

Tratamiento metodológico a la integración de los operadores lógicos y relacionales con expresiones lógicas en la estructura selectiva doble

Methodological treatment to the integration of logical and relational operators with logical expressions in the double selective structure

Pedro José Serrano Yero¹ (pedrosy@ult.edu.cu) (<https://orcid.org/0000-0002-1358-2746>)

Essenia Cruz Alfonso² (essenia@ult.edu.cu) (<https://orcid.org/0000-0001-9540-0194>)

Yunior Rodríguez Rodríguez³ (yuniorr@ult.edu.cu) (<https://orcid.org/0000-0002-0729-4000>)

Resumen

En la formación inicial del Licenciado en Educación Informática es importante estimular, desde la resolución de problemas, formas de pensamiento que conduzcan al logro de aprendizajes significativos, autorregulados y con alto índice de motivación. Todo ello para contribuir a la resolución de problemas de diversos contextos, al aplicar las metodologías relacionadas con las técnicas de programación que se estudian, a partir de los recursos que brindan los diferentes paradigmas de programación. Sin embargo, en la práctica pedagógica se pueden percibir carencias en el tratamiento metodológico en algunos contenidos de la asignatura Fundamentos de Programación, específicamente en la integración de los operadores lógicos y relacionales con expresiones lógicas para la construcción de estructuras selectivas dobles en la resolución de problemas donde se utilicen dichas estructuras. Esta insuficiencia debe ser atendida desde una alternativa metodológica que concatene coherentemente los contenidos implicados para la formación del conocimiento. Para la realización de este trabajo se utilizaron métodos de investigación pedagógica tales como: el análisis documental, analítico-sintético, el inductivo-deductivo, la observación y la prueba pedagógica. Estos propiciaron la identificación del problema metodológico y ayudaron a concebir la propuesta del proceder metodológico que debe aplicarse por el docente universitario para su accionar.

Palabras claves: pensamiento lógico, expresiones lógicas, estructuras selectivas dobles, operadores lógicos y relacionales, resolución de problemas.

¹ Máster en Nuevas Tecnologías para la Educación. Profesor Auxiliar. Facultad de Ciencias Técnicas y Agropecuarias, Departamento de Informática. Universidad de Las Tunas. Cuba.

² Máster en Nuevas Tecnologías para la Educación. Profesora Auxiliar. Facultad de Ciencias Técnicas y Agropecuarias, Departamento de Informática. Universidad de Las Tunas. Cuba.

³ Máster en Nuevas Tecnologías para la Educación. Profesor Auxiliar. Facultad de Ciencias Técnicas y Agropecuarias, Departamento de Informática. Universidad de Las Tunas. Cuba.

Abstract

In the initial formation of the Graduate in Computer Science Education it is important to stimulate, from the resolution of problems, ways of thinking that lead to the achievement of significant learning, self-regulated and with high index of motivation. All this to contribute to the resolution of problems of diverse contexts, when applying the methodologies related to the programming techniques that are studied, from the resources that offer the different programming paradigms. However, in the pedagogical practice it is possible to perceive deficiencies in the methodological treatment of some contents of the subject Programming Fundamentals, specifically in the integration of logical and relational operators with logical expressions for the construction of double selective structures in the resolution of problems where such structures are used. This insufficiency must be addressed from a methodological alternative that coherently concatenates the contents involved for the formation of knowledge. In order to carry out this work, pedagogical research methods were used, such as: documentary, analytical-synthetic, inductive-deductive, observation and pedagogical test analysis. These methods helped to identify the methodological problem and helped to conceive the proposal of the methodological procedure that should be applied by university teachers for their actions.

Key words: logical thinking, logical expressions, double selective structures, logical and relational operators, problem solving.

La formación inicial del Licenciado en Educación Informática tiene como uno de sus objetivos esenciales el logro de la dirección del proceso de enseñanza - aprendizaje con enfoque integrador que propicie la resolución de problemas de diferentes contextos como contribución al desarrollo del pensamiento lógico y heurístico, de valores morales, así como la formación del pensamiento computacional⁴ y la toma de decisiones.

La formación, desde la programación, repercute en la preparación para dirigir el proceso de enseñanza-aprendizaje del futuro egresado. Se manifiesta en su independencia cognoscitiva, organización, creatividad y autocontrol en el empleo de las técnicas de programación y de las metodologías inherentes a los diferentes paradigmas y su aplicación en la solución de problemas de variados contextos socioeconómicos. Además, se contribuye al establecimiento de relaciones de respeto y modestia consigo mismo, con los demás y en la solución de problemas mediante la utilización de un lenguaje de programación, en expresar las ideas con criterios propios, en reconocer el trabajo de los demás, en el ejercicio de la crítica y autocrítica.

⁴Consiste en la resolución de problemas, el diseño de los sistemas, y la comprensión de la conducta humana haciendo uso de los conceptos fundamentales de la informática. Es útil para todo el mundo, no sólo para los científicos de la computación (Jeannette Wing. "Computational Thinking. It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use". March, 2006. Vicepresidente corporativo de Microsoft Research y profesora de Computer Science Department Carnegie Mellon University)

Se propicia el conocimiento y los modos de actuación necesarios para mantener una adecuada disciplina durante la programación, en el uso y cuidado de los recursos informáticos en función de la solución de problemas, así como en el cumplimiento de las tareas y la actitud de no abandonarlas hasta encontrar su solución, poniendo a prueba su perseverancia y decisión.

Todo lo anterior contribuye a que los estudiantes sean capaces de resolver problemas de diversos contextos en términos de describir un algoritmo de solución o diseñar interfaces para la interacción hombre-máquina utilizando un lenguaje de programación. Esto desarrolla la iniciativa, el pensamiento computacional, las capacidades intelectuales generales, hábitos del proceder intelectual mediante el estudio de los métodos y procedimientos propios de la informática.

Es necesario destacar las siguientes insuficiencias en el orden metodológico:

- Desde la preparación de la asignatura no se concibe una estrategia que concatene coherentemente los contenidos implicados para la formación de estructuras selectivas.
- Los contenidos referentes a operadores lógicos, relacionales y expresiones lógicas se trabajan de manera aislada a la formación de estructuras selectivas.
- No se propicia la resolución de problemas relacionados con la construcción de expresiones lógicas para su utilización en las estructuras selectivas.

Estas insuficiencias propiciaron la investigación acerca del tema y la elaboración de una propuesta de alternativa metodológica que contribuya a erradicarlas. Es propósito de este artículo dar muestras de ello.

Tratamiento metodológico necesario en Fundamentos de Programación

La comprensión de las estructuras básicas de la programación es esencial para poder aplicarlas en la resolución de problemas. Estas estructuras se relacionan entre sí y con estas relaciones se logra la efectiva resolución de problemas de diferentes ámbitos a través de la programación. Es por ello que la integración de estos contenidos es de vital importancia a la hora de impartirlos para que los estudiantes puedan comprender mejor el complejo proceso de creación de algoritmos y de programar.

Dos investigadores italianos, Böhm y Jacopini (1966, p. 366), plantearon el teorema del programa estructurado, el cual expone que: cualquier *algoritmo propio* puede ser escrito utilizando únicamente tres tipos de estructuras:

1. Secuencia: ejecución de una instrucción tras otra.
2. Selección: ejecución de una instrucción (o conjunto de instrucciones), según el valor de una variable o expresión booleana (lógica).
3. Iteración: ejecución de una instrucción (o conjunto de instrucciones) mientras una variable.

De ahí la importancia de saber caracterizar y utilizar de manera adecuada y eficiente cada una de las estructuras, en cuanto a su sintaxis y función dentro de la programación, entre las que se encuentran las selectivas.

Para ello es necesario fortalecer, en el colectivo de profesores de la disciplina Lenguajes y Técnicas de Programación, el tratamiento metodológico a la integración de los operadores lógicos y relacionales con expresiones lógicas para la construcción de estructuras selectivas dobles, en la asignatura Fundamentos de Programación.

En este trabajo nos apoyaremos en la contextualización de los elementos heurísticos a la programación planteada por Aguilosocho y Crespo (2000) para la elaboración de la alternativa metodológica y ejemplificaremos a través de una clase cómo implementar dicha propuesta.

Procedimientos heurísticos:

Principios: 1. Analogía. 2. Reducción. 3. Generalización.

Reglas: Indicaciones generalmente en forma de preguntas para guiar el pensamiento (preguntas heurísticas). Permiten movilizar el pensamiento hacia la búsqueda del conocimiento.

Estrategias: 1. Trabajo hacia adelante. 2. Trabajo hacia atrás. 3. Descomposición del problema. (Permite mayor comprensión del problema)

Medios Auxiliares Heurísticos:

1. Figuras ilustrativas, esbozo o figuras de análisis.

2. Tablas que reflejan relación entre datos.

3. Compendios de definiciones y conceptos fundamentales.

4. Diagramas de flujo que ilustren el problema.

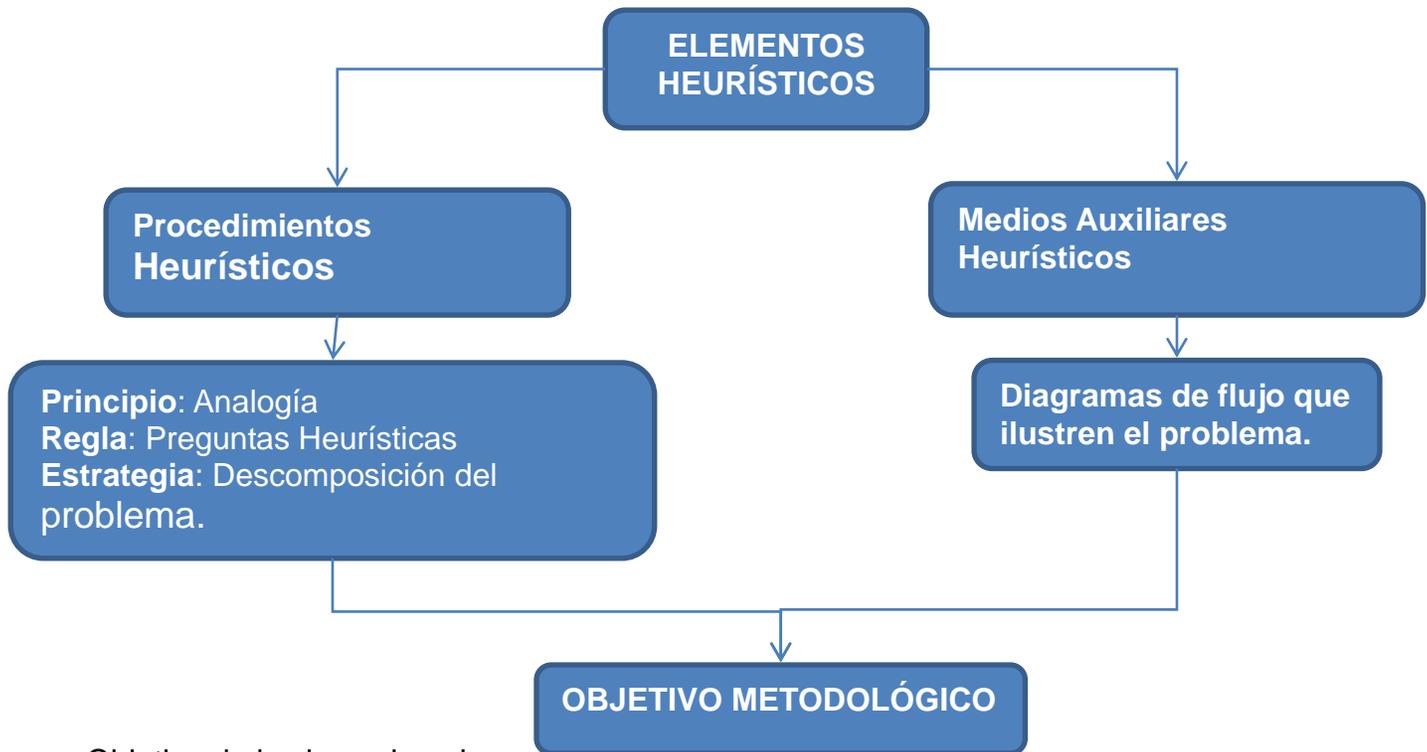
- Son una herramienta metacognitiva que facilitan la comprensión y reflexión sobre las acciones del procedimiento objeto de aprendizaje, especialmente cuando este se caracteriza por una estructura o secuencia compleja (Martí y Pozo, 2000^a, p. 8)

- Se propone como una herramienta de representación externa del conocimiento declarativo sobre procedimientos algorítmicos y heurísticos propios del área de la tecnología (Carrera, 2002).

5. Ayudas incorporadas al sistema.

6. Algoritmos básicos.

7. Bibliotecas de procedimientos y funciones, ya sean profesionales o elaboradas por el estudiante.



Objetivo de la clase ejemplo:

Caracterizar la estructura selectiva doble *Si – Entonces – SiNo* en cuanto a su sintaxis y función dentro de la programación estructurada a un nivel reproductivo como contribución a la formación del pensamiento computacional y la toma de decisiones.

Acciones de la habilidad caracterizar:

Analizar:

- Determinar los límites de la sintaxis de la estructura selectiva doble.
- Determinar los criterios de descomposición de la sintaxis de la estructura selectiva doble.
- Delimitar las partes de la sintaxis de la estructura selectiva doble. (*Si, expresión lógica, sentencias para el caso verdadero, SiNo, sentencias para el caso falso, FinSi*)
- Estudiar cada parte de la sintaxis de la estructura selectiva doble.
- Determinar el elemento esencial de la estructura selectiva (la expresión lógica).
- Comparar la estructura selectiva doble con la estructura selectiva simple.
- Seleccionar los elementos distintivos (*SiNo*) de la estructura selectiva doble.

Método: Elaboración conjunta.

Participan activamente en el proceso el profesor y el estudiante.

Se utiliza cuando:

- se quiere impartir nuevos conocimientos
- se quiere aprender a fundamentar
- se quiere que el alumno aprenda a resolver ejercicios independientemente.

Actividad del profesor:

- Dirige la actividad.
- Formular las preguntas con claridad y precisión, sin adelantar el núcleo de la respuesta o acompañadas de gestos que sugieran esta.
- No limitar las iniciativas de los estudiantes, en cada una de las acciones que este debe realizar.
- Ofrecer la posibilidad de que la mayoría de los estudiantes participen.
- Valorar rápidamente la calidad de la respuesta, analizando el núcleo positivo de la misma, invitando a otros estudiantes a completarlas

Actividad del estudiante:

- Participa de forma activa, en esta conversación, pregunta, emite criterios, realiza valoraciones y comparaciones
- Descubre los nuevos conocimientos por sí mismo.
- Aprende y comprende el nuevo contenido de manera independientemente.
- Aprende a valorar y respetar los criterios de los demás.
- Se percata de sus propios errores o aciertos.

Procedimiento: Conversación heurística.

Se caracteriza por dirigir el pensamiento de los estudiantes para que descubran, por sí mismos, determinados conocimientos.

El profesor debe:

- tener claro los objetivos,
- dominio del contenido,
- buena técnica para preguntar,
- dirigir el pensamiento de los estudiantes para que descubran por sí mismos determinados conocimientos.

Medios de enseñanza que se proponen para trabajar el tema

- Analizar y seleccionar el/los softwares para utilizar en el tema a tratar. Editores de presentaciones electrónicas o de animaciones PowerPoint, Impress, Prezzi, Macromedia Flash, etc. Herramientas de creación de algoritmos en pseudocódigo PSeInt, MarugaScript, etc.

- Emplear medios que estén disponibles en las esferas de actuación del profesional. Pizarra, cartulinas, etc.

PowerPoint

Para la presentación de las diapositivas donde se muestren conceptos, diagramas de flujos, mapas mentales, esquemas, ejemplo de algoritmos, etc.

- Está instalado y accesible en todas las PC de los laboratorios.

- Su formato es compatible con las aplicaciones móviles para estos fines.

Pizarra.

Para reflejar en esta, aquellos elementos que resultan esenciales en el contenido y como apoyo a la ilustración, ejemplificación, argumentación y explicación de aspectos específicos que son de difícil comprensión por parte de los estudiantes.

- Están presentes y accesible en todos los laboratorios.

Es esencial saber combinar su utilización con los demás medios de enseñanza seleccionados para una mejor comprensión del contenido.

PSeInt (Se sugiere la utilización de este software, Novara, 2003-2019, zaskar_84<aroba>yahoo.com.ar. Universidad Nacional del Litoral. Argentina.

Es un software muy popular de su tipo y es ampliamente utilizado en universidades de Latinoamérica y España. Para la demostración práctica como resultado de los contenidos, expuestos en las diapositivas y explicados en la pizarra, en la solución de problemas donde serán utilizados los conocimientos resultantes de todo ello.

- Permite la ejecución paso a paso con explicación textual de cada secuencia.

- Permite la creación y exportación de diagramas de flujos.

- Permite la exportación del pseudocódigo a diferentes lenguajes como el C++ que es el lenguaje a utilizar en la asignatura LTP I en 3er año.

- Cuenta con una versión para móviles.

Aseguramiento del nivel de partida (ANP)

1. Determinar con precisión el contenido y la extensión de los conocimientos y habilidades necesarias que debe poseer el estudiante.

- Operadores relacionales: >, <, >=, <=, =, <>
- Operadores lógicos: y, o, no

- Tabla de valores de verdad
 - Estructura selectiva simple
- Identificar los diferentes tipos de operadores.
 - Determinar la relación entre los operadores relacionales y lógicos.
 - Comparar los operadores lógicos según su función.
 - Caracterizar la estructura selectiva simple.
2. Controlar a través de preguntas heurísticas si los estudiantes disponen de los conocimientos y habilidades necesarias para enfrentar el nuevo contenido.
- ¿Cuáles son los operadores relacionales?
 - ¿Cuáles son los operadores lógicos?
 - ¿Cuál es la diferencia entre los operadores lógicos (y, o)? Ejemplifique
 - ¿Qué nombre reciben las expresiones donde se utilizan estos tipos de operadores y/o la combinación de ellos?
 - ¿Cuál es la sintaxis de la estructura selectiva simple?

Revisión del estudio independiente: acciones metodológicas

1. Estimular el protagonismo de los estudiantes en la propuesta de solución al estudio independiente.
2. Propiciar el debate y estimular a los estudiantes para que emitan sus criterios libremente sin temor a las respuestas no acertadas.
3. Resaltar la importancia de la investigación y la autopreparación en la búsqueda del conocimiento.

Estudio independiente:

Ejemplo: Realice un algoritmo que permita dar solución al siguiente problema:

De una persona se conoce el sexo y la edad. Realice un algoritmo que muestre el siguiente mensaje: "La persona es del sexo femenino y es mayor de edad", si la persona tiene 18 años o más y es del sexo femenino, en caso contrario, mostrar el mensaje: "La persona es del sexo masculino o es menor de edad".

Motivación: acciones metodológicas

- Mediante ella se debe despertar el interés de los estudiantes hacia el contenido que van a recibir, a ser conscientes de sus necesidades de aprendizaje y de cómo el nuevo contenido les ayudará a resolverlo.
- Se debe mantener la motivación durante toda la clase ayudando a los estudiantes a ser conscientes del avance que realizan y evitando el aburrimiento.

- Utilizar como motivación fundamental el reto de dar solución al problema.

Orientación hacia el objetivo (O.H.O).

- Es la orientación anticipada que deben recibir los estudiantes sobre los resultados que han de lograr con su actividad.
- Despertar interés, curiosidad por lo desconocido, ansias de saber.

- Con los conocimientos que hasta hoy poseen, no es posible resolver el problema planteado.

- Vamos a estudiar una nueva estructura de control, en este caso la *estructura selectiva doble*.

Bibliografía: acciones metodológicas

1. La bibliografía que se orienta debe ser accesible, variada, actualizada y suficiente.
2. Se debe hacer énfasis en el trabajo con diferentes fuentes bibliográficas para obtener un conocimiento más acabado.
3. Contribuir al desarrollo de la capacidad de análisis, síntesis y el pensamiento crítico.
4. En el caso específico de este contenido se debe aclarar que en las fuentes bibliográficas se trabajan indistintamente los términos: estructuras de decisión, alternativas o selectivas.
5. En la bibliografía en inglés la estructura selectiva doble aparece como if – then – else o if – else.

En la formación inicial del Licenciado en Educación Informática es importante estimular, desde la resolución de problemas, formas de pensamiento que conduzcan al logro de aprendizajes significativos, autorregulados y con alto índice de motivación. Todo ello para contribuir a la resolución de problemas de diversos contextos, al aplicar las metodologías relacionadas con las técnicas de programación que se estudian, a partir de los recursos que brindan los diferentes paradigmas de programación.

En función de ello, se realizó un análisis metodológico para que en la práctica pedagógica se lograra mayor efectividad en la integración de los operadores lógicos y relacionales con expresiones lógicas para la construcción de estructuras selectivas dobles en la resolución de problemas donde se utilicen dichas estructuras, a partir del proceder metodológico que se propone.

Referencias

- Böhm, C. y Jacopini, G. (1966). Flow diagrams, Turing Machines and Languages with only Two Formation Rules. *Comm. of the ACM*, 9(5), 366-371.
- Carrera, X. (2002). *Uso de diagramas de flujo y sus efectos en la enseñanza-aprendizaje de contenidos procedimentales. Área de Tecnología* (tesis doctoral). Universidad de Lleida (Lérida) dpto. Psicología pedagógica. España.

-
- Cortés, O. y Gutiérrez, J. (2017). *Estructuras de decisión, selectivas o condicionales* (tesis doctoral). Institución Universitaria "Pascual Bravo". Colombia.
- Gómez, J. J. (2015). *Fundamentos de Programación*. CD de la carrera 2015-2016. La Habana: Ministerio de Educación Superior.
- Hernández, L. (2014). *Apuntes de clase de la asignatura Fundamentos de la Programación*. España: Universidad Complutense.
- Joyanes, L. Rodríguez, L. y Fernández, M. (1996). *Fundamentos de Programación. Libro de problemas*. España: McGraw Hill /Interamericana.
- Martí, E. y Pozo, J. I. (2000). *Más allá de las representaciones mentales: la adquisición de los sistemas externos de representación*. Recuperado de <http://doi.org/10.1174/021037000760087946>