

# USANDO APPLETS PARA LA ENSEÑANZA DE ASPECTOS PROBABILISTICOS EN ANALISIS Y DISEÑO DE ALGORITMOS

*Germán Hernández  
Universidad Nacional  
Bogotá, Colombia*

*Este artículo ilustra como se usan Java Applets para simular y animar de algoritmos con el fin de ilustrar y explorar algunas propiedades, principios y conceptos probabilísticos involucrados en el análisis y diseño de algoritmos. El uso de despliegues gráficos dinámicos e interactivos que ilustran el comportamiento de un algoritmo permite al estudiante comprender de manera más adecuada, a través de exploración, conceptos probabilísticas complicados, incluso en ocasiones permite descubrir propiedades que de otro modo pasarían desapercibidas. Los Java Applets son programas que pueden ser ejecutados dentro una página web, esto hace posible que el estudiante lo utilice en el momento que lo considere más conveniente, lo que contribuye a la creación de un ambiente que estimula el aprendizaje.*

## INTRODUCCIÓN

Los algoritmos son procesos abstractos definidos mediante secuencias de instrucciones computacionales que reciben, transforman y producen datos. Un algoritmo se diseña, en general, para resolver un problema especificado como relación de entrada-salida de datos y que potencialmente será ejecutado por un computador. El diseño del algoritmo incluye también el diseño y análisis de las estructuras (estructuras de datos) que almacenan la información requerida dentro el algoritmo.

En los cursos de algoritmos se busca que el estudiante este en capacidad de analizar y diseñar algoritmos eficientes. Antes de analizar un algoritmo es necesario primero entender como funciona el algoritmo, y esto, aún en los casos más simples, requiere que el estudiante haga animaciones, bien sea en papel o mentales, de la secuencia transformaciones que ocurren en los datos a medida que se ejecutan las instrucciones del algoritmo. Este proceso es complicado para los estudiantes por que se trata de imaginar como a través de un proceso secuencial abstracto (el algoritmo) se transforman objetos abstractos (datos o estructuras de datos). En general no se dispone de análogos concretos ni para el proceso, ni para los objetos que permitan imaginar al estudiante el funcionamiento del algoritmo asociándolo con un proceso concreto conocido.

El recurso usualmente utilizado por los docentes para explicar el funcionamiento de un algoritmo es construir algún tipo de ilustración o representación esquemática para las estructuras de datos sobre los que opera un algoritmo. Luego, para un ejemplo particular de datos de entrada, se ilustra paso a paso como el algoritmo va transformando estas representaciones graficas, de esta manera se produce una secuencia de imágenes que ilustra como el funcionamiento del algoritmo. Este tipo de explicaciones tiene dos limitaciones muy fuertes:

1. *El espacio es muy limitado para una ilustración completa:* aun en los algoritmos más sencillos el número de imágenes que se requiere para una ilustración completa es en general

muy grande. Lo que ocurre en la práctica es que se ilustra el estado inicial de la estructura de datos, a partir de un ejemplo particular de datos de entrada con el que se quiere ilustrar el funcionamiento de algoritmo. Luego al seguir paso a paso las instrucciones del algoritmo se sobrescriben en la ilustración las transformaciones que va realizando el algoritmo sobre los datos. Para el estudiante es muy complicado tomar nota de la secuencia de imágenes que se produce al seguir el proceso y, si el estudiante no tiene una ilustración completa de la secuencia, es muy difícil que pueda de revisar la explicación posteriormente. Es importante anotar que una ilustración completa puede requerir varias páginas lo que obliga a que aun en los libros de texto solo se ilustren muy pocos seguimientos de algoritmos de manera completa.

2. *Limitación temporal para revisar una gama amplia de comportamientos:* como se mencionó antes, en general solo se ilustra el comportamiento del algoritmo para un ejemplo o un número muy pequeño de ejemplos de entrada de datos. Aún en los algoritmos más simples se tiene una gama muy amplia de entradas distintas y comportamientos significativamente diferentes para diferentes entradas. Para darse una idea completa del comportamiento del algoritmo sobre todo el conjunto de posibles entradas uno debería hacer seguimientos al menos sobre los casos típicos y hacer esto manualmente es prácticamente imposible por las limitaciones de tiempo.

Una vez el estudiante ha alcanzado un entendimiento mas o menos completo del algoritmo se puede proceder a etapas mas complejas del análisis que incluyen: 1. análisis de correctitud que busca probar formalmente que el algoritmo cumple la especificación del problema, i.e., probar que el algoritmo produce la respuesta correcta para cada una de las posible entradas y 2. análisis de complejidad que busca determinar la cantidad recursos que requiere el algoritmo para resolver el problema.

Dada la afinidad de los docentes y estudiantes del área de computación con el uso de medios computacionales, la enseñanza-aprendizaje de algoritmos fue una de las primeras áreas en las que se utilizaron y herramientas computacionales interactivas animadas que permiten la exploración dinámica de los objetos y procesos [1]. En la actualidad se dispone en la Internet de un buen numero de este tipo herramientas desarrolladas a lo largo de los años por la comunidad de algoritmos a nivel mundial [2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13]. Se han hecho múltiples estudios sobre el uso de estas herramientas y su efectividad e impacto en el aprendizaje de algoritmos [14, 15, 16, 17, 18, 19, 20, 21, 22, 23].

En este trabajo se aborda el problema de la enseñanza de aspectos probabilísticos de algoritmos usando simulaciones y animaciones que hacen posible que un estudiante realice un mejor aprendizaje y se divierta en el proceso.

En la sección 2, se presentan brevemente el análisis probabilístico de algoritmos y el estudio de algoritmos aleatorizados. En la sección 3, describen como funcionan los Java Applets. En la sección 4, se presentan los Applets desarrollados.

## **1. PROBABILIDAD Y ALGORITMOS**

La probabilidad se ha convertido en los últimos años de los soportes fundamentales de la computación. Hoy el análisis probabilístico de algoritmos [24] y el estudio de algoritmos aleatorizados (randomized algorithms) [25, 26] han pasado de ser una curiosidad teórica a ser herramientas amplia aplicación practica y uso generalizado [27, 28].

### **1.1 Análisis probabilístico de algoritmos**

El análisis probabilístico de algoritmos estudia el comportamiento estadístico de un algoritmo cuando sus entradas tienen una distribución de probabilidad determinada, i.e., dada una distribución de probabilidad sobre las posibles entradas del algoritmo se estudian las distribuciones de probabilidad de: las salidas, del tiempo de ejecución, de espacio requerido, etc. Realizar este tipo de análisis es mucho más complejo que analizar el comportamiento del algoritmo para todas posibles entradas.

### **1.2 Algoritmos aleatorizados**

Hasta hace unos años se tenía la concepción generalizada de que un algoritmo era un proceso secuencial determinístico, hoy se sabe que las soluciones más eficientes de una gran cantidad de problemas son los algoritmos aleatorizados. Este tipo de algoritmos incluyen aleatoriedad en la especificación del proceso que realizan [29], esto hace más complejo comprender su funcionamiento y realizar su análisis es mucho más complicado. La enseñanza-aprendizaje de estos aspectos probabilísticos de los algoritmos es por lejos más compleja que el aprendizaje de los aspectos clásicos de algoritmos y es prácticamente imposible hacerlo sin apoyo de herramientas computacionales. En este artículo se presentan los Java Applets que se han desarrollado, de manera conjunta por el autor y los estudiantes, en los curso de algoritmos [30] y de algoritmos alatorizados [31] de la UN Bogota para el análisis probabilístico de algoritmos y el estudio de algoritmos alegorizados.

## **2. JAVA APPLETS**

En los últimos años hemos tenido un crecimiento asombroso en el número de dispositivos de con capacidad de computo conectados a redes. Ademas de computadores de todos los tamaños y las capacidades conectados en red, hoy tenemos otros dispositivos conectados en red que incluyen: televisores, equipos de audio, teléfonos celulares, fax, scanners, printers, agendas digitales, relojes de pulso que tienen microprocesadores e capacidad de conectarse a redes de multiples formas. Este universo de dispositivos interconectados creo una nueva necesidad de porgramas de computador que pudieran ser enviados de manera segura a traves de las redes. El lenguaje Java se diseño para permitir el desrrollo de este tipo de programas.

### **1.3 Java applets**

Los Java Applets son programas Java, en general pequeños, que descargan (desde un servidor en Internet) y ejecutan (localmente en dipostivo) cuando se carga y despliega una página web, usando un navegador de Internet (p.e., Explorer, Netscape, Firefox, Konkeror, etc). Si se tiene un navegador funcionando en el dispositivo donde se despliega la página web para ejecutar un Applet no se requiere ningún tipo de instalacióna adiconal. La ejecución de los Applets es segura por que todo navegador bloquea cualquier tipo de acceso (lectura/esritura) a los recursos locales

en el dispositivo (discos, memoria, etc). Por tanto no hay riesgo al vusalizar páginas que incluyen Applets.

### 3. APPLETS PARA ALGORITMOS

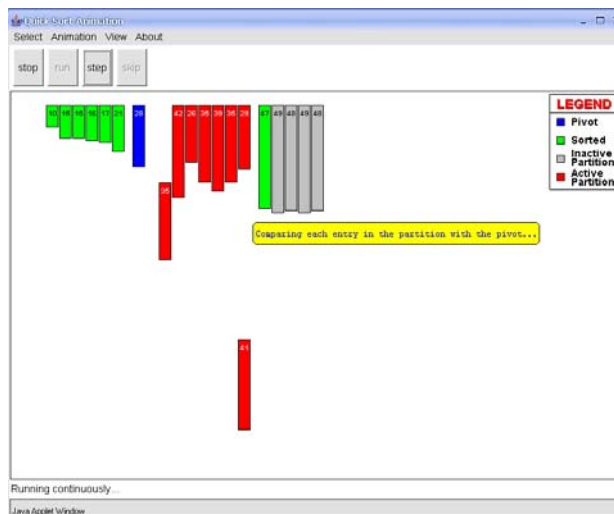
En el curso de Algoritmos se utilizan algunos Applets disponibles en Internet y otros que se han desarrollado en el curso, algunos de estos con la colaboración de los estudiantes del curso.

Figura 1



En la figura 1, se ilustra un Applet disponible desde <http://www.ee.unb.ca/brp/lib/java/insertionsort/> donde se ilustra de manera animada el funcionamiento de un algoritmo de ordenamiento de datos llamado ordenamiento por inserción (insertion sort). Como se observa el Applet tiene tres paneles, en uno de los paneles está escrito el algoritmo (code panel) y que muestra con una línea sombreada cual instrucción se esta ejecutando; un segundo panel (animation panel) que exhibe en cada paso cual es el estado de los datos y un tercer panel (animation control) donde se tiene los botones y barras de control. Es evidente la facilidad que ofrece el applet en la comprensión del Insertion sort.

Figura 2



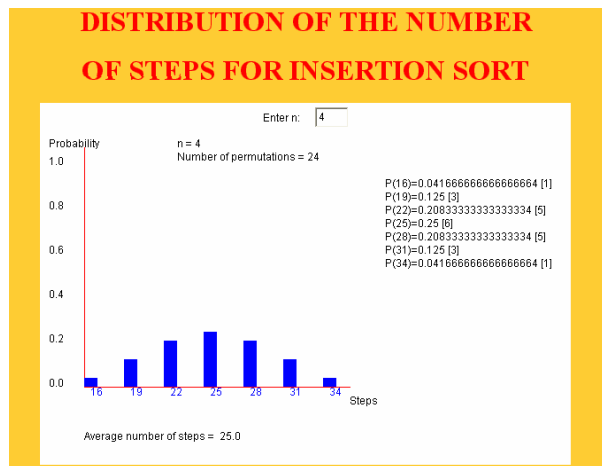
En la figura 2, se ilustra un Applet disponible desde [http://ciips.ee.uwa.edu.au/~morris/Year2/PLDS210/ds\\_ToC.html](http://ciips.ee.uwa.edu.au/~morris/Year2/PLDS210/ds_ToC.html) que explica el algoritmo Quick sort en este se muestra de manera mas dinámica las transformaciones que se van llevando acabo sobre los datos a medida que se ejecuta el algoritmo.

Figura 3



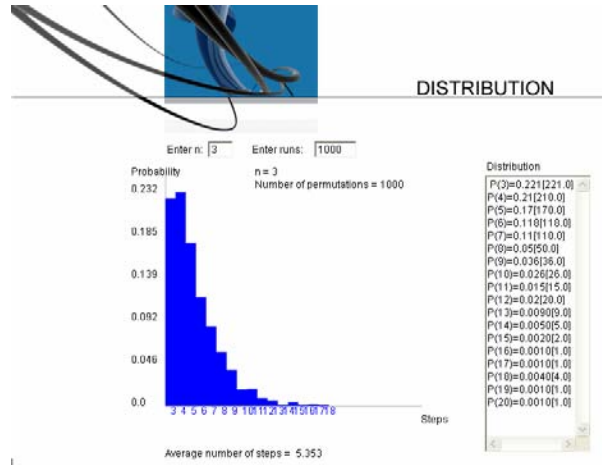
En la figura 3, se ilustra un Applet disponible desde [http://ciips.ee.uwa.edu.au/~morris/Year2/PLDS210/ds\\_ToC.html](http://ciips.ee.uwa.edu.au/~morris/Year2/PLDS210/ds_ToC.html) que explica el procedimiento de solución al para resolver el problema de las torres de Hanoi y despliega el conteo del numero de pasos para solucionarlo en función del numero discos.

Figura 5



La figura 5 muestra uno de los Applets desarrollados por el autor para el curso, disponible en <http://dis.unal.edu.co/~algorithms/applets/InsertSortTimePlot/IsortTimePlot.html>. En esta el estudiante puede simular el Insertion Sort para todos los posibles arreglos de entrada de un tamaño fijo n y estudiar la distribución de probabilidad del tiempo de ejecución.

Figura 6



La figura 6 muestra uno de los Applets desarrollados por los estudiantes del curso, este se encuentra disponible en <http://dis.unal.edu.co/~algorithms/applets/Coupon/Coupon.html>. Los mejores Applets desarrollados por los estudiantes se incorporan al material del curso, lo que ha resultado un incentivo muy poderoso que estimula el trabajo creativo de los estudiantes.

#### 4. EVALUACIÓN

A continuación se presentan los resultados de una encuesta aplicada a 57 estudiantes de dos grupos de algoritmos en I 2006, muestran las dos preguntas fundamentales de la encuesta aplicada y los porcentajes obtenidos para cada respuesta.

De que manera, piensa usted que el uso de applets que permiten animar o simular algoritmos, ha influenciado so aprendizaje de algoritmos?	
Respuestas	Porcentaje contestado
Positiva	76%
Casi no tiene influencia	19%
No tiene influencia	4%
Negativa	2%

La construcción de applets para animar o simular algoritmos incrementa su entendimiento de los algoritmos animados simulados?	
Respuestas	Porcentaje contestado
Si	79%
No	21%

Se puede concluir que el uso y construcción de simulaciones contribuye de manera significativa a mejorar el proceso de aprendizaje de algoritmos para la mayoría de los estudiantes

## REFERENCIAS

- [1] Marc H. Brown and Robert Sedgewick, Techniques for Algorithm Animation, IEEE Software, 2(1):28–39, January 1985.
- [2] Marc H. Brown, Exploring Algorithms Using BALSAS-II, IEEE Computer, 21(5):14–36, May 1988.
- [3] John T. Stasko, TANGO: A Framework and System for Algorithm Animation, IEEE Computer, 23(9):27–39, September 1990.
- [4] Marc H. Brown, Zeus: A System for Algorithm Animation and Multi-View Editing, In Proc. 1991 IEEE Workshop on Visual Languages, 1991.
- [5] J. T. Stasko, Animating algorithms with XTANGO, SIGACT News, 23(2) pp. 67–71, 1992.
- [6] Marc H. Brown and Marc A. Najork. Collaborative Active Textbooks: A Web-Based Algorithm Animation System for an Electronic Classroom. 1996 IEEE Symposium on Visual Languages, 1996.
- [7] J. E. Baker, I. F. Cruz, G. Liotta and R. Tamassia, Algorithm animation over the world wide web, Proc. Int. Workshop on Advanced Visual Interfaces (AVI '96), 1996.
- [8] Steven Robbins. The JOTSA Animation Environment. Thirty-First Annual Hawaii International Conference on System Sciences-Volume 7. 1998. <http://vip.cs.utsa.edu/java/jotsahome/>
- [9] Guido Rosling and Bernd Freisleben. ANIMAL: A System for Supporting Multiple Roles in Algorithm Animation, Journal of Visual Languages and Computing, Volume 13, pp Pages 341-354, 2002.
- [10] K.-P. Lohr and A. Vratislavsky. JAN - Java animation for program understanding. Proceedings 2003 IEEE Symposium on Human Centric Computing Languages and Environments. IEEE Press, 2003.
- [11] Giuseppe Cattaneo, Pompeo Faruolo, Umberto Ferraro Petrillo and Giuseppe F. Italiano. JIVE: Java Interactive Software Visualization Environment, Proceedings 2004 IEEE Symposium on Visual Languages - Human Centric Computing (VLHCC'04), 2004. <http://jive.dia.unisa.it/>
- [12] Susan Rodger and Thomas Finley, JFLAP - An Interactive Formal Languages and Automata Package, Jones and Bartlett Press, 2006. <http://www.jflap.org/>
- [13] Fan Chung Graham, My Top 5 Algorithm Animation Sites. <http://math.ucsd.edu/~fan/math188/bonus/park/top5.html>
- [14] Peter A. Gloor. Proceedings -AAACE- Algorithm Animation for Computer Science Education 1992, DEC Research 1992.
- [15] John T. Stasko. Using Student-Built Algorithm Animations as Learning Aids. SIGCSEB: SIGCSE Bulletin (ACM Special Interest Group on Computer Science Education) 1996.
- [16] Duane Jeffrey Jarc. Assessing the Benefits of Interactivity and the Influence of Learning Styles on the Effectiveness of Algorithm Animation Using Web-Based Data Structures Courseware, PhD. Thesis, George Washington University, 1999.
- [17] Michael Syrjakow and Joerg Berdux. Interactive Web-Based animations for teaching and learning. Proceedings of the 2000 Winter Simulation Conference, 2000.
- [18] Steven Hansen, N. Hari Narayanan. Designing Educationally Effective Algorithm Visualizations. Journal of [19] Visual Languages and Computing, Volume 13, pp. 257-354, 2002.
- [20] Christopher D. Hundhausen, Sarah A. Douglas and John T. Stasko. A Meta-Study of Algorithm Visualization Effectiveness, Journal of Visual Languages and Computing, Volume 13, pp. 259-290, 2002.
- [21] Aloka Nanjappa and Michael M. Grant. Constructing on Constructivism: The Role of Technology. Electronic Journal for the Integration of Technology in Education, Volume 2, No. 1, Spring 2003. <http://ejite.isu.edu/>
- [22] Q. He and P. Tymms. A computer-assisted test design and diagnosis system for use by classroom teachers. Journal of Computer Assisted Learning 21, pp419–429. 2005.
- [23] E. Scanlon and K. Issroff. Activity Theory and Higher Education: evaluating learning technologies. Journal of Computer Assisted Learning 21, pp430–439. 2005.
- [24] Probability and Algorithms. National Academy Press, 1992. <http://books.nap.edu/books/0309047765/html/index>
- [25] Michael Mitzenmacher and Eli Upfal. Probability and Computing: Randomized Algorithms and Probabilistic Analysis, Cambridge University Press 2005.

- [26] Hromkovic, J.,I. and Zámečniková. Design and Analysis of Randomized Algorithms: Introduction to Design Paradigms, Spriger Verlag, 2005.
- [27 ]Motwani R. and Raghavan P. Randomized Algorithms, Cambridge University Press 1997
- [28] Pierre Baldi , Paolo Frasconi and Padhraic Smyth. Modeling the Internet and the Web: Probabilistic Methods and Algorithm, Wiley, 2003.
- [29] Wikipedia. Randomized Algorithm definition [http://en.wikipedia.org/wiki/Randomized\\_algorithm](http://en.wikipedia.org/wiki/Randomized_algorithm)
- [30] <http://dis.unal.edu.co/~algorithms>
- [31] <http://dis.unal.edu.co/~hernandg/ra/>