MAGAZINE DE LAS CIENCIAS:
REVISTA DE INVESTIGACIÓN E INNOVACIÓN
https://https://revistas.utb.edu.ec/index.php/magazine

e-ISSN: 2528-8091
enero-marzo 2020
Vol. 5, No. 5, PP.70-89

Sistema básico de habilidades para algoritmizar durante la programación computacional

Basic system of skills to algorithmize during computer programming

Isabel Alonso Berenguer[1]; Antonio Salgado Castillo[2]; Alina Blanco Hamad[3]
{ialonso@uo.edu.cu}
Fecha de recepción: 23 de marzo de 2020 − **Fecha de aceptación:** 2 de mayo de 2020

Resumen: En el trabajo se propone un Sistema Básico de Habilidades para la Algoritmización Computacional, el que favorece el perfeccionamiento del proceso de enseñanza-aprendizaje de las asignaturas de programación. Este sistema fue creado mediante el Método Sistémico Estructural Funcional y tuvo por objetivo la orientación a los profesores sobre la formación de habilidades requeridas para la elaboración de algoritmos computacionales. El mismo aporta la definición de cada una de ellas, explica la importancia de su formación, las ejemplifica, precisa su relación con las otras habilidades del sistema y brinda orientaciones metodológicas para que los profesores las introduzcan en su docencia. La intención es que se comience trabajando con los estudiantes en la formación de habilidades de menor grado de complejidad, las que, al ser aplicadas e interrelacionadas, darán lugar al desarrollo de otras más complejas, hasta que sean capaces de lograr algoritmos computacionales eficaces. La viabilidad y pertinencia del sistema fueron analizadas mediante talleres de socialización con especialistas, concluyéndose que el mismo es una herramienta provechosa para orientar el proceso de formación de la algoritmización computacional.

Palabras clave − *Programación, habilidades, sistema básico, algoritmización, computacional.*

Abstract: The work proposes a Basic System of Abilities for Computational Algorithmization, which favors the improvement of the teaching-learning process of the programming subjects. This system was created by means of the Functional-Structural Systemic Method and had as objective, the orientation to the professors on the formation of abilities required for the elaboration of computational algorithms. It provides the definition of each of them, explains the importance of their training, exemplifies them, specifies their relationship with the other skills of the system and provides methodological guidelines for teachers to introduce them in their teaching. The intention is to start working with students in the formation of abilities of lower complexity degree, which, when applied and interrelated, will lead to the development of more complex ones, until they are able to achieve effective computational algorithms. The viability and relevance of the system were analyzed through socialization workshops with specialists, concluding that it is a useful tool to guide the process of computational algorithm formation.

Keywords − *Programming, abilities, basic system, algorithmization, computational*.

[1]Universidad de Oriente.

[2]Universidad de Oriente.

[3]Universidad de Oriente.

Cómo citar: Alonso Berenguer, I., Salgado Castillo, A., & Blanco Hamad, A. (2020). Sistema básico de habilidades para algoritmizar durante la programación computacional. Magazine de las Ciencias: Revista De Investigación E Innovación, 5(5), 70-89. https://revistas.utb.edu.ec/index.php/magazine/article/view/725

INTRODUCCIÓN

En la actualidad las Tecnologías de la Información y las Comunicaciones (TIC) abarcan prácticamente todos los ámbitos de la sociedad e intervienen en importantes actividades de la misma, tales como el estudio, el trabajo y las actividades comerciales, entre otras de gran importancia para el hombre. Tan es así que se ha dicho que, al igual que el español o la aritmética, el dominio de las TIC será una habilidad de aplicación universal para todas las personas (Wing, 2006).

Muchas instituciones y administraciones a nivel mundial han reconocido la necesidad del estudio formal de las competencias computacionales desde los niveles de escolaridad primario y secundario. Existen países como Inglaterra que han incluido formalmente el estudio de la programación de ordenadores en los planes de estudios de la educación primaria y secundaria (Basogain, Olabe y Olabe, 2015).

De manera particular, la formación de los profesionales de las ciencias computacionales cobra cada día mayor relevancia; la que tiene como prioridad el desarrollo de habilidades lógicas, propias de esta ciencia y específicas de cada una de las disciplinas que forman sus currículos (Díaz y Crespo, 2013).

La satisfacción de la citada prioridad implica el aprendizaje de importantes contenidos computacionales como los que brinda la Programación, asignatura que aporta las bases para el diseño y construcción de programas (softwares), a partir de los cuales se puede dar solución a disímiles problemas que surgen en la sociedad. Consecuentemente, se espera que los estudiantes de estas ciencias se apropien de las principales herramientas y técnicas de la Programación, para que puedan diseñar, escribir e implementar programas informáticos; todo lo cual requerirá de algoritmos computacionales que sirvan de base para crear programas eficientes y pertinentes (Salgado, Alonso y Gorina, 2013).

Sin embargo, en la actualidad se confrontan insuficiencias en la apropiación de contenidos de programación por parte de los estudiantes, lo que ratifican numerosos investigadores, como González, Estrada y Martínez (2006); Reyes (2010); Vargas, Pérez y Blanco (2014); Salgado, Alonso y Gorina (2014); Blanco, Salgado y Alonso (2016), quienes corroboraron la carencia de habilidades para programar, coincidiendo en concluir que en ello influyen significativamente las limitadas destrezas que manifiestan para la concepción y desarrollo de algoritmos.

Precisamente, sobre la elaboración de algoritmos y su enseñanza se han realizado numerosas investigaciones en los últimos tiempos, las que han fundamentado la necesidad de que los estudiantes aprendan a algoritmizar, por el relevante papel que juegan los algoritmos para el logro de programas eficientes. Al respecto la Asociación de Profesores de Ciencia de la Computación (Computer Science Teachers Association) y la Sociedad Internacional de Tecnología de la Educación (International Society for Technology in Education) han concluido que los estudiantes se comprometen con el pensamiento computacional cuando usan algoritmos para resolver problemas y mejoran la solución de estos con la computación (ISTE and CSTA, 2011).

Ahora bien, para que la anterior conclusión se haga realidad, el proceso de enseñanza-aprendizaje de la algoritmización computacional demanda una reconstrucción teórica y práctica, que tenga en cuenta la didáctica de la resolución de los problemas de programación computacional, en su integración con la

Matemática y otras ciencias. Una reconstrucción con esas características fue realizada en la tesis doctoral presentada por Salgado (2015), cuyos aportes se asumen como principal soporte teórico de la presente investigación.

Consecuentemente, el Sistema Básico de Habilidades para la Algoritmización Computacional, que se propone en la presente investigación, complementa los aportes de la citada tesis, al profundizar en el estudio del proceso de algoritmización computacional, develando, fundamentando y ejemplificando cada una de las habilidades que lo conforman, con lo que se constituye en un eficiente medio de orientación a los docentes de Programación para la formación de la algoritmización computacional, partiendo de un conjunto de habilidades lógicas, que al relacionarse conforman el sistema. Este es útil para dinamizar el citado proceso formative, en el cual el aprendizaje de los estudiantes transitará por habilidades que irán incrementando su nivel de complejidad, hasta llegar a la más general de todas, la de algoritmizar computacionalmente.

DESARROLLO

Al formar parte la algoritmización de las operaciones racionales que se llevan a cabo en el marco del proceso de resolución computacional de las situaciones problémicas, ésta alcanza, a la vez que nuevas funciones, también nuevos rasgos y se manifiesta como una habilidad. De manera que en la presente investigación se concibe la algoritmización computacional como una habilidad y se fundamenta como tal.

Para el desarrollo de esta habilidad se considera que, una vez apropiadas por el estudiante las habilidades de menor grado de complejidad, estas son integradas para lograr algoritmos de las situaciones problémicas que se le presentan, empleando para ello pseudocódigos, diagramas de flujo o diagramas de Nassi-Schneiderman. Lo anterior se logra por la cualidad que poseen dichas habilidades de constituir un sistema, es decir, de relacionarse entre sí para dar lugar a propiedades cualitativamente distintas a la suma de las propiedades de cada habilidad que lo conforma y síntesis de las relaciones entre ellas, las que además caracterizan al sistema y expresan la dirección de su desarrollo (Blanco, 2017).

La propuesta didáctica que se hace no pretende considerar las habilidades que conforman el sistema como exclusivas de la algoritmización computacional, ya que es sabido que las mismas están presentes en otras ciencias y son básicas para el aprendizaje de otros contenidos, pero en este caso se contextualizan y resignifican para que cumplan una función orientadora dentro de las especificidades formativas de la citada algoritmización. Tampoco se pretende asegurar que ésta sea la única opción para formar la habilidad de algoritmizar computacionalmente, pero es una alternativa que puede proporcionar buenos resultados, dado que ofrece orientaciones metodológicas al profesor para su implementación práctica.

El logro de estas habilidades sólo se podrá materializar en la misma medida en que los profesores de las diferentes disciplinas de las carrera de ciencias computacionales las conozcan, estén convencidos de la lógica interna que garantiza la formación de los profesionales de estas ciencias, capaces de resolver computacionalmente las situaciones problémicas que manifiestan los usuarios, de forma independiente

y creadora y comprendan que la vía para lograr este propósito reclama de un esfuerzo mancomunado de unidad de pensamiento y acción.

Las cualidades que distinguen al Sistema Básico de Habilidades para la Algoritmización Computacional desde su condición de sistema, son las siguientes:

- Se considera un sistema abierto, dado que está sometido a múltiples influencias externas y la dinámica que promueve permite su remodelación y mejora constante. Esto posibilita su continuo perfeccionamiento a partir de las TIC y el desarrollo de la propia didáctica de la programación.
- Se clasifica como sistema básico en tanto es un sistema generador, conformado por un mínimo de habilidades lógicas, que se consideran necesarias para lograr el objetivo formativo que se persigue, las que interactúan en dicho proceso para dar lugar a otras tres habilidades de mayor complejidad: a) representar matemáticamente la situación problémica, b) generalizar mediante pseudocódigos o diagramas de flujo la representación matemática y c) valorar sintáctica y semánticamente la representación computacional. Estas, a su vez, conforman la estructura operacional de la habilidad algoritmizar computacionalmente.
- Es un sistema recursivo ya que adquiere sentido de las habilidades que lo conforman y dichas habilidades adquieren significado a través de su integración sistémica, lo que da cuenta de su coherencia.
- Manifiesta su sinergia en la habilidad algoritmizar computacionalmente, que se configura como nueva cualidad totalizadora, alcanzada en su implementación.
- Su entropía es provocada por las insuficiencias que presentan los estudiantes para concebir un algoritmo, computacional coherente, a la hora de resolver una situación problémica, así como por la resistencia de los profesores a aceptar los cambios que impliquen la introducción del sistema en la dinámica del proceso de enseñanza-aprendizaje de la algoritmización; ya que esto exige de una mayor preparación matemática y computacional y requiere de una profundización en el trabajo con representaciones que involucran objetos, relaciones matemáticas y pseudocódigos. Otra fuente de entropía es la asociada al proceso de comunicación, el que no siempre logra la eficiencia necesaria.
- La homeostasis es favorecida cuando el profesor adquiere conciencia de la necesidad de enseñar a algoritmizar para lograr una correcta resolución de las situaciones problémicas, mediante la programación computacional, y es capaz de fortalecer en sus estudiantes la formación de habilidades que les permitan apropiarse de la algoritmización computacional. Además, se puede potenciar valiéndose del empleo de software educativos, que permitan profundizar en la interpretación de la algoritmización de diversas situaciones problémicas.
- Su autodesarrollo se expresa en el carácter flexible, abierto, dinámico que posee, el que facilita al profesor su sistemática adecuación a determinadas circunstancias contextuales, que permiten su progresivo perfeccionamiento y desarrollo.
- La estructura se conforma a partir de las tres habilidades generales: a) representar matemáticamente la situación problémica, b) generalizar mediante pseudocódigos o diagramas de flujo de la representación matemática y c) valorar sintáctica y semánticamente la representación computacional, las que en su sistemática interacción dan lugar a la habilidad de algoritmizar

computacionalmente, y que a su vez son formadas a partir del trabajo con otras habilidades de menor grado de complejidad y generalidad, las que junto a estas conforman el sistema.

Fundamentación de la habilidad algoritmizar computacionalmente

Como antecedente del presente Sistema Básico de Habilidades, se consideró el trabajo de Delgado (1999), quien fundamentó un Sistema Básico de Habilidades Matemáticas.

Para su fundamentación, desde el un punto de vista psicológico se asumió la definición dada en Álvarez (1996), y citada en Alonso (2001), la que permite considerar que algoritmizar computacionalmente es una habilidad, ya que permite:

- Asimilar los conocimientos matemáticos y computacionales involucrados y derivados de la solución de una situación problémica.
- Conservar dichos conocimientos, dado que los registra en la memoria del resolutor.
- Recuperar los mismos para aplicarlos a la resolución de una determinada situación problémica, aplicación en la que los utiliza en correspondencia con las condiciones y exigencias de esa nueva situación, actualizándolos y generalizándolos en esta sistemática utilización.
- Exponer o comunicar estos conocimientos.
- En el plano didáctico, se consideró el concepto de habilidad planteado por Fuentes (2000) y a la luz de este se concibe el algoritmizar computacionalmente como una habilidad, pues se da:
- En la interacción del resolutor con la situación problémica, configurada como problema en una actividad concreta, dentro del proceso de enseñanza y aprendizaje de cualquier asignatura de programación computacional.
- Y en la comunicación del estudiante resolutor con su profesor, con sus compañeros y consigo mismo, de modo que esta exteriorización del algoritmo computacional resultante, facilita la regulación del proceso.

La habilidad algoritmizar computacionalmente es el contenido de las acciones que el estudiante resolutor de una situación problémica realiza y está integrada por un conjunto de operaciones, que en este caso tienen por objetivo la elaboración de un algoritmo que dé solución a la citada situación, extraída de un usuario, el cual facilita la comprensión y solución de la misma y se asimila en el propio proceso de resolución. De aquí se deriva la necesidad de explicitar las operaciones que el resolutor debe realizar para llegar a la representación algorítmica de la situación problémica mediante pseudocódigos. Es por ello que debe develarse una estructura operacional para la habilidad algoritmizar computacionalmente.

En esta dirección puede observarse que la representación algorítmica de una situación problémica, mediante pseudocódigos o diagramas de flujo, requiere que el estudiante resolutor descomponga mentalmente la situación en sus partes e identifique los elementos: objetos (reales, matemáticos o computacionales), características y relaciones que intervienen en la misma. Para ello, dicho estudiante requerirá de la comparación de esos elementos de la situación problémica con conceptos y procedimientos disponibles en su base de conocimientos y experiencias (memoria), de manera que los mismos adquieran un significado objetivo. Así, al momento de la identificación, se requerirá de la

adición de información almacenada en dicha base, resultado de esa comparación, la cual será necesaria para hacer las inferencias adecuadas.

Adicionar información es, según Alonso (2001), vestir a la situación problémica con elementos que no posee, que tiene el resolutor guardados en su base de conocimientos y experiencia, los que pueden ser de diferente naturaleza: sobre la computación, la matemática, la realidad objetiva, etc. Esta adición no sólo se produce durante la identificación de los aspectos de la situación problémica, sino que está presente durante todo el proceso de algoritmización computacional. Una vez identificados los elementos involucrados, se requerirá de su comparación y del establecimiento de nexos entre ellos y con respecto a las componentes de la situación problémica, de manera que se puedan seleccionar los aspectos relevantes a los efectos de la elaboración del algoritmo que le dará solución, dejando fuera aquellos que no sean importantes para dicho algoritmo.

La selección adecuada de la información dependerá de las exigencias y condiciones de la situación problémica extraída del usuario, así como de los conocimientos y experiencias matemáticas y computacionales que posea el estudiante resolutor, sobre tipos de situaciones y métodos de solución.

Un estudiante con habilidades para resolver situaciones problémicas, reconoce muchos tipos de estas y una vez que identifica la que tiene ante sí, le es más fácil juzgar qué es lo más importante para su resolución. Así la identificación y selección de los elementos del problema que se consideran relevantes, su posterior abstracción del análisis de las partes y su integración, dan lugar a una síntesis y conclusión, al nivel superior, en el conocimiento de la situación problémica.

En resumen, el proceso de abstracción-generalización explicado hasta aquí se logra a través de un proceso de análisis-síntesis, por medio del cual se lleva a cabo el aislamiento de unos aspectos de otros, el establecimiento de relaciones entre ellos y la selección de los esenciales o relevantes. A partir de la integración de los elementos considerados esenciales, se alcanza una síntesis como resultado de la integración de los mismos en función de las condiciones y exigencias de la situación problémica, lo que constituye una primera representación del problema y un peldaño superior en su comprensión.

A través de todo este proceso se establece un canal de comunicación entre la base de conocimientos y experiencias del estudiante resolutor, y los elementos de la situación problémica, mediados por las condiciones y exigencias de la misma. Esto permite realizar comparaciones sistemáticas entre los elementos de la situación y los conceptos y experiencias almacenadas en la memoria del resolutor. El intercambio se establece mediante un proceso de análisis y síntesis como base de abstracciones y generalizaciones que se van haciendo y que permiten la adición de la información necesaria para profundizar y precisar la información que da el problema. Al finalizar el citado proceso se obtiene una representación matemática de la situación problémica que puede exteriorizarse a través de una función, un gráfico, una ecuación, u otros objetos matemáticos (Alonso, 2014).

Una vez obtenida dicha representación, el estudiante resolutor tendrá que distinguir, dentro de un conjunto de estructuras algorítmicas conocidas por él, aquellas que le serán útiles para transformar los objetos y relaciones que aparecen en la representación matemática (a partir del conocimiento de las características y funciones de dichas estructuras), para luego realizar la selección, análisis y concatenación adecuadas de las que previamente identificó para conformar el algoritmo, generalizando mediante pseudocódigos o diagramas de flujo la representación matemática. En este momento del proceso de algoritmización se destaca también la presencia del proceso lógico de análisis-síntesis y abstracción-generalización que sustenta a la habilidad algoritmizar.

Finalmente, el estudiante deberá valorar sintáctica y semánticamente la representación computacional, para garantizar la pertinencia del algoritmo realizado; lo que logrará haciendo sucesivos refinamientos durante y después de estructurado el algoritmo, a partir de tomar en cuenta la sintaxis del pseudocódigo, para luego hacer la valoración semántica, midiendo la precisión de su efectividad, en correspondencia con la intencionalidad deseada, para dar respuesta a la situación problémica. Todo lo anterior permite esquematizar la estructura operacional de la habilidad algoritmizar computacionalmente una situación problémica como se muestra en la figura 1.

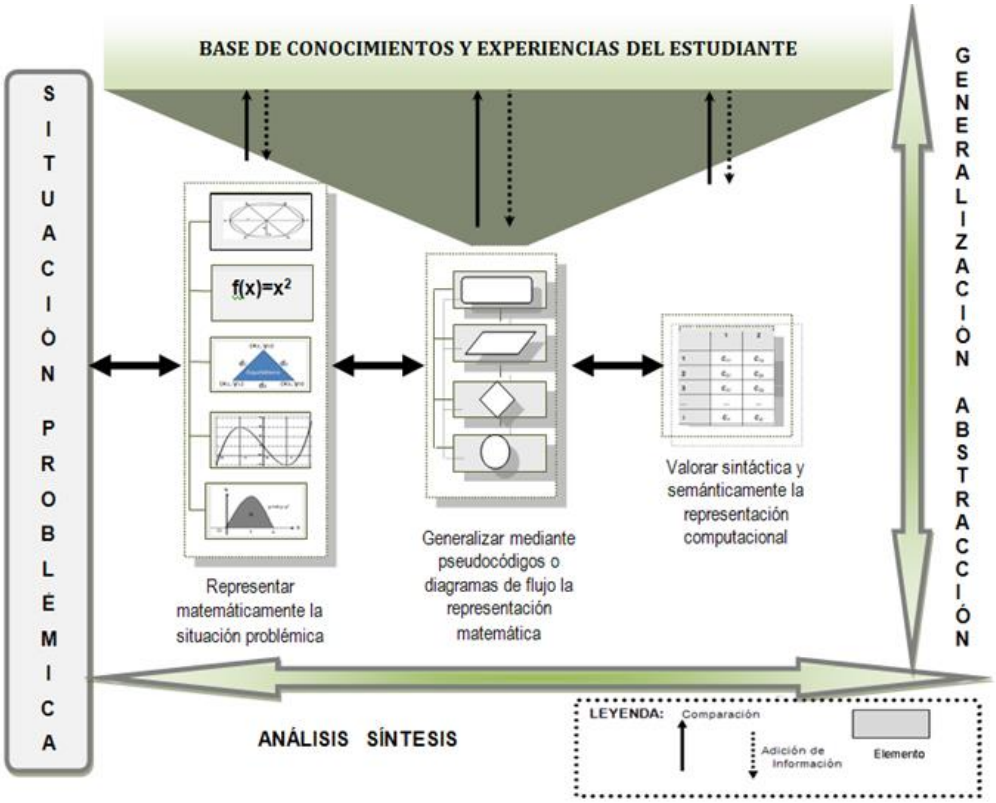

Figura 1. Estructura operacional de la habilidad Algoritmizar computacionalmente

Definición de las habilidades que conforman el sistema básico

De la habilidad de mayor grado de generalidad dentro del sistema básico, algoritmizar computacionalmente, se han derivado las tres habilidades explicadas anteriormente, las que estructuran el sistema: representar matemáticamente la situación problémica, generalizar mediante pseudocódigos o diagramas de flujo la representación matemática y valorar sintáctica y semánticamente la representación computacional. Estas tres habilidades, a su vez, se despliegan en otras de menor grado de generalidad, las que pueden interpretarse también como operaciones, y que al interactuar en el proceso formativo dan lugar a la apropiación de las más generales y complejas. De manera que en la dinámica del proceso de enseñanza-aprendizaje de la algoritmización computacional, los docentes deben comenzar trabajando las habilidades menos generales y complejas, para ir incrementando sistemáticamente esos niveles de generalidad y complejidad hasta lograr que sus estudiantes logren algoritmos computacionales eficientes y eficaces.

A continuación, se presenta la fundamentación de las habilidades de menor grado de complejidad en el sistema, dentro de las que se distinguirán cuatro habilidades, que sin pretender considerarse más importantes que las demás, se constituyen en el sustento del resto. Como se muestra en la figura 2, estas habilidades son: analizar, sintetizar, abstraer y generalizar.

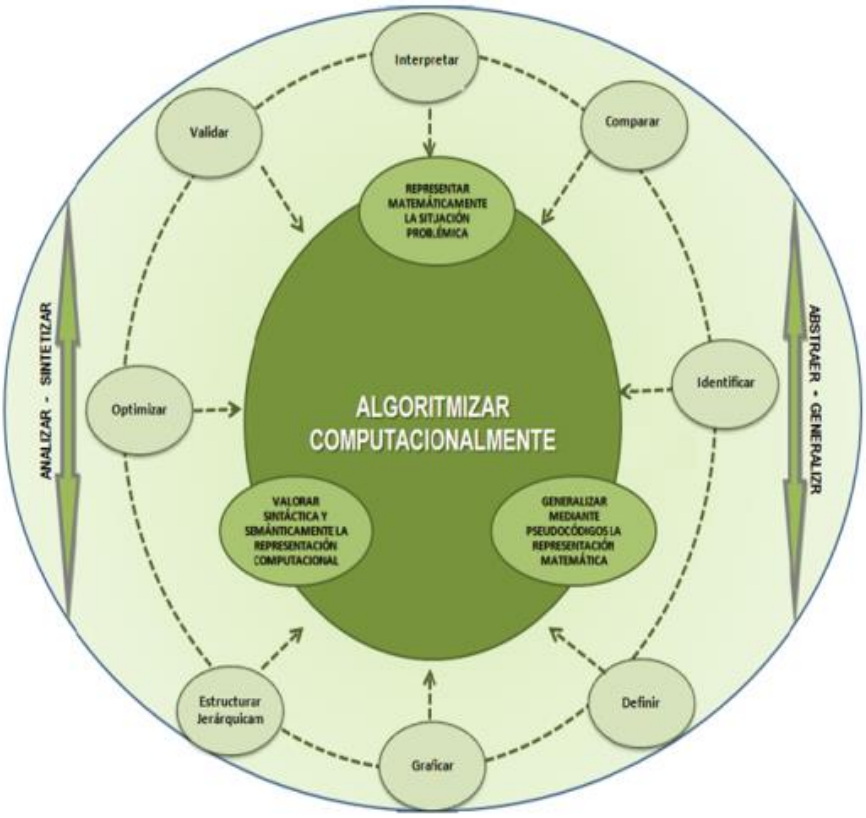

Figura 2. Sistema básico de habilidades para la algoritmización computacional

ANALIZAR

Definición: es la descomposición mental de la situación problémica en sus partes integrantes, con el objetivo de examinar detalladamente cada una de estas e identificar los objetos reales, matemáticos y/o computacionales, que las conforman, sus características y las relaciones que existen entre ellos, así como clasificarlos, conocer sus funciones y determinar las más importantes o esenciales a los efectos del objetivo que se persigue, todo para avizorar una posible algoritmización computacional de la misma.

Importancia de su formación: posibilita el estudio de los fenómenos y procesos que se manifiestan en el mundo objetivo y subjetivo con que se relaciona el hombre. Es una habilidad muy importante para el desarrollo del proceso de algoritmización computacional, pues está presente a lo largo del mismo; en un inicio permitiendo identificar los elementos de la situación problémica y establecer los aspectos que servirán de punto de partida para la creación del algoritmo y luego facilitando el proceso de identificación de las estructuras computacionales (pseudocódigos o diagramas) más adecuados para

conformar dicho algoritmo. De manera que de su apropiación, por parte de los estudiantes, dependerá el éxito que tengan en la algoritmización computacional.

Relación con otras habilidades del sistema: se desarrolla en estrecha relación con la de sintetizar, constituyéndose en la base de todas las demás habilidades y manteniendo su presencia en todo el proceso de algoritmización computacional.

Orientaciones metodológicas: durante el proceso formativo el profesor debe incitar al estudiante para que lea varias veces la situación problémica que se le plantea y trate de identificar y clasificar los objetos que la componen, las características de esos objetos y sus relaciones. Para ello podrá emplear el método inductivo-deductivo y mediante preguntas ir guiando la realización del análisis. También podrá emplear el método de trabajo en grupos para que los estudiantes menos preparados se apropien de patrones de análisis empleados por los más aventajados. El análisis deberá trabajarse también para determinar las estructuras computacionales necesarias en la construcción del algoritmo.

SINTETIZAR

Definición: es la composición, a partir del análisis previo de una situación problémica, de aquellos objetos, características y relaciones de la misma que son esenciales a los efectos del objetivo algorítmico que se persigue, relacionándolos e integrándolos en un nuevo objeto, más simplificado y significativo que la situación problémica original.

Importancia de su formación: permite que el estudiante aprenda a seleccionar información relevante de la situación problémica. Esta habilidad será muy útil en todo el proceso de representación que debe llevar a cabo durante la algoritmización computacional.

Relación con otras habilidades del sistema: forma una simbiosis con la de analizar, para dar lugar a uno de los métodos más universales y útiles en la captura y procesamiento de la información proveniente de una situación problémica: el método de análisis y síntesis. De manera que resulta básica dentro del sistema que se propone y está presente en todos los momentos de la algoritmización computacional.

Orientaciones metodológicas: a lo largo del proceso formativo deben trabajarse situaciones problémicas en las que el estudiante tenga la oportunidad de centrar la atención en el objetivo (satisfacción de los requerimientos o condiciones y de las exigencias de la citada situación) y a partir del mismo seleccionar la información que sea esencial para cumplirlo, teniendo en cuenta el resultado del análisis previo realizado. Para ello podrán emplearse los métodos orientados para el caso de la habilidad analizar. También podrá emplearse el método de elaboración conjunta y la conversación heurística.

ABSTRAER

Definición: es la identificación de los rasgos esenciales de los objetos y su utilización para determinar reglas, conceptos y otras generalizaciones propias de la algoritmización computacional. Consiste en la separación mental de una determinada faceta, cualidad o dato, de un fenómeno u objeto, aislando las cualidades, características y relaciones que son esenciales en el mismo, con la intención de lograr el objetivo algorítmico perseguido.

Importancia de su formación: facilita el estudio de los fenómenos y procesos que se revelan en el mundo objetivo y subjetivo inherente a la situación problémica. Juega un papel muy importante en el proceso de análisis y síntesis que debe llevarse a cabo en esta, así como en el resto del proceso de algoritmización computacional, siendo determinante para la asociación de objetos computacionales a los objetos matemáticos. Su formación está íntimamente relacionada al grado de conciencia con que se forman los conocimientos en el individuo, lo que la convierte en una habilidad sumamente compleja para su formación y medición. El desarrollo de la abstracción depende de la acumulación de representaciones y percepciones, para lo cual es necesario que el estudiante posea o se forme una adecuada base de conocimientos e incluso un apropiado sistema de creencias.

Relación con otras habilidades del sistema: es la base de la síntesis y se desarrolla en estrecha relación con las habilidades de analizar, identificar, comparar y generalizar, manteniendo su presencia en todo el proceso de algoritmización computacional. Cabe precisar que, aunque la síntesis lograda por medio de la abstracción es más pobre que el propio objeto, es un peldaño superior en la comprensión de dicho objeto, ya que en esa abstracción están sintetizadas sus propiedades esenciales. La abstracción facilita el proceso de comparación e identificación de las estructuras computacionales para diseñar un algoritmo pues permite su asociación con las estructuras matemáticas.

Orientaciones metodológicas: deben presentarse situaciones problémicas a los estudiantes en las que tengan que identificar rasgos esenciales de los objetos que las componen, para luego lograr una síntesis de los mismos. Para ello podrán emplearse situaciones derivadas de procesos reales conocidos por los estudiantes y aprovechar el método de elaboración conjunta para destacar cada paso que conduzca a una abstracción, explicitando su presencia en la solución. Luego podrán emplearse métodos de enseñanza como la conversación heurística y el trabajo en grupo, entre otros, que lleven al desarrollo de debates e intercambios que activen el proceso de aprendizaje y faciliten que los estudiantes tomen consciencia de la importancia de esta habilidad y trabajen para apropiarse de la misma.

GENERALIZAR

Definición: la generalización es la reunión mental de objetos en un concepto algorítmico, éste es un proceso de gran complejidad, en cuya formación la actividad del pensamiento está dirigida a revelar los rasgos esenciales y generales del objeto matemático y/o computacional; así como, establecer las relaciones y nexos entre ellos, formando nuevas propiedades, nexos y reglas, las que se concretan mediante ejemplos. Implica una profundización y enriquecimiento del conocimiento del individuo para el establecimiento de determinados caracteres comunes, y a la vez esenciales, entre varios objetos o fenómenos analizados.

Importancia de su formación: es una habilidad básica para la inducción, es decir, para obtener suposiciones de un conjunto de objetos, fenómenos o relaciones, a partir del análisis de casos particulares o especiales. Es muy importante para la algoritmización, ya que no se trata de buscar un algoritmo que dé solución a cada uno de las situaciones problémicas particulares planteadas, sino que se deben buscar modelos generales de algoritmos que resuelvan clases de problemas. La generalización no sólo puede ser en forma de conclusión o definición de conceptos, sino también en forma de explicaciones o demostraciones. Esta hace posible la adquisición por parte de los estudiantes del contenido matemático y computacional de una forma rápida y consciente.

Relación con otras habilidades del sistema: tiene elementos comunes con la habilidad abstraer, estando siempre acompañada de ésta. Se produce con la ayuda de la comparación, que subraya lo general de una serie de objetos o fenómenos y de la abstracción de lo general, para llegar al conocimiento profundo de la realidad a través de los conceptos, los juicios y las conclusiones. En ella están presentes como habilidades fundamentales, analizar, sintetizar, abstraer, identificar y estructurar. Se concreta, mediante un algoritmo, en pseudocódigos o diagramas de flujo, como formas de conclusión que la transforma en conocimientos generalizados.

Orientaciones metodológicas: una forma de contribuir a la formación de esta habilidad es proponer un problema cuyo procedimiento de solución pueda generalizarse. Luego, a partir de la solución particular dada, puede plantearse otro problema más general que se resuelva siguiendo el mismo procedimiento o uno análogo. Además pueden usarse problemas para los cuales se pueda buscar desde el principio un problema general, que resulte más fácil de resolver, y luego aplicar la solución del mismo al caso particular. El profesor puede apoyarse de preguntas como las que aparecen en Salgado, Alonso y Gorina (2014).

El resto de las habilidades que conforman el Sistema Básico de Habilidades para la Algoritmización Computacional, que se presentan a continuación, unidas a analizar, sintetizar, abstraer y generalizar, permiten la construcción de algoritmos computacionales eficientes y eficaces (ver figura 2).

INTERPRETAR

Definición: es concebir, ordenar y expresar de un modo personal los fenómenos y procesos de la realidad, que conforman una situación problémica, atribuyendo significado matemático y/o computacional a cada una de sus componentes, de modo que adquieran sentido algorítmico. Se hace posible a partir de la determinación del universo de la situación problémica y los sistemas que la integran, así como del significado que tienen las componentes que se obtienen de sus interrelaciones. Mediante la interpretación se formula la veracidad lógica y real de los juicios analíticos y sintéticos, así como de su interdependencia, con el objetivo de lograr una adecuada algoritmización computacional.

Importancia de su formación: es una habilidad lógica relevante para la algoritmización computacional, pues permite deducir los nexos o relaciones esenciales (jerárquicas y de coordinación) entre los componentes de una situación problémica, atribuyéndole un significado (estructura), así como determinar la relación entre los objetos, fenómenos y/o procesos (función) y la dinámica de dichos objetos, fenómenos y procesos como un todo íntegro, mediante la síntesis, considerando sus partes, propiedades, relaciones y las leyes de su desarrollo (relación entre estructura y función de los objetos matemáticos y/o computacionales).

Relación con otras habilidades del sistema: es básica para el desarrollo de todas las demás habilidades del sistema. Se sustenta en las habilidades de analizar y sintetizar para algoritmizar computacionalmente.

Orientaciones metodológicas: el desarrollo de esta habilidad presupone una interacción sistemática con la situación problémica que se está resolviendo, lo que debe lograr el profesor mediante preguntas que vayan permitiendo conocer como la han procesado mentalmente los estudiantes (Salgado, et all, 2014). Es por ello importante realizar actividades docentes en las que se resuelvan problemas de manera conjunta, incitando a la participación de los más pasivos, porque es la forma más precisa para

regular el proceso interpretativo. Se pueden usar los métodos de enseñanza-aprendizaje anteriormente citados, sobre todo la conversación heurística. En esta dirección el profesor debe tener en cuenta que hasta que el estudiante no interprete bien la situación que aborda, todo el trabajo que realice para resolverla puede resultar improductivo.

COMPARAR

Definición: es el establecimiento de semejanzas o diferencias entre objetos, según sus rasgos, características, propiedades, funciones, etc. El descubrimiento de semejanzas permite organizar la información conocida y relacionarla con la nueva, con el fin de establecer relaciones. La identificación de diferencias permite contrastar los conceptos, además de concatenar los resultados del análisis de la situación problémica, vinculándolos a las diversas partes de la información obtenida para establecer relaciones. Es decir; relacionar los datos, variables, estructuras de control similares, y localizar las diferencias y semejanzas, para que, una vez hecha la comparación, se puedan establecer los vínculos entre los componentes de la información. Puede realizarse entre dos o más objetos, estructuras (matemáticas o computacionales), situaciones o con la base de conocimientos y experiencias de la persona que compara.

Importancia de su formación: dominar esta habilidad hace posible obtener mejor comprensión, retención y clarificación de confusiones entre conocimientos similares. Además, fortalece el desarrollo conceptual y favorece el desarrollo de competencias tan importantes como la toma decisiones. Es importante para formar el pensamiento analítico en los estudiantes, pues les permite ver las diferencias y similitudes que existen entre objetos, variables o estructuras (matemáticos, computacionales) con respecto a un mismo objetivo (resolver la situación problémica mediante la algoritmización computacional). Además, les facilita la formación del pensamiento abstracto, al permitir obtener una interpretación matemática y/o computacional en cada paso del proceso de algoritmización.

Relación con otras habilidades del sistema: se relaciona con analizar, sintetizar, ordenar, identificar, generalizar y abstraer. En la estructura de una comparación está presente el identificar los rasgos o características de los objetos, ordenar y describir los indicios de los objetos, señalando los esenciales (abstraer), confrontar según los rasgos esenciales determinando diferencias y semejanzas y generalizar en forma de conclusión sobre la base de estos rasgos esenciales.

Orientaciones metodológicas: para enseñar a comparar es preciso partir de establecer los criterios que van a servir de base para la comparación. En el caso de la algoritmización el criterio puede ser relativo al número de variables que intervienen, las condiciones de parada, las precondiciones y pos-condiciones de la situación problémica, la necesidad de repetir acciones, etc. Es importante desarrollar actividades docentes en las que los estudiantes trabajen en la apreciación de similitudes, semejanzas y diferencias, explicándoles que la comparación puede ser incompleta cuando se toman los indicios para las semejanzas o para las diferencias solamente y será completa cuando se realicen las dos consideraciones. Además de precisarles que las semejanzas pueden ser absolutas, relativas, intrínsecas, funcionales e implícitas. Para comparar acertadamente las estructuras lógico-computacionales se podrían tener en cuenta interrogantes como las que aparecen en Salgado, et al, 2014.

IDENTIFICAR

Definición: la identificación se establece a partir de la constatación de los rasgos, características, propiedades o cualidades esenciales de los objetos, lo que permite su inclusión en una categoría o concepto dado, es decir reconocer si el objeto es el que se busca como perteneciente a un concepto de referencia.La identificación no sólo recae sobre los objetos de la situación problémica que se analiza, sino que es necesario identificar también los objetos matemáticos y computacionales que deberán emplearse en la representación de la mencionada situación, para garantizar una lógica coherente en el proceso de algoritmización.

Importancia de su formación: su formación complementa al sujeto con un recurso teórico insustituible para la toma de decisiones y la resolución de situaciones problémicas. Esta habilidad facilita que, una vez que el estudiante haya realizado una interpretación correcta de la situación problémica, tenga que distinguir dentro de un conjunto de estructuras matemáticas y algorítmicas conocidas por él, aquellas que le serán útiles para transformar los objetos y relaciones que aparecen en la citada situación, a partir del conocimiento de las características y funciones de dichas estructuras. Todo lo anterior posibilitará el logro de una representación matemática pertinente y garantizará la validez de la correspondiente representación computacional, que es la que permite concretar la solución algorítmica requerida, a partir de objetos y relaciones computacionales.

Relación con las otras habilidades del sistema: se relaciona en gran medida con las habilidades analizar y comparar, pues en la medida en que el estudiante va analizando, comienza a identificar los tipos de objetos que están presente en la situación problémica y para identificarlos debe comparar estos con su base de conocimientos y experiencias, llegando a una representación, primero usando objetos matemáticos y luego objetos computacionales como if, while, do while, for entre otras. De aquí que el representar, sea muy importante en la formación de la habilidad identificar, pues es mediante la representación gráfica, tabular, conjuntista, algebraica o geométrica, que podrá determinar cuál de las estructuras identificadas (matemáticas y computacionales) es la mejor. Por otra parte, el identificar se relaciona también con optimizar pues al comparar las posibles representaciones matemáticas y computacionales a utilizar y seleccionar la mejor, garantiza que el diseño del algoritmo sea eficiente.

Orientaciones metodológicas: dado que la identificación de estructuras matemáticas y computacionales presupone el dominio de las propiedades y funciones de las mismas, el profesor deberá facilitar la apropiación de estos conocimientos, haciendo una amplia ejemplificación de estas funciones, relaciones, ventajas y limitaciones, para que se fijen en la mente del estudiante y puedan ser recuperados cuando deban hacer una identificación. Se debe utilizar fundamentalmente el lenguaje natural, de manera que le permita al estudiante captar la esencia de las estructuras, sin tener en cuenta aun la sintaxis de los lenguajes de programación.

También será preciso que sepan diferenciar las propiedades y funciones esenciales de las no esenciales, como habilidad clave para lograr la identificación. Esto requiere del dominio del concepto de propiedad y función, además de habilidades para diferenciar, en las estructuras, diversas propiedades y características. Para lograr esto el profesor podrá:

- Inducir a la utilización de diversas estructuras lógico-computacionales para un mismo problema, con el objetivo de analizar cada una de ellas y realizar comparaciones, teniendo en cuenta los objetos utilizados, la intencionalidad del uso de cada una, sus ventajas y limitaciones.

- Emplear métodos de enseñanza participativos (elaboración conjunta, trabajo en grupos, exposición problémica, búsqueda parcial, entre otros) para propiciar la reflexión y discusión de las representaciones de las situaciones problémicas. Esto promueve la recuperación y reconstrucción de los conocimientos matemáticos y computacionales de los estudiantes.
- Ejercitar las estructuras algorítmicas necesarias para transformar los objetos de la situación problémica. Así, si se está ante una sumatoria o productoria, debe explicase a los estudiantes que lo más recomendable es identificar como estructuras computacionales posibles a emplear, las iterativas (for, while, do while, etc).

DEFINIR

Definición: es establecer mediante una proposición las características esenciales (genéricas o distintivas) de la situación problémica bajo estudio, a los efectos del objetivo algorítmico que se persigue, precisando su estructura y su función.

Importancia de su formación: contribuye a la formación de la habilidad identificar y comparar. Será necesaria su presencia en cada paso del algoritmo, para definir las precondiciones y postcondiciones que se deben cumplir; así como, para asignar y definir valores a las variables de entrada y salida. El logro de una correcta selección de las características esenciales del objeto a definir se constituye en base conceptual de la habilidad algoritmizar computacionalmente.

Relación con las otras habilidades del sistema: se relaciona con todas las habilidades del sistema y en mayor medida con las habilidades identificar y comparar, pues en la medida en que el estudiante es capaz de analizar, comprender e interpretar la situación problémica, comienza a identificar los tipos de objetos que están presentes en ella, comparándolos con su base de conocimientos y experiencias hasta llegar a la definición, primero usando objetos matemáticos; variables, condiciones, dominio, imagen, entre otras y luego objetos computacionales tales como asignaciones de variable, arreglos y estructuras lógicas computacionales. Por otra parte, durante este proceso de definición, el optimizar juega un papel muy importante, pues permanentemente el estudiante debe cuestionarse si la definición hecha contiene exactamente los objetos necesarios para representar la situación problémica y poder resolverla. De aquí que representar sea inherente a la habilidad definir, pues es mediante la representación que se podrá valorar si lo ya definido realmente logra precisar la estructura y función de los objetos que intervienen.

Orientaciones metodológicas: esta habilidad debe trabajarse en cada situación problémica que aborde el estudiante, bajo la orientación del docente, y en dos momentos; el primero encaminado a definir matemáticamente los objetos que componen la situación problémica y el segundo a definir computacionalmente los objetos matemáticos previamente precisados. Para lograr lo anterior el profesor deberá inducir a los estudiantes a que definan de manera no formal las condiciones de la situación problémica, ya sean pre-condiciones o pos-condiciones.

Posteriormente, podrá realizar preguntas a los estudiantes tales como: ¿cuántas variables será necesario utilizar para representar los objetos que aparecen en la situación problémica?, con el objetivo de que los estudiantes vayan comprendiendo que según la cantidad de objetos identificados en la situación problémica será en mayor o menor medida las variables que deberán definir. También podría preguntar ¿qué valores podrían tomar estas variables? y ¿cuál es su posible dominio?, con el objetivo de que el estudiante pueda definir matemáticamente cada una de las variables. Una vez lograda la

definición matemática será necesario realizar la definición computacional, para lo cual debe enfatizarse en la declaración de los tipos de variable (real, entero, cadena, arreglo, etc.) en correspondencia con el dominio definido matemáticamente. En este momento el profesor puede preguntar ¿Cómo se pueden declarar las restricciones matemáticas de cada variable? ¿Será necesario declarar nuevas variables? Estas preguntas incitarán al estudiante a profundizar en sus conocimientos. Es importante explicar que, desde el punto de vista computacional, las restricciones definidas matemáticamente para una determinada variable no se declaran, sino que se controlan usando estructuras lógico computacionales tales como if, while, do while, for, case y switch, entre otras.

GRAFICAR

Definición: es representar, durante el proceso de algoritmización, relaciones entre objetos matemáticos y/o computacionales, tanto desde el punto de vista geométrico, como de diagramas o tablas y, recíprocamente, colegir las relaciones existentes, a partir de su representación gráfica.

Importancia de su formación: esta habilidad permite al estudiante comunicar información de manera visual y sucinta, así como representar objetos ideales involucrados en la situación problémica. Su uso es muy importante en el proceso de construcción de un algoritmo, ya que facilita la representación de objetos y relaciones reales, mediante objetos y relaciones matemáticas, para facilitar su posterior generalización mediante pseudocódigos. Su empleo permite la visualización de analogías ocultas tras las fórmulas.

Relación con otras habilidades del sistema: esta habilidad sirve de base a la habilidad interpretar y a la de abstraer, relacionándose también con la de analizar y sintetizar.

Orientaciones metodológicas: se debe insistir en su empleo, siempre que la situación problémica que se está resolviendo lo admita, ya que ayudan a orientarse en la tarea a resolver. Hacer ver a los estudiantes que los gráficos facilitan la observación de elementos y relaciones que mentalmente son difíciles de descubrir. Para formar esta habilidad es muy importante el uso del método de ejemplificación, la elaboración conjunta y la conversación heurística.

ESTRUCTURAR JERÁRQUICAMENTE

Definición: es un proceso de ordenación, disposición e integración algorítmica, que garantiza la selección, análisis y concatenación correcta de las estructuras lógicas, previamente identificadas (ya sean matemáticas o computacionales), compatibles a partir de un criterio de comparación preestablecido, para conformar el algoritmo que constituirá la estructura funcional del proceso de programación computacional. Permite organizar y clasificar en rangos de distintas categorías, estableciendo un orden de superioridad o de subordinación entre conceptos, variables o estructuras, a partir de un orden de importancia.

Importancia de su formación: su apropiación por el estudiante asegura la eficiencia algorítmica del proceso y la solución de la situación problémica objeto de análisis. El estudiante deberá distinguir y ordenar las estructuras matemáticas y computacionales a partir de una comparación previa, constituyéndose en expresión concreta de una elección lógico-computacional-estructurada.

Relación con las otras habilidades del sistema: se relaciona con las habilidades analizar, comparar e identificar, pues en la medida en que el estudiante va analizando las estructuras lógico-computacionales identificadas, va comparándolas, teniendo en cuenta sus características, ventajas y desventajas, nivel de jerarquía y posible orden para estructurarlas de acuerdo a una situación problémica determinada.

La habilidad estructurar jerárquicamente también se relaciona con representar y optimizar, pues es mediante la representación computacional, ya sea mediante pseudocódigo, diagramas de flujo o diagramas de Nassi-Schneiderman, que se podrá determinar cuál de las estructuraciones computacionales realizadas es la mejor. También durante la estructuración jerárquica habrá que optimizar constantemente, pues al realizar varias estructuraciones, que podrían responder a una misma situación problémica, el estudiante para seleccionar la mejor deberá considerar sus características, ventajas y desventajas, así como el número de objetos que intervienen y la cantidad de variables a declarar, este proceso indirectamente evidencia una optimización de recursos computacionales en el diseño del algoritmo.

Orientaciones metodológicas: el profesor debe asegurarse que los estudiantes realicen estructuraciones logrando el algoritmo, empleando pseudocódigo, diagramas de flujo o diagramas de Nassi-Schneiderman, sistematizando las ventajas y desventajas de cada estructuración realizada. Para tales propósitos puede resultar provechoso el empleo de un software como el PSeInt (Novara, 2012), RAPTOR (Wilson, Carlisle, Humphries y Moore, 2013), Free DFP (Cárdenas, Castillo y Daza, 1998) o el Software para la enseñanza-aprendizaje de algoritmos estructurados (Arellano, Nieva, Solar y Arista, 2012), entre otros que permitan apoyar la dinámica resolutora.

También se deberá poner énfasis en las ventajas y desventajas de cada estructura lógico-computacional, propiciando que el estudiante realice comparaciones entre ellas, con el objetivo de seleccionarlas, analizarlas e integrarlas adecuadamente, en aras de formar una estructura funcional para la programación. Se sugiere realizar el análisis de diferentes formas de ordenar y concatenar las estructuras computacionales, de acuerdo con las definiciones y representaciones matemáticas obtenidas, para ofrecer patrones a seguir en la resolución de las situaciones problémicas, de manera que puedan valorar la pertinencia de cada concatenación e integración, sus ventajas, desventajas, relaciones y características, explotando el uso de casos particulares para reforzar el aprendizaje de la representación matemática-computacional mediante estructuras algorítmicas. Se debe exigir al estudiante que establezca comparaciones para justificar el uso de determinadas estructuras, así como su integración mediante un pseudocódigo en un algoritmo.

OPTIMIZAR

Definición: es emplear en la construcción de un algoritmo aquellas estructuras matemáticas y lógico-computacionales que aseguran eficiencia y eficacia en la solución de una determinada situación problémica.

Importancia de su formación: si se tiene en cuenta que la optimización es el proceso a través del cual se mejora la rapidez y la eficiencia en el funcionamiento de un sistema de información computacional, se comprenderá la importancia de educar a los estudiantes en la discusión de las diversas vías de solución algorítmica de una situación problémica, resaltando y seleccionando la más eficiente para contribuir de manera productiva al desarrollo de la habilidad optimizar.

Relación con las otras habilidades del sistema: se relaciona principalmente con las habilidades comparar, estructurar y validar, pues en la medida en que el estudiante va comparando las estructuras computacionales para formar el algoritmo, debe ir analizando sus desventajas y ventajas, teniendo en cuenta el número de operaciones a realizar, así como el número de variables y condiciones de culminación del algoritmo. Esto permite que en cada momento se esté validando y valorando el diseño lógico del algoritmo, con lo que se irá ganando en eficiencia y eficacia computacional.

Orientaciones metodológicas: el profesor debe propiciar actividades docentes en las que los estudiantes realicen varias estructuraciones de un mismo problema, de manera que se analicen a profundidad las estructuras lógico-computacionales empleadas, para que sistematicen sus ventajas y desventajas. Para tales propósitos puede resultar provechoso el empleo de un software como el PSeInt, RAPTOR o el Free DFP.

También deberá analizar con los estudiantes el funcionamiento de las estructuras computacionales y enfatizar en el número de instrucciones máquina que consume cada operación que se diseña en el algoritmo, pudiendo formular preguntas como: ¿cuáles instrucciones consideran innecesarias en el algoritmo? ¿Qué pasa si se ejecutan acciones innecesarias? ¿Cómo se pudieran disminuir las líneas del pseudocódigo sin afectar la solución? ¿Cuál es en términos computacionales la implicación de eliminar una sentencia? Estas preguntas pueden llevar al estudiante a desarrollar la habilidad optimizar de una manera natural durante el proceso de algoritmización computacional, todo ello con el propósito de elevar la eficiencia y eficacia de los resultados.

VALIDAR

Definición: es la acción y efecto de convertir la solución algorítmica en válida, dándole confiabilidad. Se refiere a la verificación, confirmación o examen de la solución computacional obtenida mediante un algoritmo, en función de las condiciones y exigencias originales de la situación problémica.

Importancia de su formación: el proceso de validación es clave para la resolución de una situación problémica computacional, pues un algoritmo o programa computacional se constituye en solución, cuando se ha logrado demostrar su validez y confiabilidad con respecto a todos los posibles rangos de datos entrantes y se cumple con las exigencias y condiciones de dicha situación problémica.

Es así que la validación se encarga de verificar la corrección, integridad y entendimiento compartido de los datos. No obstante, validar contra un esquema no garantiza totalmente que los datos sean correctos, permite detectar formatos nulos o valores fuera de rango y por tanto incorrectos.

Cabe señalar que el proceso de validación es iterativo y se debe realizar tantas veces como se considere, de acuerdo a la complejidad de la situación problémica y del propio algoritmo diseñado, siempre con el objetivo de mejorar la eficacia y la eficiencia computacional.

Relación con las otras habilidades del sistema: se relaciona principalmente con las habilidades comparar, estructurar y optimizar, pues una vez estructurado el algoritmo se debe probar con juegos de datos diseñados para ello, e ir comparando en cada ejecución manual, los resultados con lo que se espera, de acuerdo a los requerimientos de la situación problémica. Es así que si se detecta algún error durante la validación, ya sea sintáctico o semántico, se debe proceder a realizar una nueva estructuración con lo que se estaría a su vez optimizando el algoritmo.

Orientaciones metodológicas: el profesor debe evidenciar la necesidad de perfeccionar el proceso de validación como etapa esencial de la sistematización de la algoritmización computacional, que conduce a que se logren cualidades tan importantes como la pertinencia, precisión y exactitud computacional. Generalmente se prueba el algoritmo con los datos que aparecen como ejemplo en el planteamiento de la situación problémica y, si funciona adecuadamente, se supone que el algoritmo ya es correcto. Sin embargo, en numerosas ocasiones, al probarlo con datos reales, usando software de apoyo, el algoritmo no se ejecuta correctamente o no proporciona el resultado esperado. Siendo consecuente con lo anterior, se hace necesario sugerir la formación de la habilidad validar en dos direcciones, la primera dirigida a la validación sintáctica y la otra a la validación semántica.

a. Validar sintácticamente el algoritmo: debe explicarse al estudiante que el empleo de pseudocódigos no significa la ausencia de errores sintácticos. Para facilitar este proceso pueden mostrarse ejemplos en los que se evidencien errores sintácticos comunes al emplear pseudocódigos. Esto permitirá ir creando la necesidad de revisar minuciosamente el pseudocódigo del algoritmo una vez escrito, lo que contribuirá al desarrollo de habilidades para implementar algoritmos sintácticamente correctos en un lenguaje de programación. El profesor puede orientar que se confirme el flujo de control del algoritmo, lo que consiste en verificar si el orden temporal en el cual se ejecutan las operaciones del algoritmo es correcto, es decir, si no se viola la disposición lógica de las estructuras ya establecidas.

b. Validar semánticamente el algoritmo: debe indicarse la valoración de la validez de la representación computacional, a partir de la realización de una ejecución manual del algoritmo con datos significativos, que abarquen todo el posible rango de valores, para comprobar que la salida coincide con lo esperado en cada caso, lo que facilitará corroborar si la estructuración del pseudocódigo es correcta en términos del contenido.

Para esto puede emplearse el trabajo en grupos, propiciando discusiones que den la oportunidad de exponer y defender ideas sobre la intencionalidad de la estructuración algorítmica realizada. Así se potenciará que los estudiantes se apropien de patrones de análisis y perfeccionamiento del proceso de validación semántica. Deben emplearse ejemplos en los que se creen conflictos cognitivos que estimulen cambios en las estructuraciones algorítmicas realizadas, lo que favorecerá la formación de habilidades para diseñar algoritmos que permitan optimizar la memoria y el tiempo de ejecución del computador. Puede además, promover la ejercitación del algoritmo realizado, tanto manual como empleando algún software como PSeInt, Free DFP o RAPTOR.

CONCLUSIONES

El Sistema Básico de Habilidades para la Algoritmización Computacional, que se aporta, tiene la intención de favorecer y orientar a profesores y estudiantes de Programación en la conducción de la dinámica del proceso de enseñanza aprendizaje de la algoritmización computacional.

El mismo está conformado por la definición de cada una de las habilidades que lo conforman, su importancia, la relación entre las habilidades y orientaciones metodológicas para su formación, todo lo cual facilita el trabajo de los profesores y potencia la apropiación de las mismas por parte de los estudiantes.

Este no pretende presumir de exclusividad en cuanto a las habilidades que incluye, pero las mismas se contextualizan y resignifican para que cumplan una función orientadora dentro de las especificidades formativas de la citada algoritmización computacional.

REFERENCIAS BIBLIOGRÁFICAS

- Alonso, I. (2001). *La resolución de problemas matemáticos. Una alternativa didáctica centrada en la representación.* Tesis doctoral. Universidad de Oriente, Cuba.
- Alonso, I. (2014). *La enseñanza de la Matemática a través de la Resolución de Problemas*. Trabajo presentado en el IV Encontro Internacional de Ensino e Pesquisa em Ciência na Amazônia. Tabatinga. Brasil.
- Arellano, J. J., Nieva, O. S., Solar, R. y Arista, G. (2012). Software para la enseñanza-aprendizaje de algoritmos estructurados. *Revista Iberoamericana de Educación en Tecnología y Tecnología en Educación, 8*, 23-33.
- Basogain, X., Olabe, M. A. y Olabe, J. C. (2015). Pensamiento Computacional a través de la Programación: Paradigma de Aprendizaje. RED. *Revista de Educación a Distancia. 46(6)*, 1-33.
- Blanco, A. (2016). Sistema Básico de Habilidades para la Algoritmización Computacional. Tesis de maestría. Universidad de Oriente. Cuba.
- Blanco, A., Salgado, A. y Alonso, I. (2016). Habilidades para la algoritmización computacional en la Licenciatura en Educación: Especialidad Educación Laboral-Informática. *Revista Maestro y sociedad, 13(1),* 16-28.
- Cárdenas, F., Castillo, N. y Daza, E. (1998). Editor e intérprete de algoritmos representados en diagramas de flujo. *Informática educativa UNIANDES-LIDIE, 11(1),* 101-106.
- Delgado, J. R. (1999). La enseñanza de la Resolución de Problemas Matemáticos. Dos elementos fundamentales para lograr su eficacia: La estructuración del conocimiento y el desarrollo de habilidades generales matemáticas. Tesis doctoral. ISPJAE. Ciudad Habana. Cuba.
- Díaz, K. I. y Crespo, T. (2013). *Análisis del sistema de habilidades del pensamiento lógico, como vía para la conformación de las habilidades de la programación.* Trabajo presentado en el Congreso Internacional de de Matemática y Computación COMPUMAT 2013. Villa Clara. Cuba.
- Fuentes, H. C. (2000). Didáctica de la Educación Superior. Universidad de Oriente. Cuba.1-252.
- González, W., Estrada, V. y Martínez, M. (2006). Contribución al desarrollo de la creatividad a través de la enseñanza de la programación. *Revista Pedagogía Universitaria*, 9(3).
- ISTE and CSTA (2011). Computer Science Teachers Association and the International Society for Technology in Education. "Pensamiento Computacional, Caja de Herramientas". Eduteka.
- Novara, P. (2012). PSeInt. http://pseint.sourceforge.net

- Reyes, L. (2010). Estrategia didáctica para la enseñanza de la habilidad programar: una alternativa para los profesores de los politécnicos de informática. Tesis de maestría. Universidad de Oriente, Cuba.
- Salgado, A., Alonso, I., Gorina, A. y Tardo, Y. (2013). Lógica algorítmica para laresolución de problemas de programación computacional: una propuesta didáctica. *Revista Didasc@lia: Didáctica y Educación*, 4 (1), 57-76.
- Salgado, A., Alonso, I. y Gorina, A. (2014). Ejemplificación de la solución algorítmica de problemas deprogramación computacional. *Revista Didasc@lia: Didáctica y Educación*, 5(4), 15-36.
- Salgado, A. (2015). Dinámica lógico-algorítmica del proceso de resolución de problemas de programación computacional. Tesis doctora). Universidad de Oriente, Cuba.
- Vargas, A., Pérez, O. L. y Blanco, R. (2014). Desarrollo de la habilidad algoritmizar: una visión desde los estudios de Ciencia, Tecnología y Sociedad. Trabajo presentado en la 1ra Conferencia Científica Internacional UCIENCIA 2014. Cuba.
- Wilson, T., Carlisle, M., Humphries, J. y Moore, J. (2013). RAPTOR Loop Logic. https://rogersoles.com/ technology/raptor-loop-logic/
- Wing, J. M. (2006). Computational Thinking. Communications of the ACO. 49(3).