

Evaluación y uso de Software para la Enseñanza: Caso de un Tema Específico de Estadística

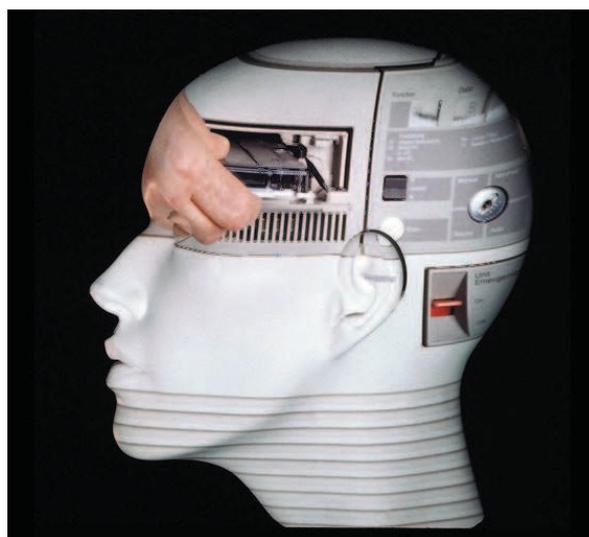
Dr. Juan Manuel Gómez Reynoso¹, MC. Estela Lizbeth Muñoz Andrade¹,
MC. Carlos Argelio Arévalo Mercado²

Palabras clave: Aprendizaje, *software* de propósito específico, métodos formales, usabilidad, calidad del *software*, enseñanza tradicional.

Key words: Learning, purposely-developed software, formal methods, usability, software quality, traditional teaching.

RESUMEN

El aprendizaje es una actividad cotidiana. Aprendemos a través de experiencias, el estudio y la práctica. La tecnología es un medio eficaz que ayuda a las personas a aprender. Existe abundante *software* comercial para entrenar a las personas en el uso de herramientas comunes, tales como las hojas de cálculo o el correo electrónico, pero existen actividades más complejas que requieren conocimientos previos. El presente estudio muestra un *software* desarrollado a la medida como una opción válida para ayudar en el aprendizaje de conocimiento que demanda un entendimiento previo del tema (específicamente, identificación de patrones). El estudio incluyó tanto el desarrollo y prueba del *software* hecho a la medida, como la conducción de un estudio piloto para probar el instrumento que midió el conocimiento adquirido. Finalmente, se condujo un experimento con dos grupos para probar si el *software* desarrollado generaba un mejor desempeño que la enseñanza con medios tradicionales.



ABSTRACT

Learning is an every-day activity. People learn through experiences, studying, and practicing among others. Technology is an effective mean to help people to learn. Commercial software has been used to help people to learn how to use some productivity tools such as a spreadsheet or email. However, there are some other activities that are more complex and require previous knowledge. We believe that purposely-developed software is a good option to help people to learn a specific knowledge that requires previous understanding of the subject (specifically, statistics). In order to study this, special software was developed and tested for usability. In addition, a pilot test was

Recibido 2 de Octubre de 2006, Aceptado 17 de Febrero de 2007

¹ Universidad Autónoma de Aguascalientes, Centro de Ciencias Básicas, Departamento de Sistemas Electrónicos, teléfono 910-84-22, correo electrónico: jmgr@correo.uaa.mx, elmunoz@correo.uaa.mx

² Universidad Autónoma de Aguascalientes, Centro de Ciencias Básicas, Departamento de Sistemas de Información, teléfono 910-84-17, correo electrónico: carevalo@correo.uaa.mx



conducted so that the measuring instrument was tested. An experiment was conducted with two groups in order to probe that purposely-developed software results in better performance than teaching with traditional means.

INTRODUCCIÓN

Tanto individuos como organizaciones enfrentan frecuentemente el reto de actualizar y adquirir nuevo conocimiento. Generalmente, en la enseñanza, el costo se considera como un factor importante (Olfman, L., & Pitsatorn, P., 2000), aunque el uso de la computadora puede ayudar a reducirlo (Brown, K. G., 2001, Buch, K. y Bartley, S., 2002). La enseñanza, con medios tradicionales, genera buenos resultados. Proponemos que el uso de tecnología puede mejorar dichos resultados.

La enseñanza, usando *software*, generalmente se ha centrado en la capacitación de herramientas simples tales como el correo electrónico, las hojas de cálculo o los procesadores de palabras. Pero en temas complejos que requieren el conocimiento previo de conceptos (como es el caso de la estadística), no existen herramientas de *software*

para la enseñanza. Entonces, en temas complejos creemos que un *software* especialmente diseñado, puede ser una mejor opción que con el uso de medios tradicionales. Para investigar esto, se diseñó un *software* especial de apoyo a la enseñanza de un tema complejo: identificación de patrones (control estadístico de procesos).

Por otro lado, es importante que el desarrollo de *software* para la enseñanza tenga calidad para que cumpla con su objetivo. La calidad del *software* puede evaluarse desde el inicio del desarrollo

(Pressman, R. S., 2005) hasta su culminación. Los usuarios perciben la calidad del *software* de diversas maneras, por ejemplo, si el *software* está libre de errores, o si éste no es complicado de aprender y utilizar (Dumas, J. F. y Redish, J. C., 1999). Esta percepción de los usuarios tiene que ver con el concepto de usabilidad, el cual es una medida de la facilidad, efectividad y eficiencia en la forma que puede usarse y aprenderse una aplicación informática. Cabe mencionar que la usabilidad es sólo un factor en el desarrollo de *software*, pero existen algunos más. Para esta investigación se desarrolló un *software* que se sometió a una prueba de usabilidad en dos fases. Adicionalmente, se realizó un experimento con dos grupos: uno usando el *software* especial y otro utilizando medios tradicionales de enseñanza. Los resultados demostraron que la calidad de este *software* para apoyo a la enseñanza de la estadística se puede mejorar evaluando la usabilidad. Además, al final del caso estudiado, se observó que se obtuvieron mejores resultados en el aprendizaje con el uso del *software* que a través de medios tradicionales. La Figura 1 muestra el proceso seguido:

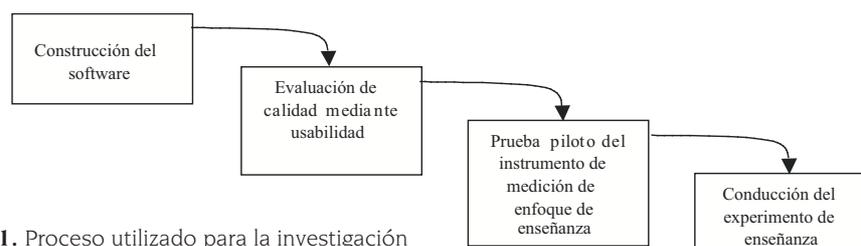


Figura 1. Proceso utilizado para la investigación

MATERIALES Y MÉTODOS

El desarrollo de *software* puede hacerse usando distintas metodologías. Sin embargo, el seguirlas no garantiza que el *software* tenga calidad. Es por eso que es importante adoptar y usar estándares y métodos que permitan asegurar la calidad (Pressman, R. S., 2005, Sommerville, I., 2005). Los sistemas con interfases gráficas (GUI) son una buena forma de prevenir errores de los usuarios (Pressman, R. S., 2005, Sommerville, I., 2006). Además, una buena interfase, entre otros factores, mejora la calidad del *software* (Hayes, P. J., 1999, Vokurka, R. J., Flores, B. E., et al., 1996). Para el presente trabajo, se desarrolló una herramienta de *software* que lleva a cabo análisis de patrones bajo un esquema de sistema experto, con un enfoque en lo visual por medio de una GUI. Aunque el desarrollo de tal herramienta es importante, también lo es probarla exhaustivamente para efecto de que el potencial usuario final realmente se beneficie de ella.

El *software* se desarrolló por prototipo incremental y en su diseño, se buscó que el *software* permitiera a los usuarios explorar a su propio ritmo el conocimiento incluido y realizar simulaciones gráficas. Durante el proceso de prototipo se sometió al *software* a dos rondas de pruebas de usabilidad con el fin de averiguar si ésta se podía mejorar.

A los evaluadores potenciales se les invitó bajo dos criterios: primero, que estuviesen trabajando en proyectos relacionados con el desarrollo de *software*; y segundo, que tuvieran conocimiento de estadística. Se invitó a veintisiete estudiantes de noveno semestre de Ing. en Sistemas Computacionales de la Universidad Autónoma de Aguascalientes y diecinueve de ellos participaron. Para probar la usabilidad, los sujetos usaron el sistema durante 3 horas y contestaron 2 cuestionarios: uno demográfico antes de usar el sistema, y otro acerca del *software* al finalizar. Ambos cuestionarios están basados en literatura previa (Dumas, J. F. y Redish, J. C., 1999, U. of W., U. o. W., 2004). El cuestionario final tiene ocho preguntas cerradas en una escala de Likert de cinco puntos e incluían una sección de sugerencias; una pregunta con 3 posibles respuestas (sí, no, y no sé); y tres preguntas abiertas. Para cada pregunta cerrada se obtuvo la media y la desviación estándar.

Las pruebas se llevaron a cabo con todos los participantes en el mismo laboratorio de cómputo al mismo tiempo. Para efecto de eliminar problemas relacionados con las especificaciones de *hardware*, todas las computadoras contaban con exactamente la misma configuración. Aunque los evaluadores tenían la libertad de cancelar la prueba, todos la completaron. Se agruparon las respuestas de las preguntas abiertas, basándose en su contenido. Al terminar la primera prueba de usabilidad, los problemas detectados y sugerencias por los evaluadores fueron atendidos.

La segunda prueba de usabilidad, se llevó a cabo para determinar si los detalles detectados por los evaluadores fueron resueltos. Los evaluadores fueron los mismos que participaron en la primera prueba, lo cual incrementa la confiabilidad y ayuda a reforzar los resultados. En la segunda ronda de evaluación, sólo se aplicó el cuestionario posterior, debido a que no existía mortalidad de evaluadores, lo cual significa que los resultados en el cuestionario preliminar hubieran sido iguales al anterior. Los resultados en la segunda evaluación presentaron cambios respecto a la primera ya que mostraron menos dispersión, lo cual indica que los usuarios reportaron evaluaciones más similares entre sí, comparados con la primera evaluación de usabilidad del sistema. Al finalizar la segunda ronda, se obtuvo el *software* que permitió evaluar el aprendizaje en el tema complejo seleccionado como prueba. Una vez que el *software* estuvo listo, se realizó un experimento con dos grupos de veinticinco integrantes cada uno. El curso se ofreció para que cualquier interesado se inscribiera. Se podían inscribir alumnos que estuvieran cursando como máximo el sexto semestre de Ing. en Sistemas Computacionales. Una vez que se tuvo el número deseado de participantes se procedió a asignarlos aleatoriamente a uno de los dos grupos. El grupo experimental (EXPG) recibió la enseñanza con el *software*, el grupo de control (CONG) a través de medios tradicionales. Cada grupo trabajó durante cinco sesiones de dos horas cada una. Al final del curso se les aplicó un examen a los participantes. El estudio midió la efectividad del enfoque de enseñanza recibido (variable dependiente) en los dos grupos aplicando un examen al final del curso. Ambos cursos se condujeron de la siguiente forma, en una base diaria durante cinco días consecutivos:

- El grupo experimental (EXPG) hizo uso del *software* para aprendizaje, el cual permitió que los participantes repasaran ejemplos almacenados, crearan nuevos y los analizaran. Además, el sistema también permitía repasar las notas electrónicas del curso. El instructor fungió como consultor puesto que sólo aclaró dudas acerca de la utilización del *software*, pero nunca impartió clase como tal.
- El grupo de control (CONG) recibió clases con medios tradicionales. Es decir, no se utilizó ningún tipo de *software*. El grupo CONG recibió un paquete impreso con las notas del curso. Durante cada sesión se dio el tema del día a los estudiantes. Posteriormente, el grupo solucionó un conjunto de diez ejemplos preparados previamente por los

investigadores, así como algunos ejemplos propuestos por los participantes. El instructor fungió en el papel tradicional puesto que impartió clase de forma usual.

Al final del experimento, se aplicó el mismo examen a los participantes con el objetivo de medir los resultados del enfoque de enseñanza recibido.

En esta investigación, se buscó encontrar indicios de que un *software* desarrollado con un propósito específico de aprendizaje genere mejores resultados que la enseñanza tradicional en el dominio de identificación de patrones. El modelo de investigación se muestra en la Figura 2.

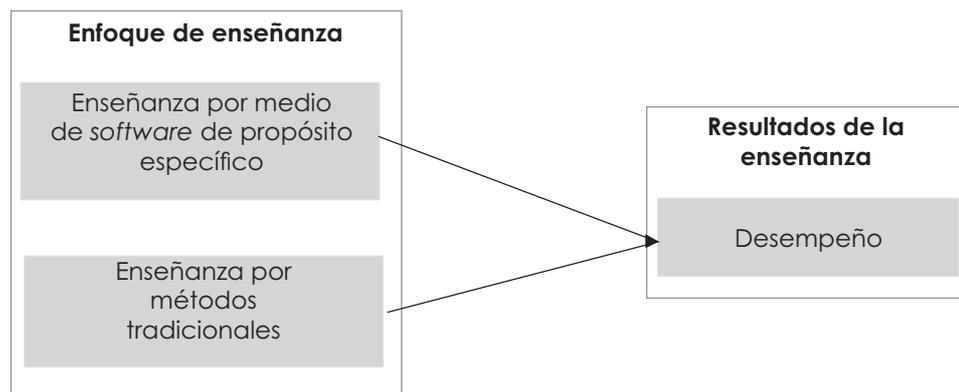


Figura 2. Modelo de la investigación del estudio

RESULTADOS Y DISCUSIÓN

Respecto a las pruebas de usabilidad, los resultados de las encuestas de la primera ronda permitieron atender nueve sugerencias y corregir siete errores del *software*. La media y la desviación estándar fueron calculadas para las preguntas cerradas

como índice comparativo; en la pregunta con tres respuestas se calculó solamente la frecuencia. Los resultados en la segunda ronda muestran menos dispersión con respecto a la primera. La comparación de los resultados se muestra en la Tabla 1. Estos resultados muestran que la calidad que se tenía en la primera ronda se incrementó en la segunda.

Preg.	Primera ronda		Segunda ronda		Cambio	
	Media	Desv. Std.	Media	Desv. Std.	Media	Desv. Std.
1	2.053	0.780	1.737	0.653	0.316	0.126
2	1.947	0.848	1.579	1.507	0.368	0.341
3	1.684	0.582	1.526	0.513	0.158	0.069
4	2.263	0.653	1.842	0.602	0.421	0.051
5	2.474	0.905	1.947	0.621	0.526	0.284
6	1.316	0.749	1.105	0.315	0.211	0.434
7	1.684	0.749	1.421	0.507	0.263	0.242
8	2.211	0.535	1.684	0.478	0.526	0.058

Tabla 1. Resultados de usabilidad

El siguiente paso en el proceso consistió en la validación del instrumento (en este caso, el examen) con el cual se midieron los resultados de los cursos de cada enfoque de enseñanza y para esto, se realizó una prueba piloto. Veinticuatro estudiantes de octavo semestre de Ing. en Sistemas Computacionales participaron en dicha prueba. Estos alumnos fueron seleccionados porque ya habían tomado varios cursos de estadística. Un día antes del examen, cada participante recibió las notas para el curso de entrenamiento que era la base para contestar examen a aplicar al final de los cursos. No se dio ninguna otra clase o sesiones a los participantes. La puntuación fue calculada utilizando la escala tradicional. El tiempo total para responder el examen se registró. La primera persona terminó 21 minutos después del tiempo de arranque, y la última terminó 41 minutos después del tiempo de arranque. La Figura 3 muestra la distribución de las calificaciones que obtuvieron los participantes en la prueba piloto.

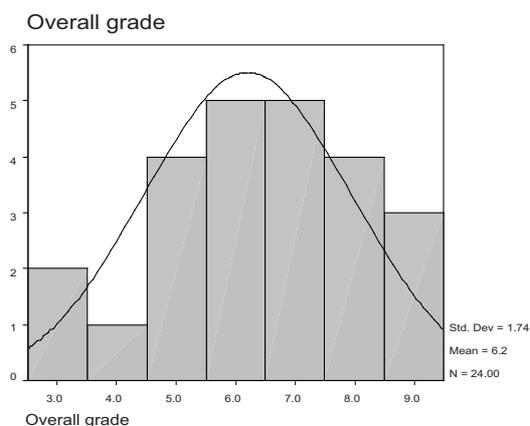


Figura 3. Resultados de la prueba piloto

La Tabla 2 muestra los resultados de la prueba del instrumento. El valor del coeficiente de asimetría calculado (-.489) para las calificaciones está por debajo del error estándar del coeficiente de asimetría (dos veces 0.472). Por lo tanto, se asume que el coeficiente de asimetría está dentro del rango esperado de probabilidad de fluctuación; esto significa que no existen problemas de asimetría. Además, el valor calculado para la curtosis de una muestra no debe exceder a dos veces el error estándar de curtosis (SEK). El valor calculado de curtosis (-.599)

está debajo del SEK (dos veces 0.918). Entonces, se asume que la curtosis está dentro del rango esperado de la probabilidad de fluctuación. Puede decirse que los resultados del instrumento exhiben una distribución normal, por lo que esto indica que el instrumento puede medir lo que se pretende.

N	24
Media	6.1979
Mediana	6.250
Moda	5.75
Desviación estándar	1.7383
Coeficiente asimetría	-.489
Error estándar de coeficiente asimetría	.472
Curtosis	-.599
Error estándar de curtosis	.918
Mínimo	2.50
Máximo	8.75

Tabla 2. Estadísticas obtenidas de la prueba piloto

La Tabla 3 muestra los resultados del examen que intenta medir los efectos del software en la enseñanza. La distribución para EXPG está negativamente cargada, por el contrario, el CONG tuvo una distribución normal (ver figura 3). Los resultados muestran que el EXPG tuvo mejor aprendizaje que el CONG.

	EXPG	CONG
Media	9.270	5.690
Mediana	9.500	5.750
Moda	10.00	2.500
Desviación estándar	0.8383	1.885
Mínimo	6.750	2.00
Máximo	10.00	8.25

Tabla 3. Estadísticas de los resultados del examen

Los resultados de los grupos se probaron usando ANOVA (ver tabla 4) y se observó que sí existe una diferencia en el aprendizaje ($p < 0.001$). Entonces, para la enseñanza de identificación de patrones usando un software especialmente desarrollado arroja mejores resultados que con medios tradicionales.

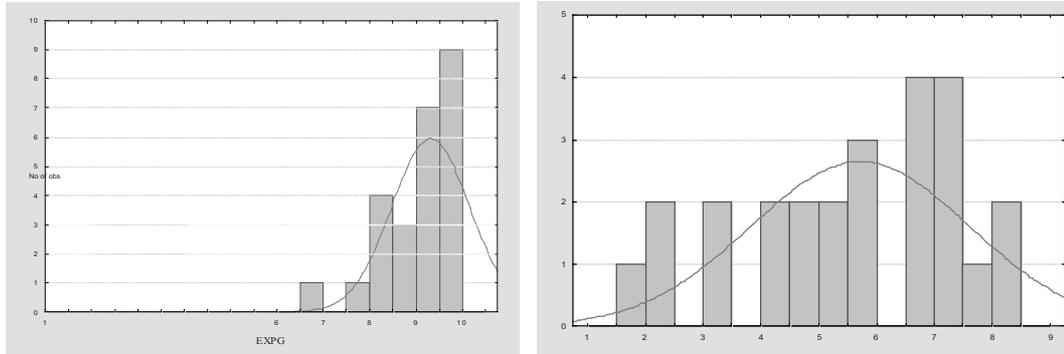


Figura 4. Histogramas de ambos grupos.

	Sum of squares	df	Mean square	F	Sig.
Between groups	160.205	1	160.205	75.280	.000
Within groups	102.150	48	2.128		
Total	262.355	49			

Tabla 4. ANOVA

CONCLUSIONES

En primer lugar, nuestra investigación intentó probar que la calidad del *software* se puede mejorar evaluándolo mediante usabilidad. Los resultados obtenidos nos podrían indicar que la usabilidad incrementa la calidad del *software*, aunque se recomienda investigar más para poder confirmar los resultados obtenidos. Además, a través de esta técnica se pueden desarrollar sistemas de *software* que al ser evaluados, se reduce sesgo que evaluadores tecnológicos puedan introducir, ya que no piensan igual que los usuarios finales. Esto es debido a que los tecnológicos tienen amplios conocimientos del uso de *software* con lo cual podrían evaluar inadecuadamente los requerimientos que los usuarios tienen acerca de un sistema en particular.

En segundo lugar, el *software* desarrollado intenta ser una herramienta que ayude a las personas a aprender. Los resultados obtenidos confirman que una herramienta de este tipo

podría ayudar a tener mejores resultados que la enseñanza usando métodos tradicionales. Analizando tanto la Tabla 3 y la Figura 3 se pudo notar que el desempeño es significativamente mejor usando el *software* que con métodos tradicionales, ya que en el grupo experimental el 64% obtuvo al menos una calificación de 9. Por otro lado, en el grupo de control no se obtuvo ninguna calificación de 9 o superior. Adicionalmente, en el grupo experimental ningún participante reprobó el examen. Por el contrario, en el grupo de control el 44% de los participantes obtuvo una calificación reprobatoria. Es importante agregar que la diferencia en la calificación media en el grupo experimental (9.27) es de 3.58 puntos mayor que el grupo de control (5.69), lo cual representa una diferencia del 38.62%. En cuanto a la dispersión en las calificaciones en el grupo experimental, están más centrados alrededor de la media (0.8383) que el grupo de control (1.885). Finalmente, es importante hacer notar que estos resultados podrían ser aplicables sólo a la investigación realizada y que en otra área los resultados podrían ser diferentes.

BIBLIOGRAFÍA

- BROWN, K. G., Using Computers to Deliver Training: Which Employees Learn and Why?, *Personnel Psychology*. 54, 271-296.
- BUCH, K. y BARTLEY, S., Learning Style and Training Delivery Mode Preference, *Journal of Workplace Learning*. 14, 5-10.
- DUMAS, J. F. y REDISH, J. C., *A Practical Guide to Usability Testing*. England: Intellect Books, 1999.
- OLFMAN, L., & PITSATORN, P., End-User Training Research: Status and Models for the Future, *Framing the Domains of IT Management. Projecting the Future Through the Past*. Cincinnati: Pinnaflex, 2000.
- PRESSMAN, R. S., *Software Engineering: A Practitioner's Approach*. 6ª ed., New York: Mc Graw-Hill, 2005.
- SOMMERVILLE, I. *Ingeniería de Software*. Madrid: Pearson Education, 2005.
- SOMMERVILLE, I., *Software Engineering*. 8ª ed., London: Addison-Wesley, 2006.
- U. of W., U. o. W., Usability test for the University of Washington libraries' information gateway. De: <http://www.lib.washington.edu/usability/redesign03-04/surveyResults.html>, October 05, 2004.
- VOKURKA, R. J., *et al.*, Automatic Feature Identification and Graphical Support in Rule-Based Forecasting: A Comparison, *International Journal of Forecasting*. 12, 495-512.