



MAPA CONCEPTUAL “TIPOS ABSTRACTOS DE DATOS” Y SISTEMA DE VISUALIZACIÓN DE PROGRAMAS SVP-SUBC: HERRAMIENTAS EFICACES EN LA FORMACIÓN VIRTUAL DEL INGENIERO INFORMÁTICO.

M. Sc. Yolanda Soler Pellicer ⁽¹⁾; PhD. Mateo Gerónimo Lezcano Brito ⁽²⁾; M. Sc. Isvani Frías Blanco ⁽¹⁾; M. Sc. Manuel José Linares Alvaro ⁽¹⁾; Ing. Edel Ángel Rodríguez Sánchez ⁽¹⁾

⁽¹⁾ Universidad de Granma, Cuba; ⁽²⁾ Universidad Central “Marta Abreu” de Las Villas, Cuba

yoly@udg.co.cu, mlezcano@uclv.edu.cu, ifriasb@udg.co.cu, cheche@udg.co.cu,
edel@udg.co.cu

Resumen: La comprensión de los diferentes niveles de abstracción en el diseño de los datos, la selección de estructuras y la implementación de algoritmos óptimos que las manipulen, constituyen un problema a solucionar en las carreras de perfil informático, más aún, al desarrollar el proceso docente en un modelo a distancia. Se propone el diseño de un lenguaje de programación que facilita el proceso de visualización (SubC), así como un Sistema para la Visualización de Programas en este lenguaje (SVP-SubC), que permite la representación gráfica de los elementos estáticos y dinámicos de programas implementados en este lenguaje. El sistema se vincula al mapa conceptual “Tipos Abstractos de Datos”, como una aplicación que permite diseñar, visualizar y probar los algoritmos; el mapa se encuentra insertado como un recurso de la asignatura Estructura de Datos en la plataforma Moodle de la Universidad de Granma, Cuba. Esta experiencia institucional proporciona un complemento de inestimable valor en el proceso de enseñanza-aprendizaje, aumentando la comprensión de conceptos de un alto nivel de abstracción y



29 de octubre al 9 de noviembre de 2008

complejidad, apoyan el autoaprendizaje y el trabajo colaborativo en un modelo de Enseñanza a Distancia.

Palabras Claves: Visualización de programas, Estructuras de Datos, Mapas Conceptuales, Enseñanza a Distancia.

Eje Temático: Calidad y Materiales educativos y Herramientas tecnológicas en Educación a Distancia.

1. INTRODUCCIÓN

La educación en entornos virtuales es una modalidad de educación a distancia, cuya particularidad es la integración de tecnología al proceso de enseñanza aprendizaje. Estas herramientas configuran de una manera particularmente diferente los procesos socio-cognitivos del aprendiz y las situaciones interactivas alumno – alumno, alumno – tutor y tutor – tutor (Kearns y Vazirani, 2001).

En el proceso de enseñanza-aprendizaje de la asignatura Estructura de Datos, generalmente, para lograr una interpretación acabada de un programa se realiza el estudio del código fuente, que es incómodo y de difícil aplicación o la construcción de casos de prueba para explicar la conducta de un programa, tarea penosa y especulativa. Según Clinton (1999), Jeffery (2001), Señas y Moroni (2003), estas dificultades motivan el desarrollo de programas especiales que son usados para ayudar a explicar la conducta de otros programas.

La educación universitaria en Cuba transita hacia un modelo de educación semipresencial, lo que exige una mayor calidad en los medios y tecnologías que se pongan a disposición de los docentes y estudiantes, así como de plataformas que apoyen el proceso de enseñanza-aprendizaje y soporten estos recursos. A través de la herramienta didáctica propuesta en el presente trabajo, es posible el manejo en ambas direcciones (MES, 2003).

Es por ello que en el Centro de Estudios de Informática Educativa (CEIE) de la Universidad de Granma y el Centro de Estudios Informáticos (CEI) de la Universidad Central “Marta Abreu” de Las Villas, Cuba, se considera necesario cambiar el enfoque desarrollado hasta el momento y aprovechar las ventajas que brindan las herramientas informáticas para contribuir a que los estudiantes puedan solucionar problemas complejos mediante el diseño óptimo de Tipos Abstractos de Datos (TAD) para representar la información y las operaciones que sobre ellos se realizan, usando técnicas de visualización de programas; consultando la información contenida en el curso de Estructuras de Datos de la plataforma Moodle de la Universidad de Granma, aprovechando las facilidades que la misma proporciona para el trabajo colaborativo en grupos y la interacción entre los sujetos del proceso de enseñanza-aprendizaje, revisando la relación entre conceptos organizados en el mapa conceptual Tipos Abstractos de Datos que se inserta como recurso a la plataforma.

Se trazan los siguientes objetivos:

- Diseñar un mapa conceptual que organice el sistema de conocimientos de la



29 de octubre al 9 de noviembre de 2008

asignatura Programación II, mostrando las relaciones entre los distintos Tipos Abstractos, las estructuras de datos, los algoritmos que implementan sus operaciones, sus características, conceptos y aplicaciones.

- Diseñar una herramienta de ayuda al diseño de estructuras de datos y la visualización de algoritmos que implementan sus operaciones, usando un lenguaje de alto nivel desarrollado para la ejecución de los mismos.
- Integrar estas herramientas en a la plataforma interactiva de Educación a Distancia de la carrera de Informática de la Universidad de Granma.

La combinación de estas herramientas da respuesta a las necesidades del nuevo Plan de Estudios "D" de la carrera de Informática, el cual tiene como objetivos fundamentales la educación virtual, semipresencial, centrada en el alumno, un modelo en el que se imparten los contenidos esenciales y el alumno debe, usando los medios y las facilidades de la plataforma de e-learning que se ha escogido, construir su propio conocimiento, en interacción con el grupo y el tutor.

2. METODOLOGÍA

2.1 Mapas Conceptuales en el estudio de la programación.

La enseñanza de la programación, como parte fundamental de la formación profesional de las carreras de Ciencias de la Computación, ha sufrido cambios importantes que han estado aparejados al desarrollo de la tecnología y la enseñanza, muchos de ellos se han basado en el surgimiento de nuevos paradigmas de programación y a cambios en los entornos educativos (Castillo y Barberán, 2000).

Entre los principales obstáculos que aparecen para el aprendizaje y aplicación de un lenguaje de diseño de algoritmos pueden puntualizarse los siguientes:

- El alumno se ve necesitado de manejar un gran número de nuevos conceptos e integrarlos de manera significativa. En los algoritmos, las acciones complejas suelen definirse en términos de otras acciones más sencillas
- Las acciones tienen dos aspectos que están estrechamente relacionados entre sí: una sintaxis (reglas de redacción de las acciones en un algoritmo), y una semántica (significado formal y preciso de una acción dada).
- El alumno se enfrenta a la necesidad de manejar un lenguaje objeto (para elaborar algoritmos) y un metalenguaje (para hablar acerca de cómo se comporta el lenguaje algorítmico).
- Existen conceptos relativamente complejos interrelacionados entre sí (Chestlevar, 2001).

Escribir un programa de computadora utilizando un Lenguaje de Programación requiere del alumno varias competencias y habilidades, que involucran básicamente la capacidad de manipular un conjunto de abstracciones interrelacionadas para la resolución de problemas.



29 de octubre al 9 de noviembre de 2008

La tarea de aprender a manipular el conjunto de símbolos asociado a un lenguaje conforme a una sintaxis, relacionándolos con una semántica, demanda un esfuerzo considerable para los alumnos de los primeros años de la carrera. A esto se suma, en muchos casos, una formación deficiente que les dificulta organizar nuevos conceptos de una manera ordenada para construir taxonomías y diferenciar propiedades que permitan establecer pautas para razonar sobre ellas, esta situación puede agravarse en un modelo de enseñanza a distancia si no se traza una estrategia adecuada.

En este contexto, Chestlevar (2001) y Stojanovic (2002), proponen el uso de mapas conceptuales para la enseñanza de conceptos básicos de programación y desarrollo de algoritmos y a este criterio nos subscribimos.

Al revisar los textos tradicionales de enseñanza de programación en el ámbito universitario se comprueba que mayoritariamente no hacen uso de un lenguaje de diseño de algoritmos para enseñar a programar y, en su lugar, apelan directamente a un lenguaje de programación. Ese acercamiento prescinde, muchas veces, de una clara identificación de cómo se interrelacionan distintos conceptos teóricos entre sí, el resultado es que muchos se presentan independientemente y sólo a través de la práctica el alumno llega a interrelacionarlos. Esto puede motivar la exploración de distintas técnicas didácticas que facilitan a los alumnos una mayor comprensión y vinculación de los temas presentados.

En programación se da el caso particular de que todo concepto expresado a través de la sintaxis de un lenguaje tiene su correlación con un significado operacional (semántica), y dicho significado estará definido de manera composicional, en término del significado de otros conceptos más elementales.

Soler y Lezcano (2007), en una primera etapa de esta investigación realizan un análisis crítico del sistema de conocimientos de la disciplina Técnicas de Programación de Computadoras de la carrera Ingeniería Informática, del nivel de precedencia de los contenidos y la necesidad de vincular en la asignatura Estructura de Datos y Algoritmos los temas de diseño y análisis para obtener algoritmos eficientes, se crea un mapa conceptual, usando la herramienta cmapTools, que muestra la relación entre los Tipos Abstractos, las estructuras de datos, los algoritmos que implementan sus operaciones, características, conceptos y aplicaciones, que incluyen simulaciones de algoritmos, pero que adolece del nivel de visualización de estas operaciones, objetivo que se cumple en una segunda etapa de investigación.

2.2 Herramientas de Visualización de Programas y Algoritmos: declarativas e imperativas.

Si bien los entornos de los lenguajes de programación ofrecen capacidades cada vez más útiles al programador, como el uso de colores y los depuradores, éstos no son aún lo suficientemente amigables con el usuario como es deseable. Se ha pensado en el desarrollo de entornos específicos que ayuden y conduzcan efectivamente a los estudiantes en el proceso de aprendizaje de la programación de



29 de octubre al 9 de noviembre de 2008

manera más gráfica, diversa y comprensible, Señas y Moroni (2003), consideran que la visualización de la conducta dinámica de los programas y de sus modelos abstractos, los algoritmos, y consecuentemente, el beneficio psicopedagógico de su aplicación en la enseñanza, constituyen una importante área de investigación que aún no ha sido completamente explorada.

Estos autores plantean su experiencia en el uso combinado de los llamados Esquemas de Ejecución Algorítmica (EEA), representaciones gráficas para visualizar la ejecución de un algoritmo y los Mapas Conceptuales Hipermediales (MCH). Los EEA ayudan a comprender la relación entre el algoritmo como entidad estática y el dinamismo de su ejecución, constituyen, además, una herramienta valiosa para la etapa de prueba y verificación. Por otra parte, el alumno crea el MCH del tema sobre el que trata el problema a resolver, lo que representa una ayuda significativa para la comprensión del mismo.

Usando estos fundamentos y las herramientas de visualización declarativas e imperativas se diseña el Sistema de Visualización de Programas SVP-SubC.

2.3 Plataforma e-learning.

Se usa la plataforma interactiva Moodle de la Universidad de Granma a la que se integra el mapa conceptual Tipos Abstractos de Datos (creado con la herramienta CmapTools) y el Sistema de Visualización de Programas SVP-SubC, así como ejercicios, actividades y materiales que apoyan el proceso de enseñanza-aprendizaje en un entorno virtual (Dougiamas, 2003).

3. RESULTADOS Y DISCUSIÓN

3.1. Se logra la organización del sistema de conocimientos de la asignatura estructuras de datos y algoritmia a través del cmaptools.

Teniendo como base los fundamentos anteriores se crea el mapa conceptual **Tipos Abstractos de Datos**, usando la herramienta cmapTools. En el nivel primario del mapa se muestra el concepto **Tipos Abstractos de Datos** como un **modelo matemático** y las **operaciones** que se ejecutan sobre él, para profundizar más en el tema al nodo se le añade una página web con la definición y ejemplos.

En este nivel de su formación los alumnos son capaces de definir, debido a los conocimientos adquiridos en las asignaturas de Álgebra y Matemática I, el modelo matemático de los números enteros, complejos o las matrices y las operaciones sobre ellos, lo que sirve de base al estudio de nuevos modelos más complejos. La Figura 1 muestra el mapa que representa la unión de estos dos conceptos conocidos y la definición de un Tipo Abstracto de Datos.



29 de octubre al 9 de noviembre de 2008

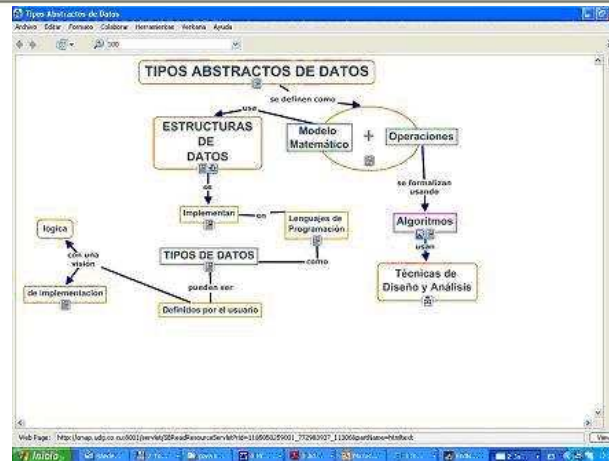


Figura 1: Mapa Conceptual Tipos Abstractos de Datos.

3.2. Se enriquece el mapa con informaciones y recursos.

El mapa conceptual se brinda información sobre todos los conceptos incluidos, a través de los recursos (documentos, imágenes, videos, páginas web, animaciones, simulaciones, mapas conceptuales) que el cmapTools permite incluir a cada nodo. De esta manera se muestran los diferentes niveles de abstracción en el diseño de los datos y su implementación.

En el caso de la operación de ordenamiento en arreglos se incluyen aplicaciones para los algoritmos de ordenamiento Bubble, QuickSort, por Inserción, Mezcla, entre otros.

En la Figura 2 se muestran las aplicaciones para los algoritmos de ordenamiento por los métodos de Bubble y QuickSort, para cada uno se ofrece un análisis de la complejidad teniendo en cuenta el mejor, peor y el caso medio, lo que permite a los estudiantes familiarizarse con el concepto de eficiencia, a la vez que va comprobando paso a paso los resultados de la ejecución y puede comprender la importancia del costo al elegir un algoritmo para solucionar un problema.



29 de octubre al 9 de noviembre de 2008

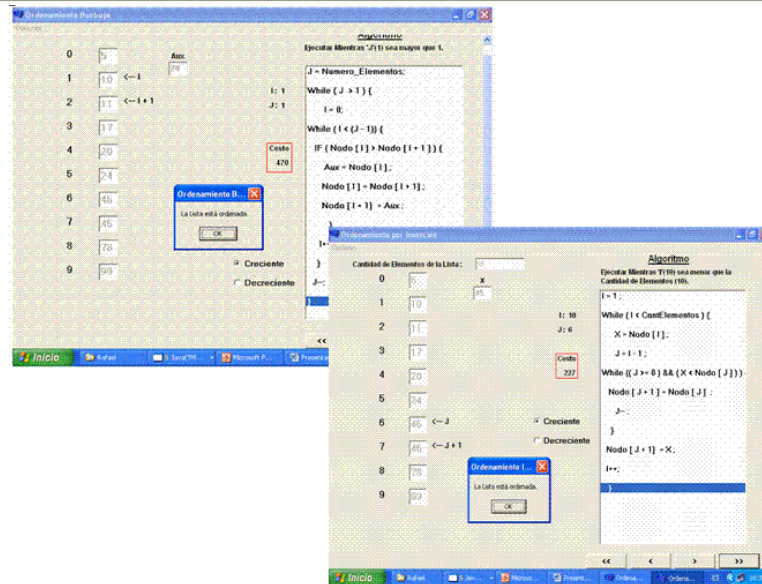


Figura 2: Aplicaciones que simulan los algoritmos de ordenamiento Bubble y QuickSort.

Se parte, para este análisis, del algoritmo de ordenamiento Bubble, ya estudiado y fácil de entender, demostrando su complejidad cuadrática y por tanto su ineficiencia al compararlo con otros. Se adiciona al nodo una página web con información general sobre esta operación (Ordenamiento.htm) y en ella se introduce la notación O y se comparan los diferentes algoritmos teniendo en cuenta esta función. Las operaciones de ordenamiento y búsqueda pueden ser más óptimas si se utilizan **Técnicas de Diseño y Análisis de Algoritmos**, es por ello que al orientar el estudio del método de Ordenamiento Rápido (QuickSort) o Búsqueda Binaria se sugiere una visita al nodo **Divide y Vencerás** de este mapa.

El nodo de ordenamiento cuenta, además, con problemas resueltos donde se utilizan algoritmos de ordenamiento, se mezclan algunos y en otros se propone a los estudiados que introduzcan variantes para mejorar su eficiencia, en cada caso se realiza el análisis de complejidad y se compara con la de otros algoritmos similares.

Una de las ventajas de la herramienta CMapTools es la posibilidad de adicionar a un nodo una lista de discusión, lo que facilita el trabajo colaborativo y la interactividad, alumno-alumno y alumno-profesor. Para un mismo concepto pueden crearse varias listas de discusión. La generación del mapa en formato html permite su uso aunque el cmapTools no se encuentre instalado como cliente, la dirección para acceder al sitio cmap de la Universidad de Granma es <http://cmap.udg.co.cu>.

En esta fase de la investigación se logra un sistema bibliográfico novedoso y una herramienta de organización que muestra la esencialidad de cada contenido y la interrelación entre ellos, entendiendo que este contenido parte de lo simple a lo complejo, teniendo en cuenta la inclusión del conocimiento.

3.3. Se implementa el Sistema de Visualización de Programas en el lenguaje

29 de octubre al 9 de noviembre de 2008

SubC, herramienta que eleva la calidad del proceso docente de la asignatura Estructura de Datos.

En la Figura 3 se muestra la inserción de un recurso importante en el mapa conceptual, el Sistema de Visualización de Programas SVP-SubC, desarrollado como una segunda etapa de esta investigación (Frías, Soler y Lezcano, 2007).

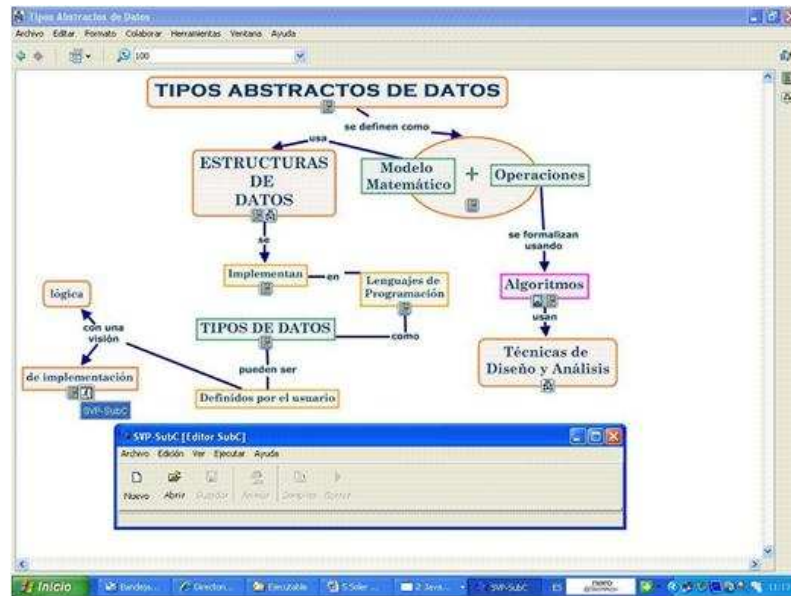


Figura 3: Sistema de Visualización de Programas SVP-Sub-C como recurso del mapa conceptual Tipos Abstractos de Datos.

SVP-SubC consta de tres funcionalidades principales: edición, ejecución y animación de programas escritos en SubC. La interfaz principal de SVP-SubC incluye facilidades de edición y opciones para ejecutar acciones relacionadas con la animación y ejecución de programas –como Animar, Ejecutar y Compilar. La ejecución de un programa incluye una ventana –Consola SubC– que realiza la función de interfaz para la entrada – salida del mismo. El proceso de animación consta de cuatro ventanas: la interfaz principal, la interfaz de entrada–salida, la ventana de visualización de programas y la ventana de visualización del árbol de llamadas a funciones, se muestra el código del programa que se encuentra en el proceso de animación, resaltando la última línea ejecutada. La interfaz de entrada – salida cumple la misma función que en el proceso de ejecución. Estas funciones pueden apreciarse en la Figura 4.



29 de octubre al 9 de noviembre de 2008

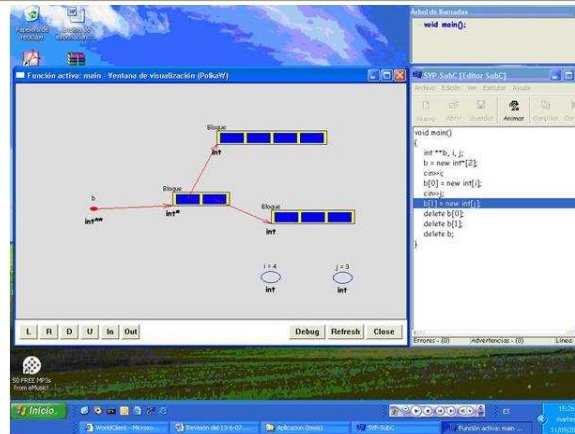


Figura 4: Visualización-ejecución de un programa escrito en SubC.

3.4. El SVP-SubC facilita la visualización del árbol de llamadas a funciones.

Uno de los problemas detectados en la comprensión de los algoritmos está relacionado con el uso de la recursividad, hasta el momento – sin el uso del sistema – para explicar este proceso se representaba el árbol de llamadas a las funciones de forma gráfica en la pizarra o usando una representación estática del mismo a través de una imagen digital. El sistema SVP-SubC brinda la opción de visualizar el árbol de llamadas y si el estudiante realiza algún cambio en el algoritmo o en los parámetros de entrada a la función podrá comprobar cómo varía el árbol de llamadas, lo que facilita, no sólo la comprensión, sino también la explicación de estos procedimientos, como se muestra en la Figura 5.

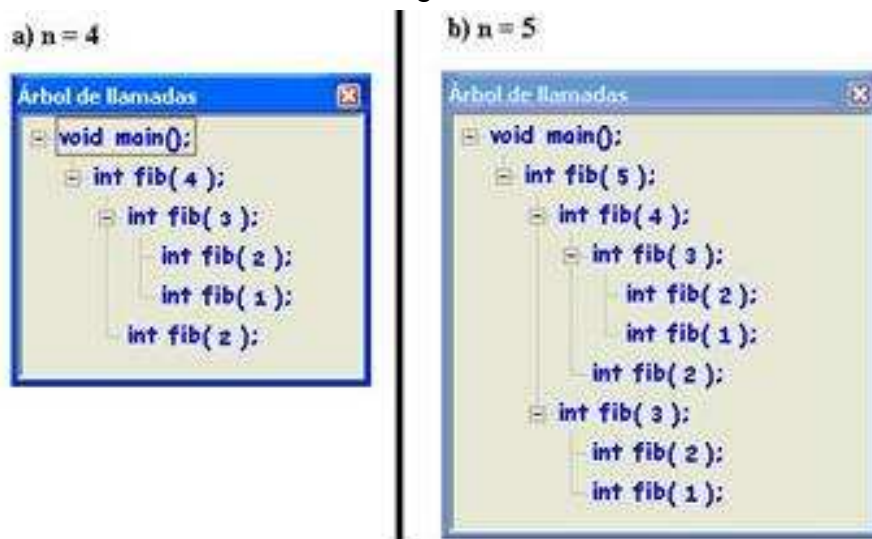


Figura 5: Árbol de llamadas a la función de Fibonacci. a) para $n=4$ b) para $n=5$.

3.5. El sistema contribuye al diseño de los datos.

El sistema cuenta con un banco de ejemplos consistentes en programas escritos

en SubC. Además da la posibilidad de editar algoritmos así como transformar el código de dichos ejemplos. El uso del banco de ejemplos permite al diseñador, a través de la visualización, concentrarse en cómo son usados los datos en el sistema para resolver el problema, sin tener que preocuparse por la representación y tratamiento de los datos en la memoria del computador; este sistema contribuye a que el estudiante comprenda la utilidad de realizar un buen diseño, partiendo del TAD, su modelo matemático y operaciones.

Los estudiantes pueden constatar, además, que la verdadera utilidad de un TAD aparece en el diseño de nuevos tipos de datos, que es la habilidad generalizadora que deben lograr en la asignatura, y que la especificación de un TAD no conlleva ninguna consideración de implementación. Estos ejemplos presentan soluciones a diferentes problemas que utilizan diversas técnicas de programación y algoritmos que manejan diferentes estructuras de datos.

3.6. Visualización de Algoritmos.

En la Figura 6 se pueden apreciar los efectos que provoca el uso de instrucciones en diferentes órdenes; dando lugar a la ejecución que se muestra en la misma figura.

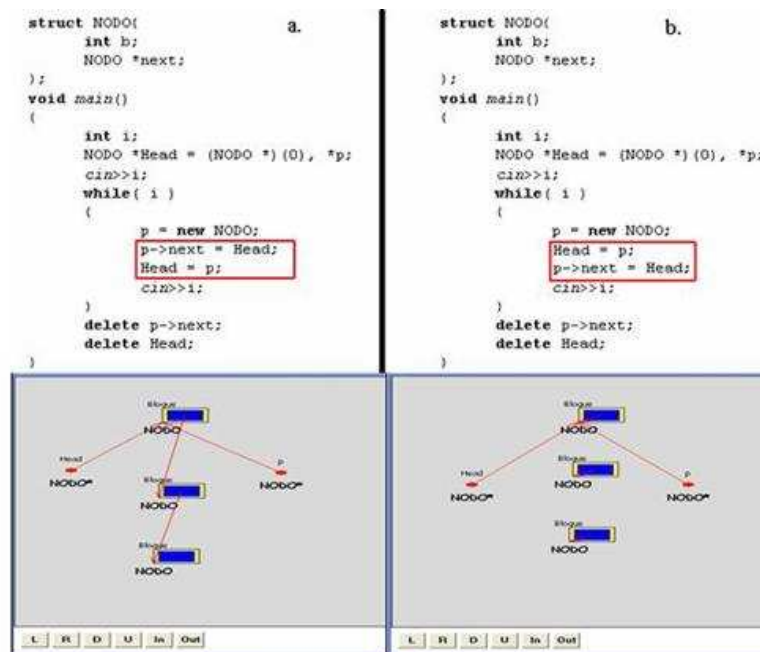


Figura 6: Resultado de la visualización – ejecución del código correcto (a) y con intercambio de instrucciones (b).

3.7. Ayuda a la selección de estructuras de datos.

A través de la visualización, SVP-SubC puede mostrar el coste que puede ocasionar una mala selección de dichas estructuras como solución a un determinado problemas, es posible la visualización de estructuras de datos más complejas como árboles, listas múltiples y grafos en general. Además, ofrece herramientas para la

edición y depuración de programas.

3.8. Se logra integrar estas herramientas a la plataforma de Educación a Distancia de la asignatura Estructura de Datos.

Los recursos y herramientas antes mencionados se han integrado a la plataforma de Educación a Distancia de la Universidad de Granma, en el curso de Estructura de Datos. (Figura 7)

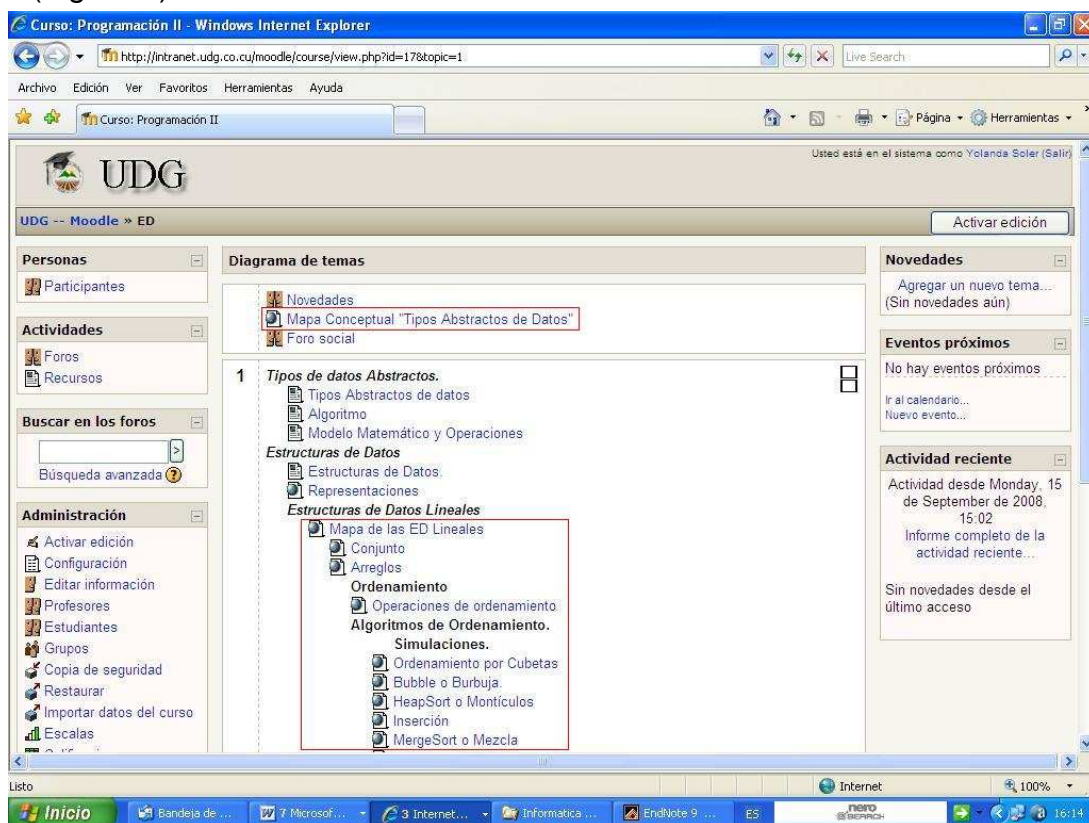


Figura 7: Curso de Estructura de Datos en la plataforma Moodle de la Universidad de Granma.

El uso de esta plataforma interactiva fomenta estrategias de enseñanza-aprendizaje constructivistas y apoya los pilares de la misma: aprender a aprender, aprender a hacer, aprender a ser y aprender a convivir con los demás, ya que en la asignatura se trabaja por proyectos, en equipos.

Se aprovechan las ventajas de la plataforma Moodle para potenciar la relación entre los sujetos del proceso de enseñanza-aprendizaje y la labor del tutor.

A partir del mes de septiembre de 2008 la carrera de Ingeniería Informática se inicia en un Modelo Semipresencial en las Sedes Universitarias de dos municipios de la provincia Granma. El curso de Estructuras de Datos montado sobre esta plataforma resulta, entonces, imprescindible para satisfacer las exigencias de este modelo, mantener y elevar la calidad del proceso de enseñanza-aprendizaje, donde



29 de octubre al 9 de noviembre de 2008

el papel del tutor es muy importante y el estudiante se convierte en verdadero protagonista y planificador de su propio proceso de aprendizaje.

4. CONCLUSIONES Y TRABAJO FUTURO.

- La vinculación de componentes multimediales y de simulación de procesos contribuyó al desarrollo del mapa conceptual Tipos Abstractos de Datos que constituye un sistema bibliográfico novedoso y una herramienta de organización del sistema de conocimientos de la asignatura, que propicia la comprensión del contenido y la interrelación entre conceptos.
- El Sistema de Visualización de Programas en el lenguaje SubC, insertado como recurso al mapa, realiza la ejecución real de programas ofreciendo facilidades para la visualización de variables, bloques de memoria y llamada a funciones en correspondencia con dicha ejecución, ofreciendo instrumentos para la edición y depuración de programas.
- Las herramientas diseñadas se integran como recursos a la plataforma de Educación a Distancia que usa la asignatura para el desarrollo del proceso de enseñanza-aprendizaje y da respuesta al Plan de Estudios “D” de la carrera que se centra en la virtualidad.
- Las Tecnologías de la Información y las Comunicaciones y las Técnicas de Programación puestas al servicio de la educación contribuyen a elevar la calidad del proceso de enseñanza-aprendizaje, a la vez que constituyen un objeto de estudio en la carrera de Ingeniería Informática.
- La carrera de Ingeniería Informática continúa trabajando en el diseño de medios y estrategias de enseñanza-aprendizaje que apoyen el modelo de Educación a Distancia.

5. REFERENCIAS BIBLIOGRÁFICAS



29 de octubre al 9 de noviembre de 2008

- Castillo, J. y Barberán, O. (2000). Mapas Conceptuales en Matemáticas. Disponible en: <http://www.cip.es/netdidactica/articulos/mapas.htm> [2005].
- Clinton, H. (1999). Program Monitoring and Visualization (Springer-Verlag ed. Vol. 2). Denver.
- Chestlevar, C. I. (2001). Utilización de Mapas Conceptuales en la enseñanza de la programación. Bahía Blanca - Argentina: Departamento de Ciencias de la Computación. Universidad Nacional del Sur.
- Dougiamas, M. (2003). Moodle. Australia: Curtin University.
- Frías, I., Soler, Y. y Lezcano, M. (2007). Sistema de Enseñanza Asistida por Computadora para la visualización de operaciones sobre estructuras de datos y animación de algoritmos. Publicada en: Maestría en Nuevas Tecnologías para la Educación, Universidad de Granma, Bayamo, 74p.
- Jeffery, C. (2001). Program Monitoring and Visualization (Springer-Verlag ed.).
- Kearns, M. y Vazirani, U. (2001). An Introduction to Computational Learning Theory: The Bactra Review. Occasional and eclectic book reviews by Cosma Shalizi. How to Build a Better Guesse, 12p.
- MES. Documento base para la elaboración de los planes de estudio (PLAN "D"), Dirección de Formación de Profesionales (2003).
- Señas, P. y Moroni, N. (2003). Herramientas no convencionales para la enseñanza de la programación. Publicada en: de la Computación, Instituto de Ciencias e Ingeniería de Computación Universidad Nacional del Sur, Bahía Blanca-Argentina, 112p.
- Soler, Y. y Lezcano, M. (2007). Organización del conocimiento de la asignatura Programación II para Ingeniería Informática basada en mapas conceptuales. Publicada en: Tesis de Maestría, 75p.
- Stojanovic, L. (2002). El paradigma constructivista en el diseño de actividades y productos informáticos para ambientes de aprendizaje "on-line". Pedagogía. Carácas, 23, 66.