

# Metodología de diseño y desarrollo del generador de tutoriales multimediales: TUTGEN – Tutorial Generator

Alicia Beatriz Paúr<sup>1</sup>; Zulema B. Rosanigo<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

Como producto de implementación del proyecto de investigación N° 383 denominado "**Construcción de tutoriales basados en componentes reusables**", se desarrolló **TUTGEN** (Tutorial Generator), el proyecto está circunscripto en el área de Informática Educativa y tiene por objetivo principal facilitar: la generación de material didáctico en general, y de tutoriales del tipo "enseñanza paso a paso" en particular, a partir de algunos componentes que soporten un conjunto de comportamientos estándares, independientemente de la funcionalidad específica para la que han sido diseñados. En este artículo describimos la metodología de diseño y desarrollo aplicada para su creación.

## Palabras claves:

Tutorial – componente reusable – framework – metodología de diseño y desarrollo

---

<sup>1</sup> Analista Programador Universitario - Investigador Cat. V – Profesora Adjunta D.S.E. - [apaur@topmail.com.ar](mailto:apaur@topmail.com.ar)

<sup>2</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>3</sup> Ingeniero Civil – Investigador Cat. IV - Profesor Titular D. S.E.. [bramati@infovia.com.ar](mailto:bramati@infovia.com.ar)

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

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

## INTRODUCCIÓN

Una realidad de nuestros tiempos es que los adolescentes no manifiestan gran predisposición a leer extensa bibliografía como parte del proceso de aprendizaje o investigación, pero sí presentan gran interés en incursionar en un tema, navegando por Internet, por una obra hipermedial o siguiendo los pasos de un tutorial, por ejemplo.

También es una realidad que esta sociedad tecnificada e informatizada, que propone y desarrolla la comunicación por medio de imágenes y de la interacción con ellas –multimedia–, formará a los profesionales futuros, los que necesitarán imperiosamente adquirir nuevos conocimientos, habilidades, destrezas y actitudes que las nuevas tecnologías les exigen dominar.

Por otro lado, todas las universidades están planteando la necesidad de acortar sus planes curriculares, para dar cabida como parte de una capacitación permanente, a los posgrados, maestrías y doctorados.

A todo ello debemos agregar que la enseñanza tradicional presenta algunas falencias, por ejemplo:

- Desigualdad entre el alumnado en la comprensión de los conceptos explicados.
- En grupos grandes de alumnos, no se puede captar la atención de todos en cualquier momento.
- Limitación de recursos (tiempo, espacio, estructuración de los contenidos, herramientas educativas) para seguir el ritmo que cada alumno necesita como parte de su aprendizaje.
- Falta de dinamismo, interacción y flexibilidad requerida para lograr el objetivo de transmitir conocimientos, habilidades, destrezas, etc. a un gran número de alumnos con captaciones distintas, realidades y capacidades disímiles.

Todas estas particularidades hacen que sea apremiante y perentorio el hecho de: investigar, evaluar y poner en práctica nuevas herramientas de enseñanza que permitan al alumnado adquirir los conocimientos y destrezas que de él se esperan, desde una óptica que le es familiar, atrayente, que domina por serle conocida.

Una herramienta aplicada a la enseñanza universitaria, con un invaluable aporte a esta nueva forma de adquirir conocimientos, es el *tutorial*: no necesita la presencia permanente del profesor, es interactivo, repetitivo, se adapta a los tiempos y madurez de conocimiento de cada alumno, admite distintos datos de entrada para simular diferentes situaciones donde aplicar el método a aprender.

Es por ello que nuestro **proyecto de investigación N° 383 “Construcción de tutoriales basados en componentes reusables”** tiene como objetivo lograr un framework que permita al docente –en cualquier dominio de la enseñanza–, la generación de tutoriales apoyados en el uso de librerías de componentes que (sin importar su funcionalidad particular) soporten un conjunto de comportamientos estándares, lo que los hace reusables en un amplio contexto.

Uno de los resultados emergentes del proyecto de investigación es: **TUTGEN** (Tutorial Generator), software aplicado en la materia: *Sistemas de Representación* del ciclo de ciencias básicas de la carrera de Ingeniería Civil.

Esta herramienta está diseñada para permitir al docente, construir tutoriales del tipo “enseñanza paso a paso”, a partir de algunos componentes primarios o básicos –desarrollados especialmente para el dominio de aplicación–. El tutorial así creado, a su vez, se convierte en un nuevo componente del dominio y por lo tanto puede ser usado para la creación de otros, debiéndose adaptar en forma inteligente al nuevo contexto.

TUTGEN permite *al profesor*: crear los tutoriales que apoyarán, guiarán y facilitarán la capacitación del alumno. De ser necesario, podrá usar otros tutoriales previamente creados como pasos intermedios del nuevo, utilizando todas las herramientas que estarán a su disposición (cuadros de texto para ingresar conceptos teóricos, recursos visuales, etc.).

La herramienta “aprende” a dominar los tutoriales creados y permite *al alumno*: interactuar con ellos, ingresando los datos iniciales, variándolos con simplemente “arrastrar” las figuras, reiterar las acciones a partir de cualquier paso, guardar su propio ejemplo con los datos de entrada elegidos, permitiendo así su entrenamiento, para la adquisición de la capacidad pretendida por el profesor –en este caso en particular, en el área de la geometría–.

En este artículo se expone la metodología de diseño y desarrollo propuesta por los autores para la construcción de TUTGEN, que combina reglas existentes en el campo de las Ciencias de la Educación, incorporando para ello, aspectos educativos en etapas tempranas del ciclo de vida del software, respetando siempre las consignas de la Ingeniería de Software que aseguran un producto de calidad. Previamente, enunciaremos algunos de los principales objetivos del uso de tutoriales en la educación universitaria.

## **OBJETIVOS DEL USO DE TUTORIALES**

### ***Desde el punto de vista del profesor (generador de tutoriales):***

- ✓ El manejo del nuevo lenguaje que proponen los tutoriales y la aplicación de los mismos: tiene por objetivo la aceptación y adaptación a una modalidad diferente de estudio (teniendo en cuenta como ya mencionamos, que al alumno le resulta familiar el uso de una computadora) y también 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 le permitan potenciar su percepción, motivación y habilidad. También debe brindar mecanismos de evaluación, tanto para el alumno como para el docente.
- ✓ 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.

### ***Desde el punto de vista del alumno (usuario de los tutoriales):***

- ✓ Con la utilización de tutoriales, uno de los objetivos del alumno es: encontrar un medio para entender un problema y ver su resolución “paso a paso”, pudiendo visualizar todos los procesos intermedios (que en la bibliografía tradicional son obviados, o en el mejor de los casos presentados en uno o dos ejemplos fijos), puede variar los datos de entrada y ver cómo el tutorial adapta el método a los nuevos datos, repetir el proceso cuantas veces necesite hasta interpretar la técnica requerida para la resolución del ejercicio.
- ✓ Su intención es encontrar una forma motivadora para acceder al conocimiento, que actúa con la dinámica que el alumno acostumbra encontrar en otros medios de comunicación (T.V., computadoras, CD-ROMs, etc.).

En el intento de diseñar material pedagógico, que cumpla con todos estos objetivos lo mejor posible utilizando las nuevas tecnologías, es que han surgido los **tutoriales multimediales**, que intentan dar a cada alumno una enseñanza personalizada y amena, captar su atención en todo momento y sobre todo adaptarse al ritmo de cada persona haciendo uso para ello de las más variadas herramientas que actualmente nos ofrece la informática, tales como: la multimedia o la inteligencia artificial.

Debemos destacar que el uso de tutoriales, no pretende en ningún momento sustituir al profesor sino que solo intenta ser una herramienta complementaria al mismo, para llegar allí donde él no puede llegar. En este tipo de herramienta, como en otras muchas multimediales, el que marca el ritmo de aprendizaje es el alumno, tratando en todo momento de acompañar y promover los distintos tiempos de maduración individual del conocimiento.

## **METODOLOGÍA DE DISEÑO PROPUESTO**

La idea “semilla” de este proyecto de investigación, siguiendo en la línea de SE, surgió de un grupo interdisciplinario de docentes del ciclo superior de Ingeniería y del Departamento de Informática, motivados en aplicar herramientas multimediales (tutoriales, específicamente) en la educación universitaria con el objeto de mejorar la calidad de enseñanza/aprendizaje, eligiendo como dominio de aplicación la materia: *Sistemas de Representación*, correspondiente al ciclo de ciencias básicas de la carrera de Ingeniería Civil.

Como propuesta para lograr la calidad necesaria y pretendida para nuestro *generador de tutoriales*, nos basamos en el modelo de *ciclo de vida en espiral* de la Ingeniería de Software, usando a UML como lenguaje de modelamiento y a RUP (Rational Unified Process) como proceso de desarrollo, cumpliendo con las etapas que a continuación se detallan:

### **Definición del proyecto:**

- ✓ Identificación de necesidades del producto educativo.
- ✓ Delimitación del alcance del producto.
- ✓ Definición de objetivos y contenidos docentes del producto (qué y por qué).
- ✓ Caracterización de los usuarios del producto, considerando a los alumnos y profesores.
- ✓ Definición de la funcionalidad del producto e identificación de los casos de uso.
- ✓ Definición de ambiente de uso, incluyendo lugar y circunstancia (dónde y cuándo).
- ✓ Definición de principios pedagógicos y estrategias didácticas aplicables.
- ✓ Definición de la arquitectura basada en los diferentes casos de uso.
- ✓ Determinación de recursos de desarrollo, incluyendo personas, herramientas, técnicas, equipos y tiempo (cómo y con qué) necesarias para lograr el producto.

### **Planificación:**

- ✓ Estimación de tiempo y costo del proyecto.
- ✓ Planificación del seguimiento, gestión y control del proyecto.
- ✓ Planificación de la garantía de calidad del software.
- ✓ Planificación de la garantía de calidad educativa.

### **Análisis de alternativas.**

### **Identificación, análisis y resolución de riesgos.**

### **Ingeniería: desarrollo del producto de “siguiente nivel” (prototipos):**

- ✓ Diseño:
  - De la información.
  - De la interacción.
  - De la presentación.

- ✓ Detección de componentes con interfaces bien definidas, que serán luego ensamblados para generar el sistema en la siguiente iteración. En cada etapa se va cumpliendo el proceso de “descubrimiento” y desarrollo de componentes, que evoluciona hacia el sistema final.
- ✓ Construcción del prototipo.

### Evaluación del prototipo:

- ✓ Desde el punto de vista de la calidad de software.
- ✓ Desde el punto de vista pedagógico y de la calidad educativa.
- ✓ Evaluación contextualizada (en el ambiente de uso, con docentes y alumnos).

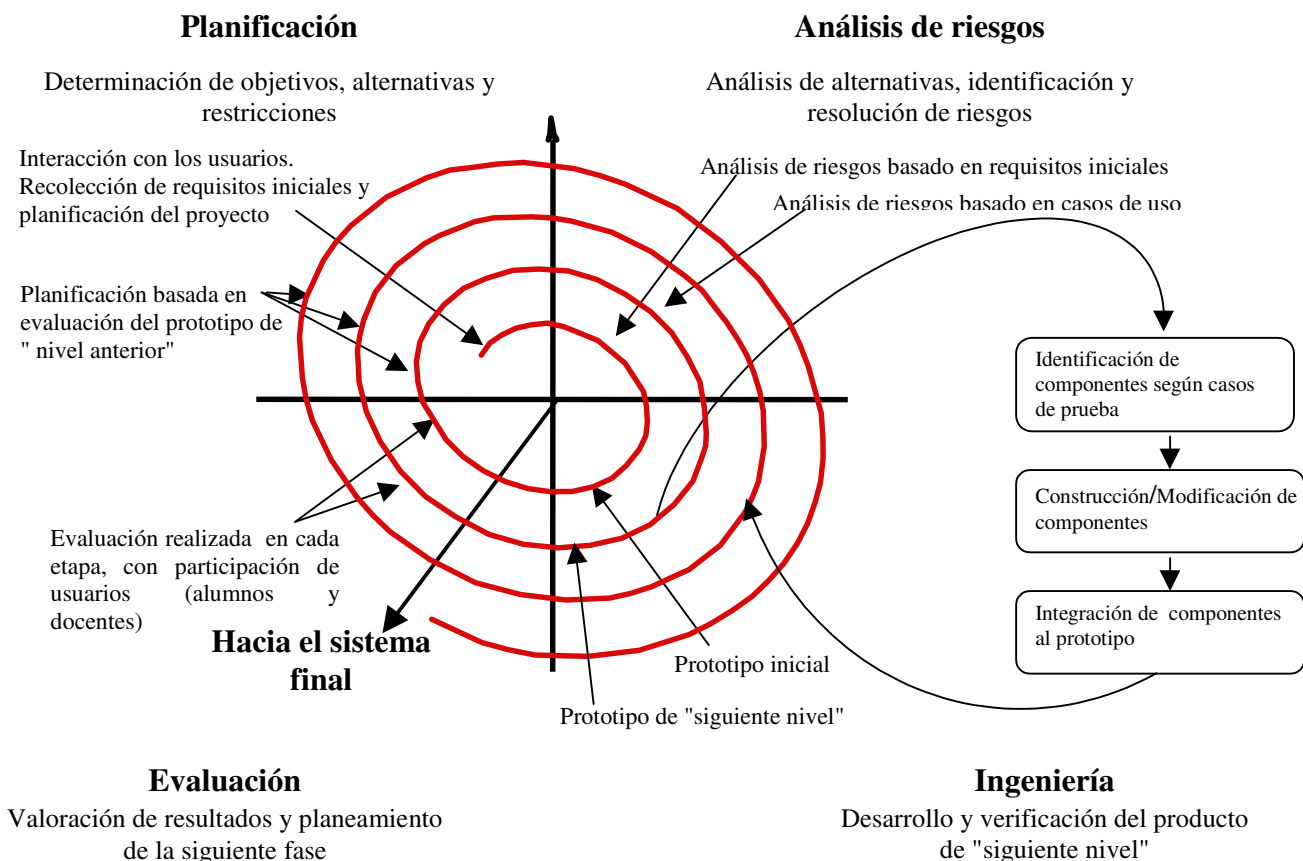
### Identificación de los cambios y ajustes a realizar.

La evolución que sigue el proyecto es básicamente en torno al desarrollo del prototipo originado en los casos de uso y basado en componentes (versión beta del producto final), que incrementa en contenido, gráfica, navegación y funcionalidad sin hacer cambios sustanciales en las estructuras y capacidades de la aplicación.

En cada etapa del ciclo de vida se identifican los componentes necesarios, se los busca en bibliotecas y se los adaptan; solo si no se encuentra un componente que satisfaga: se lo construye, se publican e integran al prototipo de la siguiente etapa. El resultado de la iteración prototipo-evaluación en terreno, y la consiguiente retroalimentación con las modificaciones sugeridas, debe finalizar en la obtención de la aplicación deseada.

A continuación mostramos una gráfica del modelo de ciclo de vida propuesto:

## MODELO DE CICLO DE VIDA EN ESPIRAL BASADO EN COMPONENTES



Los principales beneficios del modelo presentado son:

- ✓ Incorporación y contemplación de aspectos pedagógicos, educativos y comunicacionales dentro de cada etapa del ciclo de vida.
- ✓ Incorporación de casos de uso, los que permiten reemplazar la especificación funcional tradicional, transformándose en guía para las actividades que se realizan durante el proceso de desarrollo, incluyendo diseño, implementación y pruebas del sistema.
- ✓ Estructuración de las actividades llevadas a cabo en el diseño y desarrollo del software, con la incorporación de componentes que permiten y privilegian el reuso de código.
- ✓ Importante disminución en el tiempo de desarrollo.
- ✓ Pronta incorporación de un prototipo que permite evaluaciones tempranas del producto y va evolucionando constantemente hasta convertirse en el producto deseado.
- ✓ Considerando de suma importancia la etapa de diseño y dentro de ella, sus tres subdivisiones: Diseño de la información, Diseño de la interacción y Diseño de la presentación, a continuación se las describe con mayor detalle:

### ***Diseño de la Información***

Implica clarificar los objetivos y organizar el conocimiento de forma tal que se pueda cumplir con el propósito del SE. Se apunta a definir *Qué es el producto*.

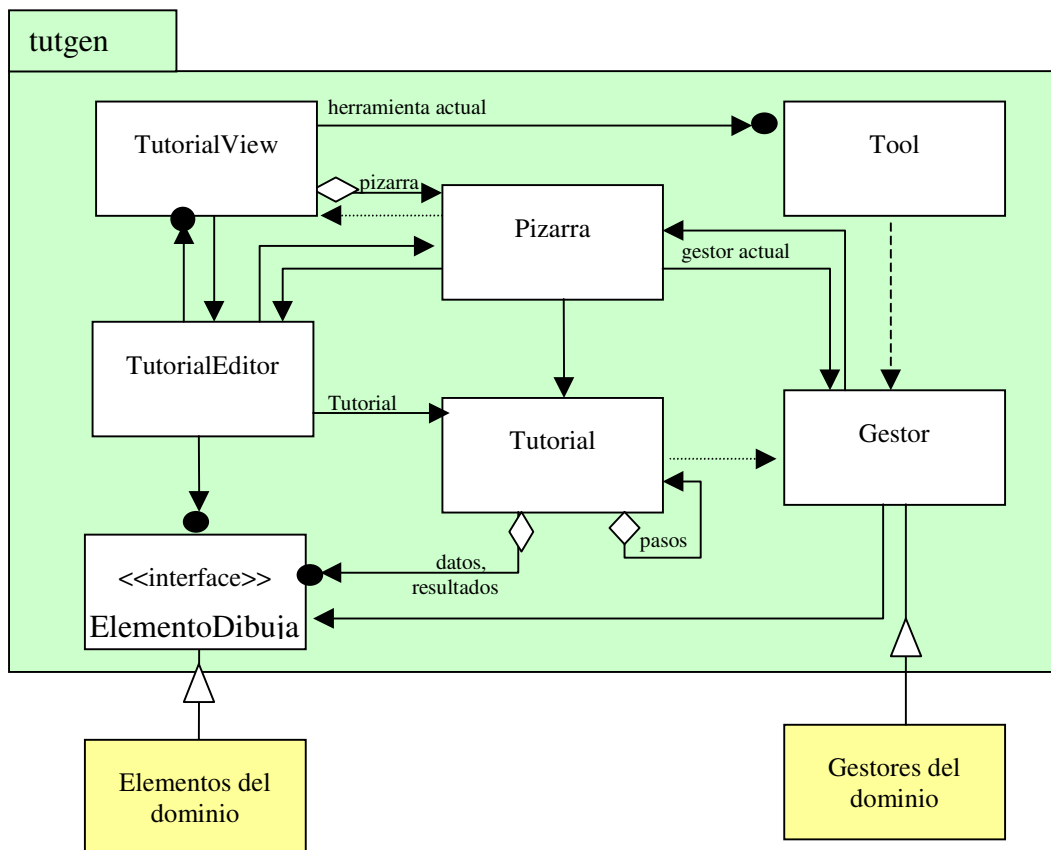
Esta etapa del diseño constituye el soporte principal de la arquitectura de todo proyecto educativo multimedial.

Siendo una etapa decisiva para el éxito final del proyecto, es de suma importancia elegir la metodología más adecuada para su formalización, permitiendo a los profesores estructurar su conocimiento lo más naturalmente posible.

En nuestro caso en particular, en primer lugar, los expertos en el tema requieren darle forma, secuencializar, estructurar y representar el conocimiento de los distintos casos de uso que luego van a plasmar en el tutorial, necesitan clarificar e identificar los conceptos básicos a partir de los cuales surgirán todos los tutoriales del dominio en desarrollo, de forma tal que el modelo mental que se forme el alumno se parezca a la realidad de los sistemas en estudio y al modelo que el profesor desea transferir.

Como resultado de esta etapa, se desarrollaron componentes específicos del dominio de la Geometría que representan entes geométricos: Punto, Recta, Arco y componentes que modelan transformaciones o cálculos: intersección entre figuras, trazados, etc.

El siguiente diagrama muestra los principales componentes del framework y sus relaciones:



### Diseño de la Interacción

Involucra tareas tales como definición de lo que sucede en cada etapa del tutorial (entrada o pasos), diseño de controles, de interacción (retroceso, arrastre de objetos, edición del texto, cambio de visualización de los objetos). Apunta al *Cómo debe funcionar*.

Es importante abordar el diseño de la interacción desde las dos ópticas que precisa el uso de TUTGEN, que son:

El *profesor* desea *crear* sus tutoriales: para él es fundamental contar con herramientas interactivas que le faciliten su tarea, permitan probar el tutorial, modificarlo, rehacerlo si es necesario, todo ello de la forma más natural posible, aún más sencillamente que si lo hiciera manualmente, esto implica que muchas tareas pueden y deben estar automatizadas, en última instancia el profesor deseará grabarlo para así generar un nuevo componente en el dominio específico de la Geometría.

Como ya dijimos, el tutorial así creado se convierte 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.

El *alumno* desea *usar* los tutoriales para practicar con ellos y aprender la técnica y destreza que se requiere en el área del conocimiento, para ello necesita poder levantar un tutorial e interactuar con él comenzando con el ingreso de los correspondientes datos de entrada, avanzando paso a paso hasta la obtención correcta del resultado deseado.

## ***Diseño de la Presentación***

Se define el estilo y el diseño de los elementos del contenido del SE y se produce un prototipo. Responde a *Qué apariencia debe tener*.

Debemos pensar que en ambas modalidades: de generación o de uso de un tutorial, debemos contemplar que cuente con una amplia gama de herramientas que faciliten ambos usos.

El generador de tutoriales será usado sin embargo con el objetivo primordial de practicar en él técnicas y destrezas que se desean adquirir, es primordial entonces dotarlo de apariencia didáctica, perceptiva, amena, motivadora, etc. para lograr el objetivo.

### **El diseño de la presentación consta de:**

- ✓ Definición formal de cada pantalla
- ✓ Objetivo
- ✓ Diagrama de la pantalla, indicando cuáles objetos tiene y dónde están ubicados (pizarra, texto explicativo, descripción del paso, etc.).
- ✓ Listado de las características tanto de la pantalla como de cada objeto (colores, tamaño de fuentes, resolución de imágenes, etc.)

Recomendaciones a tener en cuenta en el diseño de la presentación:

- ✓ Utilizar imágenes que faciliten la comprensión del mensaje del SE, iconos claros y sin ambigüedades que revelan su propósito a simple vista.
- ✓ Cuando una imagen, aclaración o mensaje no resulta indispensable en la representación del tutorial pero sí útil para algún lector, su visualización y manejo debería quedar condicionada a una acción específica del usuario, como el click sobre un botón o la activación de un ítem de menú, en lugar de ser automático.
- ✓ Mantener una estandarización de los botones en cuanto a su posición, agrupándolos por función facilitando así su reconocimiento y uso.

En esta etapa se prioriza el uso de iconos (con gráficas claras), pizarra o área de exposición de los pasos del tutorial, botones que permitan tanto a *los profesores*: crear o modificar tutoriales, personalizarlos con comentarios, colores, transparencias, etc., como a *los alumnos*: realizar sus tareas, guardar resultados, reiterar pasos no comprendidos, experimentar con nuevos datos de entradas, leer y comprender la sustentación teórica de cada paso, etc.

En concreto: se busca que una pantalla sea clara, amigable, flexible y agradable de usar, *consistente*: cuidando que los mensajes y la distribución en pantalla, el juego de colores, etc. sigan un mismo patrón; *equilibrada*: respecto al texto y los demás medios elegidos para la representación de los conceptos; también es necesario que sea altamente *interactiva*, manteniendo un equilibrio entre los mecanismos de comunicación entre usuario–aplicación y el objeto de estudio.

Es prioritario asegurar el interés del contenido y mantener la atención del usuario, si se siente atraído por la pantalla, es muy probable que sea más productivo, requiera de menos supervisión, adquiera la habilidad pretendida en menos tiempo y cometa menos errores, para ello es conveniente tener en cuenta que las principales características para un buen diseño visual son: claridad y codificación visual (*una imagen vale más que mil palabras*).

En nuestro caso, se utilizó la metáfora de la pizarra:



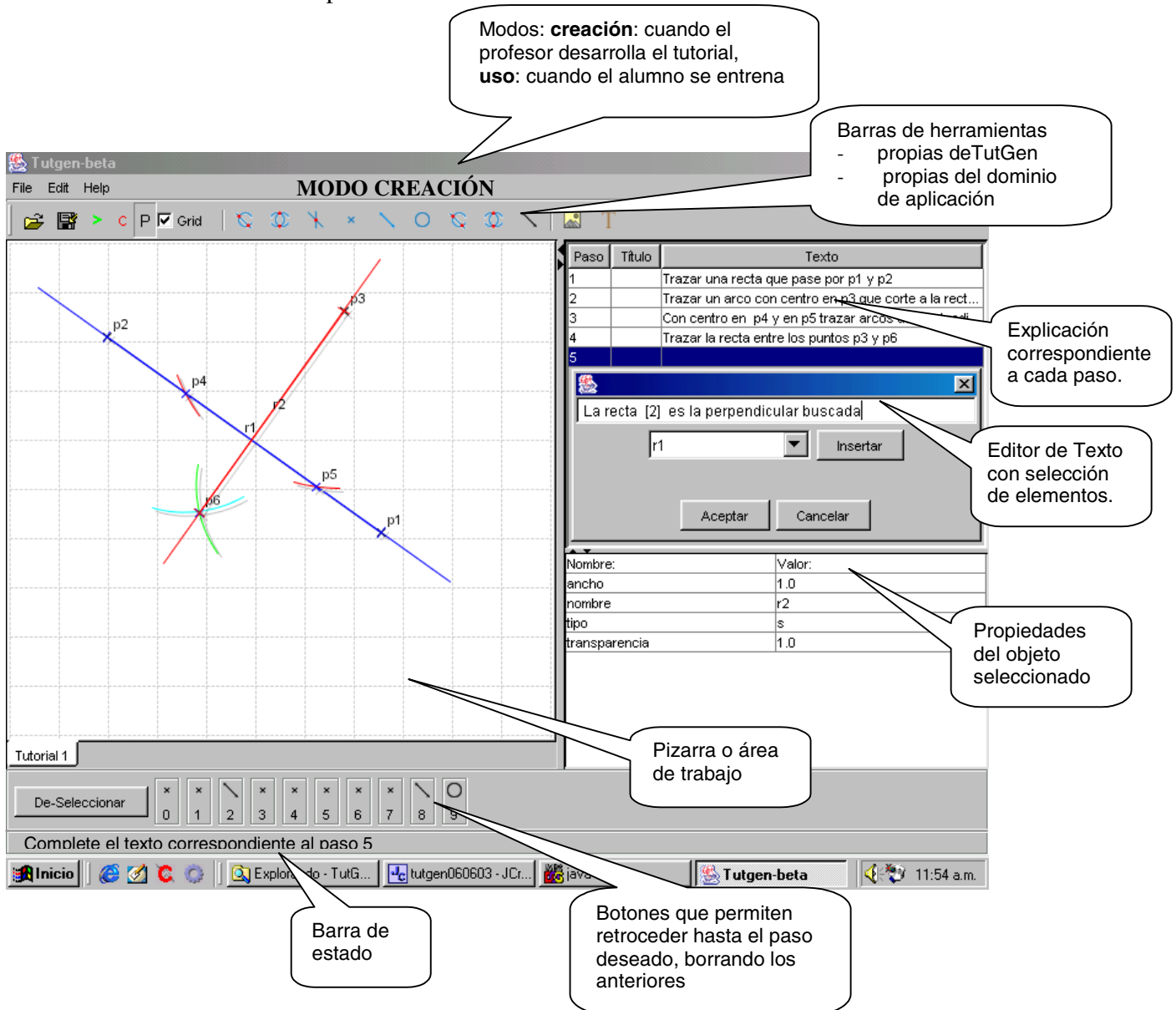
Para el modo *creación*, se cuenta con un área amplia donde el docente va construyendo el tutorial, con un sector en el que puede detallar información sobre los pasos que deben seguirse para la resolución del problema que se está enseñando, y otro sector donde se muestran las propiedades de los elementos. La amplitud de estos sectores es variable, y si no los necesita puede anularlos. Si lo desea, puede solicitarlos presionando el botón derecho del mouse en el área de la pizarra, y eligiendo la opción adecuada en un menú pop up emergente.

Para elegir lo que desea hacer tiene menús y barra de herramientas. El sistema le comunica los mensajes en la barra inferior de estado o mediante cuadro de diálogos.

Para el modo *uso*, se prioriza la comunicación de los conceptos que se desean resaltar. Se mantiene un área amplia donde se va ejecutando el tutorial, y un sector amplio destinado a las explicaciones.

En ambos modos, la interacción con el usuario es mediante acciones de teclado y de mouse, con los que el usuario puede cambiar propiedades, elegir opciones, avanzar o retroceder, recabar más información del problema, etc.

A continuación mostramos la pantalla de TutGen en modo creación:



## **DESCRIPCION DEL PRODUCTO: TUTGEN Tutorial Generator**

### ***Objetivos***

El desarrollo de tutoriales a partir de componentes reusables tiene varios propósitos:

- ✓ Brindar una herramienta muy apreciada en la educación para impartir conocimiento y adquirir destrezas en áreas de la enseñanza ya descriptas.
- ✓ Brindar a los docentes una forma relativamente sencilla de crear tutoriales que plasmen sus desarrollos de solución de problemas “paso a paso”.
- ✓ Brindar al alumno la posibilidad de interactuar con tutoriales que permitan: el ingreso de datos iniciales, variarlos cuantas veces sea necesario, reiterar las acciones a partir de cualquier paso, permitiendo así, su entrenamiento para la adquisición de la habilidad pretendida por el profesor.

### ***Recursos necesarios***

Los requisitos que a continuación se describen son los mínimos, es evidente que la ejecución resultará más ágil cuanto más potente sea el equipo utilizado.

Se requiere una PC con procesador Pentium II o superior con al menos 64Mb RAM, Video VGA, disco rígido con 50 Mb de espacio libre.

Para su ejecución se requiere un sistema operativo Windows 95 o superior.

### ***Herramientas de desarrollo***

Optamos por usar Java por ser una herramienta que provee un entorno de desarrollo orientado a objetos con lo que se puede establecer y personalizar el comportamiento de los objetos, y así lograr la flexibilidad y practicidad de uso requerido por nuestro proyecto, teniendo en cuenta los aspectos pedagógicos y didácticos.

De este lenguaje, estamos usando la versión 1.4 porque en ella está optimizado el uso de gráficos y tiene soporte para XML, con una mejora además de las librerías.

## **CONCLUSIONES**

El uso de una metodología de diseño y desarrollo, es fundamental para cualquier proceso de generación de Software, en especial en Software Educativo es muy importante además considerar todas las metas pedagógicas perseguidas.

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.

El agrupamiento en librerías y posterior organización en un framework de los componentes, permite el diseño de partes reusables las que quedan descriptas por un grupo de clases abstractas, y por la interacción y colaboración de sus instancias.

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, siendo los controles de calidad y gestión del sistema de suma importancia para lograrlo.

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 (que en la bibliografía tradicional son obviados, o en el mejor de los casos presentados en uno o dos ejemplos fijos), 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

- [Booch+98] Booch G., Jacobson I., Rumbaugh J. *The Unified Process Software Development* Addison-Wesley Publications, 1998
- [Coltell +93] Coltell Ò.; Ortells S.; Corella D.; Guillen M. “*Tutorial Inteligente Para El Aprendizaje De Epidemiología Genética Y Molecular*”–Grupo de Investigación en Integración y Reingeniería de Sistemas (IRIS). Departamento de Informática. Universitat Jaume I Basado en: Shank RC. *AI. Multimedia and Education*. In Proc. of IJCAI 93. Chambéry, France, 1993
- [Cooper98] Cooper J. W. - *Java Design Patterns: A Tutorial*, 1998 – Addison Wesley
- [Crane +97] Crane A., Clyde S. “*Extending Patterns for GUI Design*”–Research Paper. Utah State University. 1997.
- [Gamma+95] Gamma E.; Helm R.; Johnson R., Vlissides J., *Design Patterns. Elements of Reusable Software*, Addison-Wesley, 1995
- [Hinostroza+97] Hinostroza E., Hepp P., Straub P. “*Método Desarrollo Software*” – 1997 – <http://www.enlaces.el/documentos/metododesarrollo/metodo.html>
- [Kristof+98] Kristof R. –Satran A. “*Diseño interactivo*”–Ediciones Anaya Multimedia 1998.
- [Lage+99] Lage F., Cataldi Z. “*Ingeniería de software educativo*”–Actas de V Congreso Internacional de Ingeniería Informática ICIE'99 Fac. de Ingeniería, U.B.A. – Buenos Aires – 1999
- [León99] León Ortiz I. “*Sobre la elaboración de Redes Conceptuales en un proyecto multimedial*”–<http://www.umb.edu.co/curvir/postgrado/informatica/lectura/red.html>
- [Marqués95] Marqués P. “*Metodología para la elaboración de software educativo*” – 1995 – <http://www.blues.uab.es/home/material/programes/t023151/uabdisof.htm>
- [Marqués99] Marqués P. “*Programas Didácticos: Diseño y Evaluación*” –1999–  
<http://www.xtec.es/~pmarques/edusoft.htm>
- [Mullet +95] Mullet K., Sano D. “*Designing Visual Interfaces: Communication Oriented Techniques*”–SunSoft Press (Prentice-Hall). 1995.
- [Ochoa +00] Ochoa S., Fuller D. “*Una Metodología de Educación Basada en Componentes*”– Pontificia Universidad Católica de Chile. Escuela de Ingeniería. DCC Santiago, Chile 2000.
- [Pressman 97] Pressman R. “*Ingeniería del Software, un Enfoque Práctico*”– 4º Ed., Mc. Graw Hill. 1997.