

FUNDAMENTOS DE INFORMÁTICA PARA ALUMNOS DEL 1º NIVEL, APLICANDO SOFTWARE LIBRE

Blanca R. Carrizo^{*1}, Cynthia L. Corso², Jorge E. Abet³
Gustavo I. González⁴, Adriana E. Olmedo⁵

^{*1} Dpto. Ingeniería Mecánica, Industrial y Sistemas – Universidad Tecnológica Nacional
Esq. Maestro López s/n, Córdoba, Argentina - e-mail: bcarrizo@tecnicatura.frc.edu.ar

² Dpto. Ingeniería Industrial y Sistemas – Universidad Tecnológica Nacional

Esq. Maestro López s/n, Córdoba, Argentina - e-mail: cynthia@bbs.frc.utn.edu.ar

³ Dpto. Ingeniería Mecánica e Industrial – Universidad Tecnológica Nacional

Esq. Maestro López s/n, Córdoba, Argentina - e-mail: jabet@extension.frc.unt.edu.ar

⁴ Dpto. Ingeniería Mecánica – Universidad Tecnológica Nacional.

Esq. Maestro López s/n, Córdoba, Argentina - e-mail: gonzalez@iua.edu.ar

⁵ Dpto. Ingeniería Mecánica y Sistemas – Universidad Tecnológica Nacional

Esq. Maestro López s/n, Córdoba, Argentina - e-mail: aolmedo@tecnicatura.frc.utn.edu.ar

RESUMEN

Este trabajo que forma parte de un Proyecto de Investigación perteneciente al Dpto. Ing. Mecánica, está enmarcado en la asignatura “Fundamentos de Informática”, cuyos contenidos mínimos responden a la Ordenanza N° 1027 que rige el Diseño Curricular de la carrera. Su objetivo es motivar un abordaje holístico a problemas de ingeniería a través de la identificación y análisis de distintos criterios para la resolución de los mismos, desarrollando habilidades y destrezas destinadas a orientar la resolución automática mediante el uso de la computadora; donde el rol principal del alumno es involucrarse como usuario. El Programa Analítico de la asignatura, en su Unidad N° 2 “Introducción al Diseño de Algoritmo y Lógica de Programación”, comprende la resolución de problemas en computadora aplicando el Software Libre “Octave”. La metodología aplicada en el desarrollo de este tema consistió en exposiciones teóricas acompañadas de mapas conceptuales y Guías de Trabajos Prácticos con ejercicios modelos; donde cada uno de ellos tiene su representación gráfica (diagrama de flujo), *testing* (prueba de escritorio), traducción a un lenguaje natural (pseudocódigo escrito por el alumno) e implementación de los mismos a sentencias interpretadas y ejecutadas automáticamente por “Octave” (*testing* generado por este Software). El análisis de los resultados obtenidos (notas de parciales), permitió registrar una excelente retroalimentación por parte de los alumnos, quienes a través de un razonamiento lógico y aplicando la Teoría General de Sistemas, formularon alternativas de solución factibles y fomentaron su espíritu crítico a través del análisis y debate sobre la optimización de las mismas, competencia exigida para estos perfiles ingenieriles [1]. Se pretende facilitar el proceso de Enseñanza y Aprendizaje [2] a través de la incorporación de esta didáctica metodología no solo en el seno de la cátedra sino que actúe como elemento multiplicador en las asignaturas que la consideren aplicable y pertinente.

Palabras Claves: Informática – Diseño Curricular – Software Libre – Resolución Problemas.

1. INTRODUCCIÓN

La cátedra Fundamentos de Informática, asignatura perteneciente al 1er. Nivel de la carrera Ingeniería Mecánica, tiene como uno de sus objetivos brindar “nociones de programación” a los alumnos, conforme al Diseño Curricular de la carrera regido por la Ordenanza 1027; en la cual esta asignatura tuvo su origen.

En función de ello, y de los contenidos detallados en el Programa Sintético de esta Ordenanza, se incluyó en el Programa Analítico de esta asignatura, la Unidad Nro. II denominada “Introducción al Diseño de Algoritmos y Lógica de Programación” [1].

De las experiencias docentes relacionadas con esta temática, inferimos que esta Unidad es una de las que más dificultades de aprendizaje representa para los alumnos.

Es por ello que, la cátedra decidió implementar para este ciclo lectivo una estrategia de aprendizaje basada integración de los contenidos teóricos, con práctica en papel y desarrollo de destrezas en computadora mediante el uso de un Software que permita generar pseudocódigo, basado en el paradigma del Software Libre. El Software Libre seleccionado por la cátedra es “Octave” [2], el cual nos permite programar algoritmos.

Este trabajo propone cómo implementar una metodología didáctica e innovadora, que permita facilitar y afianzar el proceso de enseñanza y aprendizaje en los alumnos mediante la incorporación de Octave como herramienta de apoyo e integración.

2. SELECCIÓN DEL SOFTWARE

Cuando se analizó en el seno de la cátedra cómo motivar a los alumnos en esta temática, se planteó la necesidad de llevarlo a la práctica no sólo en papel sino en computadora, es decir automatizar el proceso a fin de poder hacer un paralelo con el Sistema Productivo; donde lo más importante son las Salidas (Productos) y que, en base a ello analizamos qué entradas son requeridas (Materias Primas), y luego planteamos cuál/es son el/los proceso/s necesario/s para obtener esas Salidas.

Basados en la premisa de que lo ideal es automatizar todo proceso a fin de optimizar los tiempos y mejorar la relación costo/beneficio, decidimos encarar la búsqueda de un Software que nos permitiera automatizar este proceso de generación de pseudocódigo.

Investigamos qué experiencias similares o diametralmente opuestas se implementaban en otras especialidades o bien en otras carreras de otras Universidades y pudimos comprobar que el abanico de posibilidades no era tan amplio y que la mayoría se inclinaba por el uso de Software Propietario, basado en estándares de cálculo de uso generalizado en todas las ramas de la Ciencia e Ingeniería como el MatLab o MatCad.

Con la finalidad de facilitar el acceso al Software a toda la comunidad educativa, decidimos elegir uno basado en la filosofía de Software Libre, como lo es Octave, que es un software de distribución gratuita que responde a la filosofía GNU, el cual facilita el acceso al programa y al código fuente del programa para modificarlo si nos interesara. [3,4]

3. METODOLOGIA

La metodología aplicada, y que previamente fue analizada en forma detallada por la cátedra, está basada en las experiencias propias y ya implementadas en otras asignaturas que requieren el uso de computadoras para el desarrollo de sus clases en laboratorios, donde surge un nuevo actor en la relación unidireccional Docente / Alumno, que es la computadora y quien se transforma en la tercera involucrada en este proceso de enseñanza y aprendizaje. [5,6,7]

Es necesario destacar que, es fundamental el desarrollo de un teórico claro y conciso, tendiente a fundamentar la importancia de diseñar y construir estrategias orientadas a la resolución de problemas, de distinto origen, bajo la óptica del enfoque de sistémico.

Esta metodología de resolución de problemas se basa en las siguientes etapas secuenciales a seguir: [8][9]

- a. Lectura inicial y análisis a priori de la situación planteada en el problema.
- b. Análisis detallado del problema, basado en el Enfoque Sistémico que delimita el problema y lo divide en subproblemas más simples para su comprensión (Diseño Top Down).
En este paso, es necesario destacar que si el problema se segmenta en subproblemas para su análisis, las Salidas de un subproblema pueden ser las Entradas de otro, y así sucesivamente; ya que nunca debe perderse la visión del todo armónico (Concepto de Sistemas); respetando la siguiente secuencia de pasos a respetar.
 - Identificar las Salidas requeridas (qué Producto necesito / Reporte o Gráfico Estadístico pide un Supervisor)
 - En base a ello, analizar qué Entradas (Materias Primas) necesito para poder obtener ese Producto y si las tengo.
 - Analizar cuál es el Proceso a implementar para transformar esas Entradas requeridas en las Salidas deseadas o esperadas. Aquí analizo cuál de las Estructuras de Control se adapta mejor al enunciado planteado, el cual optimiza el mismo.
 - Enumerar de la secuencia ordenada de pasos, para resolver la/s consigna/s solicitada/s y definir la/s Variable/s Auxiliar/es a utilizar en este problema, [9] entendiendo por variable una unidad de almacenamiento identificada por un nombre y un tipo de dato, capaz de almacenar información según el tipo de dato especificado.
- c. Graficar el Diagrama de Flujo o Algoritmo, que es la representación gráfica de la resolución del problema planteado, a través de la implementación del diseño Top Down (de lo general a lo particular), previamente analizado en la etapa anterior.
- d. Comprobar que el Algoritmo funciona para una muestra pequeña y representativa de la población, lo cual implica seleccionar aquellas entradas de datos que pueden presentarse; mediante el uso de la herramienta Prueba de Escritorio (matriz $n \times n$ variables que facilita la ratificación o rectificación de la lógica implementada) y es una actividad aúlica.

- e. Codificar el Algoritmo a un lenguaje natural denominado Pseudocódigo, como una instancia previa y de consolidación de la lógica empleada, tendiente a traducir cada acción del Diagrama de Flujo planteado a instrucciones. (Actividad Aúlica)
- f. Codificar el Algoritmo en Octave y ejecutarlo, llevando el Algoritmo y Pseudocódigo realizado en Aula; verificando que el mismo arroja los mismos resultados que la prueba de escritorio realizada en forma manual.

Cabe aclarar que, se va a Laboratorio una vez que los alumnos han realizado la Guía de Trabajos Prácticos publicada, tanto a nivel algoritmos (gráfica) como pseudocódigo (descriptiva); ya que el objetivo es comprobar que los ejercicios realizados son correctos y que automatizar el proceso agiliza los tiempos de respuesta; pero que lo trascendente es desarrollar una lógica orientada a la resolución de problemas de distinto origen.

De la implementación de esta estrategia educativa, surge la posibilidad de complementar las prácticas realizadas en el aula con las prácticas desarrolladas laboratorio mediante el uso del Octave.

4. CASO PRÁCTICO

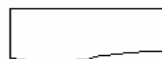
El ejercicio propuesto, es un caso práctico tomado de la realidad laboral y que todos conocemos, y forma parte de nuestra labor habitual.

Previamente, indicamos la simbología a emplear en la graficación de los Algoritmos, a través de los Diagramas de Flujo, donde: [9]

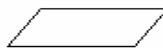
- Inicio / Fin: Representa el comienzo “inicio” y el final “fin” del algoritmo. Se representa por el símbolo de terminal.



- Salidas se representan con:



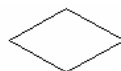
- Entradas se representan a través de:



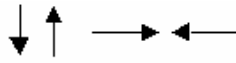
- Proceso: cualquier tipo de operación que pueda originar cambio de valor, operaciones aritméticas y de asignación.



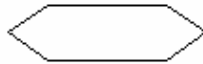
- Decisión: indica operaciones lógicas o de comparación entre datos y en función del resultado de la misma determina cuál de los caminos alternativos del algoritmo se debe seguir, normalmente tiene dos salidas. Y se representa:



- Líneas de dirección: indicador de dirección o línea de flujo, indica el sentido de ejecución de las operaciones.



- Subrutina: indica el comienzo de una subrutina o función.



Es importante destacar que los algoritmos diseñados y propuestos por los alumnos, para la resolución de los problemas, los construirán bajo el paradigma estructurado.

Es decir que, se plantean las soluciones de manera secuencial, no permitiendo el uso de instrucciones de transferencia incondicional (go to). [10]

En nuestro caso particular, partimos de un Algoritmo que se resuelve a través de la implementación de una Estructura Secuencial, luego se avanza a una instancia intermedia con Estructuras Alternativas y llegamos a una instancia de abstracción superior mediante la implementación de las Estructuras Repetitivas.

4.1. Estructura Secuencial.

4.1.1. Enunciado:

Se necesita calcular el Sueldo de un operario de una empresa metalúrgica, donde se cuenta en los siguientes datos.

- Cantidad de horas trabajadas.
- Precio por hora trabajada.

Se pide que muestre el Sueldo por pantalla.

4.1.2. Análisis del problema. (a y b)

Salida/s:

- Mostrar el sueldo empleado (por pantalla o impreso) – **Variable es Sueldo = 0**

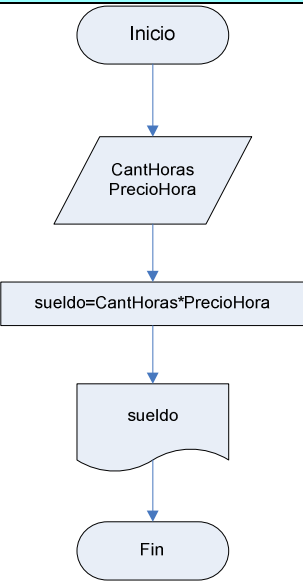
Entrada/s:

- Cantidad Horas Trabajadas. - **Variable es CantHoras = 0**
- Precio Hora. – **Variable es PrecioHora = 0**

Proceso:

- Leer los datos de entrada. (CantHoras y PrecioHora)
- Calcular el sueldo (Producto entre Cantidad Horas Trabajadas* Precio Hora)
(Sueldo=CantHoras*PrecioHora)

Tabla 1 Ejemplificación de la resolución del problema.

Diagrama de Flujo	Pseudocódigo	Octave
 <pre> graph TD Inicio([Inicio]) --> Input[/CantHoras PrecioHora/] Input --> Process[sueldo=CantHoras*PrecioHora] Process --> Output[sueldo] Output --> Fin([Fin]) </pre>	<p>algoritmo</p> <p>Inicio</p> <p>leer(cantHoras,PrecioHora);</p> <p>sueldo<- cantHoras*PrecioHora;</p> <p>escribir(sueldo);</p> <p>Fin</p>	<p>#inicio</p> <p>cantHoras=input("Ing. cant. de horas");</p> <p>precioHora=("Ingrese precio por hora");</p> <p>sueldo=cantHoras*precioHora</p> <p>disp(sueldo)</p> <p>#fin</p>

4.2. Estructura Alternativas

4.2.1. Enunciado:

Se necesita calcular el Sueldo de un operario de una empresa metalúrgica, donde se cuenta en los siguientes datos:

- Cantidad de horas trabajadas.
- Precio por hora trabajada.
- Turno de Trabajo: 1 - Diurno y 2 - Nocturno

Si el turno de trabajo es nocturno, incrementar al Sueldo un monto de \$50; de lo contrario el Sueldo no se modifica.

4.2.2. Análisis del problema. (a y b)

Salida/s:

- Mostrar el sueldo empleado (por pantalla o impreso). - **Variable es Sueldo = 0**

Entrada/s:

- Cantidad Horas Trabajadas. - **Variable es CantHoras = 0**
- Precio Hora. - **Variable es Precio Hora = 0**
- Turno – **Variable es Turno = 1 o 2**

Proceso:

- Leer los datos de entrada. (**Cantoras, Precio Hora, Turno**)

- Calcular el Sueldo (Producto entre Cantidad Horas Trabajadas* Precio Hora), discriminando si el operario trabaja en Turno Diurno o Nocturno, ya que en base a ello varía el Sueldo en cada caso. **(Sueldo=CantHoras*PrecioHora)**

Tabla 2 Ejemplificación de la resolución del problema.

Diagrama de Flujo	Pseudocódigo	Octave
<pre> graph TD Inicio([Inicio]) --> Input[/CantHoras PrecioHora Turno/] Input --> Decision{Turno=2} Decision -- Si --> Process1[sueldo=CantHoras*PrecioHora] Decision -- Sino --> Process2[sueldo=(CantHoras*PrecioHora)+50] Process1 --> Output[sueldo] Process2 --> Output Output --> Fin([Fin]) </pre>	<p>Algoritmo</p> <p>Inicio leer(cantHoras,PrecioHora, turno); Si(turno=2) sueldo<- (cantHoras*PrecioHora)+50; sino sueldo<- cantHoras*PrecioHora; Fin_si escribir(sueldo); Fin</p>	<p>#inicio cantHoras=input("Ing. cant. de horas:"); precioHora=("Ingrese precio por hora:"); turno=("Ingrese turno:"); if(turno=2) sueldo=(cantHoras*Pr ecioHora)+50 else sueldo=(cantHoras*Pr ecioHora) endif disp(sueldo) #fin</p>

4.3. Estructura Repetitivas.

4.3.1. Enunciado:

Se necesita calcular el Sueldo de 20 operarios de una empresa metalúrgica, donde se cuenta en los siguientes datos:

- Cantidad de horas trabajadas.
- Precio por hora trabajada.
- Turno de Trabajo: 1 - Diurno y 2 - Nocturno)

Si el turno de trabajo es nocturno, incrementar al Sueldo un monto de \$50; de lo contrario el Sueldo no se modifica.

4.3.2. Análisis del problema. (a y b)

Salida/s:

- Mostrar el sueldo empleado (por pantalla o impreso) – **Variable Sueldo = 0**

Entrada/s:

- Cantidad Horas Trabajadas. **Variable es CantHoras = 0**
- Precio Hora. - **Variable es Precio Hora = 0**

- Turno. - **Variable es Turno = 0**

Proceso:

- Leer los datos de entrada. (**CantHoras, Precio Hora, Turno**)
- Calcular el sueldo (Producto entre Cantidad Horas Trabajadas* Precio Hora), discriminando si el operario trabaja en Turno Diurno o Nocturno, ya que en base a ello varia el Sueldo en cada caso. (**Sueldo=CantHoras*PrecioHora**)

Tabla 3 Ejemplificación de la resolución del problema.

Diagrama de Flujo	Pseudocódigo	Octave
	<p>Algoritmo</p> <p>Inicio Para contador 1 a 20 paso 1 hacer leer(cantHoras,PrecioHora, turno); Si(turno=2) sueldo<-(cantHoras*PrecioHora)+50; sino sueldo<-cantHoras*PrecioHora; Fin_si escribir(sueldo); Fin_para Fin</p>	<p>#inicio for contador=1:20 cantHoras=input("Ing. cant. de horas:"); precioHora=("Ingrese precio por hora:"); turno=("Ingrese turno:"); if(turno=2) sueldo=(cantHoras*precioHora)+50 else sueldo=(cantHoras*precioHora) endif disp(sueldo) endfor #fin</p>

5. CONCLUSIONES

Una de las actividades profesionales reservadas al Ingeniero Mecánico identifica, como una de los elementos principales que hacen a la concepción de su formación integral, el desarrollo de capacidad de análisis y acción a partir del conocimiento profundo de los problemas de Ingeniería y Tecnología, tanto disponible como concebible.

Esto implica la construcción del conocimiento de forma integrada a la faz práctica; de tal forma que el estudiante se acerque y se forme a través de tareas de observación e interpretación de problemas reales.

Nuestra asignatura contribuye a la formación del perfil mediante la implementación de una didáctica y novedosa estrategia pedagógica que facilita el proceso de enseñanza y aprendizaje en lo referido a la resolución de problemas, mediante el diseño de algoritmos y lógica de programación.

Es necesario destacar que, esta estrategia de resolución de problemas es mucho más rica que la aplicación mecánica de un algoritmo, pues implica crear un contexto donde los datos guarden significatividad. Desde este análisis, se han de establecer jerarquías: ver qué datos son prioritarios, rechazar los elementos distorsionantes, escoger las operaciones que los relacionan, estimar el rango de la respuesta, etc.; en base a ello representar gráficamente el problema a través de un algoritmo; y verificar que el mismo es correcto mediante dos herramientas:

- Manual: Prueba de Escritorio, que se representa a través de una matriz, que se rotula en las columnas con el nombre de los datos de entrada y tantas filas como muestras de datos se analicen, y donde el registro de estas pruebas se hace en papel.
- Automática: Software Libre Octave, que es un software que permite ingresar los datos en su entorno, previamente definido, y obtener inmediatamente resultados deseados.

Es decir que, no es lo mismo hacer un ejercicio que resolver un problema. [6]

Una cosa es aplicar un algoritmo de forma más o menos mecánica, evitando las dificultades que introduce la aplicación de reglas cada vez más complejas, y otra, resolver un problema, dar una explicación coherente a un conjunto de datos relacionados dentro del contexto.

La respuesta suele ser única, pero la estrategia resolutoria está determinada por factores madurativos del alumno y criterios propios.

Nuestro objetivo es reutilizar esta estrategia implementada en la resolución de problemas de la Unidad N° 2 en otras Unidades que requieren desarrollar otras destrezas a nivel uso de Software, pero donde la metodología planteada permite resolver en forma lógica y objetiva es similar; y desatacamos la importancia de formar profesionales con juicio crítico, basados en un análisis y razonamientos evolutivos.

6. REFERENCIAS

- [1] Programa Analítico de la Asignatura: [uv.frc.utn.edu.ar/fundamentos de informática/](http://uv.frc.utn.edu.ar/fundamentos%20de%20informática/)
- [2] <http://www.octave.org>
- [3] Jesús M. González Barahona, Software Libre en la Enseñanza Informática, 2002.
- [4] Israel Herraiz, Software Libre: una oportunidad para todos, 2005.
- [5] Eduardo Cortés, Vanoli Verónica y Casas Sandra, *WorkShop de Investigadores en Ciencias de la Computación, "Bing Bang: un recurso didáctico-pedagógico en el aprendizaje de la implementación de algoritmos y pseudocódigo"*, 2006.
- [6] Material confeccionado por docentes de la cátedra Matemática Discreta. Universidad Tecnológica Nacional Facultad Regional, Carrera: Ingeniería en Sistemas de Información, *"Introducción a la Programación"*, 2003.
- [7] Joyce, B., Weil. M, *Modelos de Enseñanza. "Los organizadores previos: mejorar la eficacia del estudio y de otros modos de presentación de información"*, Anaya, 1985.
- [8] Elizabeth Jiménez Rey, Gregorio Perichinsky, *WorkShop de Investigadores en Ciencias de la Computación, "El Mapa Conceptual como Representación del Modelo de Polya para la Creación de programas"*, 2006.
- [9] Ing. Valerio Fritelli, *"ALGORITMOS y Estructuras de Datos"*, Científica Universitaria, Córdoba, 2001.
- [10] Material confeccionado por docentes de la cátedra Algoritmos y Estructuras de Datos. Universidad Tecnológica Nacional. Facultad Regional. Córdoba, Carrera: Ingeniería en Sistemas de Información, *"Algoritmos y Estructuras de Datos. Guía de Trabajos Prácticos"*, 1997.