SRec 1.2: visualizador integrado de programas recursivos generales y de Divide y Vencerás

Ángel Velázquez Iturbide
Universidad Rey Juan Carlos
C/ Tulipán, s/n
C.P. 28933 Madrid
91 664 7454
angel.velazquez@urjc.es

Antonio Pérez Carrasco
Universidad Rey Juan Carlos
C/ Tulipán, s/n
C.P. 28933 Madrid
91 488 8266
antonio.perez.carrasco@urjc.es

ABSTRACT
En esta comunicación presentamos una herramienta software, SRec, que permite desarrollar de manera automatizada la visualización animada de programas con múltiples opciones de interactividad y configuración que permiten aumentar la implicación y éxito del alumno en el aprendizaje de los algoritmos recursivos y diseñados bajo la técnica de "Divide y vencerás".

Categorías y Descriptores de Materias
D.3.3 [**Programming Languages**]: Processors – *preprocessors*, H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems – *animations*; I.7.2 [**Document and Text Processing**]: Document Preparation – *makrup languages*; K.3.1 [**Computers and Education**]: Computer Uses in Education.

Términos Generales
Algorithms, Design, Languages.

Palabras Clave
Visualización de programas, animación de programas, recursividad, divide y vencerás.

1. INTRODUCCIÓN
La animación de programas ha sido un campo inexplorado hasta hace tres décadas, cuando se comenzó a investigar sobre las posibilidades que ofrece esta aplicación de la informática educativa. Lo costoso de poner en práctica la visualización de programas [3] provocó una imposibilidad de conocer a fondo los resultados de la aplicación de las visualizaciones en la actividad docente. Por ello, la docencia de la programación y la algoritmia no ha contado de forma masiva con la asistencia de visualizaciones en las clases teóricas y prácticas. De hecho, las visualizaciones por sí mismas no parecen ser un factor decisivo en el éxito del alumno, sino que la componente interactiva, que consigue un mayor grado de implicación del alumno, es el factor clave para alcanzar el éxito en el aprendizaje de la programación y la algoritmia [2].

Existen varios enfoques que permiten reducir el coste que supone la creación de animaciones [1], uno de ellos es la visualización de programas, ligadas al código fuente del mismo. La aplicación que presentamos en esta comunicación sigue este enfoque. Acepta algoritmos programados en lenguaje Java que es capaz de compilar, ejecutar y visualizar, ofreciendo un surtido de posibilidades de interacción que serán presentadas más adelante.

En su versión 1.0, SRec sólo tenía la capacidad de realizar visualizaciones sobre algoritmos recursivos [5]. Esta versión ofrecía menos flexibilidad respecto a la disposición de los paneles en la ventana de la aplicación, ya que su ubicación estaba prefijada, si bien sí se permitía su redimensionamiento.

SRec 1.1 supuso un paso adelante en las opciones de interacción que se brindaban al usuario [4]. Así, la aplicación permite desde entonces seleccionar aquellos métodos que no se desea que formen parte de la visualización para borrarlos de la misma gracias al recálculo del árbol de recursión y de la información que debe mostrarse en todas las vistas. No obstante, la aplicación da la opción al usuario de volver a insertarlos cuando lo desee. También se introdujo la posibilidad de filtrar información dentro de cada método, manteniendo visible o haciendo invisible algunos de los parámetros de entrada de cada uno de los métodos involucrados en la visualización. La selección de información también quedó disponible basándose en la búsqueda de subllamadas recursivas en función del valor de los parámetros de entrada, lo que permite identificar y resaltar en la visualización fácilmente partes de especial relevancia en la ejecución del algoritmo. Además, quedaron disponibles nuevas funcionalidades de recuperación de información estadística.

El nacimiento de SRec 1.2 permite ampliar el tiempo de uso que se puede hacer de la aplicación durante los cursos de programación y algoritmia. Esta nueva versión mantiene las visualizaciones generales de versiones anteriores pero, siendo conscientes de que no son suficientes para ciertos algoritmos recursivos, se ha trabajado con el fin de ampliar el catálogo de vistas utilizables para los algoritmos basados en la técnica de diseño de Divide y Vencerás [6]. En concreto ofrece tres vistas específicas más para los algoritmos basados en esta técnica de diseño de algoritmos. Todas estas vistas dinámicas se muestran

sincronizadas en todo momento y ofrecen información complementaria al usuario.

Con esta incorporación, se puede ampliar el tiempo de uso de la aplicación durante la actividad docente, al ser capaz de ofrecer de manera completa toda la información necesaria sobre algoritmos recursivos generales y diseñados bajo esta técnica en un único sistema mucho más completo y que incorpora una mayor flexibilidad de visualización al mejorar la interacción con los paneles de su ventana.

A lo largo de la presentación se irán exponiendo las vistas genéricas de la recursividad (apartado 3) y las específicas de la técnica de Divide y Vencerás (apartado 4) precedidas de las características generales de la aplicación (presentadas brevemente en el apartado 2). El apartado 5 comenta algunos aspectos de la integración de la técnica de Divide y Vencerás en la aplicación así como de las mejoras colaterales que se han introducido. Tras ello, el apartado 6 ofrece algunos resultados del uso de la versión 1.0, que permitió recoger valiosa información de cara a la implementación de las versiones posteriores 1.1 y 1.2.

2. CARACTERÍSTICAS PRINCIPALES

2.1 Ventana general

SRec ofrece una ventana general (Figura 1) que se compone de una serie de menús de opciones, una barra de herramientas, una barra de animación y varios paneles que albergan las vistas del algoritmo que se desee visualizar.

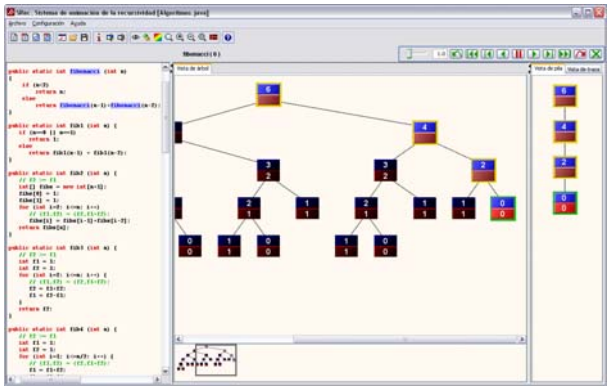

Figura 1. Ventana general de SRec

Los menús permiten cargar clases escritas en lenguaje Java, crear animaciones nuevas, cargar animaciones generadas en una sesión anterior, guardar nuevas animaciones, exportar material gráfico... Dan acceso también a una serie de opciones sobre la visualización activa, como son el control del formato y de la cantidad de información que se muestra. Otras opciones como el zoom, cuadros de información y la búsqueda de llamadas también quedan recogidas en los menús de la aplicación. Como es habitual en las aplicaciones, también se ofrece información corporativa y una aplicación de ayuda interactiva.

La barra de animación, visible en la parte derecha de la Figura 1, es la intuitiva interfaz de manejo de las visualizaciones. Permite activar y pausar las animaciones estableciendo además un periodo de transición elegido por el usuario, así como navegar por los estados de la visualización. La barra de animación ofrece también la posibilidad de rebobinar al principio la visualización, de adelantarla al final, de avanzar o retroceder manualmente un paso, o de plegar o desplegar un subárbol de llamadas por el que no queremos navegar paso a paso.

2.2 Funcionamiento de las visualizaciones

Al generar una nueva visualización, ésta queda situada en el estado inicial; esto es, con la llamada principal del algoritmo ejecutado recién iniciada. Cada paso de la animación descubre una nueva llamada recursiva o resuelve una de las subllamadas ya activadas en los pasos anteriores. De esta manera, se va obteniendo paso a paso el árbol de recursión de toda la ejecución (ejemplo de la serie de Fibonacci en la Figura 2).

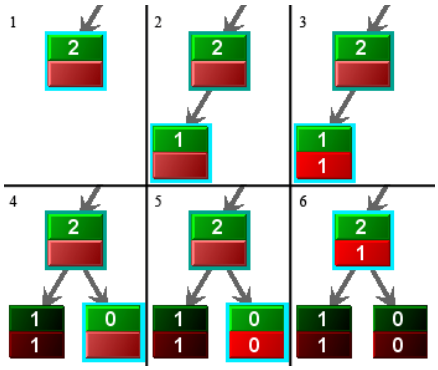

Figura 2. Pasos de transición de una animación

Todas las vistas ofrecen la información con un formato similar para que resulte muy sencillo identificar los datos que se ofrecen en cada una. Este formato es configurable, permitiendo al usuario distinguir por colores la información representada siguiendo varios posibles criterios.

3. VISTAS PARA LA RECURSIVIDAD

SRec ofrece tres vistas dinámicas para la visualización de algoritmos recursivos: árbol de recursión, pila de control y traza de ejecución. Se describen todas ellas a continuación.

3.1 Árbol de recursión

La representación del árbol de recursión (Figura 3) muestra el árbol completo de llamadas recursivas que se han ido sucediendo a lo largo de la ejecución. Cada nodo del árbol representa una llamada recursiva del algoritmo, y contiene la información de los valores de entrada de la llamada y su valor de retorno. Estos dos tipos de información se pueden mostrar en colores diferenciados para una mejor identificación.

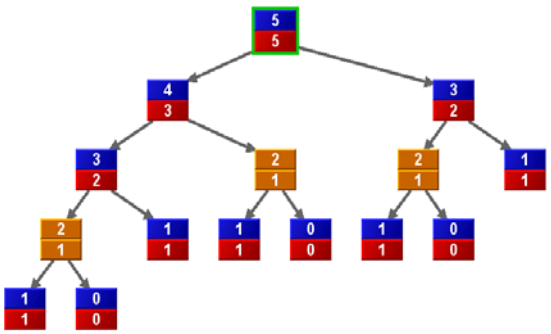

Figura 3. Árbol de recursión con nodos resaltados

Esta vista, que puede contener árboles de gran tamaño, ofrece un visor de navegación que permite contextualizar la parte del árbol que se está viendo respecto al árbol completo. Además, el visor agiliza la navegación sobre el árbol con ayuda del ratón.

La vista del árbol de recursión es la que ofrece mayores posibilidades de interacción, ya que se ayuda del ratón para dar acceso directo a información adicional. Así, mediante la pulsación de los botones del ratón sobre un nodo, se podrá acceder a la información de los valores que alberga un nodo o bien a un menú contextual con una serie de opciones de interés sobre el nodo (información extendida, búsqueda de llamadas idénticas, activación del nodo, etc.).

3.2 Pila de control

La pila de control muestra las llamadas recursivas cuya ejecución se encuentra aún pendiente de finalizar. De las subllamadas que alberga en cada momento, por tanto, no se conoce el valor que devolverán. La apariencia de los nodos mostrados es siempre idéntica a la de vista principal, la del árbol de recursión, con el fin de poder establecer más fácilmente la identificación de las subllamadas representadas en ambas vistas.

Esta vista ayuda a controlar el uso de memoria que realiza el algoritmo, al reflejar muy claramente el contenido de la misma en cada paso de la visualización.

3.3 Traza de ejecución

Esta vista tiene formato textual y muestra la sucesión de subllamadas recursivas realizadas y valores obtenidos durante el transcurso de la ejecución del algoritmo. Es decir, se puede considerar como una traza de seguimiento que permite seguir la ejecución paso a paso de todo el algoritmo. La combinación de colores que emplea también está subordinada a la de los colores de las vistas ya presentadas. La tabulación se ha empleado como recurso para representar el nivel de profundidad de la subllamada.

La traza aporta un orden cronológico que ayuda a comprender cómo se ha desarrollado la ejecución del algoritmo y el orden en que se han ido obteniendo los diferentes valores.

4. VISTAS PARA DIVIDE Y VENCERÁS

SRec 1.2 ofrece como novedad la posibilidad de visualizar de manera integrada algoritmos recursivos y algoritmos diseñados bajo la técnica de Divide y Vencerás. Así, para aquellos algoritmos diseñados bajo esta técnica, SRec es capaz de ofrecer hasta 6 vistas diferentes.

Estas tres vistas específicas otorgan gran protagonismo a la estructura de datos en la que se centra el algoritmo, ya se trate de un array o de una matriz. Las vistas se adaptan a la estructura de datos para mostrar su estado y evolución.

4.1 Árbol de recursión con estructura

Al igual que con el árbol de recursión para la recursividad, esta vista ofrece el árbol completo de subllamadas recursivas que han tenido lugar a lo largo de toda la ejecución. El planteamiento de la información y de la configuración mantiene la misma filosofía que la vista del árbol de recursión convencional. La diferencia estriba en que en cada nodo, aparte de aparecer los valores de entrada y el valor de retorno, se adjunta una representación de la estructura en la que se destaca mediante el uso del color la parte concreta sobre la que actúa el algoritmo en esa subllamada. Esta acotación vendrá dada por algunos de los parámetros de entrada, condición que habrá sido señalizada por el usuario previamente. En la Figura 4 se muestra un ejemplo para el algoritmo Mergesort, con el array de entrada {7,4,10,6}.

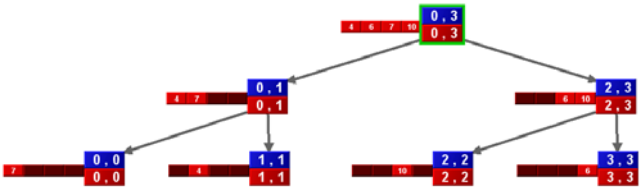

Figura 4. Árbol de recursión con un array como estructura

El array aparece en cada nodo con su valor final, ordenado tras la ejecución del algoritmo. Los valores que aparecen en las celdas son los dos parámetros más de los que consta el algoritmo y que delimitan las posiciones del array que serán objeto del algoritmo en cada subllamada. El método constaba de un parámetro más, un array auxiliar, que se decidió ocultar en la visualización por tener un interés menor.

El funcionamiento de la vista con un algoritmo que maneje matrices es totalmente análogo. En cada nodo se ofrece una representación de la matriz donde queda resaltada el área sobre la que actúa el algoritmo. Se ofrece un ejemplo en la Figura 5.

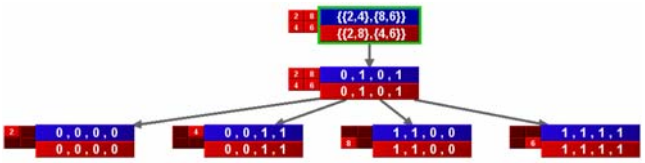

Figura 5. Árbol de recursión con una matriz como estructura

El primer nodo, el que lanza la ejecución, corresponde a un método que recibe una matriz (array de arrays) y que llama a otro diferente al que le pasa, aparte de la matriz, los parámetros que que delimitan el área de actividad del algoritmo en la matriz.

En estos dos ejemplos mostrados, los métodos no devuelven ningún valor, por lo que en las celdas inferiores se muestran los parámetros de entrada, modificados por el algoritmo si procediera.

4.2 Vista cronológica

La segunda vista específica para la técnica de Divide y Vencerás es la vista cronológica. Esta vista muestra los estados que va adquiriendo la estructura de datos en cada paso que se da en la visualización. No se muestran por tanto el resto de parámetros ni de valores que se obtienen en la ejecución. La vista da la opción de elegir entre ver en cada paso la estructura completa o sólo la parte sobre la que actúa el algoritmo en la subllamada correspondiente.

Esta vista muestra dos variantes. La primera (Figura 6, izquierda) ofrece los pasos en orden cronológico, ligando el resultado obtenido a los valores de los parámetros de entrada de esa subllamada, mientras que la segunda variante (Figura 6, derecha) muestra en pasos independientes los valores de entrada y los resultados parciales que se van obteniendo. Aparece un ejemplo en la Figura 6 para el algoritmo Mergesort y el array {8,4,6,1}.

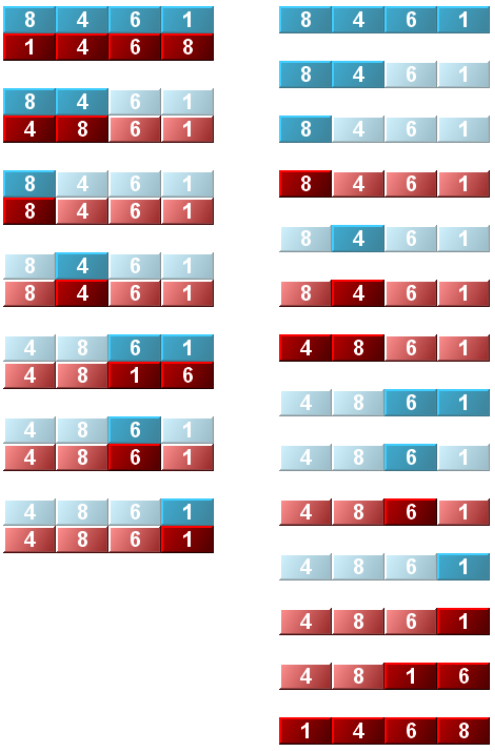

Figura 6. Vista cronológica en sus dos variantes

El orden cronológico es el principal activo de esta vista, ya que ayuda en gran medida al alumno a seguir el transcurso de la actuación del algoritmo sin distraerse en otros elementos del mismo.

La Figura 7 muestra la primera variante para matrices (ejemplo de transposición). Quedan situadas a la izquierda las matrices que se pasan como entrada de las diferentes subllamadas (ordenadas éstas cronológicamente de arriba abajo). A la derecha permanecen las matrices resultado de la subllamada correspondiente.

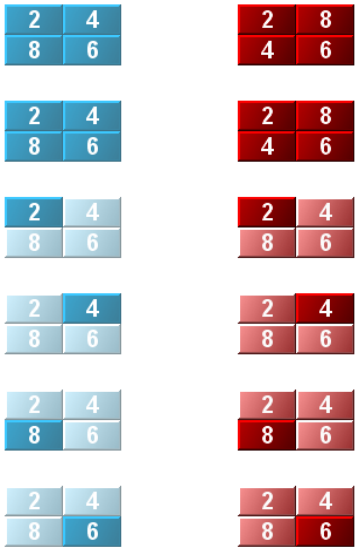

Figura 7. Vista cronológica con matrices (primera variante)

4.3 Vista de estructura

Esta vista, al igual que la vista cronológica, ofrece únicamente información sobre la estructura. Lejos de mostrar una secuencia de pasos, esta vista se limita a mostrar el estado actual de la estructura en cada paso de la visualización.

Emplea líneas y colores para representar las partes de la estructura que ya han sido tratadas en pasos anteriores, las que están siendo tratadas en el paso actual y las partes que aún no se han modificado por el algoritmo. El empleo de colores ayuda a controlar el estado de todas las partes, mientras que el empleo de líneas tiene como objetivo distinguir las distintas partes en que ha sido dividida la estructura.

Durante la visualización de arrays, las líneas dividen el array en varias partes. Con las matrices, las líneas divisorias se sitúan por encima de las posiciones de la matriz permitiendo ver de igual forma qué partes se han manejado de manera aislada y cuáles permanecen aún pendientes de ser tratadas por el algoritmo.

Justo debajo de la estructura aparece de manera esquemática un diagrama de la misma para mostrar en qué partes se ha dividido la matriz, cuáles han sido ya tratadas y cuáles se encuentran aún pendientes, con el fin de complementar al resto de información suministrada. Esta representación esquemática es capaz de representar la jerarquía completa de llamadas que se producen a lo largo del algoritmo.

La Figura 8 muestra varias capturas de esta vista en diferentes momentos del algoritmo Mergesort para el array {8,4,6,1}.

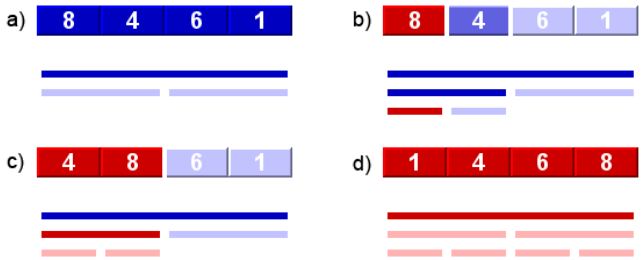

Figura 8. Cuatro capturas de la vista de estructura de array

La captura (a) muestra la estructura en el estado inicial de la visualización. Las líneas inferiores indican que hay una estructura cuyo tratamiento se dividirá inicialmente en dos partes debido a que en el método se encontraron dos llamadas recursivas que parten el array por la mitad. La captura (b) muestra el primer caso base (caso de longitud 1) resuelto, las líneas dejan ver que han existido más divisiones del array que las que estaban proyectadas en la captura (a).

La captura (c) ofrece un estado de la estructura en la que la primera mitad ya ha sido tratada por el algoritmo y queda pendiente de tratar la segunda mitad. La captura (d) ofrece la matriz en su estado final, que al tratarse del algoritmo Mergesort, queda con sus elementos ordenados. En esta captura se puede observar la jerarquía completa de llamadas que se han realizado. Éstas han tenido una profundidad máxima de nivel 3.

Para las matrices, el funcionamiento es análogo, aunque las jerarquías de llamadas se representan mediante rectángulos y no mediante líneas. Se muestra un ejemplo para transposición de matrices en la Figura 9.

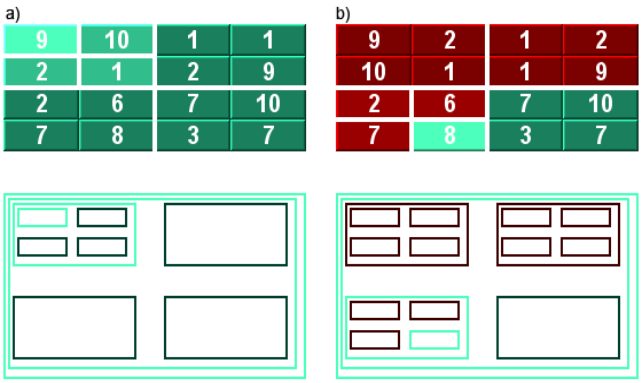

Figura 9. Dos capturas de la vista de estructura de matriz

En la captura (a) aún no se ha resuelto ningún caso base, aunque sí se ha alcanzado el primero que se resolverá (celda de la primera fila y primera columna). El diagrama inferior representa la jerarquía de llamadas que se ha proyectado que se realizará gracias al código del algoritmo recursivo.

En la captura (b) el estado de la ejecución está mucho más avanzado. Los dos cuadrantes superiores ya han sido tratados y el tercero se encuentra resolviendo su cuarto caso base.

En los diagramas cada rectángulo representa una subllamada.

5. INTEGRACIÓN DE LA TÉCNICA DIVIDE Y VENCERÁS

5.1 Integración del procesamiento de los algoritmos

Hacer de SRec un sistema capaz de manejar algoritmos recursivos y algoritmos que han sido diseñados bajo la técnica de Divide y Vencerás ha requerido durante su periodo de desarrollo varios rediseños parciales de la aplicación.

Por un lado, es necesario integrar el motor de procesamiento de clases específico para esta técnica junto al que ya tenía la aplicación para la recursividad general. El objetivo en este proceso ha sido conceder al usuario la mayor transparencia y flexibilidad en el uso de la aplicación.

SRec, para poder ejecutar un algoritmo, necesita que la clase Java que lo contiene sea cargada previamente por el usuario en la aplicación. En ese momento, SRec hace una serie de modificaciones en el código de los métodos contenidos que le permite crear a posteriori las visualizaciones de cada ejecución. Estas modificaciones son específicas de cada técnica, por lo que el usuario simplemente tiene que indicar a SRec qué métodos se desea que sean visualizados con las vistas genéricas y cuáles, además de con las genéricas, con las vistas de Divide y Vencerás.

Para los métodos de recursividad general no es necesario aportar información adicional pero para los métodos que contienen algoritmos de la técnica Divide y Vencerás el usuario debe introducir al menos el número de parámetro que contiene la estructura de datos (array o matriz) sobre la que actúa el algoritmo. Opcionalmente se pueden introducir los números de orden de los parámetros que delimitan las áreas sobre las que actúa el algoritmo en cada subllamada. En la Figura 10 se ofrece una captura del cuadro de diálogo que debe rellenar el usuario al cargar la clase en la aplicación.

Figura 10. Cuadro de carga de clase

Una vez que el usuario selecciona los métodos que desea poder visualizar e introduce la información necesaria en el caso de los algoritmos para la técnica de Divide y Vencerás, la clase queda cargada y disponible para que el usuario ejecute sus métodos tantas veces como quiera.

5.2 Integración de las vistas en la interfaz

Insertar nuevas vistas en la aplicación requería mayor flexibilidad de los recursos que proporcionan las interfaces dentro de las ventanas de la aplicación. Así, se decidió pasar de tres paneles estáticos, que no permitían la modificación de su ubicación en la ventana ni tampoco de su contenido, a los dos actuales, que sí permiten albergar cualquiera de las vistas que ofrece la aplicación. Estos paneles, asimismo, pueden ser colocados en posiciones diferentes, tomando unas proporciones donde prime la horizontalidad o la verticalidad, según las necesidades que determine el usuario.

Cada uno de estos paneles tiene la capacidad de contener varias vistas a las que se puede acceder mediante pestañas. El conjunto de vistas que contiene cada panel es configurable, por lo que es posible mantener visible al mismo tiempo cualquier par de vistas que sean de interés para el estudio del algoritmo. La transición a cualquier otra vista se limita a un click de ratón sobre la pestaña correspondiente.

En la Figura 11 se ofrece una captura parcial de la ventana de SRec donde se pueden ver las pestañas que dan acceso directo a las vistas y el cuadro de diálogo que permite gestionar los paneles tanto a nivel de contenidos como de ubicación.

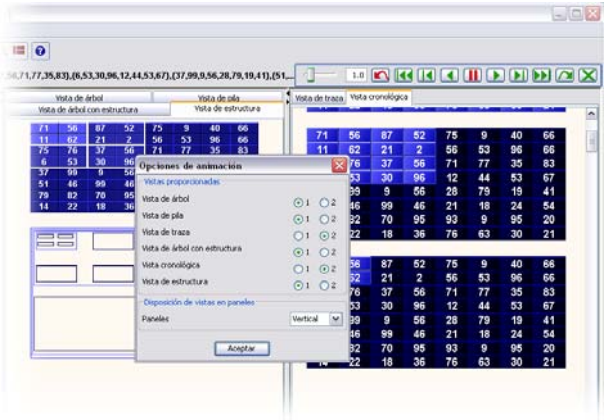

Figura 11. Paneles y cuadro de gestión de paneles

En el ejemplo, cuatro vistas quedan localizadas en el panel 1, situado en el lado izquierdo, mientras que las otras dos permanecen en el panel 2, localizado a la derecha en la imagen.

6. RESULTADOS DE SU USO

SRec ha venido empleándose en la asignatura "Diseño y análisis de algoritmos" de la carrera Ingeniería Informática de la Universidad Rey Juan Carlos desde el curso 2007/2008. En este periodo, se han llevado a cabo tres sesiones de evaluación de usabilidad sobre la versión 1.0 entre los alumnos que han servido para recabar información de gran interés, la cual ha influido positivamente en los desarrollos posteriores de la aplicación que han permitido alcanzar las versiones 1.1 y 1.2, aún no evaluadas.

Estas sesiones, de dos horas de duración, consistían en la elaboración de varios ejercicios de familiarización con la herramienta que precedían a la realización de una práctica puntuable en la asignatura.

En la Tabla 1 se pueden observar algunas de las puntuaciones medias sobre 5 cosechadas por SRec 1.0 en tales sesiones:

Tabla 1. Algunas puntuaciones medias conseguidas por SRec

Cuestión	Puntuación
SRec es fácil de usar	4,2
Calidad para analizar la recursividad	3,9
Calidad de la vista del árbol de recursión	4,2
Calidad de los controles de animación	4,2
SRec me ha gustado	4,0

Las sesiones de evaluación de usabilidad sirvieron, aparte de para recoger la opinión sobre el desarrollo de la aplicación ya realizado, para recopilar sugerencias y críticas con el fin de matizar el posterior desarrollo de SRec y conseguir una aplicación afín a las necesidades tanto del cuerpo docente como del alumnado. Por ejemplo, el visor de navegación para árboles de gran tamaño fue una de las principales reivindicaciones realizadas hasta que fue desarrollado. Otra de las capacidades de que dispone SRec, la exportación de capturas de las diferentes vistas, también fue reclamada por el alumnado para poder insertar tales gráficos en sus informes de prácticas.

En cuanto a la satisfacción personal de los usuarios tras usar la aplicación, en la tercera de las tres sesiones se logró que el 62% de los encuestados, casi dos tercios, otorgaran una nota media superior al 4. Espontáneamente, los alumnos mayoritariamente destacaron la ayuda que habían recibido de la aplicación para realizar la tarea al escribir sus informes de prácticas, donde no se les pidió apreciación alguna sobre la aplicación.

Durante el curso 2009/2010 se realizará una nueva sesión de evaluación de la usabilidad de SRec donde se medirá la aceptación de las nuevas funcionalidades que potencian la interacción de la versión 1.1 y la ampliación de vistas para Divide y Vencerás que proporciona la versión 1.2, entre otras mejoras.

7. CONCLUSIONES

Se ha presentado una aplicación que permite, de manera integrada, visualizar algoritmos recursivos generales y algoritmos diseñados bajo la técnica de Divide y Vencerás. La aplicación proporciona un conjunto de vistas complementarias que ofrecen distintos tipos y grados de interacción y que abren la posibilidad de visualizar paso a paso la ejecución del algoritmo que el usuario haya programado previamente.

SRec ofrece múltiples opciones de configuración y facilidades educativas que facilitan el manejo de la aplicación y, por tanto, predisponen al alumno o profesor a hacer un uso más intensivo de la herramienta sin caer en el temor de que la generación de las visualizaciones será un proceso tedioso gracias a la generación automatizada de las mismas que proporciona SRec.

Los resultados entre los alumnos reflejan una gran aceptación y además destacan la ayuda que les proporciona el programa durante el aprendizaje.

En el futuro próximo, SRec incorporará nuevas vistas para otras técnicas de diseño de algoritmos como la programación dinámica, un campo de visualización poco explorado hasta el momento. SRec añadirá además otras nuevas funcionalidades para completar la asistencia a la docencia de la algoritmia y al aprendizaje de los alumnos en las clases de programación y algoritmia.

Con el fin de aumentar su difusión, se ha creado una nueva web dedicada al programa: http://www.lite.etsii.urjc/srec

8. AGRADECIMIENTOS

Este trabajo ha sido financiado por el proyecto TIN2008-04103 del Ministerio de Ciencia e Innovación.

9. REFERENCIAS

[1] Ihantola, P., Karavirta, V., Korhonen, A., Nikander, J. 2005. Taxonomy of effortless creation of algorithm visualization. Proc. 2005 International Workshop oc Cog Eduaction Research, ACM Press, Nueva York, 123-133.

[2] Naps, T.L., Roessling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., Korhonen, A., Malmi, L., McNally, M., Rodger S., Velázquez, Á. 2003. Exploring the role of visualization and engagement in computer science education. ACM SIGCSE Bulletin, 35(2): 131-152.

[3] Naps, T.L., et al. 2003. Evaluating the educational impact of visualization. ACM SIGCSE Bulletin 35(4), 124-136.

[4] Velázquez Iturbide, Á., Pérez Carrasco, A. 2009. "Aumentando la Interacción con la Visualización de Algoritmos Recursivos". Actas del X Congreso Internacional de Interacción Persona-Ordenador, (Interacción 2009). Aceptado.

[5] Velázquez Iturbide, Á., Pérez Carrasco, A., Urquiza Fuentes, J. 2008. "SRec: An animation system of recursion for algorithm courses". ACM SIGCSE Bulletin, Proceedings of the 13rd Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2008).

[6] Velázquez Iturbide, Á., Pérez Carrasco, A., Urquiza Fuentes, J. 2009. "A design of automatic visualizations for divide-and-conquer algorithms". ENTCS (Electronic Notes in Theoretical Computer Science), Vol.224 (1-Enero-2009), Pág.15