

Antonio Pérez Carrasco

SRec versión 1.0: manual de uso

Número 2008-03

Serie de Informes Técnicos DLSI1-URJC

ISSN 1988-8074

Departamento de Lenguajes y Sistemas Informáticos I

Universidad Rey Juan Carlos

Índice

1. Visión general de la aplicación	3
2. Instalación de SRec	4
3. Animaciones	4
3.1 Procesar clases	4
3.2 Generar animaciones	5
3.3 Introducir valores para los parámetros.....	6
3.4 Cargar animaciones	8
3.5 Guardar animaciones	8
3.6 Utilizar animaciones	8
4. Configuración de animaciones	9
4.1 Datos que se desean mostrar	9
4.2 Ocultación de datos históricos y subllamadas.....	10
4.3 Formato de los datos	10
4.4 Zoom.....	11
5. Lectura y Edición de Código	12
6. Otras opciones	13
6.1 Selección de Máquina Virtual.....	13
6.2 Generación de Archivos Intermedios.....	13
6.3 Idioma de la aplicación	14
6.4 Manejo de Configuraciones	14
Referencias	14

SRec versión 1.0: Manual de uso

Antonio Pérez Carrasco

Universidad Rey Juan Carlos
Departamento de Lenguajes y Sistemas Informáticos I
antonio.perez.carrasco@urjc.es

Summary. SRec es una herramienta destinada a la visualización animada de algoritmos recursivos con fines educativos. La aplicación ofrece múltiples vistas que permiten hacer llegar al usuario información complementaria sobre la recursividad. SRec aporta algunas facilidades educativas, como por ejemplo un alto grado de flexibilidad en su configuración, el almacenamiento de visualizaciones generadas para su posterior recuperación o la exportación de las visualizaciones en formatos gráficos estándar.

1. Visión general de la aplicación

SRec permite visualizar de una manera gráfica, sencilla e intuitiva la ejecución de algoritmos recursivos mediante árboles de activación, aportando también otras vistas dinámicas tales como la pila de control y la traza de la ejecución. También se muestra el código del algoritmo. SRec permite configurar múltiples opciones sobre las visualizaciones. Además permite almacenarlas y cargarlas sin necesidad de generarlas cada vez, manteniendo intactos todos los parámetros de configuración junto con el estado exacto en el que se encuentra la visualización en el momento de su guardado.

SRec también permite visualizar y modificar el código de un algoritmo, bien para corregir errores, bien para modificar el comportamiento de dicho código.

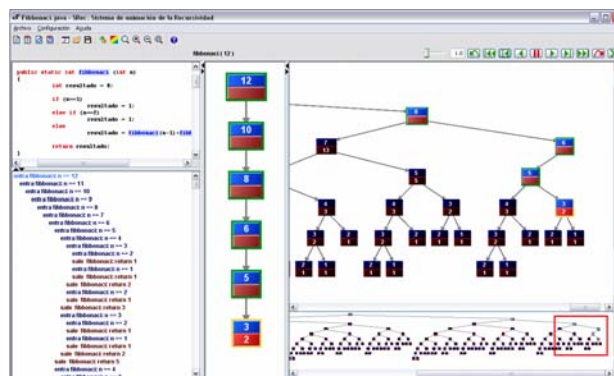


Fig. 1: Ventana de SRec

2. Instalación de SRec

Conseguir que SRec funcione en nuestro equipo es muy sencillo, puesto que no necesita instalación. Los pasos son los siguientes:

1. Descargar la última versión disponible desde la web de SRec. Ésta vendrá dada en un archivo comprimido en formato ZIP
2. Descomprimir en la carpeta deseada el contenido del archivo ZIP
3. Si se desea, se puede crear un acceso directo sobre el archivo “SRec.bat.lnk” para situarlo en el menú de inicio del sistema o en el escritorio.

Ya está instalado SRec en el equipo Windows. Para ejecutar SRec tan sólo es necesario ejecutar el archivo “SRec.bat.lnk”.

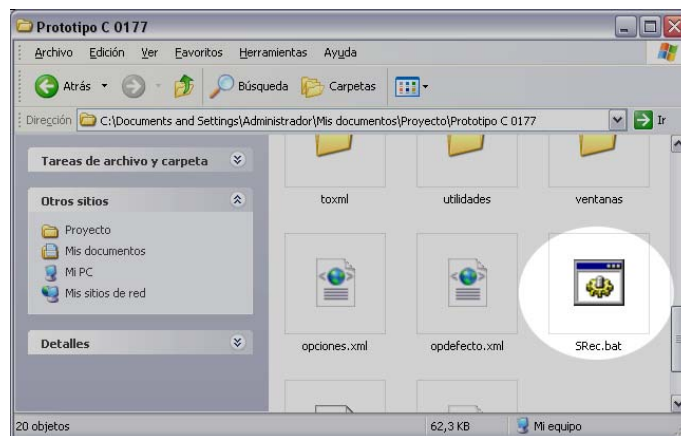


Fig. 2: Carpeta de instalación de SRec

3. Animaciones

SRec permite generar, cargar y almacenar las visualizaciones de los algoritmos que se deseen. En general, todas estas tareas resultan muy sencillas de realizar, puesto que no requieren más de 3 ó 4 pasos.

3.1. Procesar clases

Para poder generar visualizaciones nuevas, es necesario procesar la clase Java en la que se encuentra el algoritmo que deseamos visualizar. Este proceso, que tan sólo toma unos pocos segundos, permite a la aplicación generar un código adicional en Java. Éste incluye las sentencias necesarias para la creación de una estructura que

contendrá la traza de la ejecución. Para iniciar el proceso, es necesario seleccionar la opción “Archivo / Cargar y procesar clase...”.

Esta traza tiene como objetivo almacenar todas las llamadas recursivas que se realicen durante la ejecución del algoritmo seleccionado, junto con los valores de los parámetros de entrada y de su valor de retorno.

A lo largo de este proceso, la aplicación solicita a través de un listado de métodos visualizables que se marquen aquellos métodos que se desea poder visualizar posteriormente durante la sesión actual. No son métodos visualizables aquellos que no tienen ningún parámetro de entrada, los que tienen como parámetro o valor de retorno arrays de más de dos dimensiones o tipos no primitivos. Tampoco podrán ejecutarse métodos privados o aquellos que estén declarados dentro de una clase abstracta.

Los métodos seleccionados serán manipulados por la aplicación con el fin de añadir algunas sentencias adicionales que permitan el almacenamiento de dicha traza. Al finalizar esta etapa de procesado, la aplicación lanza un mensaje advirtiendo la existencia de errores o bien la completitud con éxito del proceso. A partir de entonces, podrán ser generadas tantas visualizaciones como se deseen sobre los métodos procesados.

3.2. Generar animaciones

Para crear una nueva visualización, debe dirigirse a la opción “Archivo / Nueva animación” una vez esté debidamente procesada la clase a la que pertenece. SRec mostrará un cuadro de diálogo que permitirá escoger un método de entre aquellos que fueron procesados, previa elección del usuario.

Para ejecutar el método, la aplicación requiere que se introduzca por parte del usuario el conjunto de valores para los parámetros del método escogido.

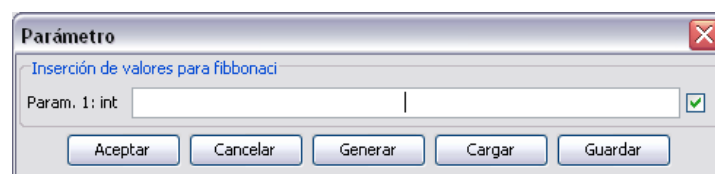


Fig. 3 Cuadro de inserción de parámetros

Es posible que no interese mostrar el valor de alguno de los parámetros durante la visualización. Para ello, el cuadro permite deseleccionar el parámetro o parámetros deseados (cajas de verificación situadas a la derecha), por lo que consecuentemente no aparecerá en la visualización. Al menos un parámetro debe permanecer siempre seleccionado para poder realizar la visualización.

El cuadro de inserción de valores permite además cargar y guardar valores. Para cargar valores almacenados previamente, tan sólo es necesario pulsar el botón

"Cargar" y seleccionar el archivo adecuado. Si el archivo guarda valores para otro método o no es un archivo válido, la aplicación lo advertirá debidamente. Para guardar los valores actualmente escritos en los campos de valores se debe pulsar el botón "Guardar" y ubicar el fichero en disco.

En este punto, la visualización ya está generada y se puede utilizar y configurar con total flexibilidad.

3.3. Introducir valores para los parámetros

Antes de realizar la ejecución del método para poder generar la correspondiente visualización, es necesario que el usuario introduzca un valor válido por cada uno de los parámetros de los que consta el método, tal y como ya se ha comentado.

La manera de introducir valores válidos para cada uno de los tipos de parámetro que admite la aplicación es la siguiente:

- Byte: cualquier número entero que entre en el rango (-128..127).
- Short: cualquier número entero que entre en el rango (-32768..32767).
- Int: cualquier número entero que entre en el rango (-2147483648..2147483647).
- Long: cualquier número entero que se encuentre dentro de los límites del rango definido como (-9223372036854775808..9223372036854775807).
- Float: número con parte decimal opcional (separada de la parte entera por un punto '.'). No se admite notación científica.
- Double: número con parte decimal opcional (separada de la parte entera por un punto '.'). No se admite notación científica.
- Char: un único carácter alfanumérico. Se admiten además ñes y vocales acentuadas con el acento utilizado en la lengua española.
- String: se permite la introducción de cadenas de caracteres alfanuméricas, ñes y vocales acentuadas con el acento utilizado en la lengua española.
- Boolean: se debe introducir la cadena "true" o "false" (sin comillas). Se admite cualquier combinación de mayúsculas y minúsculas.

Para la inserción de arrays, es necesario encerrar los valores entre llaves y separar cada elemento de los demás mediante comas (','). Se permite una cantidad variable de espacios en blanco entre elementos, comas y llaves, pero no de manera interna en un elemento. Ejemplos válidos de arrays:

- Array de enteros: { 400, 74, 28, -823 }
- Array de coma flotante: { 4, -8.7, 7.0001, 2.77778, }

Para la inserción de matrices, se debe introducir un array de arrays. Esto es, se debe encerrar una sucesión de arrays (escritos con las mismas reglas expuestas anteriormente) entre llaves, separando cada uno de los arrays por comas. Ejemplos válidos de matrices:

- Matriz de enteros: { { 400, -434, 46 } , { 4100, 0, -991 } , { 7, 1228, -78 } }
- Matriz de coma flotante: { { 177.11, -54.3 } , { 8, 432.147 } }

Para poder generar de manera aleatoria los valores de los parámetros cuando se está creando una nueva visualización, se debe pulsar el botón "Generar" cuando está visible el cuadro de inserción de valores de parámetros.

Una vez dentro del cuadro, el usuario puede seleccionar todos los parámetros o únicamente algunos de ellos: sólo se generarán valores aleatorios para los parámetros que estén seleccionados en el momento de la pulsación sobre el botón "Aceptar". El cuadro permite establecer umbrales inferiores y superiores para cada parámetro, aunque la inserción de esos valores es opcional (si bien está altamente recomendado introducirlos). Si no se introducen valores, se tomarán los límites que permiten los propios tipos de parámetros.



Fig. 4 Cuadro de generación aleatoria de valores

No se permite la introducción de límites superiores que tengan un valor menor al límite inferior introducido. No se permite tampoco la introducción de valores que pertenezcan a otros tipos (por ejemplo, no está permitido delimitar la generación de enteros mediante la introducción de un valor de coma flotante).

Para arrays y matrices se permite la introducción de valores para el tamaño. Si no se introduce ningún valor en el cuadro correspondiente, los arrays y matrices tomarán tamaños aleatorios. No está permitida la introducción de valores de coma flotante o negativos para la determinación del tamaño de arrays o matrices.

3.4. Cargar animaciones

La aplicación permite cargar una visualización guardada previamente por la aplicación. Para cargarla y poder visualizarla, tan sólo es necesario seleccionar la opción “Archivo / Cargar animación”.

Tras seleccionar el archivo en el que se encuentra almacenada la visualización, la aplicación lee todos los datos sobre la visualización y las opciones de configuración y formato, por lo que se recupera así toda la información útil para mostrar la visualización exactamente igual que aparecía en el momento del guardado. Una vez la animación está cargada, ésta se puede utilizar y configurar con total flexibilidad.

3.5. Guardar animaciones

Si al usuario le interesa guardar una visualización, con un estado concreto y una configuración de opciones y formato determinados, puede hacerlo seleccionando la opción “Archivo / Guardar animación”. Una vez hecho esto, se guarda en un archivo toda la información de interés que permite volver a mostrar la visualización en el estado exacto en el que se guardó, incluyendo las opciones de configuración.

Para exportar una animación en formato GIF tan sólo hay que seleccionar la opción “Archivo / Exportar animación como GIF”. La aplicación reiniciará la visualización y la recorrerá hasta el final de manera visible. Cuando termine, generará el archivo "animacion_SRec.gif", el cual contendrá la animación generada en formato GIF. Se recomienda no realizar un alto consumo de CPU durante este proceso.

La aplicación permite el almacenamiento de una o todas las imágenes de secuencia de la visualización en tres formatos, PNG, GIF y JPG. Para guardar el estado actualmente visible se debe seleccionar la opción “Archivo / Exportar captura...”. Para guardar una imagen por cada estado se debe escoger la opción “Archivo / Exportar capturas de animación...”.

3.6. Utilizar animaciones

Una vez que la visualización está preparada, se habilita un panel en la aplicación que muestra el nombre del algoritmo (método) junto con los parámetros que se han utilizado en la ejecución.

La visualización permite navegar a través de ella hacia adelante (séptimo botón) y hacia atrás (tercer botón), mostrando paso a paso la evolución de la ejecución. Además, se puede reinicializar la visualización colocándola al comienzo de la misma (segundo botón), o bien adelantarla hasta el final (octavo botón), estado en el que el resultado final de la ejecución está calculado y es visible.

También se pueden realizar animaciones de tal forma que la visualización, cada un determinado periodo de tiempo (normalmente menos de cinco segundos), dé automáticamente un paso hacia adelante o hacia atrás, según se desee (acceso mediante los botones sexto y cuarto, respectivamente). Ese periodo se puede ajustar mediante una barra o escribiendo el número de segundos en el cuadro de texto (se admiten valores decimales). Las animaciones pueden ser detenidas en cualquier momento (quinto botón).

Entre las acciones posibles, también está la que permite llevar a cabo un salto en la visualización de la animación que permite adelantar (noveno botón) o retroceder (primer botón) de una sola vez todas las subllamadas contenidas en el nodo activo en ese momento. Una vez que queramos cerrar la visualización, tan solo es necesario pulsar el botón de cierre correspondiente al panel de la visualización (décimo botón).

Los botones se muestran de color verde cuando su pulsación tendrá algún efecto sobre la animación. Toman color rojo cuando su pulsación no tiene sentido o no tendrá efecto alguno sobre la aplicación. Cuando el ratón entra en uno de los botones, éste toma el color naranja, color que también es tomado por los botones cuarto y sexto cuando las animaciones están activas.

Además, la aplicación muestra un visor de navegación que permite posicionar la ventana sobre la parte del árbol que se desee. También ofrece información sobre la posición del panel dentro de la extensión del árbol, lo que ayuda a contextualizar la vista.

4. Configuración de animaciones

Existen múltiples opciones que SRec permite configurar según las preferencias o necesidades del usuario.

4.1. Datos que se desea mostrar

La aplicación permite elegir qué datos deben aparecer en la visualización. Así, pueden mostrarse los valores de los parámetros de entrada junto con los valores de salida, sólo los valores de los parámetros de entrada o bien únicamente los valores de salida.

Para configurar esta opción es necesario seleccionar la opción “Configuración / Formato de árboles” y seleccionar los valores que se desea que aparezcan en la visualización. Los cambios tendrán efecto de manera inmediata sobre la visualización. Bastará pulsar "Aceptar" para salir del cuadro.

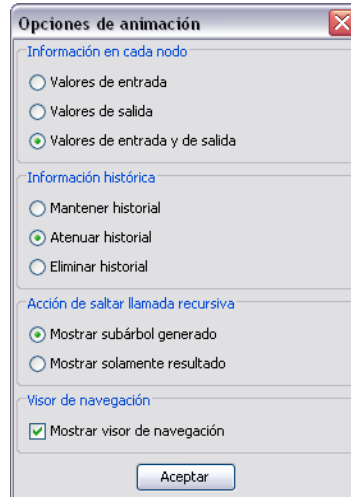


Fig. 5 Cuadro de opciones de las visualizaciones

4.2. Ocultación de datos históricos y subllamadas

La aplicación da la posibilidad de elegir entre mantener datos en la visualización utilizados en pasos anteriores o irlos atenuando o eliminando según van dejando de ser trascendentes en el estado actual de la visualización. El formato en que se muestran los valores atenuados es configurable por el usuario.

Por otro lado, la aplicación permite escoger entre mostrar el subárbol de las subllamadas que se generan cuando se da un salto sobre la llamada actual, o bien mostrar sólo el resultado. Este subárbol se mostrará siempre y cuando la configuración actual no elimine de la visualización los nodos históricos.

Para cambiar estas configuraciones es necesario seleccionar la opción "Configuración / Formato de árboles" y seleccionar en el segundo panel la acción deseada sobre los valores denominados históricos. Los cambios tendrán lugar de manera inmediata en la visualización. Una vez se pulse "Aceptar" el cuadro de diálogo desaparecerá manteniendo la configuración escogida.

4.3. Formato de los datos

Existe total flexibilidad a la hora de dar formato a los datos que se muestran en la visualización. Por un lado, a través de la opción Configuración / Formato tipográfico se pueden aplicar distintos colores y formatos a las celdas de los valores de los parámetros de entrada y a las de los valores de salida.

La primera pestaña está dedicada a la configuración de los colores para los elementos de las vistas del árbol de activación y la pila de ejecución: celdas de entrada, celdas de salida, flechas, marcos de los nodos, fondos de los paneles, etc. Además, también se puede habilitar o deshabilitar el efecto de degradado de las celdas.



Fig. 6 Cuadro de configuración de formato de las visualizaciones

La segunda pestaña muestra el mismo aspecto, pero se orienta a las vistas de traza y código. Además, permite subordinar algunos de los colores de estas vistas a los colores de las vistas de árbol de activación y pila de ejecución, para que sea más fácil asociar, por ejemplo, la visión de los valores de entrada en todos los paneles que se ofrecen.

La tercera de las pestañas, orientada a tamaños, formas y fuentes, ofrece la posibilidad de elegir un borde para las celdas, el ancho de los marcos para el nodo activo y para los nodos que forman parte de la rama actual en expansión, la distancia entre las celdas del árbol recursivo, el tipo de flecha utilizado o el ancho de la misma. En esta tercera pestaña también se puede configurar de manera independiente la fuente que se utiliza en las vistas de código y traza en cuanto a tipografía y tamaño (el color se puede graduar en pestañas anteriores).

Los cambios van teniendo efecto según se van seleccionando los distintos parámetros configurables y asignando los colores deseados.

4.4. Zoom

La aplicación permite ajustar el tamaño al que se muestran los datos en el árbol recursivo de la visualización y en la pila de control. Para ello, basta activar la opción "Configuración / Configuración de Zoom". La aplicación nos ofrecerá la posibilidad de graduar, por separado, el zoom que queremos ejercer sobre la visualización del árbol y sobre la pila de control.

A medida que se vaya graduando el zoom con la barra deslizante, se podrá ir viendo el tamaño que toman las celdas y los datos que contienen. Si se desea retirar el efecto de zoom sobre la visualización, basta pulsar el botón "Resetear" y los valores de zoom se vuelven 0 para mostrar la visualización en su tamaño y proporción original.

Si no se desea hacer uso de la barra deslizante, se puede escribir el valor deseado de zoom en el cuadro de texto. La aplicación permite además el ajuste manual del zoom mediante los botones "Zoom+" y "Zoom-", que van ampliando o disminuyendo el zoom con cada pulsación. El botón de ajuste redimensiona el grafo para que se adapte al tamaño actual del panel y se evite así la necesidad de utilizar la barra de scroll. Una vez que el tamaño de las celdas se ajusta a las necesidades, se puede pulsar el botón "Aceptar" para cerrar el cuadro de diálogo.

5. Lectura y edición de código

Ver de una manera rápida y cómoda el código de cierto algoritmo puede resultar interesante en un momento dado. La aplicación da esta posibilidad a través de la opción "Archivo / Editar clase". Si lo que desea el usuario es editar la clase actualmente procesada, debe escoger la opción "Archivo / Editar clase procesada" (para que los cambios tengan efecto sobre la visualización, será necesario reprocesar la clase y generar una nueva visualización).

Una vez seleccionado el archivo deseado que se desea ver o editar, la aplicación muestra el código, dando la opción de edición sobre el mismo. Los botones que muestra el visor de código son los siguientes:

- "Abrir": Permite seleccionar otro archivo para visualizar su código en esa misma ventana.
- "Guardar": Guarda el archivo con el código que se encuentre escrito en la ventana. El botón no se encuentra disponible cuando el archivo guardado está actualizado con el contenido de la ventana.
- "Compilar": La aplicación da la opción de compilar el código Java y de poder revisar los errores a través del panel inferior de consola.
- "Cortar": Corta el texto seleccionado almacenándolo en el portapapeles. El botón no se encuentra disponible cuando no hay ningún texto seleccionado.
- "Copiar": Copia el texto seleccionado almacenándolo en el portapapeles. El botón no se encuentra disponible cuando no hay ningún texto seleccionado.
- "Pegar": Pega el texto almacenado en el portapapeles en la posición del cursor.

Si en cualquiera de los mensajes de error que aparecen se selecciona "java:n:" (de manera exacta, dentro de una selección mayor o haciendo triple clic sobre la línea) aparecerá seleccionada la línea correspondiente en el área de código, para que así resulte mucho más sencillo encontrar los errores sobre el mismo.

6. Otras opciones

La aplicación da una serie de opciones que pueden resultar de interés para el usuario.

6.1. Selección de máquina virtual de Java

Esta opción debe ser configurada para dar a la aplicación la capacidad de compilar archivos de código Java y poder generar nuevas visualizaciones a partir de dichos códigos. Si esta opción no es configurada, la aplicación sólo podrá mostrar visualizaciones generadas previamente en otras sesiones cuando SRec contaba con una máquina virtual disponible.

Para configurar adecuadamente esta opción, se debe seleccionar la opción "Configuración / Máquina virtual Java". El usuario debe dar la dirección local en la que se encuentra ubicada la máquina virtual de Java (es necesario tener instalado el JDK -kit de desarrollo de aplicaciones Java-, no basta con tener el JRE -entorno de ejecución de Java-). En concreto, es necesario proporcionar el path completo del archivo "java.exe", que deberá contar en su mismo directorio con el archivo "javac.exe".

Aparte de escribir la dirección manualmente, existe la posibilidad de navegar por el contenido del disco duro mediante el botón "Examinar".

6.2. Generación de archivos intermedios

La aplicación crea durante la etapa de procesado de clases varios archivos que una vez terminado dicho proceso son eliminados por defecto. Sin embargo, la aplicación puede hacer que estos archivos se queden almacenados en el disco duro si así lo desea el usuario. Para configurar esta posibilidad, es necesario seleccionar la opción "Configuración / Archivos intermedios". Los ficheros que se generan son:

- Archivo con representación XML del código Java original: este archivo contiene una representación en formato XML del código Java que se quiere visualizar.
- Archivo con representación XML del código Java generado: este archivo contiene una representación en formato XML del código Java que la aplicación genera para poder almacenar la información necesaria sobre la ejecución del algoritmo deseado.
- Archivo Java generado: código Java que incluye sentencias adicionales que permiten el almacenaje de ciertos valores que serán utilizados en la visualización.
- Archivo compilado del código Java original: archivo .class del código Java seleccionado, idéntico al generado por el compilador de Java.
- Archivo compilado del código Java generado: archivo .class del código Java que genera la aplicación.

Los archivos que más interés pueden suscitar entre los usuarios podrían ser la representación XML del código Java original y el archivo compilado del código Java original (uso del programa como compilador).

6.3. Idioma de la aplicación

SRec permite ser utilizado en múltiples idiomas. Por defecto, la distribución de SRec ofrece español e inglés. Configurar el idioma en que SRec muestra los menús, los mensajes, y el resto de textos es muy sencillo.

Basta seleccionar la opción “Configuración / Idioma” para que aparezca un pequeño cuadro de diálogo que deja ver qué idioma está seleccionado en la actualidad. También se puede ver mediante un listado de selección qué otros idiomas están disponibles. Una vez que se escoja el idioma deseado, se debe pulsar el botón “Aceptar” para que los cambios tengan efecto. Tras ello, la aplicación se reinicia mostrando todos los textos en el idioma señalado.

6.4. Manejo de configuraciones

SRec ofrece una serie de opciones que permiten al usuario gestionar distintas configuraciones, las cuales puede usar cuando más le convenga mediante la carga desde el sistema de disco. A continuación se explican las siguientes opciones, accesibles desde el menú Configuración: “Configuración / Restaurar configuración”: Recarga la configuración por defecto definida por el usuario. Si hay una visualización abierta en ese instante, los resultados tienen efecto inmediato sobre la misma.

- “Configuración / Abrir configuración”: Permite cargar una configuración previamente almacenada en disco. Si hay una visualización abierta en ese instante, los resultados tienen efecto inmediato sobre la misma.
- “Configuración / Guardar configuración”: Guarda la configuración actual en disco para poder ser cargada posteriormente en otra sesión de la aplicación.
- “Configuración / Guardar configuración por defecto”: Guarda la configuración actual en disco como configuración por defecto para poder ser cargada posteriormente en otra sesión de la aplicación si se quiere recuperar la configuración por defecto.

Referencias

1. L. Fernández, A. Pérez, Á. Velázquez, J. Urquiza. “A framework for the automatic generation of algorithm animations based on design techniques”. *Creating New Learning Experiences on a Global Scale – EC-TEL 2007*, E. Duval, R. Klamma y M. Wolpers (eds.), Springer-Verlag (ISBN 3-540-75194-7), Lecture Notes in Computer Science (ISSN 0302-9743), 4753: 475-480, 2007.

2. Á. Velázquez, A. Pérez, J. Urquiza. “SRec: An animation system of recursion for algorithm courses”. *ACM SIGCSE Bulletin*, 40(3), aceptado, septiembre 2008 (ISSN 0097-8418). También en *Proceedings of the 13rd Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2008)*, ACM Press, 2008, aceptado.
3. L. Fernández, A. Pérez, Á. Velázquez, J. Urquiza. “SRec: Un sistema para la animación de árboles de recursión”. *Actas del Simposio Nacional de Tecnologías de la Información y las Comunicaciones en la Educación (SINTICE 2007)*, Isabel Fernández de Castro (ed.), Thomson, 2007 (ISBN 978-84-9732-597-4), 215-222.
4. Á. Velázquez, A. Pérez, C. Lázaro, J. Urquiza. “Un sistema con múltiples vistas para la visualización y animación de la recursividad”. *Actas do IX Simpósio Internacional de Informática Educativa (SIIE 2008)*, Escola Superior de Educação do Instituto Politécnico do Porto, 2007 (ISBN 978-972-8969-04-2), 331-332.