

XV JENUI. Barcelona, 8-10 de julio de 2009

ISBN: 978-84-692-2758-9

<http://jenui2009.fib.upc.edu/>

SRec, software de animación de la recursividad

Antonio Pérez Carrasco, J. Ángel Velázquez Iturbide

Departamento de Lenguajes y Sistemas Informáticos I

Escuela Técnica Superior de Ingeniería Informática, Universidad Rey Juan Carlos

C/ Tulipán s/n, CP 28933, Móstoles (Madrid)

antonio.perez.carrasco , angel.velazquez@urjc.es

Resumen

SRec es una aplicación que permite la generación automatizada de animaciones de programas basados en la recursividad. Ofrece una gran variedad de facilidades educativas como cargar y guardar animaciones ya generadas, configuración de la información que se muestra en cada momento y una interfaz muy intuitiva. Mitiga el esfuerzo de elaboración de animaciones para los profesores y ayuda a analizar los algoritmos recursivos a los alumnos. Los resultados obtenidos de su utilización en las clases indican una gran aceptación por parte de los alumnos y una mejora en sus evaluaciones académicas.

1. Introducción

La actividad docente se apoya cada vez más en el uso de herramientas tecnológicas para conseguir su objetivo, transmitir nuevos conocimientos a los alumnos. En lo que se refiere a la enseñanza de la programación y la algoritmia, el empleo de los equipos informáticos está cada vez más presente, no sólo para mostrar transparencias y ejecutar programas, sino también para visualizar el desarrollo de tales ejecuciones.

Es en este contexto en el que el proyecto de SRec toma forma, intentando satisfacer la demanda, tanto de profesores como de alumnos, de una herramienta que genere de manera totalmente automatizada animaciones de programas sin esfuerzo. SRec permite crear a los profesores tantos ejemplos como necesiten para la impartición de sus clases magistrales mientras que a los alumnos les ofrece la posibilidad de repasar los algoritmos expuestos en clase o de desarrollar otros nuevos para realizar prácticas y ejercicios propuestos.

La aplicación genera animaciones desde algoritmos codificados en clases escritas en el

lenguaje Java. En apenas un par de sencillos pasos SRec muestra en su ventana el árbol de activación junto a otras vistas de interés tales como la pila de control. El usuario tiene total libertad para manejar las animaciones de manera automática o manual así como para configurarlas, determinando la cantidad de información que se muestra o el formato de la misma. La aplicación también permite la exportación de imágenes en formatos estándar que pueden ser incrustados en documentos o páginas web.

Hasta ahora, se cuestionaba la capacidad de las animaciones de programas para apoyar al profesor durante las clases magistrales ya que no existía ninguna facilidad que permitiera asegurar que las animaciones suponen un recurso útil y de valor añadido para las clases. Sí existen resultados que indican que la interactividad con las animaciones despierta la motivación e implicación del alumno en el aprendizaje [4], por lo que existen indicios de que esta aplicación realmente puede ser útil para el alumnado. Es ésta por tanto, otra de las tareas que cumple SRec, confirmar que el empleo de estas animaciones basadas en la técnica de diseño realmente ayuda en la enseñanza de la programación recursiva. No existe una literatura extensa que se enfoque en las visualizaciones basadas en la técnica de diseño salvo [2][1], terreno en el que se adentra la aplicación que aquí se presenta.

Se han realizado tres experimentos similares que exigían a los participantes (alumnos de diferentes asignaturas de algoritmia) la realización de varios ejercicios didácticos y rellenar un cuestionario sobre la aplicación. Los resultados dejaron ver una alta receptividad de los participantes a las animaciones y también notables mejoras en los resultados académicos.

En el segundo apartado presentamos las funcionalidades más importantes de la

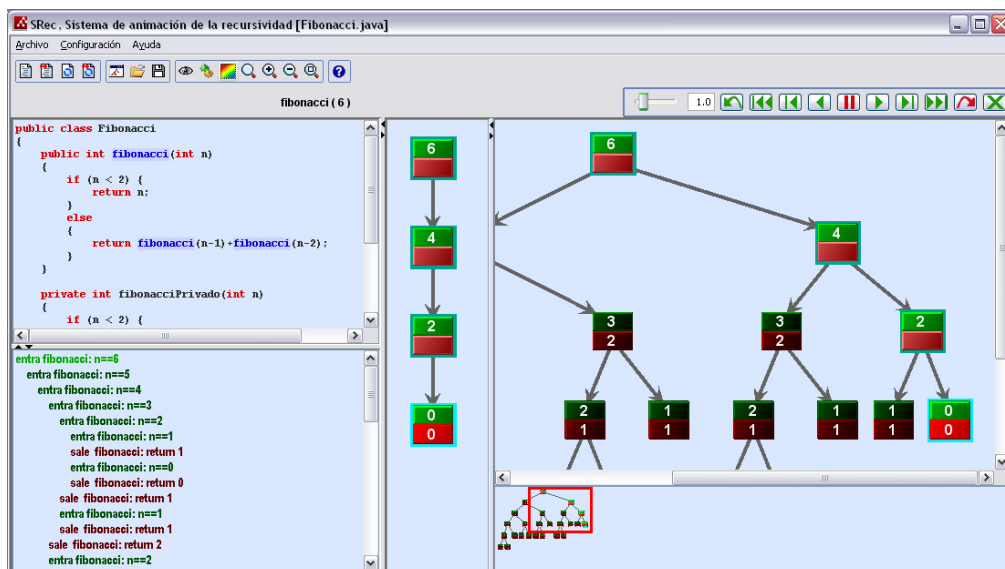


Figura 1. Ventana principal de la aplicación, con las cuatro vistas activadas

aplicación, mientras que en el tercer apartado realizamos un pequeño repaso de otras características que hacen SRec especial frente a otros sistemas similares. En el cuarto apartado detallamos las características de SRec como recurso docente y además analizamos los resultados obtenidos tras el empleo de SRec en nuestras aulas y la realización de varias evaluaciones de su usabilidad para terminar con unas breves conclusiones y comentarios sobre algunos de los trabajos futuros que están pendientes para aportar a SRec nuevas funcionalidades.

2. Uso básico de SRec

La ventana de SRec (figura 1) ofrece una serie de elementos que permiten interactuar con la aplicación y visualizar los algoritmos. Así, cuenta con un menú y una barra de herramientas que facilitan el acceso a todas las funcionalidades que ofrece la aplicación. También se cuenta con una barra de animación, que con unos mandos similares a los de un reproductor multimedia permite manejar la animación con gran grado de flexibilidad.

La parte más importante de cara a la visualización de los algoritmos la componen los

cuatro paneles que conforman el área de visualización. El primer panel (situado en la parte superior izquierda) ofrece el código de la clase que se encuentra cargada en la aplicación.

El resto de paneles ofrecen vistas dinámicas que aportan información complementaria y sincronizada sobre la animación en curso. El primero de ellos, situado justo debajo del panel de código, presenta en modo de texto una traza de las subllamadas que se han ido generando y resolviendo. El siguiente panel alberga la pila de control, que deja ver en cada momento qué subllamadas se encuentran residentes en memoria por no haber finalizado aún su ejecución.

El último de los paneles, situado en el lado derecho de la ventana, contiene la vista principal, el árbol de recursión, donde cada nodo representa una subllamada recursiva. Esta vista ofrece una idea global y en detalle de toda la ejecución. Para el manejo de árboles grandes se incluye un visor de navegación en la parte inferior que permite contextualizar la parte del árbol que se está observando en detalle en el panel, facilitando además la navegación por todo el árbol.

En cada nodo aparecen representadas dos celdas, una dibujada encima de la otra. La superior se corresponde con los valores de los parámetros de entrada para esa subllamada recursiva, mientras que la segunda, situada justo debajo, representa el valor de retorno en métodos que devuelvan algún tipo de valor o, en caso contrario, el valor de los parámetros tras la ejecución del algoritmo.

El manejo de SRec es muy sencillo e intuitivo. Generar una nueva animación requiere iniciar un proceso de dos únicos pasos, mientras que el manejo de las animaciones se realizan con unos mandos que se asemejan a los de un reproductor multimedia, lo que reduce considerablemente el tiempo de aprendizaje del usuario a la hora de iniciarse en el uso de la aplicación.

Para que SRec pueda generar animaciones primero debe tener cargada la clase escrita en lenguaje Java que contiene el algoritmo o algoritmos que se desean visualizar. Mediante una opción de menú se puede elegir qué clase de entre las almacenadas en el sistema de archivos se desea cargar. Tras unos pocos segundos, en los que la aplicación introducirá código Java adicional que le permitirá extraer datos de las futuras ejecuciones, SRec estará listo para generar las animaciones. El código de la clase resulta visible desde entonces en uno de los paneles habilitados en la ventana de la aplicación.

Generar una animación es muy sencillo, basta seleccionar la opción de menú necesaria y la aplicación lanza un cuadro de diálogo con los métodos contenidos en la clase cargada que están disponibles para su animación. El primer paso, por tanto, consiste en elegir qué método se desea visualizar. Podremos determinar qué métodos (llamados desde la ejecución que pretendemos lanzar) queremos que resulten visibles o invisibles en la animación.

A continuación, será necesario introducir los parámetros de entrada para el algoritmo escogido habilitando un cuadro de diálogo adicional. Este cuadro de diálogo además permite elegir qué parámetros de entrada deseamos visualizar u omitir en la animación. Una vez confirmados los datos introducidos, la aplicación invoca el método seleccionado y realiza la ejecución completa del algoritmo.

En ese preciso instante el resto de paneles de la ventana se activan para ofrecer las diferentes vistas dinámicas que SRec es capaz de visualizar. La animación se sitúa al comienzo de la llamada inicial, mostrando tan sólo el nodo raíz del árbol de recursión.

El usuario puede mediante la barra de animación (presentada en la figura 2) activar la función de animación automática (botones cuarto y sexto) o bien adelantar o retroceder la animación paso a paso de forma manual (botones tercero y séptimo). Los controles de SRec permiten también realizar saltos sobre todas las subllamadas de una llamada recursiva (botones primero y noveno) para que de un único paso se muestre o elimine todo el subárbol de llamadas.



Figura 2. Barra de animación

En todo momento los controles adicionalmente permiten adelantar la animación hasta el final (botón octavo), donde se ve completo el árbol de recursión, o retrocederla hasta el comienzo (botón segundo), cuando sólo se encuentra visible la llamada principal que lanza toda la ejecución recursiva. La velocidad de las animaciones automáticas se puede graduar mediante un control deslizante que varía los periodos entre transiciones de 1 a 5 segundos o bien introduciendo el valor exacto de tiempo en el pequeño cuadro de texto que contiene la barra de animación. Cuando el usuario lo desee podrá cerrar la animación (manteniendo la clase Java cargada en la aplicación) pulsando el último botón de la barra.

El transcurso habitual de las animaciones del árbol de recursión se muestra en la figura 3, donde se presenta el algoritmo de Fibonacci con un valor inicial de 2 (caso recursivo), que requiere de dos subllamadas recursivas para poder ser resuelto (subllamadas que se corresponden con el caso base).

La vista de la pila de control y de la traza de ejecución se mantienen sincronizadas paso a paso con la vista principal del árbol de recursión, por lo que resulta muy sencillo consultar qué llamadas recursivas permanecen pendientes o cuál es el historial de llamadas

recursivas y valores obtenidos hasta el momento.

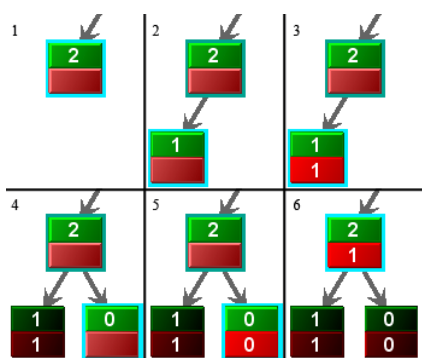


Figura 3. Transiciones de las animaciones

3. Otras funcionalidades

Existen otras muchas funcionalidades de SRec que permiten flexibilizar su uso para adaptarse a todo tipo de necesidades educativas.

La aplicación da la opción de seleccionar qué nodos se muestran y cuáles se ocultan en función del método al que representan. Se puede elegir igualmente cuáles de los parámetros de los métodos visibles deseamos que permanezcan en la animación o queden ocultos, lo que permite estudiar en cada momento diferentes aspectos de los algoritmos (valores de índices, datos...). Las únicas restricciones consisten en que siempre debe permanecer visible el método que lanza la ejecución y en que debe haber al menos un parámetro visible de cada método mantenido en la animación.

Por otro lado, se puede elegir entre visualizar únicamente valores de parámetros de entrada, valores de retorno o bien ambos (representación que se corresponde con la ya citada figura 1). Todas estas configuraciones buscan que el alumno se pueda centrar fácilmente en la parte o aspecto del algoritmo que desea estudiar.

Con la finalidad de hacer más claros el seguimiento de la ejecución del algoritmo y la distinción entre subllamadas finalizadas y subllamadas pendientes las diferentes celdas pueden tener un aspecto diferente. El usuario puede configurar SRec para que atenúe el color de los nodos finalizados o para que directamente

desaparezcan de la vista. SRec mantendrá en el mismo color aquellos nodos que representan subllamadas aún en ejecución. Se presentan las tres posibilidades mediante la figura 4.

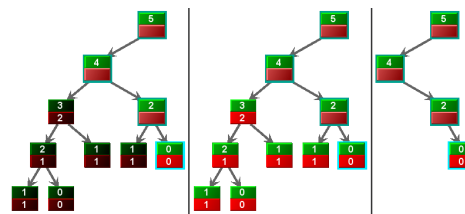


Figura 4. Modos de visualización de nodos de subllamadas ya finalizadas

La aplicación tiene implementada una funcionalidad de zoom que ofrece la posibilidad de ajustar en tamaño el árbol visible en ese momento al panel de la ventana. También se puede graduar numéricamente el tamaño del árbol según las necesidades que se tengan en cada momento.

SRec ofrece un alto grado de flexibilidad a la hora de permitir variar el formato de los datos que se desean mostrar. Así, el usuario puede escoger para cada elemento de la animación (celdas de parámetros de entrada, celdas de valores de retorno, flechas, marcos de las celdas, fondo de panel...) el color que más se ajuste a su gusto y que ayude a resaltar los datos que más le interesen de la animación. La ubicación de las celdas, el estilo de las mismas o algunos tipos de letra son otros de los elementos que SRec permite configurar.

Hay que tener en cuenta que SRec debe ser lo suficientemente flexible como para tener un comportamiento adecuado si se utiliza durante las clases magistrales, empleando pantallas grandes y proyectores, o si se emplea de manera individual por el alumno o profesor en un ordenador de carácter personal, por lo que la configuración del color y de los tamaños cobra especial relevancia.

Por último, cabe destacar que SRec incluye un sencillo editor de código que permite al alumno depurar rápidamente el código que está visualizando. Para ello habilita una ventana que presenta una barra de herramientas, un área de edición y un área de compilación, donde aparecerán los errores que el compilador de Java

encuentre en el código introducido cuando el usuario decida emplearlo.

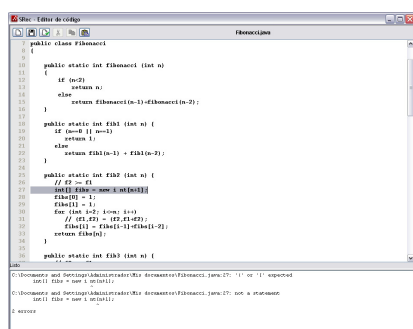


Figura 5. Ventana de edición de código

Basta seleccionar el error en el área de compilación para que SRec marque, tal y como se aprecia en la figura 5, la línea en la que se ha localizado el error pertinente. La ventana de la aplicación también identifica cada una de las líneas del archivo fuente con su número en el margen izquierdo, lo que facilita la localización de los errores indicados en el área de compilación.

Una vez que el usuario haya realizado todas las modificaciones deseadas, podrá cerrar la ventana y cargar de nuevo la clase en la aplicación. A partir de entonces el comportamiento de las animaciones se regirá por el nuevo código introducido.

Si bien no es la funcionalidad objetivo de SRec funcionar como entorno de desarrollo, esta posibilidad que plantea la aplicación evita abrir terceros programas para realizar modificaciones sobre el código, permitiendo una depuración y modificación de los programas ágil, rápida y totalmente integrada en la aplicación.

4. SRec como recurso docente

SRec es una aplicación orientada a ayudar a los profesores y los alumnos de las asignaturas de algoritmia, ofreciéndose como una herramienta de uso sencillo que aporta diferentes tipos de visualizaciones de los algoritmos recursivos.

En los siguientes subapartados desarrollamos algunas de las características más interesantes desde el punto de vista docente y un resumen de los resultados obtenidos.

4.1. Características orientadas a la docencia

La aplicación abre la posibilidad a los profesores de complementar de una manera atractiva y eficaz sus clases magistrales, en las que no tendrán que limitarse a ejemplos estáticos o animaciones pobres de ejemplos concretos para transmitir los conocimientos. SRec permite generar la animación de cualquier algoritmo recursivo que resulte interesante para la clase en apenas unos segundos de manera totalmente automatizada, aumentando la libertad del profesor a la hora de mostrar ejemplos y proponer ejercicios a los alumnos.

También ellos podrán hacer uso de la herramienta con el fin de repasar los ejemplos vistos en clase, desarrollar otros nuevos, realizar ejercicios o depurar prácticas. Así, los alumnos podrán emplear SRec tanto en clases de prácticas que se desarrollen en las aulas de ordenadores como en su hogar mientras estudian los contenidos de la asignatura. Notarán por tanto un importante salto cualitativo frente a las representaciones gráficas estáticas de los libros, a menudo demasiado sobrecargadas con el fin de representar procesos secuenciales que en SRec quedan mostrados mediante las animaciones interactivas.

SRec es una aplicación que se ha desarrollado empleando el lenguaje Java de Sun Microsystems por múltiples motivos. Este lenguaje permite crear interfaces gráficas de gran calidad sin excesivo esfuerzo, cuenta con multitud de librerías que extienden fácilmente su repertorio de funciones y posibilidades y ofrece portabilidad entre diferentes plataformas, lo cual permite universalizar el uso de la aplicación por todo tipo de profesores y alumnos, independientemente de sus preferencias.

Con el fin de hacer llegar SRec al mayor número de personas posible, además, se ha desarrollado para la aplicación un sitio web (<http://www.lite.etsii.urjc.es/>) que permite descargar la última versión de la aplicación, el manual de uso, múltiples publicaciones estrechamente relacionadas con la aplicación y su filosofía así como medios de contacto con el equipo de desarrollo. El sitio web, igual que la aplicación, se encuentra disponible en español y en inglés. De esta manera se consiguen cumplir algunas de las recomendaciones de [3] respecto

a la difusión, sitio web, publicidad y soporte prestado para los usuarios.

De cara al uso diario de SRec, éste da la facilidad de cargar y guardar animaciones con el fin de poder crear una colección de animaciones. Así, la aplicación guarda en un formato propio basado en el lenguaje XML el árbol completo, almacenando información adicional sobre la animación, como por ejemplo, el estado en el que se encuentra (enfocada en la llamada inicial, con todo el árbol de recursión desplegado, en un estado intermedio...).

Además se almacena información adicional sobre la configuración de la aplicación referente a la cantidad de información que se muestra en las vistas y el formato con que se muestra (qué métodos y parámetros deben aparecer en la vista, qué colores deben emplearse...).

Cuando el usuario, en otra sesión de uso de la aplicación, desee cargar esa animación creada por SRec, no tendrá más que seleccionar la opción correspondiente de menú, seleccionar el archivo en el sistema de ficheros, y tras unos pocos segundos aparecerá en la ventana la animación situada en el mismo estado y con la misma apariencia que en el momento en que fue guardada. La animación seguirá siendo totalmente funcional, por lo que no hay diferencia alguna en el funcionamiento de la aplicación al manejar animaciones recién generadas o cargadas.

Esta posibilidad es muy útil cuando el profesor desea preparar con antelación un algoritmo determinado con unos parámetros concretos para impartir una clase posteriormente o bien para que el alumno pueda mostrarle al profesor una duda concreta o un ejercicio realizado con apoyo de la aplicación. De esta manera se agiliza el uso de la aplicación y se fomenta la transferencia de animaciones entre profesor y alumnos.

Como complemento, SRec presenta la capacidad de generar capturas de la vista del árbol de recursión. De esta forma, se pueden generar imágenes en formato electrónico que pueden incrustarse posteriormente en documentos de ofimática y páginas web, permitiendo a los profesores elaborar material docente y a los alumnos informes de prácticas o proyectos de estudio de una manera mucho más sencilla, cómoda y elegante. Además, SRec

también es capaz de generar animaciones estándar en formato GIF, animaciones que pueden ser empleadas en transparencias o páginas web y que muestran de manera lineal el transcurso de la ejecución del algoritmo.

4.2. Sesiones de evaluación

SRec ha sido empleado en clases magistrales desde el curso 2007/2008 para apoyar al profesor en sus exposiciones ante el alumnado. Éste a su vez también ha tenido oportunidad de utilizar el software para la realización de prácticas y repaso de ejemplos y ejercicios propuestos.

A lo largo de los cursos 2006/2007 a 2008/2009 se han realizado tres evaluaciones de usabilidad entre diferentes grupos de alumnos para realizar una medición tanto del grado de aceptación de SRec y de las animaciones que produce como de los cambios registrados en los resultados académicos del alumnado tras su comienzo de utilización.

Estas sesiones tuvieron una duración de dos horas. En ellas se plantearon varios ejercicios relacionados con la programación y análisis de algoritmos. El primero de ellos siempre fue realizado por el profesor con el fin de presentar la aplicación e introducir al alumnado a un manejo básico de la misma. Los siguientes ejercicios se enfocaban en familiarizar al alumno con la aplicación y sus funcionalidades.

Posteriormente se les planteaba un ejercicio que les exigía cierta capacidad de análisis y diseño de algoritmos. En la primera sesión se les propuso la depuración del algoritmo de ordenación *mergesort*, mientras que en la segunda se les planteó el análisis del algoritmo de *competición*, con recursividad múltiple. En la tercera sesión, que tuvo lugar con una versión más avanzada de SRec, similar a la que aquí se presenta, se les planteó un problema donde tenían que eliminar la redundancia debida a la recursividad múltiple empleando técnicas de memorización y tabulación, ofreciendo posteriormente un cálculo de la complejidad de los algoritmos resultantes.

A continuación se adjuntan los datos medios (sobre 5) obtenidos en las preguntas realizadas en el cuestionario de cada una de las sesiones realizadas. Comenzamos por la tabla de resultados elaborada desde los cuestionarios de

la primera sesión de evaluación. Ésta se hizo sobre una versión preliminar de SRec, que posteriormente fue mejorada teniendo en cuenta los puntos flacos observados durante la sesión.

Pregunta	Media
SRec es fácil de usar	3,88
SRec me ha ayudado a analizar algoritmos y encontrar el error	2,63
SRec me ha ayudado a analizar algoritmos para comprobar que la solución propuesta es la correcta	4,13
Calidad general de SRec para analizar la recursividad	3,38
Calidad del menú principal	3,75
Calidad de los controles de animación	4,38
Calidad de la vista de traza	3,75
Calidad de la vista de la pila de control	4,00
Calidad de la vista del árbol de recursión	4,25
Calidad de la configuración de las visualizaciones	3,88
Interacción con los paneles	3,63
SRec me ha gustado	3,63

Tabla 1. Valores del cuestionario de la sesión 1

A continuación se presenta la tabla de la segunda sesión de evaluación.

Pregunta	Media
SRec es fácil de usar	4,50
SRec me ha ayudado a analizar algoritmos para analizar qué llamadas se realizan en tiempo de ejecución	4,29
SRec me ha ayudado a analizar algoritmos para identificar la dependencia de llamadas recursivas	4,36
Calidad general de SRec para analizar la recursividad	4,29
Calidad del menú principal	4,07
Iconos	3,86
Calidad de los controles de animación	4,50
Calidad de la vista de traza	4,00
Calidad de la vista de la pila de control	4,04
Calidad de la vista del árbol de recursión	4,43
Calidad de la configuración de las visualizaciones	3,82
Interacción con los paneles	3,89
SRec me ha gustado	4,26

Tabla 2. Valores del cuestionario de la sesión 2

Se ofrece ahora la tabla de valores medios extraídos desde el cuestionario de la tercera sesión de usabilidad de SRec, empleando una versión mucho más avanzada.

Pregunta	Media
SRec es fácil de usar	4,2
SRec me ha ayudado a analizar algoritmos y determinar las llamadas recursivas que se realizan	4,19
SRec me ha ayudado a analizar algoritmos para identificar la dependencia de llamadas recursivas	4,05
Calidad general de SRec para analizar la recursividad	4,00
Calidad de iconos	3,57
Calidad de los controles de animación	3,71
Calidad de la vista del árbol de recursión	4,00
Calidad del visor de los árboles grandes	3,86
Configuración de formatos	3,76
Configuración de zoom	3,71
Interacción con los paneles	3,80
Proceso de generación de una animación	4,00
Proceso de almacenar / cargar una animación	4,14
Visualización almacenada en un fichero de captura	4,10
SRec me ha gustado	3,95

Tabla 3. Valores del cuestionario de la sesión 3

Las sesiones de usabilidad se realizaron separadas en el tiempo y sobre versiones de SRec cada vez más completas y complejas. Esto requiere de un mayor esfuerzo por parte del usuario a la hora de aprender el manejo de la herramienta, si bien destaca el dato de que la pregunta con mayor puntuación de los cuestionarios 2 y 3 es la que refleja si SRec resulta fácil de usar.

Respecto a los resultados académicos, en la sesión número 2 se apreció que 26 de los 28 alumnos que llevaron a cabo la práctica la entregaron correctamente, y de ellos 23 emplearon SRec para apoyarse en la realización de la misma. En la sesión 3 los alumnos entregaron una práctica similar, y destaca el hecho de que se dobló el número de aprobados con nota *Sobresaliente* respecto a prácticas anteriores de la asignatura junto con que espontáneamente muchos de los alumnos indicaron en el apartado de conclusiones del informe de prácticas que SRec les había ayudado en gran medida a entender mejor el funcionamiento de los algoritmos propuestos así como a depurar los algoritmos que ellos mismos debían implementar.

En general, el alumnado ha sido muy receptivo a SRec, encontrando una herramienta de apoyo que le ha ayudado en la consecución de buenos resultados académicos gracias a las animaciones, que permiten realizar un seguimiento paso a paso muy visual de la ejecución de los algoritmos.

5. Conclusiones y trabajos futuros

Se ha presentado una aplicación informática, SRec, que genera de forma automatizada la animación de la ejecución de programas recursivos. Como ya se ha visto, SRec ofrece numerosas facilidades educativas orientadas a facilitar la labor docente de los profesores y el aprendizaje de los alumnos. Algunas de ellas fueron incorporadas después de las sesiones de evaluación de la herramienta, ya que el objetivo es que la aplicación sea totalmente afín a las necesidades tanto del profesorado como de los alumnos.

Actualmente se encuentra en desarrollo una extensión que incorpora vistas específicas para la técnica de *Divide y Vencerás*. Esta segunda aplicación, que próximamente quedará integrada en SRec, ofrece vistas nuevas, adaptadas a los datos que se desean mostrar, que se muestran en la figura 6.

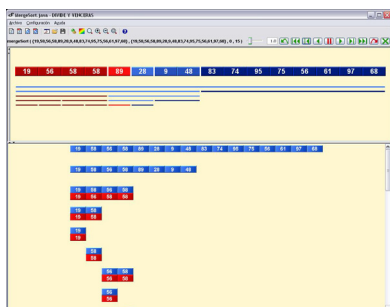


Figura 6. Visualización de otra técnica de diseño de algoritmos: Divide y vencerás

Como se puede apreciar en la figura, la aplicación ofrece una vista cronológica en la parte inferior de la ventana, que identifica en cada paso sobre qué parte de la estructura de datos se está trabajando y cuáles son los resultados que se obtienen. Además ofrece una

segunda vista en la parte superior centrada en la estructura de datos, mostrando mediante líneas y colores qué partes han sido ya tratadas en color rojo y cuáles quedan pendientes de tratar por el algoritmo en color azul.

Otro de los trabajos que quedan planificados para el futuro es permitir que SRec pueda exportar sus animaciones en un lenguaje de animación de algoritmos estándar como SVG. Hasta ahora, SRec emplea un formato propio, basado en XML, pero resultaría de gran interés hacer uso de SRec para generar animaciones sin esfuerzo y poder emplearlas gracias a su exportación en otras aplicaciones.

Agradecimientos

Este trabajo ha sido financiado por los proyectos TIN2004-07568 y TIN2008-04103 del Ministerio de Educación y Ciencia.

Referencias

- [1] Ciesielski, V., MacDonald, P. *Using animation of state space algorithms to overcome student learning difficulties*. Proc. 6th Annual Conference on Innovation and Technology in Computer Science Education. ACM Press, 97-100, 2001.
- [2] MacDonald, P., Ciesielski, V. *Design and evaluation of an algorithm animation of state space search methods*. Computer Science Education 12, 4. 301-324, 2002.
- [3] Naps, T.L., Rößling, G., Anderson, J., Cooper, S., Dann, W., Fleischer, R., Koldehofe, B., Korhonen, A., Kuittinen, M., Leska, C., Malmi, L., McNally, M., Rantakokko, J., Ross, R.J. *Evaluating the Educational Impact of Visualization*. Working group reports from ITiCSE on Innovation and technology in computer science education, 124-136 (2003).
- [4] Naps, T.L., Roessling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., Korhonen, A., Malmi, L., McNally, M., Rodger S., (y autor). *Exploring the role of visualization and engagement in computer science education*. SIGCSE Bulletin, 35(2): 13-152, 2003.