

UNA VÍA PARA LA ATENCIÓN DE ALUMNOS CON DIFICULTADES EN LA SOLUCIÓN DE PROBLEMAS ITERATIVOS

A WAY TO ATTEND THE STUDENTS WITH DIFFICULTIES IN THE SOLUTION OF THE INTERACTIVE PROBLEMS

Eduardo Hernández Martín;¹ Liosbel Fleites Cabrera;² Gladilis Barrera Garrido³

¹Licenciado en Cibernética Matemática. Máster en Didáctica de la Computación. Profesor Auxiliar de la Universidad de Sancti Spíritus “José Martí Pérez” Cuba. Email: emartin@uniss.edu.cu; ²Licenciado en Educación. Especialidad Informática. Máster en Nuevas Tecnologías para la Educación. Profesor Asistente de la Universidad de Sancti Spíritus “José Martí Pérez” Cuba, Email lfleites@uniss.edu.cu.; ³Ingeniera en Ciencias de la Informática. Profesor Instructor de la Universidad de Sancti Spiritus “José Martí Pérez” Cuba, Email: gladilis@uniss.edu.cu.

¿Cómo referenciar este artículo?

Hernández Martín, E. Fleites Cabrera L. y Barrera Garrido, G. (marzo–junio, 2018).

Una vía para la atención de alumnos con dificultades en la solución de problemas iterativos. *Pedagogía y Sociedad*, 21 (51). Recuperado de

<http://revistas.uniss.edu.cu/index.php/pedagogia-y-sociedad/article/view/605>

Resumen

Los ciclos son una temática esencial dentro de la programación estructurada, ellos se estudian en un acápite denominado estructuras de control, donde esencialmente: se clasifican, se explican cómo funcionan y se ejemplifica su uso. La anterior forma de explicar los contenidos no es suficiente para un buen aprendizaje. Los autores del artículo publicaron en el año 2016 una solución para la anterior problemática, sobre la base de la experiencia pedagógica y como parte del proyecto de investigación “La formación didáctica del profesor para dirigir la solución de problemas desde las formas organizativas del proceso docente educativo”. Dicha solución fue efectiva para alumnos de aprendizaje medio y alto, pero no lo fue para alumnos de bajo aprovechamiento, sobre este presupuesto. En este artículo, se declara como objetivo: proponer una vía, obtenida mediante el método de modelación para facilitar el aprendizaje de los ciclos por los alumnos con dificultades en el estudio de esta materia. Esta propuesta fue aplicada en las consultas impartidas al grupo de tercer año de la carrera Licenciatura en Educación, especialidad Educación Laboral e Informática con resultados favorables.

Palabras clave: algoritmo; ciclos; estructuras de control; ingeniería informática; problemas iterativos; programación estructurada.

Abstract

The cycles are an essential theme in the structured programming. They are studied in an item called control structures, where the cycles are essentially classified, explained how they function and exemplified with their use. This former way of explaining the contents is not enough for a good learning. The authors of this scientific article published in 2016, one solution to the mentioned problem, based on their pedagogical experience and as a part of the following research project: “The teacher’s didactic formation to address the problem solution from the organized ways in the teaching – learning process”. This solution was effective in students of average and high level of learning, but it wasn’t in students with low advantage of learning. The objective of this article is to propose a way, using the model method to make easy the cycles learning of students with difficulties in the study of this subject. This proposal was put into practice in consultant lessons with students of third year in Bachelor’s degree of Education, Labor Education and Informatics specialty, and the results were favorable.

Key words: Algorithm; cycles; control structures; computer engineering; interative problems; structured programming.

INTRODUCCIÓN

La disciplina Lenguajes y Técnicas de Programación (LTP) tuvo sus antecedentes, en las carreras de perfil pedagógico, en el plan de estudios A, como consecuencia de un grupo de medidas del Estado Cubano para la introducción de la computación en la educación. En este momento, se concibió el estudio de la computación, como medio de enseñanza, herramienta de trabajo y objeto de estudio; en esta última modalidad su propósito fue la resolución de problemas mediante computadoras haciendo uso de un lenguaje de programación. La asignatura tuvo en sus inicios un marcado énfasis teórico en el trabajo con algoritmos, que con la dotación de la tecnología de tableros inteligentes a los Institutos Superiores Pedagógicos posibilitó su codificación en el lenguaje MSX-BASIC.

Con posterioridad en el plan de estudio B, en la carrera Matemática y otras carreras se amplía el contenido informático a estudiar y se incluyen contenidos relativos al Sistema Operativo y Sistemas de Aplicación, además de los Lenguajes de Programación.

En la concepción del plan de estudio C, aparece la formación de profesionales de la educación para impartir la Informática en la educación general y politécnica, con la introducción de la carrera Matemática-Computación en curso diurno. En esta carrera dentro de las disciplinas del área de formación técnica, aparece la disciplina LTP para el estudio de la teoría de algoritmos y la programación estructurada.

En las modificaciones del plan C, en el curso 2001-2002 se establece la formación de profesores de Informática en curso por encuentros, donde en el caso particular de la disciplina LTP se estructura el sistema de contenidos siguiendo la lógica de iniciar con el estudio de la algoritmia, después la programación estructurada, para culminar con la programación orientada a objetos y la programación conducida por eventos que confluían en la programación visual.

En el plan de estudios D, la carrera Educación Laboral-Informática asume la formación del profesional con doble perfil, que el caso de la Informática se prepara para impartir la asignatura en la educación general y politécnica.

En el plan de estudios E, vuelve a resurgir la carrera Educación Informática, con la diferencia que por primera vez aparece en el curso regular diurno y donde se empieza el estudio de la disciplina en segundo año.

Dentro de la programación estructurada la temática de los ciclos, lazos, bucles, iteraciones o repeticiones es primordial, pues esta estructura algorítmica básica incluye dentro de ella la lineal y la alternativa.

Disímiles autores se han referido a la programación estructurada y al aprendizaje de los ciclos: (Vildósola et al, 1990; Hernández, Hondal, y González, 2005; Borjas, 2013; Trejos, 2013; Hernández, Machín y Fleites, 2015; Hernández, Fleites y Valdés, 2016; Neira, 2016; Oviedo, 2016; Taha, 2017 y Capetillo, 2017) la mayoría de ellos hacen referencia a la temática de estructuras de control y dentro ella realizan alusión a los ciclos, clasificándolos y explicando cómo funcionan.

Los autores Hernández, Fleites y Valdés (2016) hacen referencia a un sistema de indicaciones de carácter algorítmico (SICA) que puede ser aplicado en la docencia y

facilita el aprendizaje de esta estructura algorítmica básica, utilizadas en el contexto de la enseñanza de la carrera Educación Laboral e Informática de la UNISS “José Martí Pérez”, que fue útil para alumnos de un nivel de aprendizaje medio y alto, pero que no logró el mismo resultado con alumnos desventajados en el estudio de los ciclos, por ende el presente trabajo tiene el objetivo de proponer una modificación al SICA , obtenida mediante el método de modelación, que al ser aplicada en las consultas impartidas al grupo de tercer año de la carrera Licenciatura en Educación, especialidad Educación Laboral e Informática, facilitó el aprendizaje de los ciclos por los alumnos con dificultades en el estudio de esta materia.

MARCO TEÓRICO O REFERENTES CONCEPTUALES

Los autores Hernández, Fleites y Valdés (2016) realizaron una amplia consulta a disímiles autores y concluyeron que todos ellos desarrollan la temática de los ciclos dentro de las estructuras de control del flujo del programa, planteando esencialmente que: el ciclo o lazo en Computación permite que una secuencia de pasos se repita tantas veces como sea necesario, bajo condiciones o conociendo el número de repeticiones; no es programable un ciclo que se repita indefinidamente, por lo tanto, es necesaria una condición para terminar la ejecución del lazo; se plantea que existen ciclos determinados (cuando se conoce la cantidad de repeticiones); indeterminados cuando esa cantidad es desconocida, a su vez, estos últimos, pueden ser evitables cuando se controla la condición de parada al inicio del ciclo e inevitable cuando esa condición se chequea al final del ciclo.

Desde el punto de vista de la enseñanza de los ciclos se puede sintetizar que los autores analizados después de dar una idea primaria acerca de lo que son los ciclos, en donde algunos de ellos incluyen una noción de la posibilidad de clasificarlos acorde a, si se conoce o no, la cantidad de repeticiones, plantean a continuación ejemplos resueltos donde tratan de agotar las variantes más típicas, esto puede ser considerado suficiente para el aprendizaje de esta estructura, pero la praxis demuestra que no es así, que solo con tener una idea teórica de cómo funcionan los ciclos y ver algunos ejemplos no basta para poder resolver nuevos ejercicios por parte de los estudiantes, por ejemplo: en el texto Programación Estructurada de la UAI, Universidad Abierta Interamericana (2017), se hace un análisis de los tipos de estructuras y dentro de éste, se analizan las

iteraciones, manteniendo el precepto de la sintaxis de las sentencias y posibles ejemplos.

Por su parte Capetillo (2017) mantiene, en su explicación la línea de: describir sintaxis y poner ejemplo, es bueno destacar que en la explicación de sus ejemplos es bien exhaustivo, incluso presentando niveles de refinamiento, lo cual es muy útil, para la persona que quiera aprender utilizando la analogía como vía principal para apropiarse de la idea esencial de una estructura de programación.

A su vez Taha (2017) también se afilia a la línea metodológica de plantear la teoría de cada tipo de ciclo, después la sintaxis y por último plantea y comenta ejemplos.

En el sitio Web Apuntes de programación, Neira (2016), mantiene la línea clásica que parte de conceptualizar, escribir sintaxis y ejemplificar, en su caso mantiene el mismo ejemplo en las tres estructuras de ciclo, aspecto interesante, pues permite al alumno comprender las diferencias entre una estructura y otra.

Borjas (2013), realiza un estudio acerca de las estructuras de control del flujo del programa, aunque normalmente se consideran 2, la alternativa y la repetitiva, esta autora propone incluir la lineal como otra estructura, esto es un detalle novedoso.

Por su parte Trejos (2013) aborda la importancia de la programación estructurada, haciendo hincapié en que esta técnica se basa en tres estructuras básicas: lineal, alternativa y repetitiva, con respecto a esta última plantea que:

La estructura cíclica o iterativa permite que se pueda ejecutar un conjunto de varias instrucciones tantas veces como una condición lo permita, de manera que la evaluación de dicha condición consienta que se realicen las tareas iterativas que se hayan propuesto. (p. 97)

Se considera que las dos anteriores clasificaciones, que incluyen tres estructuras de control del flujo, se apegan al teorema de la programación estructurada, referenciado por Harel, David (1980) quien plantea:

El teorema de la programación estructurada establece que toda función computable puede ser implementada en un lenguaje de programación que combine sólo tres estructuras lógicas. Esas tres formas (también llamadas estructuras de control) específicamente son:

1. Secuencia: ejecución de una instrucción tras otra.

2. Selección: ejecución de una de dos instrucciones (o conjuntos), según el valor de una variable booleana.

3. Iteración: ejecución de una instrucción (o conjunto) mientras una variable booleana sea 'verdadera'. Esta estructura lógica también se conoce como ciclo o bucle.

Este teorema demuestra que la instrucción GOTO no es estrictamente necesaria y que para todo programa que la utilice, existe otro equivalente que no hace uso de dicha instrucción. (p. 383)

En la actualidad la mayoría de los autores no incluyen dentro de las estructuras de control, la lineal, sobre la base de dos fundamentos lógicos: primero, que esta estructura es de por sí, la forma elemental de flujo de cualquier programa al ser ejecutado, y en segundo lugar, debido al hecho de que mientras las estructuras alternativas y repetitivas, tienen instrucciones del lenguaje propias que las representan, la lineal adolece de este tipo de sentencias específicas de ella.

De todas formas, es válido destacar que la anterior clasificación no influye en el análisis que se está haciendo, pues para explicar cada estructura, se utiliza la forma tradicional: teoría, sintaxis y ejemplificación.

La propuesta del SICA, elaborada Hernández, Fleites y Valdés (2016), se introduce para que el alumno, además de la teoría y los ejemplos, tenga un modelo que los ayude a elaborar el algoritmo, fue probada con estudiantes en el curso 2015-2016, en la tabla que aparece a continuación pueden analizarse los resultados obtenidos.

Tabla 1. Resultados en el trabajo de control de la temática de ciclos.

Curso escolar	Cantidad de alumnos	Aprobados 4 y 5	Aprobados con 3	Desaprobados
10-11	16	6	5	5
11-12	14	4	9	6
12-13	10	3	4	3
13-14	18	7	5	6
14-15	15	6	4	5

Total antes del SICA	73	26	27	25
Porcentaje del total		31,50	36,98	34,24
15-16 (SICA aplicado)	16	10	2	4
Porcentaje		62.25	12.50	25,00

Tabla 1: Elaboración propia.

Se debe aclarar que los resultados históricos planteados anteriormente fueron obtenidos con grupos de similares características, lo que permite decir que las siguientes variables ajenas estuvieron controladas: todos aprobaron las pruebas de ingreso a la universidad; todos habían recibido como antecedente la programación conducida por eventos, mediante el IDE Visual Basic en grado 12; ninguno había recibido algún tipo de curso de programación aparte al Visual Basic; ninguno recibió atención extra docente por parte de algún profesor de programación; el profesor de la asignatura siempre fue el mismo.

Al analizar la tabla puede apreciarse que hubo una mejoría en los porcentos de alumnos aprobados con buenas notas de un 31,50% histórico, se avanzó a un 62.25% y a su vez una disminución de alumnos con bajos resultados de 36,98% histórico de alumnos aprobados con calificación de regular, se disminuyó a 12,50% y de 34,24% histórico de alumnos desaprobados; se disminuyó a 25,00%.

Esto permite aseverar que el SICA permitió mejorar los resultados de los estudiantes, pero en un estudio más refinado, teniendo como base la caracterización de los estudiantes en las categorías de Alto Aprovechamiento, Medio Aprovechamiento y Bajo Aprovechamiento se puede analizar para qué tipo de alumnos fue más efectivo el SICA. Tabla 2. Resultados en el trabajo de control de ciclos para cada tipo de alumno al aplicar el SICA.

Curso	Cantidad alumnos	Nivel Alto	Nivel Medio	Nivel Bajo	Comentario
	16	6	5	5	Los 6 alumnos de nivel alto y 4 de los alumnos de nivel medio

15-16	Obtuvieron 4 y 5	6	4	0	obtuvieron altas calificaciones, 1 alumno de nivel medio obtuvo regular y de los 5 alumnos de nivel bajo, 1 obtuvo regular y los otros 4 desaprobaron.
	Obtuvieron 3	0	1	1	
	Obtuvieron 2	0	0	4	

Tabla 2. Elaboración propia.

El análisis de la anterior tabla permite aseverar que el SICA fue útil para alumnos de alto y medio aprovechamiento, pero no así para alumnos de bajo aprovechamiento.

Además en la experiencia se pudo constatar que la SICA es útil para revisar un programa elaborado y que, a alumnos de un aprendizaje medio y alto, les resulta útil para organizar sus ideas, llegar a elaborar la solución de los problemas y comprobar su respuesta, en cambio a los alumnos de bajo aprovechamiento, les brinda una vía para comprobar si lo que hicieron está correcto o no, al igual que al resto de los alumnos, pero no logran encontrar soluciones acertadas con su uso, esto se puede corroborar analizando los resultados que se aprecian en la siguiente tabla, que son resumen de una encuesta aplicada a los estudiantes.

Tabla 3. Respuestas a las preguntas relacionadas con el uso de SICA brindadas por alumnos que lo utilizaron.

Curso	Cantidad de alumnos encuestados	Nivel alto	Nivel medio	Nivel bajo
15-16	16	6	5	5
	El SICA te ha servido para organizar ideas.	6	4	1
	El SICA te ha servido para hallar la solución del problema	6	4	0
	El SICA te ha servido para comprobar la solución encontrada por ti.	6	5	5

Tabla 3: Elaboración propia.

El anterior SICA está elaborado en función del pensamiento lógico del alumno, tiene la característica de que no responde a un lenguaje de programación determinado, sino a las técnicas de programación en general, sobre todo está enfocado hacia la elaboración del algoritmo de solución del problema, al respecto muchos autores son partícipes de

que ahí radica la verdadera esencia de la enseñanza de la programación, al respecto: Oviedo (2016) plantea:

El profesor de programación no debe olvidar que su mejor esfuerzo lo debe orientar siempre a que el estudiante adquiera el conocimiento sobre las técnicas de programación, a la definición adecuada de los datos y al desarrollo de la habilidad necesaria para el diseño y construcción de los algoritmos con un estilo disciplinado que les dé orden y claridad. (p. 6).

Las consultas. Un espacio para la atención diferenciada a estudiantes.

Según la resolución que rige el trabajo docente metodológico en la educación superior cubana, existen diferentes formas organizativas para organizar el proceso docente educativo: Resolución 210/2007

ARTÍCULO 104: Las formas organizativas fundamentales del proceso docente-educativo en la educación superior son:

La clase.

La práctica de estudio.

La práctica laboral.

El trabajo investigativo de los estudiantes.

La autopreparación de los estudiantes.

La consulta.

La tutoría. (Ministerio de Educación Superior, p. 22)

Dentro de estas formas organizativas, las consultas son un espacio que por su tipología responden esencialmente a los intereses de los autores del artículo, en aras de apoyar en el aprendizaje a los alumnos que tienen dificultades en el aprendizaje del contenido de los ciclos, esto se aprecia en la propia esencia de este espacio. Resolución 210/2007.

ARTÍCULO 128: La consulta es una de las formas organizativas del proceso docente que tiene como objetivo fundamental que los estudiantes reciban orientación pedagógica y científico-técnica mediante indicaciones, orientaciones, aclaraciones y respuestas de los profesores a las preguntas formuladas en relación con la auto preparación. Puede realizarse de forma individual o colectiva, presencial o no presencial utilizando las tecnologías de la información y

las comunicaciones. Su frecuencia depende de las necesidades individuales y grupales de los estudiantes.

El profesor puede citar a consulta a los estudiantes que a su juicio lo requieran, en atención a las diferencias individuales. A su vez, los estudiantes podrán solicitar al profesor la realización de consultas cuando así lo consideren. (p. 27)

MATERIALES Y MÉTODOS

Desde los momentos iniciales del estudio de la técnica de programación estructurada, es decir, cuando se emprende la enseñanza de las estructuras lineales y alternativas se recomienda utilizar los siguientes modelos:

Planteado por Expósito et al. (2001), se puede apreciar el siguiente modelo general, que es válido tanto para la enseñanza de los sistemas de aplicación como de las técnicas de programación.

Pasos del programa heurístico general de informática,

1. Determinar los elementos formales que integran el problema.
2. Determinar y describir los pasos principales de la solución.
3. Seleccionar las opciones necesarias y suficientes (para Sistemas de Aplicación), o codificar los pasos principales del algoritmo (para Lenguajes de Programación).
4. Controlar los resultados.
5. Introducir las acciones correctivas. (p. 62)

Para concretar los anteriores pasos generales, en la programación estructurada, los autores del artículo se acogen al siguiente modelo planteado por Vildósola et al. (1990), que se ha hecho tradicional en Cuba:

Pasos para resolver problemas en la programación estructurada.

1. Análisis del problema
 - a. [¿Qué dan como datos de entrada?
 - b. ¿Qué debo obtener como resultado final?
 - c. ¿Qué datos auxiliares debo utilizar?
 - d. ¿Cuál sería una idea primaria de solución o plan?]
2. Determinación del algoritmo de solución.
3. Escritura del programa en un lenguaje de programación [Codificar].

4. Prueba del programa [Puesta a punto].
5. Documentación del programa [Solo para nivel empresarial o resultado científico, no para ejercicios de clase].
6. Ejecución y mantenimiento del programa (p. 9) [solo para nivel empresarial].

A continuación, se describe la idea esencial que se aplicó con los alumnos de bajo aprovechamiento, partiendo del presupuesto de que el alumno se va a enfrentar a la solución de un problema de la estructura de ciclos, perteneciente al paradigma de la programación estructurada.

Lo tradicional en la enseñanza de la programación estructurada es aplicar los pasos expresados anteriormente, siempre brindando como soporte al alumno conceptos esenciales y ejemplos resueltos, muchas veces con comentarios explicativos, en el caso del SICA, es una vía que se introduce entre los pasos 1 y 2, es decir, es un apoyo para elaborar el algoritmo, lo novedoso aplicado a alumnos de bajo aprovechamiento son dos pasos a realizar antes de utilizar la SICA: el primero consiste en resolver el problema manualmente con un juego de datos lo más exhaustivo posible y analizando el modelo elaborado; responder unas preguntas de impulso al pensamiento lógico, las respuestas a esas preguntas le permiten al alumno poder aplicar el SICA y por ende elaborar el algoritmo de solución del problema, entonces el SICA quedaría transformado de la siguiente forma.

1. Análisis del problema.

- a. Los alumnos deben establecer un juego de datos. Siempre se les sugiere que escojan entre 6 y 8 para la cantidad de datos, pues esa cantidad no hace tedioso el trabajo y permite asimilar el proceder matemático. También se les plantea a los alumnos que sean exhaustivos al escoger los datos.
- b. Analizar los datos involucrados en la entrada de información y establecer su tipología, acorde a las posibilidades que existan en el lenguaje de programación a utilizar.
- c. Analizar los datos involucrados en la salida de información y los auxiliares de cálculo para establecer su tipología, acorde a las posibilidades que existan en el lenguaje de programación a utilizar.

- d. Establecer una idea para solucionar el problema (Plan).
2. Elaborar el algoritmo de solución.
 - a. Resolver manualmente el problema.
 - b. Responder las siguientes preguntas de impulso al pensamiento lógico analizando el modelo matemático elaborado.
 - ¿Se conoce la cantidad de repeticiones del ciclo?, en caso de que no se conozca ¿se sabe quién es el último dato?
 - ¿Qué tipo de ciclo está presente?
 - ¿Qué datos hay que entrar dentro del ciclo?
 - ¿Qué datos parten de un valor inicial para irse construyendo paso a paso?
 - ¿En qué valor deben ser inicializados? ¿Por qué?
 - ¿En qué parte deben ser inicializados? ¿Por qué?
 - ¿Hay que hacer impresiones para cada juego de datos? ¿Cuáles?
 - ¿Hay que hacer condiciones para cada juego de datos? ¿Cuáles son?
 - ¿Qué cálculos finales hay que hacer?
 - c. A partir del modelo construido, las respuestas a las preguntas de impulso y utilizando el SICA elaborar el algoritmo de solución.
3. Codificar el algoritmo.
 4. Prueba del programa (Poner a punto).

RESULTADOS, ANÁLISIS Y DISCUSIÓN

La experiencia fue aplicada, con los alumnos de bajo aprovechamiento, del grupo de tercer año de la carrera Educación Laboral e Informática de la facultad de Ciencias Técnicas; esta población consta de 9 alumnos de un grupo de 19 estudiantes que en total tiene el grupo, todos los alumnos muestran las mismas condiciones para las variables ajenas que los grupos analizados desde el curso 2010-2011.

La esencia de la aplicación de la experiencia se describe a continuación: desde las clases iniciales se realizó una caracterización del grupo y se detectaron los alumnos con dificultades para enfrentar la programación estructurada en general, esta caracterización se basó en los resultados obtenidos en la asignatura Fundamentos de

Programación que dichos alumnos cursaron el año anterior, pero a pesar de tener ese diagnóstico, se aplicó el SICA directamente a todos los estudiantes, dando un margen a incorporar o eliminar alumnos de la lista de los alumnos con deficiencias.

Después de tres clases prácticas aplicando el SICA, se creó el listado de alumnos, en este momento inicial se incorporaron 5 alumnos, es decir desde las estructuras más simples (se imparten en las 6 primeras clases de la asignatura), ya se habían detectado 5 alumnos con dificultades, al adentrarse en la asignatura, estudiando la estructura de ciclos, se incrementó ese listado con 4 alumnos, para el total de 9.

Estos 9 alumnos fueron citados a consulta, en la primera consulta se explicó el método a aplicar, esto se hizo en un momento inicial de la consulta y se logró el convencimiento de los alumnos que necesitaban tiempo adicional al resto del aula y que al final, aunque requería de ellos un esfuerzo extra, eso redundaría en poder apropiarse del conocimiento con un nivel de calidad que les permitiría aprobar la asignatura, incluso con buenas calificaciones y les aportaría de un proceder para aprender a elaborar algoritmos de programación.

En un segundo momento dentro de la consulta el profesor les ejemplificó, cómo aplicar el método con dos ejemplos, uno sencillo y otro de complejidad media. Después se orientó a los alumnos resolver un ejercicio de nivel medio y otro que había sido orientado dentro del estudio independiente de la clase anterior.

Después de esta primera consulta los estudiantes fueron citados a cinco consultas más, donde se siguió la idea de analizar lo elaborado por los alumnos y la orientación de nuevas tareas.

La anterior experiencia fue montada mediante un preexperimento, donde el tratamiento introducido fue la modificación al SICA, como medición inicial se tomó el diagnóstico que permitió seleccionar los alumnos de bajo aprovechamiento y como medición final los resultados obtenidos en el trabajo de control de la temática de ciclos.

Tabla 4. Resultados en el trabajo de control de ciclos, de la aplicación de la modificación del SICA a alumnos de bajo aprovechamiento.

Curso	Cantidad de alumnos	Bajo aprovechamiento	Obtuvieron 5 y 4	Obtuvieron 3	Obtuvieron 2
16-17	19	9	2	6	1
Porcentaje			22,22	66,66	11,11
Comentario:			2 alumnos de bajo aprovechamiento obtuvieron buenas notas, solo 1 alumno de esa categoría desaprobó y el resto aprobó con calificación de regular.		

Tabla 4: Elaboración propia.

A continuación, se puede observar un ejemplo de la vía aplicada.

El problema orientado a resolver es el siguiente: Se tiene un listado conformado con el nombre y apellidos, la edad, el sexo y las calificaciones obtenidas en los dos trabajos de control, en la asignatura computación, de un grupo de n alumnos de 10. grado de un preuniversitario. Se desea obtener: el promedio de notas de los trabajos de cada uno de los n alumnos y el porcentaje de alumnas desaprobadas en ambos trabajos de control. Recuerde que en el caso de computación los trabajos de control se califican en base a 40 puntos como máximo, así que el alumno está aprobado si tiene una calificación mayor o igual a 24.

Esta es la solución al problema.

Algoritmo: Promedio Total y Porcentaje con Filtro.

1. Análisis del problema.

a. Los alumnos deben establecer un juego de datos. Siempre se les sugiere que escojan entre 6 y 8 para la cantidad de datos, pues esa cantidad no hace tedioso el trabajo y permite asimilar el proceder matemático. También se les plantea a los alumnos que sean exhaustivos al escoger los datos, en este caso hay que tener varones y hembras y en el caso de las hembras debe haber desaprobadas en ambos trabajos de control.

Un juego de datos posible es:

$N=7$

Alumno	1	2	3	4	5	6	7
Nombre y apellidos	Pedro Pérez López	Yuniesky Hernández García	Marta Ferrer Martín	Yanela Prado Rosado	Luis Marín Sánchez	Ana Del Sol Mesa	Kirenia Morán País
Sexo	m	m	f	f	m	f	F
Nota 1	28,5	22,5	30,0	24,5	21,5	21,5	37,0
Nota 2	36,0	38,0	21,5	28,0	20,0	22,5	39,5

Tabla 5: Elaboración propia.

b. Analizar los datos involucrados en la entrada de información y establecer su tipología, acorde a las posibilidades que existan en el lenguaje de programación a utilizar.

En este ejemplo serían:

N de tipo entero.

Nombre y apellidos de tipo string (texto).

Sexo de tipo char (letra).

Nota1 de tipo real (número con decimales).

Nota 2 de tipo real (número con decimales).

c. Analizar los datos involucrados en la salida de información y los auxiliares de cálculo para establecer su tipología, acorde a las posibilidades que existan en el lenguaje de programación a utilizar.

En este ejemplo serían:

Promedio: suma entre cantidad, es un número de tipo real, puede tener decimales.

Suma: es un dato auxiliar, como va a sumar notas de tipo real, entonces también es de tipo real.

Cantidad: es la propia n, es decir la cantidad de datos

Porcentaje: $\text{parte}/\text{total} \times 100$, es un número de tipo real, puede tener decimales.

Parte: es un contador que se incrementa cada vez que sea una alumna, que desaprobó ambos trabajos de control. Los contadores son de tipo entero.

Total: es un contador que se incrementa cuando es una alumna. Los contadores son de tipo entero.

d. Establecer una idea para solucionar el problema (Plan).

En este caso sería:

Promedio: es una suma entre una cantidad. Hay que sumar las dos notas de cada alumno y dividir las entre 2. Esto hay que hacerlo n veces.

Porcentaje de alumnas desaprobadas en los dos trabajos de control: Hay que ir contando a las alumnas (contador 1), después se comparan las notas de cada alumna, si las dos son menores de 24 se cuenta a esa alumna (contador 2), al final se divide el contador 2 entre el contador 1, se multiplica la división por 100, y ese es el resultado.

2. Elaborar el algoritmo.

a. Modelaje matemático.

Alumno 1 (notas: 28,5 y 36,0)

Suma = 64.5

Promedio = 32.25

Sexo = m

no hay que hacer más nada.

Alumno 2. (notas: 22,5 y 38,0)

Suma = 60.5

Promedio = 30.25

Sexo = m

no hay que hacer más nada.

Alumno 3 (notas: 30,0 y 21,5)

Suma: 51.5

Promedio = 25.75

Sexo = f

hay que incrementar el contador 1 (1)

nota 1 > 24 y nota 2 < 24 (no hay que incrementar el contador 2)

Alumno 4 (notas: 24,5 y 28,0)

Suma: 52.5

Promedio = 26.25

Sexo = f

hay que incrementar el contador 1 (2)

nota 1 > 24 y nota 2 > 24 (no hay que incrementar el contador 2)

Alumno 5: (notas: 21,5 y 20,0)

Suma = 41.5

Promedio = 20.75

Sexo = m

No hay que hacer más nada.

Alumno 6: (notas: 21,5 y 22,5)

Suma: 44,0

Promedio = 22,0

Sexo = f

Hay que incrementar contador 1 (3)

Nota 1 < 24 y Nota2 < 24 hay que incrementar el contador 2 (1)

Alumno 7: (Notas: 37,0 y 39,5)

Suma = 76,5

Promedio: 38,25

Sexo = f

Hay que incrementar el contador 1 (4)

Nota 1 >24 y Nota 2 > 24 (no hay que incrementar el contador 2)

Porcentaje = contador2 (1)/ entre contador 1 (4) = 0,25

b. Sobre la base del modelo obtenido responder las siguientes preguntas.

- ¿Se conoce la cantidad de repeticiones del ciclo?, en caso de que no se conozca ¿se sabe quién es el último dato?

Respuesta

- Se conoce la cantidad de repeticiones.

- ¿Qué tipo de ciclo está presente?

Respuesta.

- Ciclo determinado.

- ¿Qué datos hay que entrar dentro del ciclo?

Respuesta.

- Sexo
- Nota 1
- Nota 2

- ¿Qué datos parten de un valor inicial para irse construyendo paso a paso?

Respuesta.

- Los dos contadores.
- El sumador.

- ¿En qué valor deben ser inicializados? ¿Por qué?

Respuesta.

- Los contadores en cero, porque el primer valor a incrementar es 1 y se van incrementando de uno en uno.
- La suma, también empieza en cero para evitar que se acumule lo obtenido por un alumno cuando se van a sumar las dos notas del siguiente alumno.

- ¿En qué parte deben ser inicializados, por qué?

Respuesta.

- Los contadores, al inicio, pues se utilizan en cálculo general.
- El sumador al inicio del ciclo, pues es privado de cada alumno.

- ¿Hay que hacer impresiones para cada juego de datos? ¿Cuáles?

Respuesta.

- El promedio de cada alumno.

- ¿Hay que hacer condiciones para cada juego de datos? ¿Cuáles son?

Respuesta.

- Si $sexo = f$ entonces
 - incrementar contador 1
 - Si $nota1 < 24$ y $nota 2 < 24$ entonces incrementar contador 2.

- ¿Qué cálculos finales hay que hacer?

Respuesta.

- El porcentaje.

c. Establecer el algoritmo de solución, aplicando el SICA, sobre la base del modelo matemático y las respuestas a las preguntas de impulso, relacionadas con el modelo.

SICA para ciclos determinados	Algoritmo
a. Antes del ciclo: <ul style="list-style-type: none"> i. Inicializaciones ii. Entrar n b. Dentro del ciclo <ul style="list-style-type: none"> i. Entrar datos ii. Procesos repetitivos <ul style="list-style-type: none"> 1. Inicializar acumuladores privados de cada juego de datos 2. Hacer acumuladores 3. Los if para mayor y menor 4. Los if para algún criterio 5. Impresiones para cada juego de datos. c. Fuera del ciclo <ul style="list-style-type: none"> i. Cálculos generales (promedios, porcentajes) ii. Impresiones generales 	Contador 1 \leftarrow 0 Contador 2 \leftarrow 0 Entrar n Para i:=1 hasta n hacer Suma \leftarrow 0 Entrar sexo Entrar nota 1 Entrar Nota 2 Suma \leftarrow Nota 1+ Nota 2 Promedio = suma /2 Si sexo = f entonces Contador 1 \leftarrow Contador 1 +1 Si Nota 1 < 24 y Nota 2 < 24 entonces Contador 2 \leftarrow Contador 2 + 1 Fin Si Fin Si Mostrar Promedio. Fin Para. Porcentaje \leftarrow contador 2 / contador 1 * 100 Mostrar porcentaje.

Tabla 6: Elaboración propia.

3. Codificar (es transcribir el algoritmo a un lenguaje de programación, en este caso se escogió el Pascal).

Algoritmo encontrado	Programa Pascal.
Contador 1 \leftarrow 0 Contador 2 \leftarrow 0 Entrar n Para i:=1 hasta n hacer	Var c1,c2:integer; s:real; i,n:integer;

<p>Suma \leftarrow 0</p> <p>Entrar sexo</p> <p>Entrar nota 1</p> <p>Entrar Nota 2</p> <p>Suma \leftarrow Nota 1+ Nota 2</p> <p>Promedio = suma /2</p> <p>Si sexo = f entonces</p> <p> Contador 1 \leftarrow Contador 1 +1</p> <p> Si Nota 1 < 24 y Nota 2 < 24 entonces</p> <p> Contador 2 \leftarrow Contador 2 + 1</p> <p> Fin Si</p> <p>Fin Si</p> <p>Mostrar Promedio.</p> <p>Fin Para.</p> <p>Porcentaje \leftarrow contador 2 / contador 1 * 100</p> <p>Mostrar porcentaje.</p>	<pre> sexo:char; n1,n2: real; prom:real; porc:real; begin c1:=0; c2:=0 write('Entre n '); readln(n); for i:=1 to n do begin s:=0; write('Entre nota 1 ');readln(n1); write('Entre nota 2 ');readln(n2); s:=n1+n2; prom:=s/2; if sexo = 'f' then begin c1:=c1+1; if (n1<24) and (n2<24) then c2:=c2+1; end; end; por:=n2/n1*100; writeln('El porcentaje de alumnas desaprobadas es ',por:6:2); readln; end. </pre>
--	---

Tabla 7: Elaboración propia.

4. Para probar el programa se orienta a los alumnos que acudan al laboratorio 1 de la UNISS “José Martí Pérez” y utilicen allí el IDE Delphi, en su modo consola.

CONCLUSIONES

Con este trabajo se ofrece una vía, que consiste en una modificación a un SICA, obtenida con el método de la modelación que permite a los estudiantes de bajo aprovechamiento alcanzar los objetivos deseados en cuanto al estudio de la programación estructurada, específicamente en lo referido a la estructura repetitiva.

Esta vía, para ese tipo de alumnos, necesita más tiempo que la estrategia común, que es aplicar la analogía y resolver problemas nuevos a partir de ejemplos resueltos, por eso se utilizó la consulta como forma de docencia para su aplicación.

REFERENCIAS BIBLIOGRÁFICAS

Borjas, V. (2013). *Unidad III. Estructuras de control*. Recuperado de <https://profvanessaborjas.files.wordpress.com/2013/02/unidad-iii-estructuras-de-control.pdf>

Capetillo, I. (2017). *Lógica de Programación. Estructura mientras–hacer*. Recuperado de <http://www.israelcapetillo.com/2015/04/logica-de-programacion-16-esctructura.html>

Expósito, C., Cruañas, J., Gener, E., De la Noval, N., Rivero, A.y Peñalver, L. (2001). *Elementos de la Metodología de la Informática*. La Habana, Cuba: Pueblo y Educación.

Hareld, D. (1980). On Folk Theorems. *Revista ACM*, 23(7), 379–389. doi:10.1145/358886.358892.

Hernández, E., Hondal, V. y González, N. (2005). *Los lazos: Su didáctica bajo la luz de la programación visual*. 10 Congreso Internacional de Educación Superior. Universidad 2016. Sancti Spíritus Recuperado de <http://biblioteca.uniss.edu.cu/sites/default/files/CD/2015%20Universidad%202016/personal-injury/c9/c5.pdf>

Hernández, E., Martínez, Y. y Fleites, L. (2012). Utilización de Problemas en las clases de Programación para la motivación de los alumnos. *Pedagogía y Sociedad*, 15(35). Recuperado de <http://revistas.uniss.edu.cu/index.php/pedagogia-y-sociedad/article/viewFile/290/221>

Hernández, E., Machín, I. y Fleites, L. (2015). *Los lazos. Su didáctica bajo la luz de la programación estructurada*. [CD]. Memorias de Universidad 2016. Sancti Spíritus, Cuba: Universidad de Sancti Spíritus “José Martí Pérez”.

Hernández, E., Fleites, L. y Valdés, L. (2016). Indicaciones con carácter algorítmico para la enseñanza de los ciclos. *Pedagogía y Sociedad*, 19(46). Recuperado de <http://revistas.uniss.edu.cu/index.php/pedagogia-y-sociedad/article/view/455/383>

Neira, A. (2016). *Condicionales y Ciclos en C++*. Recuperado de <http://programacionjand.blogspot.com/p/condicionales-y-ciclos-en-c.html>

Martínez, A. (2012). *Concepto de algoritmo, Diagrama de flujo y pseudocódigo*. Recuperado de <https://andresmtzg.wordpress.com/2012/09/27/concepto-de-algoritmo-diagrama-de-flujo-y-pseudocodigo>

Ministerio de Educación Superior (2007). *Resolución 2010/2007*. La Habana, Cuba: Autor. Versión Digital.

Oviedo Galeano, M. (2016). *La enseñanza de la programación*. Recuperado de <http://bibliotecadigital.conevyt.org.mx/colecciones/documentos/somece2002/Grupo4/Oviedo.pdf>

Tahual, M. (2017). *Los tipos de ciclos de programación*. Recuperado de <https://prezi.com/yxagx3-osr1w/los-tipos-de-ciclo-de-programacion/>

Trejos, O. (2013). *Relaciones de aprendizaje significativo entre dos paradigmas de programación a partir de dos lenguajes de programación*. *Revista Tecnura*, 18(41), 91-102. Recuperado de <https://revistas.udistrital.edu.co/ojs/index.php/Tecnura/article/view/7025>

Universidad Abierta Interamericana (UAI) (2017). *El futuro sos voz: Programación Estructurada*. Recuperado de <http://pestructurada-uai.blogspot.com/2011/10/tipos-de-estructuras-en-la-programacion.html>

Vildósola, S., Del Cañal, E., González, S., Mateu, M., Toledo, M. y De la Torres, E. (1990). *Fundamentos de Programación*. La Habana, Cuba: Pueblo y Educación.