

Evaluación sistematizada de software educativo: estudio de un caso de aplicación en grafos

Patricia Calvo¹, Zulma Cataldi¹, Fernando Lage¹

¹ Facultad de Ingeniería. Universidad de Buenos Aires. Ciudad Autónoma de Buenos Aires, Argentina.

Resumen

Se presenta la evaluación de un software educativo siguiendo la metodología elaborada por Cataldi (2001). Se toma el caso del software para el aprendizaje de los recorridos en profundidad y en anchura para el tema de Grafos. Se trabajó con una muestra de 15 alumnos de la Tecnicatura en Informática del Instituto Superior de Profesorado Técnico. Los alumnos debían resolver los ejercicios planteados por el docente y luego completar las plantillas que se presentan para evaluación de los contenidos y de la interface. Las indicaciones y sugerencias de los alumnos permiten considerar algunos cambios para facilitar el uso de la aplicación.

Palabras clave: software educativo, evaluación de software educativo.

1. Introducción

El concepto de aprendizaje y las metodologías utilizadas en el proceso de aprendizaje han dado un vuelco en las últimas décadas, debido a los avances producidos por distintas corrientes de psicología sumadas a las nuevas tecnologías entre las cuales ocupan un lugar predominante las que se valen de la computadora, incorporada definitivamente como recurso de enseñanza. En este sentido, se debe tener en cuenta que los productos de software educativo pueden constituir una vía privilegiada para la aplicación directa de los principios planteados por las distintas escuelas de psicología del aprendizaje, o bien repetir viejos esquemas ya superados, en los cuales el alumno es un mero receptor del conocimiento que imparte el docente. Se considerarán entonces, algunos conceptos básicos de las principales corrientes teóricas y sus implicancias en el desarrollo de software para la enseñanza, a fin de conformar referencias que permitan establecer un marco para la evaluación del software educativo. Según la teoría de aprendizaje conocida como *cognitivismo*, el aprendizaje se produce a partir de la experiencia, como una representación de la realidad. La mente del sujeto que aprende constituye el agente activo en este proceso, construyendo y adaptando los esquemas mentales o sistemas de conocimiento, en un proceso reflexivo.

Aprender trae consigo un cambio en las teorías y modelos del alumno (que se reestructuran), no meramente en sus respuestas. En este sentido, un recurso de aprendizaje desarrollado considerando estos principios debe tener como eje *el conjunto de procesos del hacer y del reflexionar que el alumno debe recorrer para lograr el aprendizaje*. Para ello, debe procurarse la ‘movilización’ de los esquemas previos de pensamiento, de modo que el nuevo material de aprendizaje solamente provocará la transformación de las creencias y pensamientos del alumno [1]. La teoría *constructivista* considera que, para el alumno, lo aprendido está constituido por sus estructuras mentales, experiencias previas y sistema de creencias; en el proceso de aprendizaje reinterpreta su realidad en base a la experiencia propia. Es por ello que según estas consideraciones [2,3] los recursos deberían estar planteados de modo tal de ir de lo simple a lo complejo, incorporando lo conceptual y lo procedimental, teniendo en cuenta las relaciones de efectos positivos que se dan entre los alumnos. Para la *psicología genético-dialéctica* (representada fundamentalmente por Vigotzkii, y sus seguidores Leontiev, Luria, Rubinstein [4] desde un modelo sociocultural se destacan las actividades de aprendizaje con un sentido social, atribuyendo gran importancia al entorno sociocomunicativo del sujeto para su desarrollo intelectual y personal. Se sostiene que el cambio cognitivo, se da en la ZDP (zona de desarrollo próximo), zona determinada por la distancia entre el nivel real de desarrollo y el nivel posible; en esta transformación se realiza la resolución de problemas mediados por un adulto o tutor. De estas ideas se deriva el concepto fundamental de Bruner

[5] (representante de la *psicología genético cognitiva*, de gran influencia piagetiana) denominado *andamiaje* educativo. Para esta corriente es fundamental la importancia de la acción en el desarrollo del aprendizaje (concepto de aprendizaje por descubrimiento). Implica tender un ‘andamio’ entre quien enseña y quien aprende, para permitir ampliar el alcance del sujeto y que éste pueda realizar tareas que de otro modo serían imposibles que llevara a cabo. Esta estructura de andamio o agente mediador puede estar constituida por aplicaciones educativas desarrolladas de modo apropiado. Esto implica que el software educativo debe

presentar problemas concretos que capten la atención de los alumnos y representen un desafío que promueva a la transferencia de aprendizaje, utilizando experiencias diversas. Rogers [6], desarrollando los conceptos del *aprendizaje cooperativo* sostiene que el aprendizaje significativo tendrá lugar cuando el sujeto perciba al tema como relevante o satisfaciendo alguna de sus características o necesidades personales sociales. Perkins [7], codirector del Proyecto Zero del Centro de Investigación para el Desarrollo Cognitivo de Harvard, en su *Teoría Uno* afirma que “*la gente aprende más cuando tiene una oportunidad razonable y una motivación para hacerlo*”.

En consecuencia, algunas de las condiciones necesarias para que el aprendizaje se lleve a cabo de un modo satisfactorio, son las siguientes:

- El alumno debe poseer una fuerte motivación intrínseca y extrínseca
- La información debe clara y precisa
- Debe darse una práctica reflexiva
- Asimismo se requiere una retroalimentación informativa positiva

Las aplicaciones computacionales ofrecen a los alumnos experiencias de aprendizaje con un amplísimo rango de diversidad. Para ser eficaces, deben estar desarrolladas en función de los procesos que debe llevar a cabo el individuo para adquirir conocimientos mediante la presentación de los recursos y actividades necesarias desde la perspectiva del aprender mediante la acción y la reflexión.

1.1. ¿Qué es el software educativo?

Durante las últimas décadas, la computadora ha tomado cada vez mayor relevancia como recurso en los ámbitos educativos. Los programas de aplicación didáctica son actualmente uno de los medios más atractivos para la presentación de nuevos contenidos, la ejercitación guiada de los alumnos, y la evaluación de los mismos. Se denomina software educativo a todo programa de computación realizado con el objetivo de ser utilizado como facilitador del proceso de enseñanza y consecuentemente de aprendizaje. Este software por lo general tiene particularidades tales como facilidad de uso, interactividad y posibilidad de adaptación a ciertas características de los alumnos. Se denomina programa educativo a aquel programa de computadora que ha sido creado con la finalidad específica de ser usado como medio didáctico para facilitar los procesos de enseñanza y de aprendizaje [8]. Estos programas se caracterizan por llevar a cabo diversas funciones, entre las cuales por lo general se mencionan [8]:

- Una función informativa estructuradora de la realidad.

- Una función instructiva, que promueve actividades de los alumnos orientadas al logro de los objetivos educativos.
- Una función motivadora, al considerar elementos para motivar a los alumnos y dirigirlos hacia los aspectos más importantes de las actividades.
- Una función lúdica se incorpora a menudo para aumentar la motivación.
- Además, por lo general se considera también una función evaluadora del trabajo realizado por los alumnos.

Por otro lado Galvis [9] (con la metodología de Ingeniería del Software Educativo, ISE y los micromundos activos) indica que un buen software educativo debe cumplir estas características:

- Que el programa sea adecuado para los niveles de dominio diferenciado de los usuarios. – Que considere características de la población a la que se dirige. – Que permita reconstruir experiencias a los alumnos. – Que pueda contribuir al desarrollo de habilidades, conocimientos y destrezas de los estudiantes.
- Que propicie la participación activa de los alumnos para realizar búsquedas, generación, apropiación y reconstrucción del conocimiento.
- Que pueda detectar y completar los vacíos conceptuales.

1.2. El proceso de construcción del software

El modelo tradicional del proceso de construcción de software, como en todo proceso de elaboración de un producto, inicialmente considera una necesidad, y luego las especificaciones de los requisitos, para posteriormente obtener el diseño de la aplicación, desarrollarla, implementarla y probarla. Dada la limitación de la vida útil de cualquier aplicación, se establece que “*se denomina ciclo de vida a toda la vida del software, comenzando con su concepción y finalizando en el momento de la desinstalación del mismo*” [10]. Se puede describir de modo resumido enumerando estos pasos:

- Obtener requisitos del software.
- Realizar el diseño preliminar y el diseño detallado.
- Realizar la implementación.
- Realizar las pruebas.
- Llevar a cabo la instalación.
- Ejecutar las eventuales tareas de mantenimiento, y actualización del sistema.

Por otra parte, se destaca también el modelo en cascada de construcción de software, inicialmente presentado por Royce [11], y refinado e integrado por Boehm [12],

Sommerville [13] y Sigwart [10]. Según esta perspectiva, el producto sufre una evolución a lo largo de una secuencia de etapas ordenadas linealmente, las cuales permiten iterar al estado anterior.

1.3. La calidad de un software

El concepto de calidad en el ámbito de las aplicaciones de software, hace referencia a un conjunto de características que el software debería reunir. Estas características son:

- Funcionalidad o capacidad del software de cumplir los objetivos para los cuales fue desarrollado.
- Eficiencia, la cual implica la minimización de los recursos que utiliza la aplicación.
- Portabilidad, lo cual indica que puede ser ejecutado en distintas plataformas.
- Confiabilidad, que es la capacidad de mantenerse en ejecución frente a datos o resultados imprevistos.
- Reusabilidad
- Mantenibilidad, es decir, que el sistema permite sin dificultad realizar modificaciones deseadas por el usuario o el programador de aplicaciones.

2. El proceso de evaluación de software educativo

La evaluación del software educativo constituye un proceso que permite determinar el grado de adecuación de dicho software a un contexto educativo particular. Por lo general, en el proceso evaluativo se consideran los aspectos técnicos, pedagógicos y funcionales [14]. El análisis de los aspectos técnicos alcanza hasta el diseño de pantallas e interfase de comunicación. La consideración de los aspectos pedagógicos se enfoca en los objetivos educativos, los contenidos y las trayectorias pedagógicas. Los aspectos funcionales se orientan a la determinación de las ventajas de la aplicación como recurso didáctico, como facilitador de aprendizajes y como favorecedor de funciones de pensamiento. Por lo general se consideran dos tipos de evaluación [14]: una interna, que desarrollan los responsables del desarrollo del software, y una externa, en la cual intervienen profesores y alumnos una vez terminado el sistema. La evaluación interna, realizada por el equipo responsable del desarrollo del software, puede considerarse una evaluación del proceso y permite realizar los ajustes necesarios antes de presentar el producto a alumnos y docentes para llevar a cabo la evaluación externa. La evaluación externa suele ser llevada a cabo por un equipo de evaluación distinto que construye listas de preguntas, de tipo cerrado (generalmente “*checklist*”) y abierto y recurre a uno o más grupos de prueba o “*focus group*” que interactúa con el producto a evaluar para completar los

cuestionarios, con cuyas respuestas luego se podrán realizar distintas inferencias. Durante la evaluación externa se presenta el producto a los potenciales alumnos para verificar que sean alcanzados los objetivos propuestos, y detectar errores imprevistos. La evaluación externa permite obtener las sugerencias de los alumnos potenciales, quienes serán en definitiva los usuarios del software y de los docentes que lo utilizarán como material didáctico. De existir documentación, o archivos de ayuda, también éstos serán sometidos a la evaluación. El resultado final de ambas evaluaciones es la primera versión del programa con su respectivo manual de usuario. Esta contiene todos los aspectos que se consideren indispensables para el uso docente, con detalles técnicos, y del entorno pedagógico y didáctico para el que se desarrolló el programa. [15] Esta documentación tiene un rol fundamental en la posterior etapa de mantenimiento y de actualización; dado que la posibilidad de un producto de no perder vigencia y que dar fuera de mercado, depende de sus posibilidades de incorporación de nuevas funciones y/o modificación de funcionalidades previas.

2.1. Los instrumentos de evaluación

Entre los instrumentos más usados están los cuestionarios de valoración en los cuales el resultado indica el grado de conformidad del usuario con las afirmaciones propuestas. Es conveniente incluir en las plantillas de evaluación preguntas abiertas que permitan al usuario referirse a aspectos de la aplicación no considerados en las categorías propuestas en las preguntas cerradas. Los aspectos a evaluar se deben ordenar y clasificar en distintas categorías tales como logro de los objetivos, aspectos técnicos, desarrollo de contenidos. Por otra parte, los cuestionarios utilizados en la evaluación deben estar redactados con un vocabulario sin ambigüedades y adecuado al de los integrantes de cada “*focus group*” que se considere. El cuestionario “genérico” es una guía que debe ser readaptada a cada contexto educativo particular.

2.2. Alcance del trabajo realizado

Se ha trabajado en la evaluación externa de un software educativo que tiene como destinatario el alumnado de cursos superiores con conocimientos previos de estructuras de datos. El producto está orientado al aprendizaje de los recorridos en profundidad y en anchura de grafos y ha sido desarrollado con fines didácticos en el Departamento de Matemática Aplicada de la Facultad de Informática de la Universidad Politécnica de Madrid (España). Para la evaluación de la aplicación se ha usado un conjunto de planillas que se describe más adelante, en las cuales el alumno debe marcar respuestas en base a una codificación informada previamente (ponderación), a lo que se agregó la posibilidad de escribir sugerencias que surgieran en

forma espontánea. Se ha trabajado con un grupo de prueba formado por 15 alumnos (grupo disponible) de nivel terciario, de *Tecnicatura en Informática* como parte de los contenidos de la asignatura Estructuras y Bases de Datos. Los alumnos recibieron las instrucciones para utilizar el programa de búsqueda luego de haber realizado una lectura minuciosa de los conceptos teóricos del tema y de las especificaciones para el uso del *applet*, durante una semana. Se solicitó a los alumnos que representaran y recorrieran en amplitud un grafo como el que se indica a continuación:

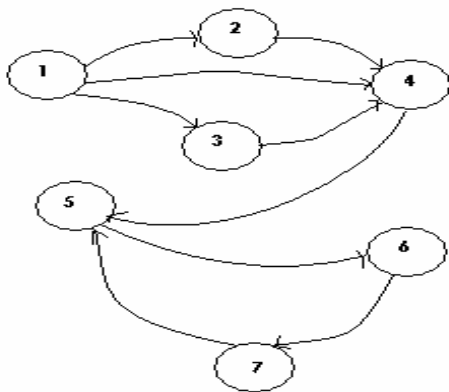


Figura 1. Grafo propuesto a los alumnos

3. Descripción del software evaluado

El programa abarca el tema de recorridos en amplitud y en profundidad en grafos. El producto es una herramienta que utiliza hipertexto y multimedia, de inteligencia convencional (no experto). La aplicación analizada es una herramienta para el aprendizaje de un tema específico capaz de detectar errores cometidos por el estudiante. Ha sido desarrollado con el objetivo de facilitar la adquisición de elementos conceptuales y procedimentales, exigiendo del alumno análisis, observación y razonamiento en la resolución de recorridos que utilizan un grafo diseñado por el mismo alumno.

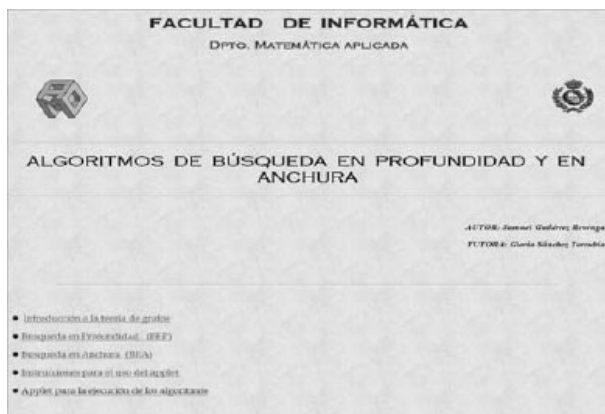


Figura 2. Pantalla principal del sitio del programa

Se ha trabajado con el software para aprendizaje de recorridos en profundidad y en anchura de grafos desarrollado con fines didácticos en el Departamento de Matemática Aplicada de la Facultad de Informática de la Universidad Politécnica de Madrid (España). En la Figura 2 se puede apreciar la página inicial.

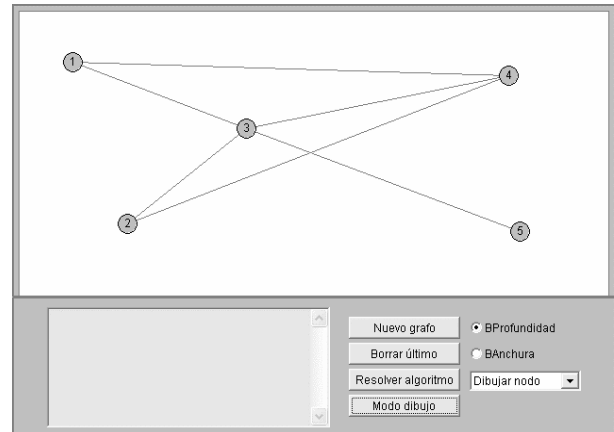


Figura 3. Construcción del grafo

La aplicación guía al alumno en la construcción de un grafo correcto (Figuras 3 y 4) Si el alumno comete errores, se le indica que debe realizar una corrección (Figura 5). Una vez construido el grafo, se puede llevar a cabo una búsqueda en profundidad o en anchura. Los colores utilizados en los nodos del grafo orientan al alumno acerca de cual debe ser el segundo nodo seleccionado a continuación (Figuras 4 y 5)

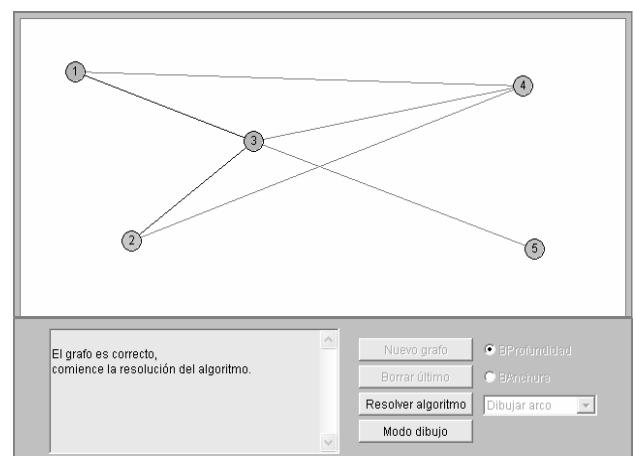
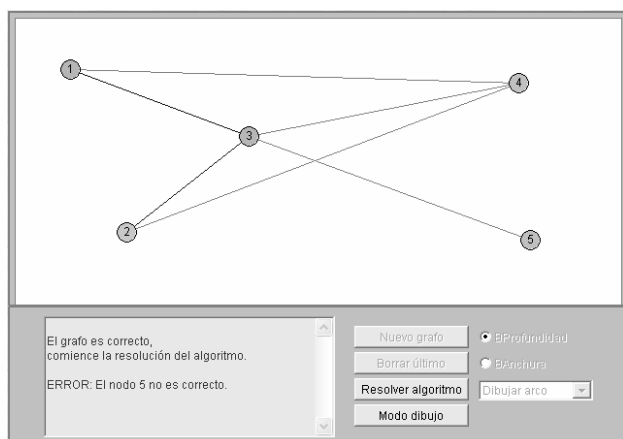


Figura 4. Pantalla de creación del grafo.



1. Figura 5. Mensaje ante errores

Si el alumno comete algún error, se despliega un mensaje de advertencia, y el proceso de resolución se detiene hasta que se realice la corrección (Figura 5).

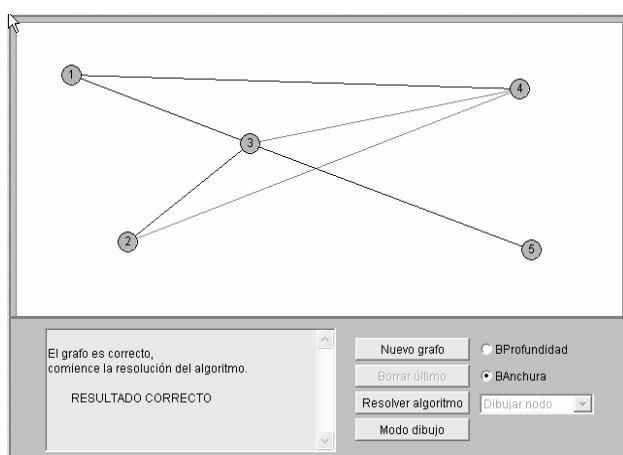


Figura 6. Diferentes recorridos

En ambas búsquedas (recorridos b.e.a. y b.e.p.), un mensaje indica la finalización correcta del recorrido deseado (Figura 6).

3.1. Planillas usadas para la evaluación

La codificación utilizada para graduar la consecución de los objetivos, se muestra en la Tabla 1.

| | | | | |
|-----------|-----------|-------|---------|------|
| 5 | 4 | 3 | 2 | 1 |
| excelente | muy bueno | bueno | regular | malo |

Tabla 1: Codificación

El modelo de la plantilla usada para la evaluación de la interface de comunicación, se desarrolla en la Tabla 2.

| Ítem | Sí/No |
|---|-------|
| 1. ¿Considera adecuado el diseño general de la pantalla? | |
| 2. ¿Considera adecuado el uso de las ventanas? | |
| 3. ¿Considera adecuado el uso de los botones? | |
| 4. ¿Considera adecuado el uso de los colores? | |
| 5. ¿Considera adecuado el uso de los tipos de letras? | |
| 6. ¿Considera que el programa es interactivo? | |
| 7. ¿Considera la interface amigable? | |
| 8. ¿Le da buena información acerca del recorrido? | |
| 9. ¿Considera criteriosa la secuenciación de las pantallas? | |
| 10. ¿Es de fácil manejo? | |
| 11. ¿Considera correcto el uso de los iconos? | |
| 12. ¿Ha despertado interés en usted? (a partir de la interface) | |
| 13. ¿Considera apropiado agregar sonido? | |
| Sugerencias de cambio Sí - No | |

Tabla 2: Plantilla de evaluación de interface

En la Tabla 3 se muestra el listado de las sugerencias de cambio para la interface. Las listas de sugerencias fueron pautadas previamente, pero se dejó un espacio abierto para que introdujeran otras nuevas que surgieran durante el proceso de aprendizaje. La Tabla 5 se presenta la plantilla para la evaluación de contenidos y pertinencia, mientras que la Tabla 6, se presentan las sugerencias sobre pertinencia. En las tablas 4 y 8, el contenido de la celda fila F columna C indica cuántos alumnos eligieron el código C para calificar el ítem F.

| Número de Sugerencia | Sugerencias de Cambio | Sugerencia Elegida (Si-No) |
|----------------------|---|----------------------------|
| 1 | Cambiar los colores en pantalla para que resulte más atractivo | |
| 2 | Cambiar el indicador de flecha a mano. | |
| 3 | Agregar cambios en el relieve a medida que se desarrolla la aplicación. | |
| 4 | Aumentar el contraste entre los colores | |
| 5 | Interconectar los elementos por medio de dibujos que representen cables en la pantalla de presentación.(Tipo diagrama de flujo) | |
| 6 | Aumentar el tamaño de la letra para usar en la notebook. | |
| 7 | Agregar un icono de retorno en la pantalla. | |
| 8 | Agregar algún tapiz de fondo | |
| 9 | Modificar los colores de las letras de los botones (intensidad) | |
| 10 | Colocar el icono de “volver” fijo en la pantalla para retornar a ventanas anteriores. | |
| 11 | Agregar subventanas u opciones de algunos temas | |
| 12 | Agregar sonido al momento de activar algún elemento | |

Tabla 3: Sugerencias de cambio de la interface de comunicación

| Item | Si/No |
|--|-------|
| 1) ¿Considera adecuada la selección de contenidos? | |
| 2) ¿Considera adecuado el uso del programa terminado en otros niveles? | |
| 3) ¿Quisiera que el programa fuera un tutorial? | |
| 4) ¿Le facilita la comprensión del tema? | |
| 5) ¿Los contenidos son claros? | |
| 6) ¿La motivación es adecuada? | |
| 7) ¿Considera adecuada la documentación y las ayudas? | |
| Sugerencias de cambio Sí – No | |

Tabla 5: Plantilla de evaluación de pertinencia

| Número de Sugerencia | Sugerencia de Cambio | Sugerencia Elegida (S- N) |
|----------------------|--|---------------------------|
| 1 | Poner las direcciones web y la bibliografía en las pantallas explicativas. | |
| 2 | Sugiero problemas resueltos como modelo | |
| 3 | No agregar audio con explicaciones; prefiero la explicación del docente | |
| 4 | Agregar audio con explicaciones | |

Tabla 6: Sugerencias sobre la pertinencia

La Tabla 7 resume las sugerencias seleccionadas para la interface de comunicación, mientras que la Tabla 8 lo hace con los resultados obtenidos en la evaluación de contenidos y pertinencia.

| Número de sugerencia | Cantidad de veces que fue seleccionada |
|----------------------|--|
| 1 | 5 |
| 2 | 2 |
| 3 | 6 |
| 4 | 5 |
| 5 | 5 |
| 6 | 4 |
| 7 | 8 |
| 8 | 1 |
| 9 | 5 |
| 10 | 8 |
| 11 | 10 |
| 12 | 4 |

Tabla 7: Resumen de las sugerencias para la interface de comunicación

| Ítem → ↓Código | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-------------------|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 2 | 0 | 0 | 0 | 0 |
| 2 | 0 | 2 | 2 | 0 | 2 | 4 | 1 |
| 3 | 5 | 2 | 2 | 6 | 5 | 4 | 6 |
| 4 | 6 | 4 | 2 | 2 | 2 | 1 | 2 |
| 5 | 0 | 0 | 2 | 1 | 2 | 1 | 0 |

Tabla 8: Resumen de resultados de la evaluación de contenidos y pertinencia

En las Tablas 4 y 8 la fila superior indica el número correspondiente a cada ítem evaluado. La columna de la izquierda contiene los códigos correspondientes a los valores seleccionados. El contenido de cada celda nos indica cuántos alumnos calificaron a un código (fila) con el valor que encabeza la columna. La Tabla 9 muestra las sugerencias seleccionadas por el grupo de prueba para los contenidos.

| Número de Sugerencia | Cantidad de veces que fue seleccionada |
|----------------------|--|
| 1 | 9 |
| 2 | 11 |
| 3 | 7 |
| 4 | 4 |

Tabla 9: Resumen de sugerencias para los contenidos

3.2. Sugerencias adicionales realizadas espontáneamente por los alumnos

Los alumnos hicieron las sugerencias siguientes de forma espontánea, respondiendo a la solicitud del docente:

- Agregar imágenes en la explicación de cómo armar el grafo.
- Agregar imágenes en la definición de componente conexas.
- Agregar una animación ejemplificadora de la resolución de las búsquedas en profundidad y anchura.
- Evitar repetición de términos en las explicaciones.
- Agregar instrucciones sobre el uso del applet en la ventana misma del applet.
- Explicar mediante un texto detallado en qué consiste el error cometido cuando el alumno incurre en él al realizar los pasos del algoritmo en cuestión.
- Agregar la posibilidad de que las etiquetas de los nodos sean de otro tipo (distinto de tipo entero) con valor creciente en cada nodo.
- Agregar teclas aceleradoras.
- Agregar una ayuda interactiva con ejercicios modelos resueltos.
- Agregar como opción que el applet genere grafos de manera aleatoria.
- Agregar opción de resolución automática.

4. Análisis de los resultados obtenidos

Según se desprende del análisis de los resultados, el software analizado ha logrado despertar el interés de los alumnos por el tema ya que se observa una mejor predisposición que cuando debían resolver los ejercicios sin uso del software. En general el grupo ha considerado adecuado el uso de pantallas, ventanas, botones, íconos y fuentes de letras, pero los alumnos sugieren que la aplicación brinde información adicional acerca del recorrido asociado a las pantallas. Han sugerido también aumentar el contraste de colores y agregar sonidos asociados con las distintas etapas de resolución del algoritmo. En cuanto a las cuestiones de pertinencia, el grupo de prueba ha considerado apropiada la selección de contenidos, los cuales aprecian expuestos de modo claro. Consideran regular el tratamiento del aspecto motivacional, ya que se podían haber incluido algunos ejemplos usando otros recursos disponibles que le dieran una mayor amigabilidad a la interface.

Algunos miembros del grupo sugirieron una ayuda para el manejo con el applet, mientras otros consideraron apropiada la que se daba. En general han señalado la

conveniencia de agregar más imágenes y gráficos acompañando a las definiciones y explicaciones teóricas, así como agregar bibliografía y links a sitios relacionados con el tema, y en particular han sugerido agregar ejercicios modelo resueltos, o bien una opción de resolución automática de un grafo propuesto por el alumno.

5. Conclusiones

Las cuestiones de interface han sido resueltas de un modo apropiado. En cuanto a la pertinencia, si bien los contenidos han sido seleccionados del modo más satisfactorio, puede ser oportuno:

- agregar situaciones motivacionales dirigidas a aumentar el interés de los alumnos
- explorar los conocimientos previos de los mismos
- aumentar la cantidad y calidad de los gráficos en la conceptualización teórica previa
- suministrar a los alumnos ayuda detallando los errores cometidos
- considerar ejercitación que permita el trabajo en equipo con otros alumnos
- plantear situaciones que funcionen como disparadores motivacionales la continuación del estudio y la profundización del tema (o temas relacionados)

6. Referencias

- [1] Cataldi, Z. (2000) Metodología de Diseño, desarrollo y evaluación de software educativo, Tesis de Magíster en Informática, U.N.L.P.
- [2] Araujo, J.B., Chadwick, C. B. (1993). Tecnología educacional. Teorías de instrucción. Paidós.
- [3] Ertmer, P. y Newby, T. (1993). Conductismo, cognitivismo y constructivismo: Una comparación de aspectos críticos desde la perspectiva del diseño de la instrucción Performance Improvement Quarterly
- [4] Vigotzkii, L. (1978): Mind in Society. The development of higher psychological process. Cambridge. M. A. Harvard University Press.
- [5] Bruner, J. (1988) Realidad mental y mundos posibles. Ed. Gedisa
- [6] Rogers, C., Kutnick, P. (1992): Psicología social de la escuela primaria, Paidós.
- [7] Perkins, D. (1995) La escuela inteligente. Gedisa
- [8] Marquès, P. Criterios de evaluación Documento en línea www.xtec.es/~pmarques/edusoft.htm
- [9] Galvis, A.(1992). Ingeniería de Software Educativo Ediciones Uniandes. [10]Sigwart, C. (1990) Software Engineering: a project oriented approach. Franklin, Beedle y Associates Inc.
- [11]Royce W. (1970): Managing the development of Large software systems: Concepts and techniques. Proceedings, Wescon, agosto, 1970, citado en Piattini (1996).
- [12]Boehm B. (1988): A spiral model of software development and enhancement. Computer 1988 IEEE págs. 61-72.
- [13]Sommerville I. (1985): Software Engineering. Addison Wesley.
- [14]Marquès Plantilla para el análisis de webs docentes <http://dewey.uab.es/pmarques/webdocen.htm>. Documento en línea
- [15]Cataldi, Z. (2005). Evaluación de programas hipermedia educativos de producto final y en un contexto similar al de aplicación, Revista Latinoamericana de Tecnología Educativa, 4 (2), 27-52.
[http://www.unex.es/didactica/RELATEC/sumario_4_2_.htm].

Dirección de Contacto del Autor/es:

Patricia Calvo

Paseo Colón 850 4º Piso. C1063ACV.

Ciudad Autónoma de Buenos Aires. Argentina

e-mail: pat_calvo@yahoo.com

sitio web: <http://www.fi.uba.ar/laboratorios/lie/lie.htm>

Zulma Cataldi

Paseo Colón 850 4º Piso. C1063ACV

Ciudad Autónoma de Buenos Aires. Argentina

e-mail: liemal@fi.uba.ar

sitio web: <http://www.fi.uba.ar/laboratorios/lie/lie.htm>

Fernando Javier Lage

Paseo Colón 850 4º Piso. C1063ACV

Ciudad Autónoma de Buenos Aires. Argentina

e-mail: flagel@fi.uba.ar

sitio web: <http://www.fi.uba.ar/laboratorios/lie/lie.htm>

Ing. Patricia Calvo es Ingeniera en Sistemas de la UTN-FRBA. Profesora de Departamento de Computación de la Facultad de Ingeniería de UBA y del Instituto Superior del Profesorado Técnico.

Dra. Zulma Cataldi es investigadora y profesora de Departamento de Computación de la Facultad de Ingeniería de UBA. Directora del Laboratorio de Informática Educativa y Medios Audiovisuales.

M. Ing. Fernando Javier Lage es Mg. En Informática, investigador y profesor de Departamento de Computación de la Facultad de Ingeniería de UBA. Co-Director del Laboratorio de Informática Educativa y Medios Audiovisuales.
