corresponda/n (de acuerdo a los datos de entrada), se debe revisar el proceso y volver a la etapa a). Ya que los errores detectados en esta etapa responden mayormente a errores lógicos (mal diseño del algoritmo).

3. CONCLUSIONES

El Ingeniero por la esencia de su quehacer, está motivado y formado para hacer y crear bienes y/o servicios; a través de la resolución de problemas reales de ingeniería.

Entendemos por problema una situación propia de un objeto, que provoca una necesidad en un sujeto, el cual desarrollará una actividad para transformar la situación mencionada.

La palabra ingeniero proviene del latín "ingenium" en el concepto de algo que se mueve por sí sólo, entendiendo por tal a una máquina, mecanismo, motor, etc., lo cual podría implicar a una persona como maquinista o mecánico; donde "in" significa "en" y "genium" de "gene", "engendrar" o "generar"; es decir "el ingeniero hace".

Esta asignatura se plantea como uno de los objetivos la formación del perfil del Ingeniero mediante la implementación de una didáctica pedagógica que facilita el proceso de enseñanza y aprendizaje en lo referido a la resolución de problemas mediante el diseño de algoritmos y la lógica de programación.

De allí que, la aplicación de algoritmos computacionales en la resolución de diversos problemas mediante la generación de programas básicos de computadora es muy utilizada en diversos campos interdisciplinarios; particularmente en el contexto industrial en campos como: Cálculo Numérico, Automatización y Robótica.

En un contexto industrial, se puede definir la automatización, como una tecnología que está relacionada con el empleo de Sistemas mecánicos-eléctricos basados en computadoras para la operación y control de la producción.

En consecuencia, la robótica es una forma de automatización industrial.

Los robots son utilizados por una diversidad de procesos industriales, como: la soldadura de punto y soldadura de arco, pinturas de spray, transportación de materiales, molienda de materiales, moldeo en la industria plástica, máquinas-herramientas, entre otras.

Por ello, en los procesos de acreditación de las carreras de grado, los pares evaluadores destacaron el desarrollo de estrategias para la resolución de problemas basadas en soluciones automáticas.

Este trabajo plantea una propuesta metodológica, que facilita el aprendizaje, relacionado con la capacidad de resolución de problemas, mediante la aplicación de un enfoque propuesto por la heurística creada por G. Polya, aplicando la herramienta PSeInt en una etapa temprana del desarrollo de contenidos, con la finalidad que las destrezas desarrolladas sean reutilizadas en Unidades Temáticas más avanzadas.

En nuestro caso bajo análisis, esta herramienta apoya el proceso de enseñanza y aprendizaje de la Unidad Temática N° 2 del Programa Analítico de "Informática I", denominado: "Introducción al Diseño de Algoritmos y Lógica de Programación".

La metodología a seguir, está basada en los siguientes pasos o etapas: [5]

- Representación del algoritmo diseñado mediante la herramienta Pseudocódigo.
- Seguimiento y verificación de la sintaxis propuesta para las instrucciones de Pseudocódigo.
- Visualización de Diagrama de Flujo del pseudocódigo diseñado por el alumno.
- Automatización de prueba de escritorio del algoritmo diseñado.
- Ejecución paso a paso, lo que permite y facilita el seguimiento de la lógica planteada del algoritmo con el objetivo de localizar de manera sencilla los errores de lógica que pudiera existir en el diseño del algoritmo.

El objetivo de "Informática I" es realizar un abordaje holístico a problemas de Ingeniería en general y de la especialidad en particular, a través del análisis de posibles soluciones basadas en la lógica de programación así como propiciar la clarificación de contenidos reutilizables en otras instancias y la adquisición de destrezas relacionadas al uso de utilitarios. [1]

Todas las instancias de enseñanza se elaboran con la finalidad de brindar contenidos significativos, independientes de la tecnología de moda y un lenguaje informático adecuado que le permitan al estudiante comprender la importancia de la Informática en la administración productiva de las Organizaciones.

4. REFERENCIAS

- [1] Pimentel Jesús, García Omar, González Rocío, López Getsemaní. (2012) "*Software para el aprendizaje de algoritmos estructurados*", Revista Iberoamericana de Educación en Tecnología y Tecnología en Educación. Pág. 23-33. ISSN: 1850-9959
- [2] P. Novara. (2012). "*PSeInt*". Disponible en: <http://pseint.sourceforge.net/>
- [3] Molero María, Salvador Adela. (2002). "*Resolución de problemas. Estrategias Heurísticas*". Disponible en: <http://www2.camino.upm.es/Departamentos/matematicas/Fdistancia/PIE/Problemas/Estrategiasheurísticas.pdf>. Fecha de última consulta: 01/05/2015.
- [4] V. Frittelli. (2001). "*Algoritmos y Estructuras de Datos*". Córdoba: Universitat.
- [5] Carrizo Blanca R, Corso Cynthia L. (2010). "*Introducción al Diseño y Lógica de Programación para Ingenierías Industrial y Mecánica*". ISBN: 978-987-27648-0-7. Editorial Carola. 1er. Ed. Córdoba. Argentina

- [6] López García, Juan Carlos. (2009/2010). *“Algoritmos y Programación. Guía para Docentes.”* 2da. Edición, <http://www.eduteka.org.ar/GuíaAlgoritmos.php>
- [7] G. Polya. (2004). *“How to Solve It”*. Princeton Science Library.
- [8] G. Levine. (1989). *“Introducción a la computación y programación estructurada”*. McGraw-Hill.