

## Diseño de un entrenador

**Zulema Beatriz Rosanigo**

[brosanigo@infovia.com.ar](mailto:brosanigo@infovia.com.ar)

**Alicia Beatriz Paur**

[apaur@ar.inter.net](mailto:apaur@ar.inter.net)

**Pedro Bramati**

[pedrobramati@speedy.com.ar](mailto:pedrobramati@speedy.com.ar)

**Hernán Bramati**

[hernan.bramati@gmail.com](mailto:hernan.bramati@gmail.com)

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

### Resumen

En este artículo se expone el diseño de un software educativo, ETG, entrenador y generador de tutoriales, producto resultante del proyecto de investigación PI N° 530 "*Generador de entrenadores basados en componentes reusables*" de la Universidad Nacional de la Patagonia San Juan Bosco, Argentina.

ETG facilita la tarea de crear tutoriales como recurso didáctico, que colaboren y potencien los procesos de aprendizaje en la educación superior y sirvan de apoyo al proceso de enseñanza-aprendizaje. Al mismo tiempo, el alumno al interactuar con esos tutoriales, puede analizar el problema y la solución propuesta, tantas veces como lo desee, observando el comportamiento, incrementando así su capacidad de asimilación y fijación de conceptos. También permite que el alumno realice prácticas sobre el tema, y en caso de cometer errores, puede intervenir guiándolo hacia la solución correcta, sin necesidad de la permanente presencia del docente, a la vez que le brinda un mecanismo de auto evaluación.

**Palabras claves:** Tutorial interactivo – entrenamiento - framework – software educativo

## 1 Introducción

La innovación en los ambientes de enseñanza-aprendizaje con la utilización de nuevas tecnologías motoriza la investigación, análisis y evaluación de nuevas herramientas que permitan mejorar la formación de los futuros profesionales, y provoca la actualización de los profesores de las diferentes áreas curriculares con la finalidad de incorporar estos elementos a sus actividades docentes.

Por esta razón, nuestro grupo de investigación tiene como principal línea de investigación, la aplicación de las nuevas tecnologías en la educación. En esta dirección se han desarrollado varios proyectos en los que se evaluaron diferentes medios y se diseñaron contenidos y recursos didácticos a aplicar en la enseñanza superior, como apoyo a la tarea docente tanto en modalidad presencial como no presencial.

Siguiendo con la línea de investigación del proyecto previo: “Construcción de tutoriales basados en componentes reusables”, cuya finalidad fue facilitar la construcción de herramientas educativas, en este proyecto se diseñó un framework integrado a TutGen (generador de tutoriales) para abarcar el entrenamiento controlado y asistido que el alumno necesita realizar para ejercitarse y adquirir la destreza pretendida.

Se trata de un entrenador genérico e inteligente, denominado ETG, que interpreta de cada tutorial los diferentes caminos que conducen a la solución, los reconoce como válidos sin necesidad de que el docente deba expresarlos uno por uno, y asiste al alumno en sus errores guiándolo hacia la solución correcta, sin necesidad de la permanente presencia del docente, a la vez que le brinda al alumno un mecanismo de autoevaluación.

El tipo de problemas que se abarca es de naturaleza procedimental y de dominios que implican la modelización de procesos físicos y conocimiento procedural en el cual, los aspectos de comportamiento, resultan cruciales. Se puede utilizar en aquellas áreas del conocimiento donde se presentan problemas cuya solución requiere seguir determinado procedimiento o proceso paso a paso o también para adquirir destrezas.

La dificultad de construir un entrenador reside en que el sistema debe tener conocimiento del dominio, así como un mecanismo de interacción con el estudiante que le permita transmitirlo y verificar que esta comunicación fue eficaz, y de no ser así, efectuar alguna acción correctiva. Porque se trata de un software educativo, además de los requisitos funcionales y operativos, es sumamente importante considerar los aspectos pedagógicos en etapas tempranas del ciclo de vida, ya que tendrán incidencia a lo largo de toda la vida del mismo.

En este artículo se describe el diseño del generador de tutoriales y entrenadores.

## 2 Fundamentos

### 2.1 Teorías de aprendizaje

El propósito de las teorías educativas es el de comprender e identificar los procesos de adquisición de conocimiento y a partir de ellos, tratar de describir métodos para que el aprendizaje sea más efectivo.

Nuestro deber como docentes es aprovechar de cada teoría lo que mejor convenga a la situación, al sujeto, y al tipo de aprendizaje a lograr, enseñando al alumno a pensar y actuar en forma autónoma a fin de que adquiera más conocimientos con creciente capacidad y destreza

Según Salcedo Lagos [23] las aproximaciones al fenómeno del aprendizaje oscilan entre dos extremos: del conductismo al cognitivismo y viceversa.

En un extremo, la teoría del conductismo [25], lo único que ve del sujeto que aprende son las condiciones externas que favorecen su aprendizaje. Habla de un modelo de "caja negra" donde lo fundamental es la programación en pequeños pasos que llevan al logro del objetivo esperado, lo que se manifiesta por la respuesta del sujeto y su reforzamiento.

En el otro, para la teoría del cognitivismo [14] lo que cuenta es el sujeto con todo su campo vital, su estructura cognitiva y las expectativas que tiene. Se habla de un modelo de "caja traslúcida" donde lo que cuenta es el sujeto dentro de su entorno psicológico y social.

En medio de los enfoques anteriores, es posible encontrar una combinación de ambos, donde no se ignora por completo al sujeto, pero tampoco se lo hace eje principal.

Las teorías más representativas del proceso de enseñanza-aprendizaje son:

- Conductismo (Skinner 1958)
- Aprendizaje significativo (Ausubel 1983)
- Cognoscitivismo (Gardner 1987)
- Constructivismo (Piaget 1989)

La distinción básica entre estas grandes corrientes radica en la forma en que se concibe el conocimiento. Para el conductismo, el conocimiento consiste fundamentalmente en una respuesta pasiva y automática a factores o estímulos que se encuentran en el ambiente. Ausubel plantea que el aprendizaje depende de la estructura cognitiva previa del alumno que se relaciona con la nueva información, entendiéndose por "estructura cognitiva", al conjunto de conceptos e ideas que un individuo posee en un determinado campo del conocimiento, así como su organización. El cognoscitivismo, por su parte, considera el conocimiento básicamente como representaciones simbólicas en la mente de los individuos. Finalmente, el constructivismo, como el término lo sugiere, concibe al conocimiento como algo que se construye, algo que cada individuo elabora a través de un proceso de aprendizaje.

## 2.2 Estilos de aprendizaje

Los estilos de aprendizaje son una combinación de características cognoscitivas, afectivas y una conducta psicológica que sirven como indicadores relativamente estables sobre cómo los aprendices perciben, interactúan y responden a su ambiente [1]. Nuestra herencia, experiencias de vida y las demandas del ambiente determinan en parte cómo percibimos y procesamos la información. Los estilos de aprendizaje se ajustan dependiendo de la tarea a realizar y al estilo de enseñanza que se utiliza.

Se ha descubierto que las personas se concentran más en una determinada etapa del proceso de aprendizaje, y en función de la etapa dominante se puede hablar de cuatro estilos de aprendizajes:

- Vivir la experiencia → **ESTILO ACTIVO**: Son de mente abierta, nada escépticos y acometen con entusiasmo nuevas tareas. Crecen con los desafíos y se aburren con los plazos largos. Les gusta trabajar en grupo, se involucran en los asuntos de los demás y centran a su alrededor todas las actividades. Aprenden mejor cuando la actividad les presenta un desafío, con actividades cortas de resultado inmediato. Les cuesta más trabajo aprender cuando tienen que adoptar un papel pasivo, cuando tienen que asimilar, analizar e interpretar datos o cuando tienen que trabajar solos. La pregunta que quieren responder con el aprendizaje es **¿Cómo?**
- Reflexión → **ESTILO REFLEXIVO**: Son prudentes y consideran todas las alternativas antes de dar un paso. Recogen datos y los analizan antes de llegar a una conclusión. Aprenden mejor cuando pueden adoptar la postura de observador u ofrecer observaciones y analizar la situación

y cuando pueden pensar antes de actuar. Les cuesta más aprender cuando son el centro de la atención y cuando se les apresura de una actividad a otra o tienen que actuar sin poder planificar previamente. La pregunta que quieren responder con el aprendizaje es **¿Por qué?**

- Generalización, elaboración de hipótesis → **ESTILO TEÓRICO**: Adaptan e integran las observaciones (hechos) dentro de teorías coherentes. Enfocan los problemas de forma vertical escalonada, por etapas lógicas. Son perfeccionistas y les gusta analizar y sintetizar. Aprenden mejor a partir de modelos, teorías, sistemas, con ideas y conceptos que presenten un desafío y cuando tienen oportunidad de preguntar e indagar. Les cuesta más aprender con actividades que impliquen ambigüedad e incertidumbre. La pregunta que quieren responder con el aprendizaje es **¿Qué?**
- Aplicación → **ESTILO PRAGMÁTICO**: Su punto fuerte es la aplicación práctica de las ideas. Descubren el aspecto positivo de las nuevas ideas y aprovechan la primera oportunidad para experimentarlas. Aprenden mejor con actividades que relacionen la teoría y la práctica, viendo a los demás hacer algo y cuando tienen la posibilidad de poner en práctica inmediatamente lo que han aprendido. Les cuesta más aprender cuando lo que aprenden no se relaciona con sus necesidades inmediatas o no tienen una finalidad aparente o no está relacionado con la 'realidad'. La pregunta que quieren responder con el aprendizaje es **¿Qué pasaría si...?**

Según Kolb [17], ningún modo de aprender es mejor que otro y la clave para un aprendizaje efectivo es ser competente en cada modo cuando se requiera.

Una buena experiencia de aprendizaje reta las capacidades del aprendiz, por lo tanto, se debe establecer un balance al propiciar maneras alternas para aprender y así lograr que los estudiantes ganen confianza, muestren interés en aprender, desarrollen destrezas de razonamiento, análisis, solución de problemas y exploren otros estilos de aprendizajes.

El conocer sobre los estilos de aprendizaje nos permite facilitar el proceso de enseñanza - aprendizaje y nos ayuda a reflexionar sobre nuestras funciones y responsabilidades como educadores ante las necesidades e intereses de los estudiantes. Para mejorar el aprovechamiento académico, aumentar el interés, participación y nivel de satisfacción en el alumno, los estilos de enseñanza del profesor y los estilos de aprendizaje de los estudiantes deben corresponderse y complementarse. Por lo tanto, debemos ser conscientes sobre las diferencias entre los estilos de aprendizaje y las estrategias de enseñanza que tenemos disponibles. El balance entre ambos aspectos estimula la colaboración y la participación del aprendiz en el proceso de enseñanza - aprendizaje, haciendo que sea más significativo y efectivo.

Teniendo en cuenta que un aprendizaje óptimo, según Kolb, requiere de las cuatro fases, debemos presentar el material a aprender de forma tal que garanticemos actividades que cubran las distintas fases de la rueda de Kolb.

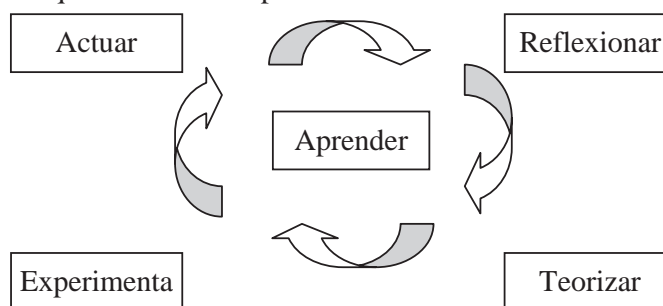


Figura 1 Fases del aprendizaje

De esta manera, facilitaremos el aprendizaje de todos los alumnos ya que habrá actividades en su estilo preferido y, además, les ayudaremos a potenciar las fases con los que se encuentran menos cómodos.

## 2.3 Enfoques para el diseño de material educativo

Existen diversos enfoques para el diseño de material educativo hipermedia [12, 21]:

- Basado en el diseño de los contenidos educativos: se articulan en cursos, lecciones, ejercicios y tests. El modelo de contenido está orientado de manera similar a la organización de las bases de datos y centrado en la idea de la estructuración del dominio educativo.
- Basado en el modelo hipertexto, en el que se modeliza un dominio educativo como una red de componentes de una granularidad determinada y donde las interacciones del alumno vienen dadas por las decisiones que este realiza durante la navegación por el material.
- Centrado en el estudiante y en sus necesidades, el diseño se realiza adaptándolo a los conocimientos previos del estudiante y a las interacciones potenciales de éste con el entorno. Hay un análisis previo de las interacciones con el entorno desde un punto de vista pedagógico y esto permite incorporar algunos nuevos paradigmas de aprendizaje en el sistema.
- Entornos integrados de enseñanza, basados en la creación integrada de políticas de acceso a servicios conocidos en el ámbito de la red: foros de debate, sistemas de conferencia electrónica, servicios para compartir archivos, aplicaciones de comunicación sincrónica. Se orientan fundamentalmente hacia el soporte de trabajo en grupo, generalmente para dar servicio de intercambio de material entre los alumnos.

## 2.4 Los Sistemas Tutores Inteligentes (STI)

Un STI es un sistema experto en una materia, diseñado con el fin de impartir conocimiento y que exhibe alguna forma de inteligencia para guiar al estudiante en el proceso de aprendizaje. Permite emular el proceso de enseñanza - aprendizaje, adaptando el tipo y el contenido de la enseñanza a las necesidades específicas del alumno, decidiendo cuándo introducir nuevos conceptos o repasar los anteriores si éstos no han sido asimilados. Estos sistemas tienen en cuenta los conocimientos a enseñar (contenido pedagógico), la forma de enseñarlo (estrategia pedagógica), así como la información relevante sobre el alumno que está siguiendo el tutorial.

Los STIs tienen la capacidad de asesorar al alumno y guiarlo en su proceso de aprendizaje, pero es fundamental la detección del conocimiento erróneo para que el comportamiento del STI sea el esperado. En dominios declarativos en los que sólo se diagnostica conocimiento teórico se suelen realizar pruebas de tipo test. En cambio, en dominios procedimentales, es necesario comprobar las habilidades del alumno al abordar problemas concretos. En este caso, el diagnóstico correcto de las acciones del alumno es totalmente dependiente del conocimiento que tenga el módulo correspondiente del STI sobre el entorno virtual y del grado de integración entre ambos.

Varios autores [8,9,11,16,24] coinciden en que un STI debe cumplir ciertos requisitos:

- Poseer conocimiento tanto sobre la manera como sobre los conceptos a enseñar.
- Ser adaptativo. Debe adaptar el nivel de enseñanza y el nivel de conocimiento a las necesidades y ritmo de estudio del alumno.
- Capacidad de reacción. Debe ser capaz de intervenir si el alumno responde de forma imprevista o su evolución de conocimiento entra en conflicto con los objetivos del tutor.
- Eficacia. Debe disponer de un abanico de estrategias didácticas suficiente y de un mecanismo de selección de dichas estrategias lo más acertado posible para conseguir que el alumno aprenda apropiadamente.

- Motivador. Debe animar al alumno mediante mensajes de aliento para que su evolución sea positiva y rápida.
- Evaluador. Debe evaluar los conceptos enseñados mediante ejercicios y cuestionarios para comprobar si el alumno está realmente entendido los conceptos que se le están presentando.

Según Kaplan [16] la arquitectura de un STI está compuesta por cuatro módulos interconectados:

- La interfaz, o Módulo de Diálogo que permite a los usuarios, alumnos y docentes interactuar con el sistema.
- El modelo instruccional o pedagógico, que se encarga de definir la estrategia para transmitir el conocimiento a los usuarios. Se encarga de adaptar el sistema a las características concretas del alumno, seleccionando el material didáctico y evaluando los conceptos aprendidos mediante ejercicios y cuestionarios.
- El modelo experto o del dominio, el cual versa sobre la materia o curso que se impartirá. Representa el dominio que se pretende enseñar al alumno. Este dominio debe estar organizado de forma que su enseñanza resulte sencilla y clara. También debe mantener relaciones entre conceptos y prerrequisitos de aprendizaje.
- El modelo del estudiante, que refleja cuánto conoce el estudiante sobre el dominio, así como las experiencias cognitivas y de aprendizaje que ha llevado, del cual puede obtenerse un diagnóstico. Este módulo es el que permitirá STI adaptarse a las necesidades del aprendiz.

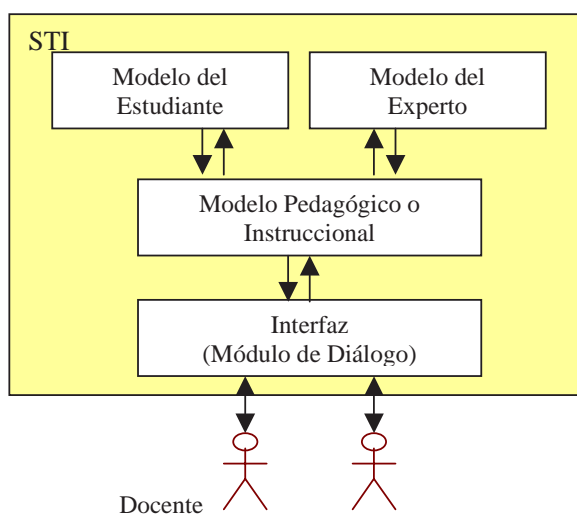


Figura 2 Arquitectura de un STI según Kaplan

**Nuestro enfoque:** Consideramos que el desarrollo efectuado se ubica en la categoría de software educativo interactivo e inteligente, o al menos con cierto grado de inteligencia, ya que tiene capacidad de aprender de las interacciones y pasos que el docente realiza para crear tutoriales y mostrar el proceso constructivo de algún tema, los recuerda y relaciona, y puede asistir al alumno en el entrenamiento, interviniendo y ayudando durante el proceso de ejercitación.

### 3 DISEÑO DEL ENTRENADOR

El diseño del sistema ha adoptado un enfoque orientado a objetos siguiendo la metodología de proceso unificado [4], aplicando patrones de software [5,6,13] y considerando los aspectos pedagógicos en etapas tempranas del ciclo de vida [12,15,18,22].

Los principales beneficios del modelo utilizado 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.

### 3.1 Arquitectura

La arquitectura del sistema es modular, de gran flexibilidad, con módulos independientes e interfaces bien definidas entre bloques, de forma tal que permite fácilmente cambiar los componentes que representan datos o transformaciones de datos de un dominio del conocimiento en particular. Se aplicó el patrón arquitectónico MVC (Model View Controller) [5], 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. El Modelo representa los datos o los objetos de la aplicación que son accedidos y presentados al usuario. La Vista es la representación en pantalla del estado actual del Modelo, y el Controlador es el objeto que manipula el Modelo y define la forma en que la interfaz de usuario reacciona ante las acciones 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 la definición de la interfaz gráfica se tuvieron en cuenta las conclusiones arribadas al respecto en investigaciones previas.

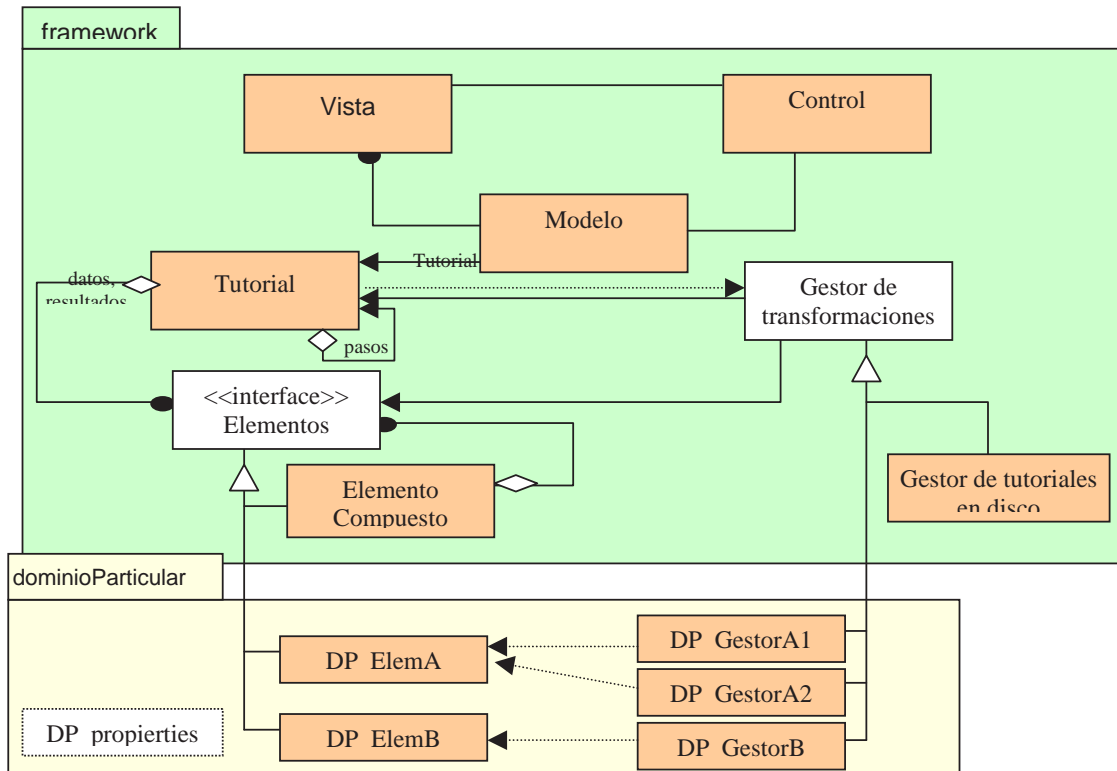


Figura 3 – Esquema de clases

Para permitir que pueda utilizarse en una variedad de dominios, su arquitectura tiene una clara separación entre las capacidades independientes del dominio y el conocimiento del dominio específico.

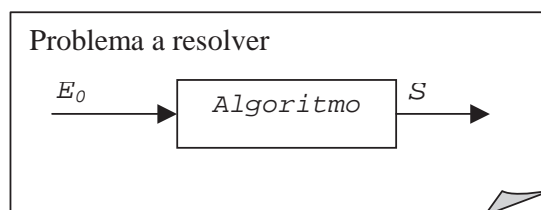
Se pueden distinguir módulos generales, comunes a todos los dominios de utilización de tutoriales y entrenadores, y módulos específicos que tienen en cuenta el comportamiento especial del dominio de aplicación. Los módulos generales describen un marco o “framework” y conforman el esqueleto de la aplicación, proveen la funcionalidad común, definen las abstracciones fundamentales y sus interfaces estableciendo 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 y entrenadores que pueden variar de una aplicación a otra, y es allí donde la arquitectura debe proveer flexibilidad.

### 3.2 Características del Dominio de Conocimiento

Las tareas del dominio candidato o elegible deben ser descomponibles en subtareas y el campo de conocimiento debe ser estable, es decir, no deben existir demasiadas estructuras de conocimiento que se vuelvan obsoletas en el corto plazo o antes de que el sistema alcance el estado de utilización completa.

En nuestro caso, pretendemos que el sistema sea capaz de:

1. Generar procedimiento de resolución de un problema, a partir de un ejemplo que desarrolla el docente (experto del dominio), que comienza con la declaración de datos iniciales y la ejecución de sucesivas transformaciones hasta llegar al resultado esperado.



2. Ser capaz de reproducir el método con otros datos iniciales o como paso intermedio en la resolución de otro problema.
3. Guiar al alumno durante el aprendizaje del procedimiento.

Con esta herramienta se pretende enseñar a resolver un problema mediante un ejemplo, mostrando a partir de un conjunto de datos iniciales, la forma de llegar al resultado explicitando todos los pasos intermedios necesarios y enfatizando los conceptos relacionados.

Lo que se pretende mostrar son las transformaciones necesarias a realizar sobre  $E_0$  para producir la salida  $S$ . Cada una de estas transformaciones es un paso del tutorial y habrá que realizarlas en un cierto orden, el cual puede no ser único.

Partiendo de esta base, es necesario identificar en el dominio los conceptos básicos que pueden intervenir como datos iniciales, y aquellos otros conceptos que representan transformaciones elementales sobre los primeros para producir nuevos conceptos, posiblemente más complejos.

### 3.3 Representación del conocimiento

Distinguimos de esta manera dos clases: *Elementos*, que representan los posibles datos de entrada y *Primitivas de Transformación*, que representa al algoritmo por el cual la entrada  $E_i$  del paso  $P_i$  se transforma en el elemento  $S_i$ .

El proceso completo de resolución de un problema queda encapsulado en la clase *Tutorial*, la cual tiene conocimiento de los Elementos que actúan como datos de partida y del conjunto de pasos que



conforman la resolución, así como el o los resultados que produce. Con colaboración de otras clases, el tutorial es capaz de almacenarse en disco y recuperarse para ser usado en otro contexto o como paso en la resolución de otro problema.

Mientras que la clase Tutorial es independiente de las características de un dominio, los Elementos y las Transformaciones tienen que especializarse en el dominio específico. Estas son las clases que aportan el comportamiento específico en un dominio particular.

Las interacciones de tutorización en un sistema de entrenamiento deben incluir explicaciones que describan el comportamiento de los componentes y los procesos físicos del sistema. Las explicaciones de comportamiento requieren la representación de modelos complejos o profundos en cooperación con los modelos conceptuales clásicos. Siguiendo esta idea, Vadillo Zorita et al. [26] proponen una representación del conocimiento del sistema formada por la coexistencia de múltiples modelos capaces de proporcionar diferentes puntos de vista del mismo con el objetivo de generar distintos tipos de explicaciones, identificando tres vistas diferentes del dominio:

- **Vista conceptual:** Agrupa todas las descripciones de los conceptos del sistema (componentes, procesos, procedimientos,...), y las relaciones entre ellos.
- **Vista estructural:** Indica cómo está organizado el sistema mediante la descripción de los componentes que lo constituyen y las conexiones físicas entre ellos.
- **Vista de comportamiento:** Agrupa las descripciones de comportamiento de las diferentes partes del sistema. Muestra cómo trabajan los componentes en función de sus procesos asociados.

### 3.4 Conocimiento pedagógico

Los conceptos del dominio, tanto teóricos como de entrenamiento, además de estar representados adecuadamente deben organizarse de una manera pedagógica que ayude en el proceso de enseñanza de los mismos. Es necesario tener en cuenta aspectos relativos al orden en que deben ser presentados al alumno y las relaciones de comparación entre ellos, distinguiendo si un concepto es prerequisite de otro o no, si es equivalente o no, si es más abstracto o concreto que otro, etc.

En nuestro sistema, algunas de estas relaciones son deducibles por el sistema en el proceso de generación del tutorial, pero muchas otras, requieren que los componentes propios del dominio, implementados como Elementos o Transformaciones, las establezcan, y así poder inferir, por ejemplo la equivalencia o no de dos procedimientos compuestos.

### 3.5 Adquisición de conocimiento

Los Elementos y Primitivas de Transformación aportan el conocimiento básico del dominio. La adquisición de más conocimiento por parte del sistema, se realiza durante el proceso de resolución de un problema mediante un ejemplo por parte del docente (experto). Mientras éste va seleccionando los datos y los pasos necesarios para su solución, el sistema va registrando estas acciones, tiene en cuenta precedencias y precondiciones y va generando un tutorial. Finalmente, cuando el docente indica que ha finalizado y desea guardarlo, el sistema genera dos archivos: un archivo XML y otro HTML. El primero, es utilizado por el propio sistema cuando se requiere utilizar el tutorial creado, como paso de solución de otro problema, o cuando el alumno requiere hacer prácticas de resolución en esa temática. Este archivo, tiene la información necesaria para reproducir la solución del problema o resolver otro problema del mismo tipo, como así también, reconocer si la solución que está planteando el alumno es la correcta o equivalente, y en caso contrario intervenir con algún tipo de ayuda o refuerzo. El segundo archivo, es la explicación del tutorial creado, que se utiliza cuando ejecuta el tutorial para mostrar o enseñar los pasos de

resolución, y que también puede ser utilizado como Objeto de Aprendizaje para ser compartido y reutilizado en otro contexto.

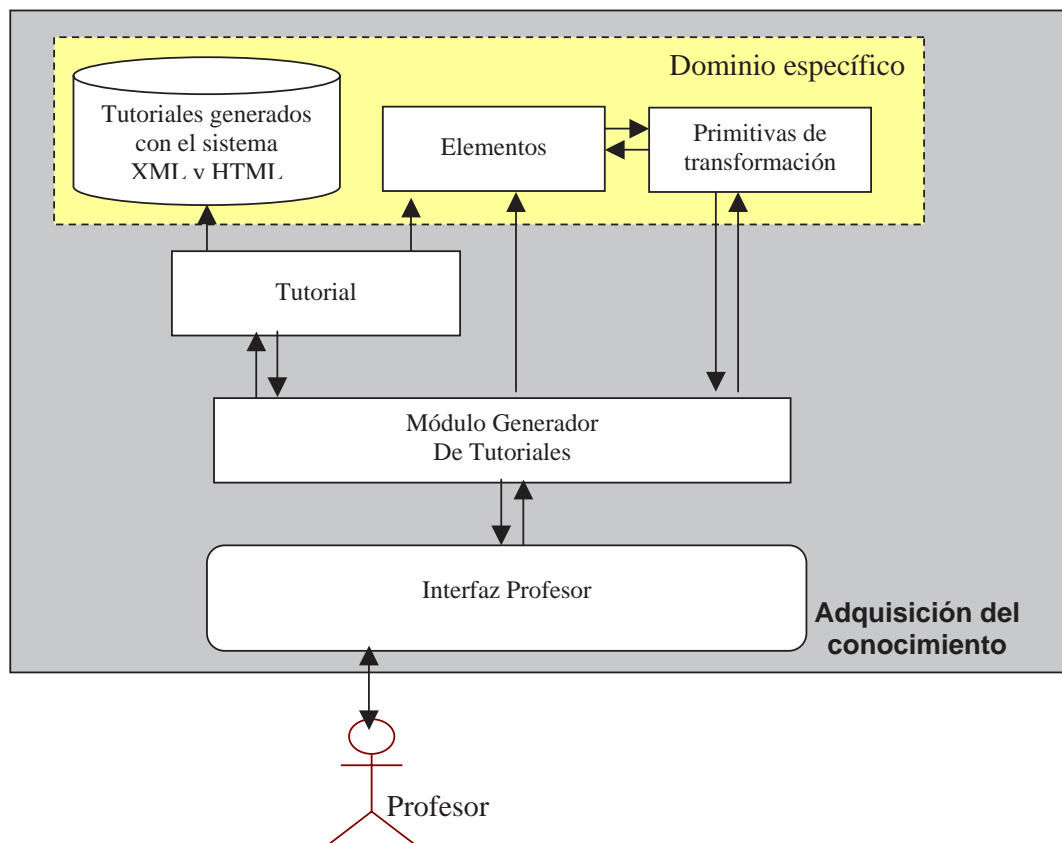


Figura 4 – Adquisición del conocimiento

### 3.6 Entrenamiento

Para el entrenamiento, el núcleo principal es el Tutor quien decide y guía el proceso de enseñanza y adiestramiento de cada alumno en particular. Este proceso de enseñanza y adiestramiento requiere una estrecha interacción entre el Tutor y el alumno en pos de conseguir los objetivos planteados. En este sentido las explicaciones dadas por el tutor son de una gran importancia debido a que su adecuación marcará el éxito en la consecución del objetivo primordial, que el alumno aprenda.

Cuando el alumno decide realizar una práctica, el tutor le ofrece elegir entre las disponibles para su perfil. Una vez elegida, el Tutor genera un tutorial con la información recuperada de la base XML, y lo tiene como procedimiento base contra el cual comparar las acciones del alumno como así también para mostrar la solución del problema en forma total o parcial, si el alumno lo solicitara. A medida que el alumno interacciona con el sistema para realizar los pasos u operaciones sobre el procedimiento en que está entrenando, el módulo Generador de Tutoriales, va registrando todas esas operaciones, de igual manera que lo hacía con el docente y creando un nuevo tutorial con los datos y pasos que el alumno selecciona. Por su parte, el Tutor está continuamente observando y por cada paso realizado por el alumno, compara con el tutorial base, analizando si es correcto o es factible de serlo, o si es un paso innecesario o si es definitivamente incorrecto. En función de ello, y la configuración de errores establecida, decide o no intervenir para ayudar y guiarlo en su aprendizaje.

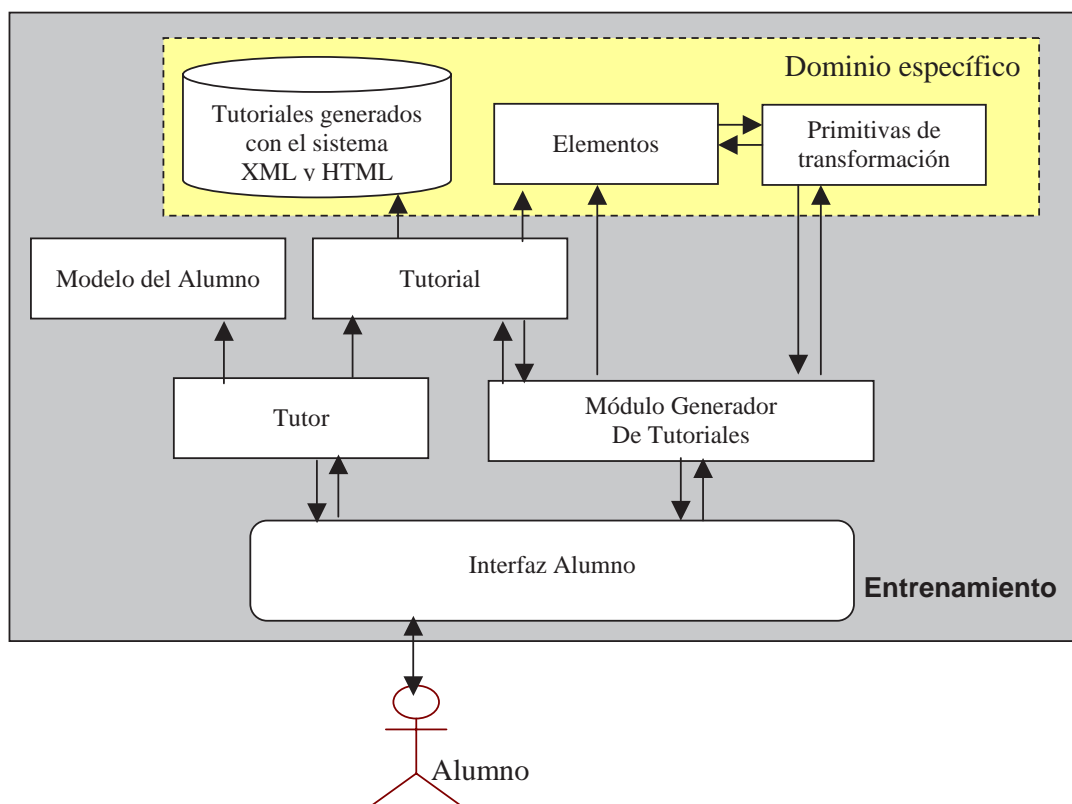


Figura 5– Entrenamiento

## 4 Conclusiones

El uso de tutoriales y entrenadores presenta importantes beneficios educativos ya que estimula y ayuda a los alumnos en las distintas situaciones del aprendizaje.

En nuestro desarrollo, el entrenador siempre puede interpretar los diferentes caminos que surgen de alterar el orden de las operaciones requeridas, teniendo en cuenta aquellas sub-secuencias que no podrían ser alteradas sin producir error, y también puede distinguir algunas formas equivalentes de realizar una tarea. De todas maneras, se encuentra restringido al mundo acotado que determina la inteligencia dada a los componentes específicos, ya que no sería capaz de inferir otras soluciones al mismo problema si no fueron previstas por el desarrollador del componente.

Más allá de las facilidades provistas por herramientas con características similares a ésta, el docente debe mantener su rol de mediador en el proceso de enseñanza-aprendizaje, ya que ninguna herramienta supliría su función.

## 5 Bibliografía

- [1] Alonso C, Gallego D., Honey P. (1999). *Los estilos de aprendizaje: procedimientos de diagnóstico y mejora*, Ediciones Mensajero, Bilbao.
- [2] Ausubel-Novak-Hanesian (1983.) *Psicología Educativa: Un punto de vista cognoscitivo* .2º Ediciones TRILLAS México
- [3] Booch G., Jacobson I., Rumbaugh J. (1998). *The Unified Modeling Language*, Addison-Wesley Publications

- [4] Booch G., Jacobson I., Rumbaugh J. (1998). *The Unified Process Software Development* Addison-Wesley Publications,
- [5] Buschmann F., Meunier R., Rohnert H., Sommerland, P., Stal, M. (1996) *Pattern-Oriented Software Architecture: a system of patterns*. Ed. Wiley
- [6] Cooper, James W. (1998) - *Java Design Patterns: A Tutorial* – Addison Wesley
- [7] Cruz Feliú, J. (1997) *Teorías del aprendizaje y tecnología de la enseñanza*. Trillas. México.
- [8] Cuevas Vallejo, Carlos Armando (1996) “*Sistemas Tutoriales Inteligentes*” INVESTAV, I.P.N., consultado en [www.matedu.cinvestav.mx/~ccuevas/STI96.htm](http://www.matedu.cinvestav.mx/~ccuevas/STI96.htm)
- [9] Cumming G. D. y Self J. A. (1991), “*Learner models in collaborative intelligent educational systems*”, <http://citeseer.ist.psu.edu/cumming91learner.html>
- [10] Felder R. M. & Henriquez E. R. (1995) *Learning and Teaching Styles* in Foreign and Second Language Education, Foreign Language Annals, 28, No. 1.
- [11] Fred, A. (1994) “*Expert Systems: Conceptual, Methodological and Management Aspects,*” en Project HC 1014 – ENN – European Neurological Network.
- [12] Galvis, Alvaro (1994) *Ingeniería de Software Educativo*. Ediciones Uniandes.
- [13] Gamma, Eric; Helm, Richard; Johnson, Ralph and Vlissides, John (1995) *Design Patterns. Elements of Reusable Software*, Addison-Wesley.
- [14] Gardner H. (1987) “*La nueva ciencia de la mente: Historia de la psicología cognitiva*”. Barcelona.
- [15] Gómez, R., Galvis, A. y Mariño, O. (1999) *Ingeniería de software educativo con modelaje OO*. Brasil. Consultado en: <http://www.minerva.uevora.pt/simposio/comunicacoes/riomezmarino.html>
- [16] Kaplan Randy, Rock Denny (1995): “*New Directions for Intelligent Tutoring*”. AI Expert, Febrero.
- [17] Kolb D. (1984) *Experiential Learning: Experience as the Source of Learning and Development*, cit. por Felder Richard & Henriquez Eunice (1995)
- [18] Marqués, Pere (1999) – *Programas Didácticos: Diseño y Evaluación* –Consultado en <http://www.xtec.es/~pmarques/edusoft.htm>
- [19] Novak J. D., y Gowin D. B., *Aprendiendo a aprender*, 1988 Ed. Martínez Roca. Barcelona.
- [20] Pressman, Roger S.(1998), *Ingeniería de software: Un enfoque práctico*. Ed. Mc Graw Hill.
- [21] Rodríguez Artacho Miguel (2000) *Una arquitectura cognitiva para el diseño de entornos telemáticos de enseñanza y aprendizaje* - Tesis doctoral Universidad Nacional de Educación a Distancia - Escuela Técnica Superior de Ingenieros Industriales.
- [22] Rosanigo, Z.B., Paur, A., Bramati, P. (2000) *Metodología de desarrollo de software educativo*. Actas de VI Congreso Internacional de Ingeniería Informática ICIEY2K Fac. de Ingeniería, U.B.A. – Bs. As.
- [23] Salcedo Lagos P. (2002) *Revista Ingeniería Informática*, edición 6, <http://www.inf.udec.cl/revista/edicion6/psalcedo.html>
- [24] Sancho, L. (2002) *Sistemas Tutores Inteligentes: Una alternativa para el uso de computadoras en educación*. Education Net. Red Global de Educación a Distancia. (DistEdNet) Universidad Estatal a Distancia. Consultado en [www.uned.ac.cr/biblioteca/global/ensenanza/instruccion/articulos/sistemas.html](http://www.uned.ac.cr/biblioteca/global/ensenanza/instruccion/articulos/sistemas.html)
- [25] Skinner B. F., (1958): *Teaching Machines*, Science, publicado en 1958
- [26] Vadillo Zorita J. A., Díaz de Ilaraza A., Fernández I., Gutiérrez J. e Elorriaga J. A. *Explicaciones en Sistemas Tutores de Entrenamiento: Representación del Dominio y Estrategias de Explicación* Departamento de Lenguajes y Sistemas Informaticos Universidad del País Vasco España Consultado en: [http://www.niee.ufrgs.br/ribie98/CONG\\_1994/volume\\_II/C30/II\\_289\\_309.html](http://www.niee.ufrgs.br/ribie98/CONG_1994/volume_II/C30/II_289_309.html)