

# **TutGen: Una propuesta para facilitar la generación de tutoriales**

Zulema B. Rosanigo<sup>1</sup>; Alicia Paur<sup>2</sup>; Pedro Bramati<sup>3</sup>; Alfredo Ortega<sup>4</sup>; José P.Cerra<sup>5</sup>

Facultad de Ingeniería – Sede Trelew – U.N.P.S.J.B. Te-Fax (02965) 42 84 02

## **Resumen**

En este artículo se presentan los resultados del proyecto de investigación “Construcción de tutoriales basados en componentes reusables”, Nro. 383 de la Facultad de Ingeniería de la U.N.P.S.J.B. cuyo objetivo es buscar nuevas alternativas que mejoren la calidad educativa, facilitando al docente la construcción de nuevos tutoriales, del tipo enseñanza “paso a paso”, para que el alumno pueda ejercitarse tanto como lo necesite. Como producto de implementación del proyecto se desarrolló un software, denominado TutGen (Generador de Tutoriales), cuya extensión al dominio de la Geometría sirve de apoyatura y complemento en el proceso de enseñanza-aprendizaje de la cátedra “Sistemas de Representación”.

## **Palabras claves**

Software educativo – Tutorial – Componente reusable – Framework

## **Introducción**

Una de las inquietudes de todos los equipos docentes consiste en encontrar, investigar y probar métodos que sean más adecuados para transmitir los conocimientos de un modo eficiente, intentando adaptar los contenidos de las materias a las necesidades presentes y futuras de los alumnos. De esta manera se trata de disminuir los problemas emergentes de la dicotomía que presenta la realidad universitaria actual que por un lado, se realizan recortes en la cantidad de horas de la currícula y por el otro, se necesitarían más para suplir las deficiencias con las que llega el alumnado.

La posibilidad de individualizar el proceso de enseñanza-aprendizaje que nos ofrece la computación representa importantes ventajas, pues puede permitirnos optimizar las capacidades de cada individuo al no tener que imponerles a todos los estudiantes el mismo ritmo, lo que es, para los más hábiles, motivo de aburrimiento, y para los que tienen mayor dificultad, frustración y decepción en dicho proceso.

En nuestro proyecto de investigación, denominado "Construcción de tutoriales basados en componentes reusables" nos propusimos facilitar al docente la construcción de tutoriales del tipo enseñanza paso a paso para que el alumno pueda ejercitarse tanto como lo necesite. Para ello diseñamos una herramienta lo suficientemente flexible como para poder crear tutoriales, de manera sencilla y amigable, para diferentes dominios del conocimiento.

---

<sup>1</sup> Ingeniera Civil – Analista Programador Universitario – Magister en Ingeniería de Software - Investigador Cat. III - Profesor Asociado D.E. [brosanigo@infovia.com.ar](mailto:brosanigo@infovia.com.ar)

<sup>2</sup> Analista Programador Universitario - Investigador Cat. V – J.T.P. D.S.E. - [apaur2@hotmail.com](mailto:apaur2@hotmail.com)

<sup>3</sup> Ingeniero Civil – Investigador Cat. IV - Profesor Titular D.S.E.. [bramati@infovia.com.ar](mailto:bramati@infovia.com.ar)

<sup>4</sup> Analista Programador Universitario - Licenciado en Informática [cachito@ortega.net.ar](mailto:cachito@ortega.net.ar)

<sup>5</sup> Alumno de Analista Programador Universitario – [jopa@infovia.com.ar](mailto:jopa@infovia.com.ar)

Estudiamos y analizamos las características y comportamientos de tutoriales en diferentes áreas tomando en cuenta qué comportamientos son compartidos y cuáles son específicos. Diseñamos y desarrollamos un framework de generación de tutoriales y lo especializamos en el dominio de las construcciones geométricas. A partir de la utilización de tutoriales, se espera resolver una de las dificultades que se plantea en el aprendizaje tradicional de algunas materias demasiado técnicas, donde en forma usual el alumno se encuentra con la resolución impresa en un solo paso, con todos los procesos intermedios obviados. Al poder seguir el proceso “paso a paso”, se eliminará la dificultad antes expuesta.

Los resultados de esta investigación se fueron plasmando en un software, denominado TutGen (Generador de Tutoriales), cuya extensión al dominio de la Geometría sirve de apoyatura y complemento en el proceso de enseñanza-aprendizaje de la cátedra “Sistemas de Representación”.

## Descripción del proyecto

Consideramos que la aplicación que permita generar tutoriales debía ser interactiva, con una interfaz gráfica para visualizar cada paso del tutorial y permitir tratar los eventos generados por las acciones del usuario, contar con un editor de tutoriales bastante semejante a un editor gráfico, un panel de dibujo o pizarra donde se puedan reflejar los datos iniciales y las transformaciones necesarias para llegar al resultado del problema, una barra de herramientas con la cual indicar el tipo y valor de los datos iniciales (Punto, Recta, Arco,...), pasos intermedios (arco que corta a recta, arco con centro y radio, arco con centro y radio que corta a arco,...), selección, desplazamiento, etc. y que varíe según el dominio de aplicación del tutorial.

Entre los requerimientos funcionales que nos hemos propuestos podemos mencionar:

- o Contemplar dos modalidades de uso bien definidas: *creación* de un tutorial y *aprendizaje* de un tutorial.
- o El trabajo de creación de un tutorial mediante la herramienta, debe resultarle al docente “tan natural” como si lo desarrollara manualmente, y en general, mucho más sencillo. Esto implica que muchas tareas pueden y deben estar automatizadas.
- o El tutorial que acaba de crear debe poder convertirse en un nuevo componente del dominio y por lo tanto, podrá ser utilizado para la creación de otro, debiéndose adaptar en forma inteligente, al nuevo contexto. Esto implica que debe existir un mecanismo especial de persistencia para los tutoriales creados.
- o En modalidad de aprendizaje de un tutorial, puede ser practicado individualmente tantas veces como se quiera, en forma completa o parcial, permitiendo volver atrás cuando el alumno lo requiera.

Teniendo en cuenta esto, definimos un marco para el generador de tutoriales, el cual forma el esqueleto de la aplicación, provee la funcionalidad común, define las abstracciones fundamentales y sus interfaces, establece las interacciones entre los objetos, dejando en determinados lugares espacios en blanco o puntos de articulación (hot spot). Cada espacio se refiere a los aspectos de los tutoriales que pueden variar de una aplicación a otra, y es allí donde la arquitectura debe proveer flexibilidad.

Quedó así definida una arquitectura flexible para generación de tutoriales, basada en el patrón de diseño arquitectónico “Model-View-Controller”, que divide el problema en tres componentes: el modelo que contiene el corazón de la funcionalidad, la vista que despliega la información al usuario y el controlador que maneja la entrada del usuario. De esta manera se desacoplan los problemas y se logra mayor reusabilidad.

La vista y el controlador definen la interfaz gráfica, con la que tendrán que interactuar el docente, en el proceso de creación de un tutorial, y el alumno, en el proceso de aprendizaje, repaso o práctica. En su definición se tuvieron en cuenta las conclusiones arribadas al respecto en el proyecto anterior: “Generalización del modelo de Hipermedia para la construcción de redes conceptuales”.

La arquitectura propuesta se implementó sobre un prototipo desarrollado en Java sobre el cual se hicieron pruebas que sirvieron para validación y retroalimentación.

Básicamente consiste en las siguientes clases:

**TutorialEditor:** coordina los diferentes objetos que participan en un editor. Desacopla los participantes del editor y actúa como un coordinador de ellos.

**Pizarra:** panel donde se despliega el tutorial. Está atenta a las intervenciones del usuario, ya sea el alumno que cambia un dato o el docente en el proceso de creación del tutorial.

**TutorialView:** gerencia toda la visualización, muestra la Pizarra y escucha sus cambios. Recibe los eventos de la interfaz de usuario y los delega a la herramienta actual. La respuesta depende de la herramienta activa en ese momento.

**Tool:** barra de herramientas para edición y ejecución de tutoriales.

**Tutorial:** provee el comportamiento genérico de todo tutorial, puede ejecutarse paso a paso, retroceder o avanzar un paso, sabe representarse, gestionar sus datos, calcular el resultado.

Para crear tutoriales en un dominio en particular, se requiere definir los elementos que pueden ser datos y las acciones primitivas que pueden efectuar transformaciones de esos datos, generando nuevos elementos. Este comportamiento es contemplado por:

**ElementoDibujable:** Los elementos son objetos que poseen comportamiento propio y pueden ser vistos como una típica caja negra, con una entrada, salida y función de conversión, y que además pueden ser representados en la pizarra donde se ejecuta el tutorial.

**Gestor:** Reúne los datos necesarios para la creación del elemento, y lo relaciona con el resto del tutorial. Por cada elemento del dominio existirá un gestor, pudiendo existir más de uno. El gestor representa las transformaciones necesarias para convertir los eventos de usuario en objetos del dominio.

Puesto que un paso de un tutorial a crear puede ser tanto un componente primitivo (ya desarrollado) como un tutorial creado con anterioridad por el docente, nos enfrentamos a la situación en que no sólo el alumno debía aprender, sino que el sistema diseñado también debía “aprender” a utilizar los tutoriales creados, de la misma manera que sabía utilizar las primitivas.

Esto implica que cada tutorial debe:

- ✓ tener un mecanismo de persistencia.
- ✓ autoejecutar los pasos en función de las entradas del nuevo contexto en que se ejecuta y las salidas de los pasos anteriores.

Este es un aspecto fundamental para garantizar la reusabilidad de un tutorial.

Se analizaron diferentes aproximaciones de solución: objetos serializables, base de datos de órdenes y propiedades, pero se descartaron por no satisfacer todos los requerimientos. Finalmente definimos el diseño conceptual del modelo que cumple con todos los requerimientos y al que denominamos “Paso Inteligente”, ya que representa un paso del tutorial y encapsula el conocimiento y provee los mecanismos que le permiten adaptarse “inteligentemente” al ser utilizados en nuevos contextos.

Para asegurar que la arquitectura propuesta es flexible y reusable, se hicieron pruebas con componentes de dos dominios distintos, el de la geometría y el de la aritmética, y desarrollados por

diferentes personas, las cuales resultaron muy satisfactorias, ya que se pudo cambiar de dominio de aplicación sin tener que modificar nada en los componentes comunes del generador de tutoriales. Esto valida que las interfaces de comunicación entre componentes han sido bien establecidas y respetadas.

La herramienta desarrollada, a la que denominamos TutGen (Generador de Tutoriales) resulta ser sencilla de utilizar tanto para el docente (modo Creación) como para el alumno (modo Aprendizaje).

En modo creación, cada tutorial creado es guardado en disco, junto con el título, el enunciado del problema que resuelve y la explicación conceptual del mismo. Cada tutorial conoce cuáles son los datos que requiere como entrada y sabe realizar automáticamente todos los pasos intermedios para lograr el resultado. Al mismo tiempo que se va creando, se genera la explicación de cada paso de resolución.

En modo aprendizaje se logra la integración de todos los componentes del tutorial creado: por un lado la visualización paso a paso, gráfica y textual, y por otro, la explicación completa del problema.

A medida que se puso en marcha cada nivel del prototipo, se fue observando y evaluando el proceso de enseñanza-aprendizaje, cuyas conclusiones sirvieron de retroalimentación para el nivel siguiente.

## Conclusiones

- o El uso de tutoriales interactivos en la educación universitaria permite el logro de objetivos tales como:
  - ✓ Enseñar una nueva forma de estudio (teniendo en cuenta como ya mencionamos, que al alumno le resulta familiar el uso de una computadora) e influir en el proceso y tipo de pensamiento, favoreciendo la formación del alumno.
  - ✓ Un tutorial, como todo Software Educativo (SE), debe poner énfasis en lograr aprendizajes significativos, brindando posibilidades de vincular los nuevos conceptos con los que ya se tienen adquiridos, estableciendo relaciones no arbitrarias entre ellos. Debe entonces, aplicar estrategias centradas en el alumno, poniendo en práctica técnicas que permitan que potencie su percepción, motivación y habilidad.
  - ✓ Las situaciones de aprendizaje requieren interacción entre profesor y alumno y hay que verlas como un proceso de comunicación, en donde se crea y se elabora la información mediante mensajes, por eso es conveniente usar un lenguaje total: es decir integración del lenguaje matemático, simbólico, gráfico, icónico.
  - ✓ Encontrar una forma motivadora para acceder al conocimiento, que actúa con la dinámica que el alumno acostumbra encontrar en otros medios de comunicación (TV., computadoras, CD-ROMs, etc.).
- o Este tipo de herramienta, como en otras muchas multimediales, trata en todo momento de acompañar y promover los distintos tiempos de maduración individual del conocimiento del alumno. Con su utilización no se pretende sustituir al profesor sino más bien complementar y potenciar su labor.
- o El uso de una metodología de diseño y desarrollo, es fundamental para cualquier proceso de generación de software, en especial en SE es muy importante además considerar todas las metas pedagógicas perseguidas.
- o El diseño y desarrollo basado en componentes, implica su detección y posterior integración al prototipo para generar el sistema, esto permite que el SE se vaya creando a medida que se

obtienen, desarrollan y maduran sus componentes. Es recomendable utilizar tempranamente en el ciclo de vida, casos de uso para la generación del prototipo.

- o El especial cuidado en las etapas de diseño: de la Información, de la Presentación y de la Interacción aseguran mantener una calidad del producto final que nunca debe perderse de vista. La integración, evaluación y testeo temprano de estas etapas permiten detectar y corregir defectos antes que se vuelvan problemáticos, decidiendo a tiempo, la conveniencia o no del uso de determinados recursos.
- o Con la utilización de tutoriales multimediales, se cuenta con una herramienta que permite entender un problema y ver su resolución “paso a paso” pudiendo visualizar todos los procesos intermedios, variando los datos de entrada y apreciando como el tutorial adapta el método a los nuevos datos, repetir el proceso cuantas veces se necesite hasta interpretar la técnica requerida para la resolución del problema

## Bibliografía

- [1] Bartolomé Pina, A., *Preparando un nuevo modo de conocer*, Edutec'96. Revisit Electronic de Tecnología Educativa, nº 4, Diciembre 1996
- [2] Buschmann F., Meunier R., Rohnert H., Sommerland, P., Stal, M. *Pattern-Oriented Software Architecture: a system of patterns*. Ed. Wiley 1996
- [3] Cooper, James W. - *Java Design Patterns: A Tutorial*, 1998 – Addison Wesley
- [4] Gamma, Eric; Helm, Richard; Johnson, Ralph and Vlissides, John, *Design Patterns. Elements of Reusable Software*, Addison-Wesley, 1995
- [5] Hinostroza E., Hepp P., Straub P. - *Método Desarrollo Software – 1997 –* <http://www.enlaces.el/documentos/metododesarrollo/metodo.html>
- [6] K. Mullet & D. Sano. “*Designing Visual Interfaces: Communication Oriented Techniques*”, SunSoft Press (Prentice-Hall). 1995.
- [7] Kristof, Ray –Satran, Amy. *Diseño interactivo*. Ediciones Anaya Multimedia 1998.
- [8] Marqués, Pere – *Metodología para la elaboración de software educativo – 1995 –* <http://www.blues.uab.es/home/material/programes/t023151/uabdisof.htm>
- [9] Ortega Alfredo, Cerra José P. , Rosanigo Zulema B., Bramati Pedro, Paur Alicia - *Implementación de componentes multi-propósitos y un entorno de desarrollo - Actas IX CACIC - Facultad de Informática – UNLP- La Plata – 2003.*
- [10] Paur Alicia, Rosanigo Zulema B., Bramati Pedro, Ortega Alfredo, Cerra José P. - *Metodología de diseño y desarrollo del generador de tutoriales multimediales: TUTGEN - Actas IX CACIC Facultad de Informática – UNLP- La Plata - 2003.*
- [11] R. Pressman “*Ingeniería del Software, un Enfoque Práctico*”, 4º Ed., Mc. Graw Hill. 1997.
- [12] Rosanigo, Z. B., Paur, A.B., Bramati, P. *Metodología de desarrollo de software educativo*. Actas de VI Congreso Internacional de Ingeniería Informática ICIEY2K Fac. de Ingeniería, U.B.A. - Buenos Aires – 2000
- [13] Sergio Ochoa, David Fuller. *Una Metodología de Educación Basada en Componentes*. Pontificia Universidad Católica de Chile. Escuela de Ingeniería. DCC Santiago, Chile 2000.