

Experiencia para la evaluación de VisBack, una herramienta para la visualización de algoritmos de *backtracking*

Carmen Lacave^a, Maximiliano Paredes-Velasco^b, J. Ángel Velázquez-Iturbide^b, Isidoro Hernán^b

^aDepartamento de Tecnologías y Sistemas de Información

Universidad de Castilla-La Mancha (España)

carmen.lacave@uclm.es

^bDepartamento de Ciencias de la Computación, Arquitectura de Computadores, Lenguajes y Sistemas Informáticos y Estadística e Investigación Operativa

Universidad Rey Juan Carlos (España)

{maximiliano.paredes@urjc.es, angel.velazquez@urjc.es, isidoro.hernan@urjc.es}

Resumen En este trabajo se describe la experiencia realizada en la Universidad Rey Juan Carlos y en la Universidad de Castilla – La Mancha para evaluar VisBack, una aplicación que visualiza gráficamente la traza de la ejecución de los algoritmos básicos de combinatoria, basados en la técnica de *backtracking*. El objetivo de VisBack es facilitar el aprendizaje de *backtracking*, una de las técnicas de diseño de algoritmos más difíciles de asimilar por los alumnos de las titulaciones de Informática. Presentamos los resultados obtenidos sobre las emociones de los alumnos, que muestran una disminución de emociones negativas tras su uso. En trabajos futuros presentaremos el análisis de su efecto sobre el aprendizaje.

Palabras clave Visualización, *backtracking*, combinatoria, traza.

Abstract This work describes the experience conducted at the Universidad Rey Juan Carlos and at the University of Castilla - La Mancha to evaluate VisBack, an application that visualizes the execution trace of the basic combinatorial algorithms based in the backtracking technique. The aim of VisBack is to assist students in learning backtracking, which is one of the most difficult algorithm design techniques for Computing degrees students. We present the results regarding students' emotions, which show a reduction of negative emotions. In the near future, we will also present the results of its effect on students' learning performance.

Key Words Visualization, *backtracking*, combinatorial, tracing.

1. Introducción

La recursividad es una técnica de programación que puede ser utilizada para la resolución de diversos y complejos problemas computacionales mediante implementaciones cortas, a partir de la solución del mismo problema para datos más pequeños. La importancia de su aplicación radica en la estrecha relación que tiene con la programación de computadores, las estructuras de datos, las matemáticas discretas, la teoría de grafos y la inteligencia artificial, entre otras. Sin embargo, es considerada como un tema de difícil comprensión para los estudiantes y uno de los principales obstáculos que éstos tienen que enfrentar en los cursos de programación. Es por ello que, como soporte al proceso de enseñanza de la recursividad, se han propuesto representaciones gráficas del

funcionamiento de la recursividad [5], [7], se han desarrollado tutores interactivos diseñados para que los estudiantes visualicen el código y el tiempo de ejecución de los algoritmos recursivos [12], videojuegos [1], [14] y software específico de animación de algoritmos [4] o de programas [2] [17].

Una de las tendencias principales es utilizar *técnicas de visualización*, ya que a pesar de que no siempre mejora el aprendizaje [6], [15], sí que se puede considerar un elemento motivador para el estudiante [9].

En este sentido, los esfuerzos han ido encaminados al desarrollo de diferentes herramientas de visualización de programas recursivos (véase [10] para una revisión completa). Puesto que la tarea de desarrollar este tipo de aplicaciones requiere mucho trabajo [3], conviene determinar previamente qué elementos de la

recursividad son los que más trabajo les cuesta entender a los estudiantes, según la etapa de aprendizaje en la que se encuentren, con el fin de elegir convenientemente las técnicas de visualización que mejor se adapten a sus necesidades.

En [8] se han identificado empíricamente las principales dificultades que tienen los alumnos cuando se enfrentan al aprendizaje de la recursividad, lo que nos ha servido de base teórica para el desarrollo de este trabajo. Por otro lado, la experiencia de los profesores de las asignaturas de “Metodología de la Programación” de la ESI de la UCLM y de “Diseño y Análisis de Algoritmos” de la ETSII de la URJC, les ha permitido concluir que la técnica de programación conocida como *backtracking* o “*vuelta atrás*” es una de la que más dificultades entraña a los alumnos, ya que su codificación suele basarse en recursividad múltiple [8].

Así, nuestro objetivo inicial estaba centrado en el desarrollo de una herramienta para la visualización de algoritmos de *backtracking* con el objetivo de facilitar su comprensión mediante la inclusión de elementos de la recursividad que puedan ser entendidos por el alumno, como puede ser la traza de ejecución del algoritmo.

Por otra parte, *backtracking* está muy relacionado con los problemas de combinatoria, ya que muchos de los problemas tienen como solución una combinación, variación o permutación de un conjunto de elementos. Es por ello que decidimos focalizar el ámbito de este trabajo en la visualización de los algoritmos básicos de combinatoria, generando como resultado la aplicación llamada VisBack [11].

Una vez concluido el proceso de implementación y prueba, la siguiente fase de nuestra investigación consistió en evaluar si, efectivamente, el uso de VisBack mejoraba el aprendizaje, por parte de nuestros alumnos, de la técnica de *backtracking*. Para ello, hemos diseñado una experiencia para comprobar empíricamente si se alcanzaba dicho objetivo, que es la que describimos en este trabajo.

El objetivo de este trabajo es *evaluar el efecto que tiene el uso de la herramienta VisBack para el aprendizaje de backtracking, tanto desde el punto de*

vista de las emociones del alumnado como de la mejora de su comprensión.

El artículo se organiza de la siguiente manera. En la Sección 2 se presenta la aplicación VisBack. A continuación, la Sección 3 describe en profundidad la experiencia desarrollada. Los resultados se muestran en la Sección 4. Finalmente, en la Sección 5 se presentan las conclusiones más relevantes.

2. La herramienta: VisBack

La herramienta VisBack es una aplicación que permite visualizar gráficamente la traza de la ejecución de los algoritmos que generan las variaciones, permutaciones y combinaciones de una serie de elementos, con y sin repetición. Aunque otros sistemas, como SRec, muestran el proceso recursivo de *backtracking* [16], dependen de la codificación de los algoritmos, mientras que VisBack siempre muestra el mismo árbol de exploración.

VisBack visualiza los árboles de exploración generados por las distintas llamadas recursivas, mostrando en cada nodo la información que elige el usuario. Así, se obtiene una visualización de forma clara y sencilla para los alumnos, adaptada a sus necesidades en función de los problemas que les supone la comprensión de *backtracking*, especialmente todo lo relacionado con los mecanismos de retroceso en las llamadas recursivas (vuelta atrás). En consecuencia, la aplicación puede ser utilizada tanto por los alumnos como por los docentes como material de apoyo en sus clases.

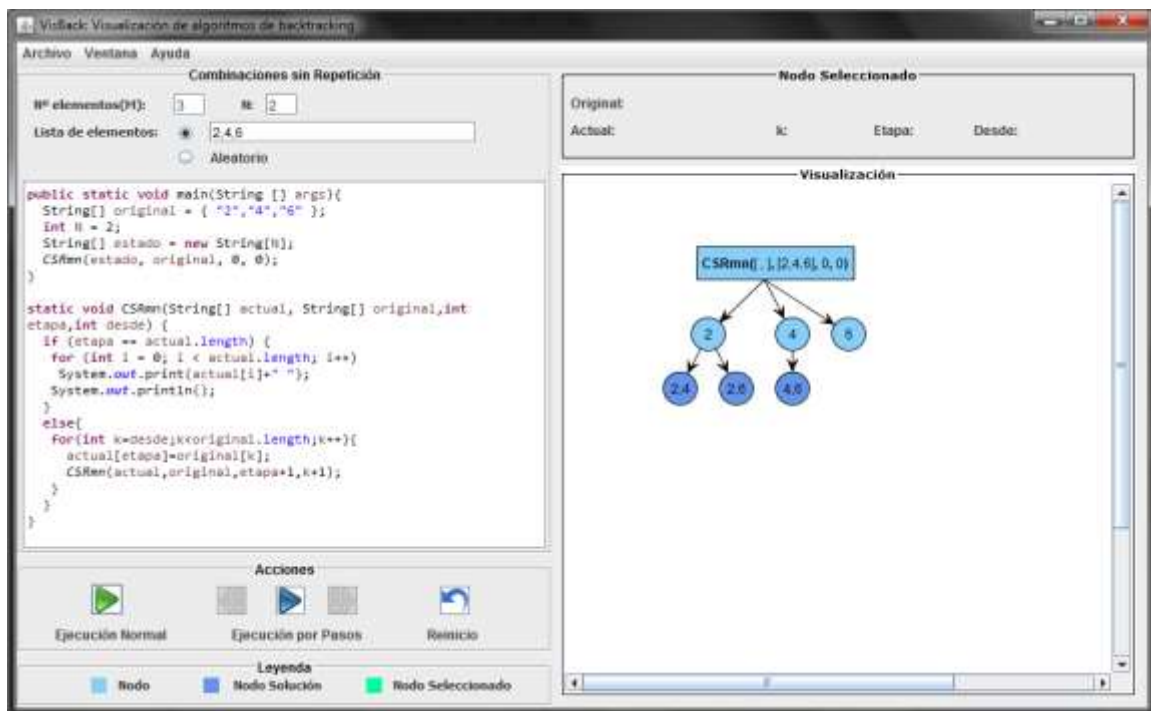


Figura 1. Combinaciones sin repetición de los elementos 2,4 y 6 tomadas de 2 en 2 en VisBack.

VisBack se ha desarrollado en Java utilizando la librería JUNG¹ para visualización de grafos y se puede ejecutar o bien como una aplicación de escritorio o bien como plugin de Eclipse, tanto en inglés como en castellano. Entre otras ventajas de VisBack, merece la pena destacar las facilidades que ofrece al usuario, y que se muestran en la Figura 1, para:

- elegir el algoritmo cuya traza se pretende visualizar, así como cambiarlo en cualquier momento que se requiera, incluyendo su código fuente si así lo desea;
- definir los parámetros de entrada del algoritmo e introducir sus valores, o bien generarlos de forma aleatoria (nº elementos, cardinalidad de los subconjuntos, lista de elementos, etc.);
- decidir si la visualización de la traza desea que se genere en una sola iteración (ejecución Normal), o paso a paso (Ejecución por Pasos); en este último caso, se permite ir también hacia atrás;
- identificar gráficamente las líneas del código fuente que se están ejecutando en cada paso, tal y como ilustra la Figura 2;
- fijar el modo en el que se desea visualizar los nodos; si se elige el modo avanzado, se muestran no solo los valores de los mismos sino los valores de los distintos parámetros del método en ejecución (Figura 3);
- reconocer gráficamente el nodo que esté seleccionado en ese momento puesto que se destaca en distinto color de los demás (verde, en este caso en la Figura 2);
- configurar otros aspectos de visualización como el color y el tamaño de algunos elementos;
- aumentar y disminuir el tamaño de los árboles que constituyen la representación gráfica mediante el uso del zoom;
- guardar y exportar las representaciones gráficas generadas en distintos formatos;
- obtener ayuda tanto para el uso de la aplicación como sobre el código fuente (Figura 4).

¹ <http://jung.sourceforge.net>

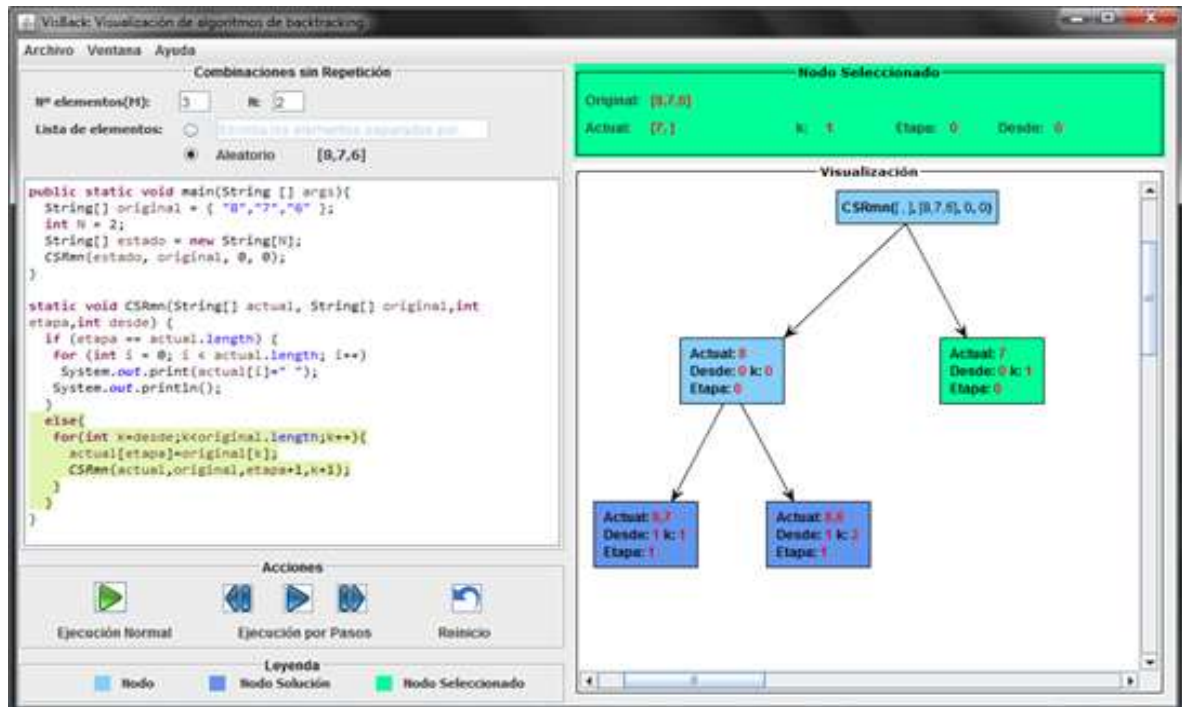


Figura 4. Combinaciones sin repetición de los elementos 4, 7 y 6, tomadas de 2 en 2 en VisBack con visualización avanzada de los nodos.



Figura 2. Pantalla para la definición del modo de visualización de los nodos del árbol, así como del idioma en el que se desea ejecutar VisBack.



Figura 3. Pantalla de ayuda para la definición del modo de visualización de los nodos del árbol, así como del idioma en el que se desea ejecutar VisBack.

3. Diseño experimental

El diseño del experimento ha sido realizado por los autores del trabajo de forma conjunta, por lo que la experiencia ha sido la misma en las dos universidades, variando simplemente en algunos detalles, como el número de participantes o la fecha de realización. En cada universidad, el mismo profesor ha impartido clase al grupo experimental y al grupo de control. En esta sección describimos en qué ha consistido, reflejando las cuestiones en las que ha habido alguna diferencia.

3.1. Participantes

La experiencia se ha llevado a cabo de forma paralela en la Universidad Rey Juan Carlos (URJC) y la Universidad de Castilla-La Mancha (UCLM). En la URJC, han participado voluntariamente 87 alumnos, de los Grados de Diseño y Desarrollo de Videojuegos y de Ingeniería Informática. En la UCLM han participado un total de 89 alumnos de la asignatura *Metodología de la Programación* del Grado en Ingeniería Informática. De ellos, únicamente 70 han participado en todas las fases.

Como factor motivador a los alumnos de la UCLM que realizasen la experiencia completa se les recompensó con el 80% de la nota que pueden obtener en la asignatura por su participación en actividades de clase, que es de 1 punto sobre 10. Por el contrario los alumnos de la URJC no han tenido ninguna compensación en la nota por su participación. Esta decisión ha sido tomada por los profesores en base a su experiencia.

3.2. Metodología

En cada universidad, el alumnado de los distintos grupos de matrícula se asignó indistintamente y de forma aleatoria a dos grupos: uno, llamado *Grupo Experimental (GE)*, que utilizó la herramienta VisBack para resolver los problemas; el otro, llamado *Grupo de Control (GC)*, que no utilizó VisBack para la realización de los ejercicios. Además, los participantes de cada grupo se ubicaron en aulas distintas para que no pudieran tener interacción unos con otros durante la realización de la experiencia.

En la URJC, el GE estuvo formado por 57 alumnos y el GC por 33. La experiencia la finalizaron completamente 36 en el GE y 26 en el GC. En la UCLM, 44 alumnos se asignaron al GE y 35 al de control, aunque realizaron la experiencia completa 41 y 28, respectivamente.

La experiencia se llevó a cabo en las siguientes cuatro fases:

FASE 1 (F1). En la primera, o fase previa de *conocimientos*, el profesorado presentó mediante clase magistral los conceptos teóricos relacionados con el esquema algorítmico de *backtracking* y la estrategia de resolución. Estos conceptos fueron ilustrados con algunos ejemplos clásicos, como la colocación de n reinas en un tablero sin que se amenacen y la obtención de subconjuntos de un conjunto dado de enteros que sumen una cierta cantidad.

FASE 2 (F2). En la segunda, o fase *pretest*, se recopilaron los datos sobre el conocimiento previo de los alumnos participantes, tanto desde su propio punto de vista, como de forma objetiva mediante la resolución de distintas actividades

FASE 3 (F3). La tercera fase, de *experiencia*, consistió en la resolución de varias actividades sobre varios problemas. También se recogieron datos sobre la motivación de cada estudiante a la hora de realizar dichas actividades.

FASE 4 (F4). Por último, en la fase de *postest*, se volvieron a recoger datos para evaluar el conocimiento adquirido y analizar si la motivación había cambiado después de realizar la experiencia.

Todos los cuestionarios así como los enunciados de las actividades a realizar se proporcionaron a través de la herramienta Formularios de Google².

3.3. Recogida de datos

Como ya hemos indicado, se prepararon varios cuestionarios y actividades para obtener la información deseada en cada sesión. Para garantizar el anonimato de las respuestas proporcionadas, los participantes generaron un código que solo ellos podrían conocer pero que nos serviría para poder relacionar los diferentes test que realizasen. Este código lo generaban combinando algunos dígitos de su

² <https://www.google.es/intl/es/forms/about/>

DNI, su número de teléfono móvil y el número del portal o vivienda. Para ello, el encabezado de cada cuestionario incluía el texto siguiente que indicaba al alumno cómo deducir su código particular:

“Las respuestas son tratadas anónimamente y no tienen ninguna repercusión sobre las notas obtenidas en la asignatura. Como la experiencia es totalmente anónima, debes introducir un código (que solo tú sabrás), para poder compararlos. El código se formará introduciendo: último número de tu DNI + letra de tu DNI + tu número del portal donde vives (número de la calle) + dos últimos dígitos de tu número de teléfono. Por ejemplo: Si mi DNI acaba en 1R, vivo en el portal 24 y mi móvil acaba en 34, mi código sería 1R2434. Gracias por tu colaboración”.

Además del código, los estudiantes debían introducir su sexo, edad, grado que estaban estudiando y curso más alto en el que estaban matriculados.

Las respuestas de los cuestionarios fueron presentadas en una escala de tipo Likert de 1 a 5, representando el 1 el valor “Muy poco” o “En desacuerdo”, y el valor 5, “Extremadamente” o “Totalmente de acuerdo”.

En el caso de las actividades, se diseñaron ejercicios de respuesta de tipo test y otras de respuesta corta abierta.

Todos los alumnos contestaron los mismos cuestionarios y resolvieron los mismos ejercicios. Los pertenecientes al grupo experimental, además, respondieron un cuestionario sobre la herramienta VisBack.

3.3.1. Conocimiento previo

La recogida de información sobre la creencia de los alumnos respecto a su grado de conocimiento de los conceptos subyacentes en *backtracking* constituyó la primera parte de la **fase pretest (F2)**. Para ello, los alumnos respondieron al cuestionario que se muestra en la Tabla 1.

Conozco los <i>principios teóricos</i> de la recursividad
Conozco el <i>funcionamiento</i> de la recursividad
Comprendo el funcionamiento de la recursividad
Conozco los conceptos teóricos sobre <i>árboles n-arios</i>
Conozco los algoritmos de <i>recorrido en profundidad de árboles n-arios</i>
Comprendo los algoritmos de <i>recorrido en profundidad de árboles n-arios</i>
Conozco los <i>principios teóricos</i> de los algoritmos <i>backtracking</i>
Conozco el <i>funcionamiento</i> de los algoritmos <i>backtracking</i>
Conozco la <i>codificación</i> de los algoritmos <i>backtracking</i>
Comprendo <i>cómo se genera el árbol de búsqueda de backtracking</i>

Tabla 1. Cuestionario para medir la percepción subjetiva de los alumnos sobre sus conocimientos de *backtracking*

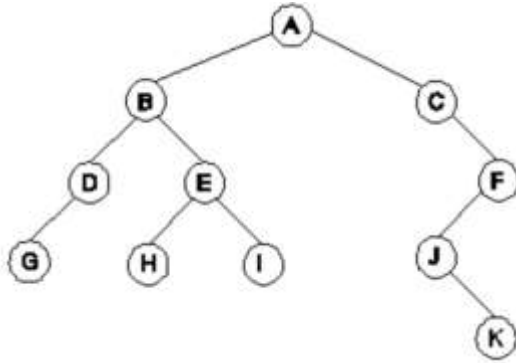
A continuación, se les plantearon dos actividades (Tabla 2) para medir los conocimientos previos sobre dos conceptos básicos asociados a *backtracking*: recursividad y recorrido de árboles.

1. Sea el vector $VE=\{2,5,1,3,4\}$ y el método *pruebaX*, cuya implementación en Java es la siguiente:

```
int pruebaX(int[] v, int li, int ls){
    int pos=-1;
    if (li==ls){
        if (v[li]==li) pos=li;
    }
    else {
        int med=(li+ls)/2;
        if (v[med]==med)
            pos=med;
        else if (v[med]<med)
            pos=pruebaX(v,med+1,ls);
        else pos=pruebaX(v,li,med-1);
    }
    return pos;
}
```

- a. ¿Qué resultado proporciona la llamada *pruebaX(VE,0,4)*?
A. 2 B. 3 C. 0 D. -1
- b. ¿Qué resultado proporciona la llamada *pruebaX(VE,1,2)*?
A. 2 B. 3 C. 0 D. -1

2. ¿Cuál es el recorrido en preorden del árbol de la figura?



- A. ABCDEFGHIJK B. ABDEGHICFJK
 C. ABDGEHICFJK D. GDBHEIACJKF

Tabla 2. Actividades para medir el conocimiento de los alumnos sobre recursividad y recorrido de árboles

3.3.2. Conocimiento sobre backtracking

Acto seguido, y también como parte de la **fase pretest (F2)** se les planteó un problema y, sobre él, diversas preguntas tipo test, que se detallan en la Tabla 3.

Un vendedor quiere visitar N ciudades. Si no quiere repetir ciudades, ¿cuántas rutas distintas puede elaborar si puede empezar y acabar en cualquiera de las ciudades? Suponiendo que $N=3$ y los nombres de las ciudades son R, M, C, ¿qué algoritmo de combinatoria aplicarías para resolver el problema?

- A. Permutaciones sin repetición
 B. Combinaciones sin repetición
 C. Variaciones con repetición
 D. Cualquiera de las opciones anteriores

Teniendo en cuenta el algoritmo que has seleccionado, pinta en un papel el árbol de exploración que se genera al realizar la llamada correspondiente al algoritmo escogido y contesta a las preguntas:

- Número de soluciones
 A. 3 B. 6 C. 9 D. 12
- ¿En cuántas etapas (siendo la llamada externa la primera etapa) se obtiene la o las soluciones?
 A. 0 B. 1 C. 3 D. 4

- ¿Cuántos nodos hay que generar antes de obtener la primera solución (incluido el mismo nodo solución)?
 A. 0 B. 4 C. 6 D. 16
- ¿Cuántos nodos hay que generar para obtener todas las soluciones?
 A. 1 B. 4 C. 6 D. 16
- ¿Qué nodo solución se genera primero?
 A. "C,R,M" B. "M,R,C" C. "R,M,C" D. "C,M,R"
- En el nivel 1 de profundidad del árbol (cuando la variable etapa vale 0) se generan los siguientes tres nodos por este orden:
 A. "C", "M" y "R" B. "R", "M" y "C"
 C. "M", "R" y "C" D. "R", "C" y "M"
- El nodo "M,R" se genera como completación del nodo:
 A. "R" B. "M"
 C. "C" D. Ese nodo no se genera
- Cuando la variable etapa vale 2, estoy en:
 A. nodo hoja B. nodo padre de un nodo hoja
 C. un nodo hijo de la raíz D. nodo raíz
- ¿Qué representa cada etapa?
 A. Añadir una ciudad válida a la solución
 B. Añadir cualquier ciudad a la solución
 C. Tener una solución de tres ciudades
 D. No poder generar más compleciones
- ¿Cómo se generan los descendientes de un nodo?
 A. Añadiendo una ciudad
 B. Añadiendo una ciudad no incluida previamente en la solución
 C. Añadiendo todas las ciudades
 D. Añadiendo todas las ciudades no incluidas previamente en la solución

Tabla 3. Actividades para medir el conocimiento inicial de los alumnos sobre backtracking

Todos los participantes, tanto del grupo de control (GC) como del grupo experimental (GE) tuvieron acceso a un archivo PDF que contenía el código Java de los algoritmos propuestos en el enunciado para la resolución del problema.

3.3.3. Actividad experimental 1

La fase experimental (F3) consistió en resolver varias preguntas asociadas a un problema. El GE lo resolvió haciendo uso de VisBack, por lo que previamente se les explicó brevemente las funcionalidades del programa. El enunciado proporcionado al GE fue el que se presenta en la Tabla 4.

¿De cuántas formas distintas pueden sentarse Paco, José y Ana en una fila de tres butacas? Utiliza la herramienta VisBack con el algoritmo Permutaciones sin repetición e introduce como datos las iniciales de sus nombres (P, J y A) para generar el árbol de búsqueda. Realiza en primer lugar estas actividades:

1. Introduce los datos y haz una ejecución normal del algoritmo. Observa el árbol que se genera.
2. Configura el tipo de nodo en modo avanzado.
3. Ejecuta paso a paso hasta el nodo "P, A". Interpreta los valores de las variables que muestra el nodo.
4. Ejecuta paso a paso hasta el nodo "A". Interpreta porqué la variable k vale 2 y Etapa vale 0.
5. Ejecuta paso a paso hasta el final. Piensa por qué se genera antes el nodo "A, P" que el nodo "A, J".

Tabla 4. Enunciado de la actividad proporcionada al GE

El enunciado del problema a resolver para el GC fue el mismo (Tabla 4) excepto lo relacionado con el uso de VisBack. Al GC se le proporcionó el código Java del algoritmo de las permutaciones, así como la representación gráfica habitual en los materiales docentes sobre *backtracking*, es decir, el árbol de la traza generado al realizar la llamada inicial al algoritmo sobre un caso de uso diferente al del enunciado del problema (Figura 5).

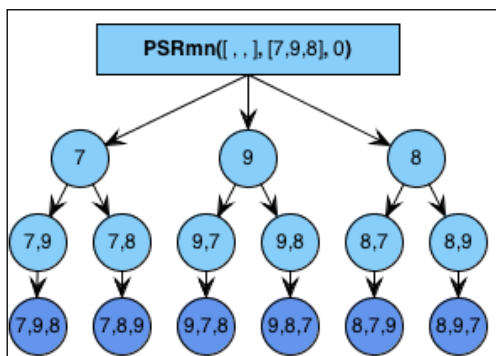


Figura 5. Árbol de la traza de la llamada al método que genera las permutaciones sin repetición de 7, 9 y 8

A continuación, ambos grupos tuvieron que resolver a modo de práctica las preguntas abiertas que se enumeran en la Tabla 5, aunque no se les recogieron los resultados.

Teniendo en cuenta estos datos, se pide que reflexiones y contestes a las siguientes cuestiones para los datos de entrada del problema [P, J, A]:

1. ¿Cuántas soluciones se generan?
2. ¿En cuántas etapas se obtiene la o las soluciones?
3. ¿Cuántos nodos hay que generar antes de obtener la primera solución?
4. ¿Cuántos nodos hay que generar para obtener todas las soluciones?
5. ¿Qué representa cada etapa?
6. ¿Cómo se detecta si un nodo es solución?
7. Los descendientes de un nodo se generan añadiendo....
8. ¿Qué valor tiene la variable Etapa cuando se crea el nodo "P,A"?
9. ¿Qué valor tiene la variable k en el nodo "J,A,P"?
10. ¿Es el nodo "JA" un descendiente del nodo "J" o del nodo "A"? ¿Por qué?
11. ¿Por qué no se crea el nodo "PP"?
12. Dibuja el árbol resultante para los datos de entrada [P,J,A]. No hace falta que lo dibujes entero.

Tabla 5. Enunciados de las preguntas proporcionadas a los dos grupos, GE y GC

3.3.4. Cuestionario PANAS de emociones

A continuación, como parte de la fase experimental (F3) se les propuso a los participantes que completasen el cuestionario PANAS [13] para conocer las emociones y sentimientos que habían percibido durante la realización de la experiencia. Dicho cuestionario consta de 20 ítems que están relacionados con las emociones enumeradas en la Tabla 6.

Interesado (curioso)	Irritable o malhumorado
Tenso o estresado	Dispuesto o despejado
Animado	Avergonzado
Disgustado o molesto	Inspirado
Enérgico, con vitalidad	Nervioso
Culpable	Decidido o atrevido
Asustado	Atento o esmerado
Enojado o enfadado	Intranquilo o preocupado
Entusiasmado	Activo
Orgullosa o satisfecho	Temerosa, con miedo

Tabla 6. Cuestionario PANAS de emociones

3.3.5. Actividad experimental 2

La segunda actividad de la fase experimental (**F3**) fue similar a la primera (4.3.4.) solo que el enunciado del problema a resolver fue distinto, ilustrado en la Tabla 7.

A un concurso literario se han presentado 4 candidatos (P, J, A, C) con sus novelas. El cuadro de honor lo forman el ganador (1º), el finalista (2º) y un accésit (3º), por lo que se desean conocer los distintos cuadros de honor que se pueden formar.

Tabla 7. Problema de la actividad experimental 2

3.3.6. Cuestionario de conocimiento posterior

La última fase de la experiencia (**F4**) consistió, en primer lugar, en responder un cuestionario con preguntas tipo test, similar al realizado en la fase inicial (Sección 4.3.2), para comprobar si el nivel de conocimiento de los alumnos sobre *backtracking* había aumentado y, en el caso del GE, si VisBack había contribuido a la mejora. El enunciado del problema propuesto en el cuestionario es el que se ilustra en la Tabla 8.

Un entrenador de fútbol dispone en la plantilla de su equipo de 5 delanteros (A,B,C,D,E) de la misma calidad y que pueden actuar indistintamente en tres puestos de ataque (A1, A2, A3) del equipo. ¿Qué algoritmo aplicarías para obtener todas las posibles delanteras que puede formar el entrenador?

Tabla 8. Problema del *postest*

3.3.7. Percepción subjetiva sobre la actividad

También se recogió la opinión de los estudiantes sobre cómo habían percibido la experiencia realizada mediante el cuestionario que se muestra en la Tabla 9.

*Considero que la tarea realizada ha sido difícil
He estado muy concentrado resolviendo la actividad
He tenido que esforzarme bastante para resolver la actividad
Me ha resultado difícil resolver la actividad
Creo que he realizado bien la actividad*

Tabla 9. Cuestionario de opinión sobre la experiencia

3.3.8. Opinión sobre VisBack

Por último, para finalizar la fase *postest* (**F4**), los participantes completaron nuevamente el cuestionario PANAS de emociones (Tabla 6) y los del GE, además, respondieron las cinco preguntas que se indican en la Tabla 10 sobre la opinión que les ha merecido la aplicación VisBack.

*Creo que VISBACK es fácil de usar
VISBACK me ayuda a comprender el backtracking
Considero que VISBACK es útil para estudiar backtracking
Usaré VISBACK para estudiar backtracking
Recomendaré VISBACK para estudiar backtracking*

Tabla 10. Cuestionario de opinión sobre VisBack

3.4. Temporalidad

Las cuatro fases se han desarrollado en tres sesiones de 90 minutos: una sesión para la fase 1, otra sesión para la fase 2 y parte de la fase 3; y otra sesión para el resto de la fase 3 y la fase 4, tal y como ilustra la Tabla 11.

La sesión 1 se dedicó íntegramente a la primera fase (**F1**) de la experiencia, que consistió en la exposición de los conceptos teóricos de *backtracking* así como a la resolución de varios problemas en el aula.

La segunda sesión se dedicó tanto a la realización de la segunda fase (**F2**) y parte de la tercera (**F3**). Así, los primeros 5 minutos se dedicaron a recordar los conceptos de combinatoria; a continuación los estudiantes dispusieron de 30 minutos para completar los cuestionarios de conocimiento previo (Tablas 1 y 2) y el *pretest* sobre conocimientos de *backtracking* (Tabla 3). También en este momento se realizó el test de emociones de los alumnos (Tabla 6). Los 30 minutos siguientes se dedicaron a realizar la primera actividad experimental de la tercera fase (Tablas 4 y 5).

La tercera sesión se dedicó a finalizar la tercera fase (**F3**) y completar la fase cuarta (**F4**). Así, durante los 25 primeros minutos los estudiantes realizaron la segunda actividad experimental (Tabla 7) para pasar a

continuación a responder los cuestionarios de conocimiento posterior (Tabla 8), de percepción de la actividad realizada (Tabla 9), de emociones y los del GE además proporcionaron su opinión sobre la herramienta VisBack (Tabla 10).

	Fecha	27-31/3		3-7/4		18-21/4
URJC	Semana nº	10		11		
	Sesión	S1	S2	S3		
	Fase	F1	F2/F3	F3/F4		
UCLM	Semana nº			9		10
	Sesión			S1	S2	S3
	Fase			F1	F2/F3	F3/F4

Tabla 11. Temporalización de la experiencia

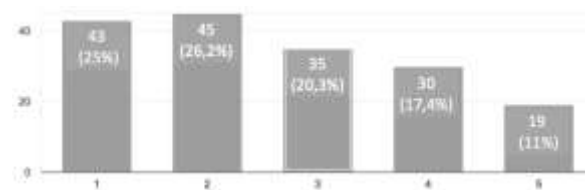
4. Resultados

En relación al tamaño de la muestra de la experiencia, debemos indicar que el número de alumnos que realizaron el *pretest* de emociones PANAS fue de 172, mientras que el *postest* lo realizaron 152 alumnos. La experiencia completa (es decir, que un mismo alumno realizó tanto el *pretest* como el *postest* y las actividades) fue realizada por 112 alumnos. Como se puede ver, la muestra se redujo de 172, que comenzaron la experiencia, a 112 que la finalizaron por completo. Esta reducción puede ser debida, por un lado, a que hay alumnos que no asisten de forma constante a las clases (20 alumnos hicieron el *pretest* y no realizaron el *postest*); y por el otro lado, a que hubo alumnos que se equivocaron al generar el código, no coincidiendo el del *pretest* con el del *postest*. En total hubo 40 códigos del *postest* que no correspondían con el código de ningún *pretest*, bien porque no habían escrito el mismo código o bien porque no habían hecho el *pretest* (este último caso es poco frecuente).

El 47,7% de los alumnos participantes fueron de la Universidad de Castilla – La Mancha, mientras que el restante, 53,3% fueron de la Universidad Rey Juan Carlos. En el aspecto de género, fue mucho mayor la presencia masculina que femenina (79,1% de chicos frente a 20,9% de chicas) y la mayoría (el 59,9%) tenían una edad comprendida entre 19 y 20 años. La mayoría de los alumnos eran de los cursos de 2º o 3º (el 95,3%).

Si analizamos los resultados de forma conjunta entre los dos grupos (tanto experimental como de control) podemos ver que hay emociones que han bajado su nivel durante la experiencia, como por ejemplo la emoción de estar tenso o estresado. Antes de empezar la actividad, el 11% de los alumnos sienten estar “extremadamente” tensos o estresados (ver Figura 6). Como podemos observar en la gráfica que muestra la Figura 6 este porcentaje de alumnos baja más de la mitad situándose tan solo a un 3,9% a la finalización de la experiencia. Esto se confirma si nos fijamos en que aumenta el número de alumnos que no están nada o casi nada estresados a lo largo de la realización de la actividad, ya que antes de empezar la actividad había un 51% que indicaban no estar tensos o estresados (es decir, los que responden 1 ó 2 en el ítem “Tenso o Estresado”) y al finalizar ha aumentado a un 62%.

Inicio actividad



Finalización actividad

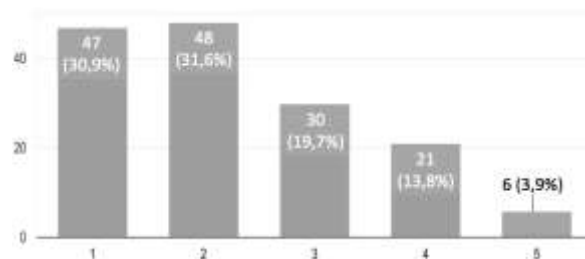
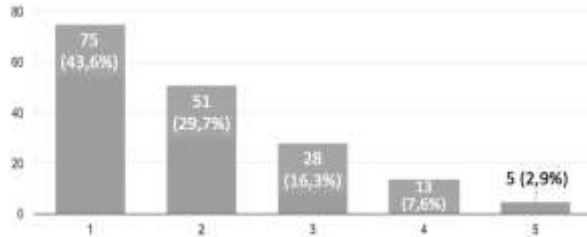


Figura 6. Comparación del nivel de la emoción de tenso o estresado al inicio v final de la experiencia.

La sensación de estar disgustado o molesto también mejora sensiblemente a lo largo de la experiencia (ver Figura 7). Como se puede ver, el número de alumnos que no están “disgustado o molesto” sube sensiblemente de 73% (es decir, los que responden 1 ó 2 en el ítem “Disgustado o molesto”) antes de empezar la experiencia a un 81% a la finalización.

Inicio actividad



Finalización actividad

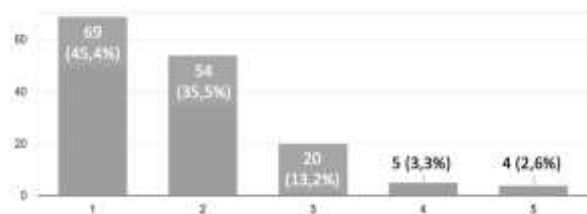


Figura 7. Comparación del nivel de la emoción de disgusto o molesto al inicio y final de la experiencia.

Centrando el foco ya en los resultados *intragrupo*, en concreto en el grupo experimental, se puede ver en la Tabla 12 la media y desviación típica de este grupo para las emociones negativas. El análisis de estos resultados se ha estructurado por tipo de ítem, de tal forma que se ha calculado la media de los ítems de emociones positivas y los de las emociones negativas. La Tabla 12 ilustra que las emociones negativas (como tenso, molesto, etc.) en los alumnos al inicio de la sesión tenían un valor de media de 25,7, mientras que a la finalización de la actividad ésta se redujo a 18,13. Esto nos hace pensar que este tipo de actividades, en las que usaron VisBack, les hace sentirse emocionalmente menos negativo.

Estadísticos descriptivos			
	Media	Desviación estándar	N
Afecto Negativo PRE	25,7538	4,83487	65
Afecto Negativo POST	18,1385	4,02677	65

Tabla 12. Descripción estadística del afecto negativo

5. Conclusiones

En este artículo hemos presentado un estudio sobre la implicación de la herramienta VisBack de

visualización de ejecución de algoritmos en relación a la eficiencia de aprendizaje y al estado emocional del alumno en el aprendizaje de *backtracking*. El estudio se ha centrado en realizar una experiencia con los alumnos en el aula organizados en dos grupos: uno de ellos (grupo de control) ha utilizado los medios clásicos en el aprendizaje de algoritmos (el algoritmo en papel y visualización de ejemplos del árbol de dicho algoritmo), mientras que el otro (grupo experimental) ha utilizado la herramienta VisBack. Aún es pronto para afirmar conclusiones y resultados finales ya que no hemos finalizado de analizar y estudiar los datos estadísticamente. Sin embargo, en una aproximación exploratoria de los resultados, hemos podido ver que este tipo de actividades, disminuye algunas de las emociones negativas, sintiéndose los alumnos menos estresado, tenso o molesto al finalizar la experiencia que cuando la comenzaron.

Como trabajo futuro realizaremos el estudio estadístico de los datos obtenidos, no sólo en el ámbito de las emociones de los alumnos, sino también en el ámbito de la eficiencia de aprendizaje de *backtracking*.

Agradecimientos

Los autores desean agradecer al alumnado de la URJC y de la UCLM su participación en la experiencia.

Este trabajo se ha financiado con los proyectos de investigación TIN2015-66731-C2-1-R y TIN2015-66731-C2-2-R del Ministerio de Economía y Competitividad, S2013/ICE-2715 de la Comunidad Autónoma de Madrid, y 30VCPIGI15 de la Universidad Rey Juan Carlos.

Referencias

- [1] A. Chaffin, K. Doran, D. Hicks, T. Barnes, “Experimental evaluation of teaching recursion in a video game”, *Proc. 2009 ACM SIGGRAPH Symposium on Video Games*, vol. 1, pp. 79–86, 2009.
- [2] W. Dann, S. Cooper, R. Pausch, “Using Visualization To Teach Novices Recursion”, *ACM*

- SIGCSE Bulletin*, vol. 33, no. 3, pp. 109–112, 2001.
- [3] H.L. Dershem, D. Erin Parker, R. Weinhold, “A Java function visualizer”, *Journal of Computing in Small Colleges*, vol. 15, 1999.
- [4] C.E. George, “EROSI - Visualising Recursion and Discovering New Errors”, *ACM SIGCSE Bulletin*, vol. 32, no. 1, pp. 305–309, 2000.
- [5] W. Hsin, “Teaching recursion using recursion graphs”, *Journal of Computing Sciences in Colleges*, vol. 23, no. 4, pp. 217–222, 2008.
- [6] C. Hundhausen, S. Douglas, I. Stasko, “A Meta-Study of algorithm visualization effectiveness”, *Journal of Visual Languages and Computing*, vol. 13, pp. 259-290, 2002.
- [7] C. Kann, R. Lindeman, R. Heller, “Integrating algorithm animation into a learning environment”, *Computers & Education*, vol. 28, no. 4, pp. 223–228, 1997.
- [8] C. Lacave, A.I. Molina, J. Giralt, “Identificando algunas causas del fracaso en el aprendizaje de la recursividad: Análisis experimental en las asignaturas de programación”, *Actas de las XIX JENUI*, pp. 225-232, 2013.
- [9] T. Naps, G. Rößling, V. Almstrum, W. Dann, R. Fleischer, C. Hundhausen, A. Korhonen, L. Malmi, M. McNally, S. Rodger, J.A. Velázquez-Iturbide, “Exploring the Role of Visualization and Engagement in Computer Science Education”, *ACM SIGCSE Bulletin*, vol. 35, no. 2, pp. 131-152, 2003.
- [10] A. Perez-Carrasco, *Sistema Generador de Animaciones Interactivas para la Docencia de Algoritmos Recursivos*. Tesis Doctoral, Universidad Rey Juan Carlos. 2011.
- [11] J.F. Pérez-Mena, “VisBack: Herramienta para la visualización de algoritmos de backtracking”. Trabajo Fin de Grado, Universidad de Castilla-La Mancha. 2015. Disponible en <https://ruidera.uclm.es/xmlui/handle/10578/6665> [Último acceso: 1-8-2017].
- [12] I. Pevac, “First experiences with tutor for recursive algorithm time efficiency analysis”, *Journal of Computing Sciences in Colleges*, vol. 28, no. 1, pp. 56–65, 2012.
- [13] B. Sandín, P. Chorot, L. Lostao, T.E. Joiner, M.A. Santed, R.M. Valiente, “Escalas PANAS de afecto positivo y negativo: validación factorial y convergencia transcultural”, *Psicothema*, vol. 11, no. 1, pp. 37-51, 1999.
- [14] J. Tessler, B. Beth, C. Lin, “Using Cargo-Bot to Provide Contextualized Learning of Recursion”, *Proc. Ninth Annual International ACM Conference on Computing Education Research* pp. 161–168, 2013.
- [15] J. Urquiza-Fuentes, J.A. Velázquez-Iturbide, “A Survey of Successful Evaluations of Program Visualization and Algorithm Animation Systems”, *ACM Transactions on Computing Education*, vol. 9, no. 2, pp. 9:1-9:21, 2009.
- [16] J.A. Velázquez-Iturbide, A. Pérez Carrasco, “How to use the SRec visualization system in programming and algorithm courses”, *ACM Inroads*, vol. 7, no. 3, pp. 42-49, 2016.
- [17] J.A. Velázquez-Iturbide, A. Pérez Carrasco, J. Urquiza-Fuentes, “SRec: An animation system of recursion for algorithm courses”, *Proc. 13rd Annual Conference on Innovation and Technology in Computer Science Education*, pp. 225–229, 2008.