

TECNOLOGÍA EDUCATIVA



Software para la evaluación *Online* bajo el enfoque de procesos

MARLENE ARIAS¹

DEISY CARRERO²

FRANCISCA GRIMÓN³

OTNIEL MORALES⁴

Universidad de Carabobo

¹ marias@uc.edu.ve

² deisycarrero@hotmail.com

³ fgrimon@uc.edu.ve

⁴ odmoralesb@hotmail.com

Resumen

La evaluación dentro del proceso de enseñanza y aprendizaje constituye una etapa que permite determinar los logros del estudiante. Generalmente, la evaluación del aprendizaje se ha centrado en el producto, mas no en el proceso que se sigue para llegar a la construcción de los conocimientos. El objetivo de esta investigación es conocer los beneficios de la evaluación centrada en procesos, mediante el desarrollo e implantación de un *software* de evaluación *online*. El caso de estudio se refiere a las operaciones de suma y producto de curvas en el plano cartesiano de la asignatura Matemáticas I de la Facultad Experimental de Ciencias y Tecnología (FACYT) de la Universidad de Carabobo. Se utiliza la propuesta metodológica de Alcalde y Salas, la cual es un híbrido entre *eXtreme Programming* o Programación Extrema (XP), con la Metodología de Diseño Instruccional de Materiales Educativos Computarizados (MDI-MEC) y el Proceso Unificado de Rational (RUP). Las conclusiones de esta investigación indican que al utilizar el *software* de evaluación *online* bajo el enfoque de procesos, el alumno es consciente de las fallas y aciertos que ha logrado durante la construcción de su conocimiento. Adicionalmente, el *software* facilita el proceso de aprendizaje del estudiante, mientras que al profesor le permite evaluar su práctica docente, así como también obtener estadísticas con respecto al proceso de aprendizaje de sus estudiantes.

Palabras clave: *Software*, matemáticas, teoría constructivista, evaluación.



Abstract

The evaluation in education and learning process constitutes a stage that allows determining the student profits. Generally, the learning evaluation has been centred in the final product but not in the knowledge construction process. The objective of this research is to know about the benefits centred in the evaluation process, by means of the development and implantation of evaluation software online. The case of study, talks about Sum and Product Operations of Curves in Mathematical the Cartesian plane of the subject I of the Sciences and Technologies Faculty (FACYT) at Carabobo University. It is used a methodological proposal Alcalde y Salas, which is a hybrid between carries far eXtreme Programming (XP), and an Instructional Methodology to Educative Materials Computerized Design (MDI-MEC) and Rational Unified Process (RUP). Some of the conclusions of this work indicate that using the software of evaluation online based on processes approach, the student is conscious of his own fails and successes obtained during the knowledge construction. Additionally, software facilitates the student learning process, whereas teachers can also evaluate educational practices, as well as, to obtain statistics based on students improve and learning.

Key words: Software, Mathematics, Constructivism Theory, Online Evaluation.

Introducción

Al momento de efectuar una evaluación, se están verificando los conocimientos y las habilidades del estudiante. La evaluación por procesos implica observar y analizar una serie de estados progresivos del estudiante, que permitan conocer e identificar las actuaciones del alumno, promover su potencial desarrollo y humanizar la evaluación. Aunado a esto, se tiene el uso de la computación como apoyo a los procesos de enseñanza y aprendizaje, haciéndose cada día mayor la asimilación de procesos informáticos dentro de las instituciones educativas (Shih-Chung, 2001). Para realizar *software* de calidad (Parzinger, Nath & Lemons, 2001), se debe considerar la incorporación de aspectos didácticos, pedagógicos, de evaluación y los inherentes a la ingeniería del desarrollo de *software*.



Esta investigación desarrolla un *software* para la evaluación *online* bajo el enfoque de procesos, para apoyar la enseñanza y aprendizaje sobre las operaciones de suma y producto de curvas en el plano cartesiano de la asignatura Matemáticas I de la FACYT. Para la elaboración del *software* se utiliza la propuesta metodológica de Alcalde y Salas, la cual consiste en la combinación de varias metodologías: *eXtreme Programming* o Programación Extrema (XP) (Charte, 2002), Metodología de Diseño Instruccional de Materiales Educativos Computarizados (MDI-MEC) (Arias, 2001) y el Proceso Unificado de Rational (RUP) (Kruchten, 2001). En las secciones de marco teórico y marco metodológico se amplían los detalles de las metodologías.

El resultado de la investigación es un *software* de evaluación *online*, en el área de las matemáticas, para el tema objeto de estudio, que permite al estudiante conocer sus fallas y aciertos durante la construcción de su conocimiento. Adicionalmente, el *software* facilita el proceso de aprendizaje del estudiante, mientras que al profesor le permite evaluar su práctica docente, así como también obtener estadísticas con respecto al proceso de aprendizaje de sus estudiantes.

Uno de los aportes de esta investigación es apoyar el proceso de enseñanza-aprendizaje mediante el uso del *software* de evaluación, dando la posibilidad de hacer un seguimiento a los estudiantes sobre sus avances y deficiencias en el estudio de los contenidos.

La estructura de este artículo se presenta en los siguientes capítulos, de la siguiente forma:

Capítulo I: Se establece la justificación sobre la investigación de evaluación *online* con enfoque de procesos, para apoyar el proceso de enseñanza y aprendizaje en la asignatura Matemáticas I.

Capítulo II: Se presenta un marco teórico sobre: *eXtreme Programming* o Programación Extrema (XP), Metodología de Diseño Instruccional de Materiales Educativos Computarizados (MDI-MEC), el Proceso Unificado de Rational (RUP), SCRUM, la propuesta Metodológica de Alcalde y Salas, Teoría Constructivista y la evaluación centrada en procesos.



Capítulo III: Se expone el marco metodológico de la investigación, el cual consiste en tres etapas: diagnóstico, propuesta metodológica y aplicación de la propuesta.

Capítulo IV: Se presentan los resultados de la investigación de acuerdo con la metodología aplicada; se puede observar en la Tabla N° 1 los artefactos generados en cada una de las fases de la metodología aplicada a esta investigación.

Al finalizar, se muestran las conclusiones de la investigación.

Capítulo I: **Justificación**

La utilización de las Tecnologías de la Información y la Comunicación (TIC) en el ámbito de la Educación Superior plantea cambios significativos y nuevas opciones en la búsqueda de mejorar el proceso de enseñanza-aprendizaje. Actualmente se pueden aprovechar los medios: video, audio, computadores, *software* entre otros, para atraer la atención del estudiante y ayudarlo a construir su conocimiento (Salcedo, 2002), aunado con una evaluación *online* que le permita una retroalimentación sobre sus aciertos y fallas.

En las universidades existe interés en el uso de World Wide Web (www) como una herramienta de aprendizaje y de evaluación, porque permite al estudiante acceder a una gran variedad de contenidos que favorecen la construcción de su conocimiento (Bouras, Lampsas & Spirakis, 2002).

Esta investigación surge por la necesidad de mejorar el rendimiento estudiantil en la asignatura Matemáticas I, en las diferentes carreras que ofrece la Facultad de Ciencias y Tecnología de la Universidad de Carabobo (FACYT). La investigación parte de un diagnóstico de la situación actual, mediante la opinión de los estudiantes con respecto al futuro uso del *software* y, por otra parte, detectar los errores más frecuentes que presentan los estudiantes cursantes de la asignatura Matemáticas I de la FACYT. Adicionalmente se estudió en conjunto con los profesores de la asignatura el plan de evaluación, con la posibilidad de realizar un *software* que permita



realizar evaluaciones *online*, bajo un enfoque de procesos. Además, en una investigación previa se desarrollaron los contenidos de la materia en un *software* educativo, que permite a los estudiantes reforzar sus conocimientos en el área de Matemáticas I, específicamente en el tema Curvas en Plano Cartesiano. Como resultado de este diagnóstico se elaboró un *software* de evaluación *online*, donde se toman aspectos concernientes a varias metodologías: XP (Metodologías Ligeras o *Lightweight Metodologies*), RUP (Metodologías Pesadas o *Heavyweight Metodologies*) y MDI-MEC (Diseño Instruccional), con el propósito principal de desarrollar una herramienta computacional en la web que cumpla con las necesidades que llevaron a su creación, tanto en el entorno computacional como educativo.

Esta investigación dará un aporte a las áreas de evaluación del aprendizaje, de desarrollo de *software* y a los MEC porque integra lineamientos para realizar evaluación por procesos, diseñar *software* educativo y utilizar metodologías de desarrollo de *software*. Asimismo, permitirá apoyar el proceso de enseñanza y aprendizaje en la asignatura Matemáticas I de las carreras ofertadas por la FACYT, a través del uso de tecnologías en conjunto con métodos pedagógicos, ofreciendo técnicas y variedad de recursos que hacen más efectivo y flexible el proceso de enseñanza y aprendizaje.

De no realizarse esta investigación, la FACYT no contaría con un *software* de evaluación *online* bajo el enfoque de procesos, que le permitiría apoyar el proceso de enseñanza y aprendizaje en la asignatura Matemáticas I. En este orden de ideas, la herramienta computacional que se desarrolló se plantea como una nueva alternativa de evaluación en los alumnos, siendo un soporte económico y actualizable; otorgando un importante grado de libertad tanto al alumno como al profesor. Permitirá al estudiante valorar de forma individual sus conocimientos cuando quiera, donde quiera y cuantas veces lo desee. Asimismo, generará confianza en el alumno, mayor libertad en la programación de sus estudios, le permitirá la interacción y aprender de sus errores. El profesor podrá actualizar la información en cualquier momento, sin grandes esfuerzos y costos económicos de edición; utilizará el *software* como medio evaluador; analizará las características didácticas de las preguntas en función de los resultados.



Capítulo II: Marco teórico

En este apartado se realiza la descripción de las metodologías de desarrollo de *software* utilizadas en esta investigación: *eXtreme Programming* o Programación Extrema (XP), Proceso Unificado de Rational (RUP), la Metodología de Diseño Instruccional de Materiales Educativos Computarizados (MDI-MEC), metodología SCRUM y la propuesta metodológica de Alcalde y Salas.

1. Programación Extrema o *eXtreme Programming* (XP): Es una metodología ligera de desarrollo de *software* que se basa en la simplicidad, la comunicación y la reutilización del código desarrollado. Enfatiza el trabajo en grupo, se basa en la retroalimentación continua entre el usuario y el equipo de desarrollo (Letelier & Sánchez, 2003), colocando especial hincapié en la evaluación de los programas. Se crean *test* antes de la implementación, durante y cuando se haya concluido el *software*; tienen como objetivo principal la satisfacción del usuario y potenciar al máximo el trabajo en equipo (Charte, 2002). Las fases que contemplan esta metodología son:

Fase de Planificación. El objetivo de esta fase es planificar el proyecto. Entre las actividades a realizar se tienen: redactar las historias de usuarios, crear un plan de entregas, hacer pequeñas entregas, pero frecuentes, controlar la velocidad del proyecto.

Fase de Diseño: En esta fase se deben crear soluciones puntuales para reducir los riesgos. Además se utilizan tarjetas CRC (Cargo, Responsabilidad y Colaboración) en las reuniones de diseño.

Fase de Desarrollo: En esta fase se escribe el código del *software*.

Fase de Pruebas: Todo el código debe pasar las unidades de prueba antes de ser implantado. Se deben ejecutar pruebas de aceptación y publicar los resultados.

2. El Proceso Unificado de *Rational* o *Rational Unified Process* (RUP): Es un proceso de desarrollo de *software* con un conjunto de actividades necesarias para transformar los requisitos de un usuario en un



software. El Proceso Unificado está basado en componentes, es decir, que está formado por componentes *software* interconectados a través de interfaces bien definidas. El Proceso Unificado utiliza el Lenguaje Unificado de Modelado (*Unified Modeling Language*, UML) para preparar todos los esquemas de un sistema *software* (Kruchten, 2001). Las fases que contemplan este proceso son:

Fase de Inicio: El objetivo de esta fase es hacer un análisis para justificar la puesta en marcha del proyecto. Para ello, primero se debe determinar el alcance del sistema propuesto y proponer una arquitectura. Además, se definen los límites, los riesgos críticos y los costos del sistema a desarrollar.

Fase de Elaboración: Tiene como objetivo formular los requisitos funcionales (casos de uso). Además, establecer una base de la arquitectura del sistema.

Fase de Construcción: El propósito primordial de esta fase es dejar listo un producto de *software* en su versión operativa inicial, a veces llamada "versión beta". El *software* debería tener la calidad adecuada para su aplicación y asegurarse de cumplir los requisitos.

Fase de Transición: Esta fase se centra en implantar el *software* en su entorno de operación. El objetivo de esta fase es cumplir los requisitos establecidos en las fases anteriores, hasta la satisfacción de todos los usuarios.

3. Diseño Instruccional de Materiales Educativos Computarizados (MEC): Se basa en la necesidad de concebir el medio instruccional, es decir, el computador, como un medio dinámico. El medio instruccional se concibe aquí como cualquier persona, dispositivo o material que transmita el mensaje requerido para el logro de un aprendizaje. Debe considerarse asimismo que el medio puede utilizarse para desarrollar destrezas requeridas para el procesamiento de la información transmitida en el contenido del mensaje; por ejemplo, estrategias de codificación y de elaboración. Es decir, que el medio puede tener dos funciones: transmitir mensajes y desarrollar las destrezas requeridas para procesar la información contenida en el mensaje.



La metodología está compuesta por cuatro fases y un eje transversal. No se requiere la culminación de una fase para pasar a la otra, es posible obtener rápidamente un prototipo que permita hacer validaciones parciales y correcciones, de ser requeridas (Arias, 2001). Las fases que se deben cumplir en esta metodología son:

Fase de Diseño Educativo: Permite resolver las interrogantes que se refieren al alcance y tratamiento que debe ser capaz de apoyar el MEC.

Fase de Producción: Se realizan los guiones de contenido, didáctico y técnico que resaltarán para identificar los aprendizajes que se establecieron en el diseño educativo.

Fase de Realización: Realización del *storyboard* y posibilidad de realizar correcciones al prototipo.

Fase de Implementación: Proceso de construcción del *software*.

Eje Transversal de Evaluación: Evaluación constante en función de los resultados obtenidos durante el proceso.

4. SCRUM: Es un proceso interactivo e incremental para el desarrollo de *software*, creado por Ken Schwaber y Jeff Sutherland. Scrum se plantea como objetivo principal producir la mayor cantidad de *software* de calidad posible a través de una serie de períodos de tiempo breves denominados *Sprints*. Estos períodos duran aproximadamente 30 días.

Las fases de desarrollo tradicionales (requisitos, análisis, diseño, evolución y entrega) son realizadas también por medio de *Sprints*. Cada *Sprint* no sigue un proceso definido, sino que es elaborado por medio de reuniones diarias denominadas *Scrum Meetings*. Este es el principal método de comunicación del equipo y en ellas cada desarrollador explica el trabajo realizado desde la última reunión y los problemas con que se ha encontrado, así como las tareas que va a realizar hasta la siguiente reunión.

Un aspecto importante de *Scrum* es que al final de cada *Sprint* hay una demostración, con el objetivo de mostrar al cliente las funcionalidades finalizadas y de esta manera involucrarlo en el proyecto.



5. Propuesta metodológica de Alcalde y Salas (Alcalde & Salas, 2003): El objetivo es proporcionar un ambiente de trabajo en el que el equipo de desarrollo esté conformado por personas del área tanto computacional como educativa, formando de esta manera un equipo multidisciplinario para la generación de materiales educativos computarizados.

El esquema de la metodología muestra la integración de XP, RUP y la MDI-MEC, en la cual se generan una serie de artefactos en cada fase iterativa de desarrollo y donde la revisión constante de la misma está controlada por un eje transversal llamado Evaluación, que representa una revisión continua de los productos entregables.

La metodología consta de cinco fases, dentro de las cuales se realiza un conjunto de actividades, generándose artefactos para cada fase de desarrollo de *software*.

Las fases se describen a continuación:

Fase Inicial: Se comienza con una reunión inicial entre clientes y desarrolladores en donde se recopila información con respecto al proyecto; se identifica qué actores participan de forma directa o indirecta en el proceso de desarrollo; se realiza un estudio de factibilidad tanto económica, operativa y técnica; se esboza una lista inicial de los posibles riesgos que pueden presentarse durante la puesta en marcha del proyecto a fin de realizar un plan de contingencia que los minimice; y se hace una descripción de la arquitectura inicial a utilizar.

Fase de Planificación: A partir de una serie de reuniones con los clientes se crean las historias de usuarios, las cuales son descripciones cortas pero concretas, escritas por el cliente en el lenguaje del cliente y que definen los requisitos del *software*. Son una guía inicial para realizar valoraciones y estudios de riesgo sobre el proyecto. A partir de éstas los desarrolladores realizan una planificación para la finalización de las mismas en cada iteración del proyecto. También se concreta la arquitectura a emplear en las fases de diseño y desarrollo.

Fase de Diseño: Para cada iteración se utilizan las tarjetas Clase, Responsabilidad y Colaboraciones (CRC) para realizar el diseño de las



historias de usuarios que se implementan en dicha iteración. Se enfatiza el trabajo en equipo y la colaboración de todos los miembros del equipo de desarrollo en la creación del material educativo computarizado.

Fase de Desarrollo: Durante la codificación se lleva a cabo la implementación de las historias de usuarios que han sido fijadas como objetivo para la iteración actual. Se crean las unidades de pruebas a partir de las unidades de aceptación. Se usa la técnica de refactorización, es decir, se rediseña todo el tiempo dejando el código siempre en el estado más simple posible; esto ayuda a mantener el diseño simple y limpio, eliminando redundancias y complejidad innecesaria, mejorando la legibilidad, reusabilidad e introducción a mejoras. Todo el código desarrollado se crea por parejas de programadores, para luego hacer uso de la integración secuencial en conjunción con la propiedad colectiva del código.

Fase de Pruebas: Una vez desarrolladas las historias de usuarios y que las nuevas funcionalidades añadidas en la iteración superan todas las unidades de prueba y las pruebas de aceptación, se considera terminada la iteración; en el caso de que aún existan funcionalidades inacabadas, se vuelve a realizar una planificación de las mismas para ser implementadas en una nueva iteración.

Eje Transversal de Evaluación: La evaluación se debe hacer constantemente en función de los resultados que se van obteniendo durante todo las fases.

6. Teoría Constructivista: Esta teoría indica que el conocimiento se construye en forma activa por parte del estudiante, quien viene a ser un actor dinámico, responsable y consciente de su aprendizaje (Sosa, García, Sánchez, Moreno, & Reinoso, 2005). Esta teoría ha sido utilizada en varias disciplinas, entre las cuales cabe destacar matemáticas y computación, que representan los ámbitos relacionados con el desarrollo de este trabajo. En concordancia con esta teoría y la investigación realizada, se tiene que el conocimiento construido por el estudiante viene dado por las interacciones que realiza al trabajar con el *software* para evaluación *online*.



En Pujol, 1999 se indica que el constructivismo como filosofía educativa sostiene que cada estudiante construye su significado de los eventos que aprende. Asimismo, hace referencia a dos aspectos:

1. el aprendizaje es un proceso activo de construcción más que de adquisición de conocimiento;
2. la instrucción es un proceso de apoyo a esa construcción más que comunicar conocimientos (Duffy y Cunningham, 1996, tomado de Pujol 1999).

7. Evaluación centrada en procesos: Partiendo del hecho que el conocimiento se construye, se deben considerar evaluaciones que permitan conocer los aprendizajes adquiridos por el estudiante; por lo tanto, es necesario realizar evaluaciones centradas en el proceso. Según Ahumada, la evaluación debe centrarse mayormente en las vivencias y evidencias del aprendizaje de proceso, más que de resultados. En este caso, la evaluación permite comprobar que las estrategias de aprendizaje son las más indicadas para los fines formativos.

En Ahumada, 2001 se indica que la evaluación debería ser considerada un proceso y no un suceso; interesa conocer evidencias centradas en el proceso de aprendizaje más que en los productos. El proceso evaluativo debe relacionarse con la naturaleza del aprendizaje, y debería pasar inadvertido para el estudiante, porque sería el resultado del desarrollo de distintas actividades de aprendizaje.

Cada vez que el estudiante se enfrenta a nuevas situaciones hay un flujo de información que se construye ante la situación y la forma de resolver el problema planteado; todo esto conlleva a una evaluación de procesos (Blanco, 2004).

Capítulo III: **Marco metodológico**

Se dividió la investigación en tres etapas: Diagnóstico, Propuesta Metodológica y Aplicación de la Propuesta Metodológica. A continuación se detallan los aspectos más importantes en cada etapa.

1. Diagnóstico: En esta etapa se elaboraron los instrumentos para conocer la opinión de los estudiantes con respecto al futuro uso del *software* y por otra parte detectar los errores más frecuentes que presentan los estudiantes cursantes de la asignatura Matemáticas I de la FACYT de la Universidad de Carabobo. Posteriormente se aplicaron los instrumentos en las muestras seleccionadas y se analizaron e interpretaron los datos obtenidos luego de aplicar los instrumentos.

Las unidades de estudio fueron los alumnos cursantes de la asignatura Matemática I de los departamentos Computación y Química de la FACYT en la Universidad de Carabobo. Para los efectos de este estudio se realizó una consulta de opinión:

En el caso de los alumnos cursantes de la asignatura Matemáticas I de la FACYT: se seleccionó una muestra intencional no estadística de alumnos cursantes de la asignatura Matemáticas I para ser encuestados. Para tal fin se distribuyeron 30 cuestionarios en cada departamento: Computación y Química.

En el caso de los profesores de la asignatura Matemáticas I de la FACYT: se seleccionó una muestra intencional no estadística de profesores de la asignatura, para ser encuestados. Para tal fin se distribuyeron siete encuestas a los diferentes profesores del área de matemáticas.

Además se recolectaron datos extraídos de Control de Estudios Sectorial de la FACYT, donde se generaron reportes con respecto a los índices académicos de los alumnos cursantes de la asignatura Matemáticas I.

2. Propuesta metodológica: Se realizó un estudio de la Metodología planteada por Alcalde y Salas, la cual es un híbrido entre la Metodología de Diseño Instruccional de *Software* Educativo, la Metodología *eXtreme Programming* y algunos artefactos del proceso de desarrollo de *software* de la *Rational Unified Process*.

3. Aplicar la propuesta metodológica: El enfoque metodológico que se siguió es el propuesto por Alcalde y Salas, haciendo unos ajustes para



cubrir aspectos que no se contemplaban, entre los cuales se tiene la elaboración del Documento Visión, el Plan de Proyecto, el Glosario y Modelo de Datos (E-R). El enfoque metodológico seguido para desarrollar el *software* consta de: Fase Inicial, Fase de Planificación, Fase de Diseño, Fase de Desarrollo y Fase de Prueba. En cada una de las fases antes mencionadas se genera un conjunto de objetos (artefactos) que se presentan en el capítulo de resultados (Ver Tabla N° 1).

La fase inicial permite hacer un análisis para justificar la puesta en marcha del proyecto. En la segunda fase se realiza la planificación del proyecto. En la fase de diseño se realiza el diseño del sistema que se utilizará en la fase de desarrollo. La fase de prueba permite realizar un conjunto de pruebas de aceptación.

Capítulo IV: **Resultados**

A continuación se presentan los resultados obtenidos siguiendo el marco metodológico planteado en el capítulo anterior.

1. Diagnóstico: Las respuestas obtenidas por los estudiantes mediante el instrumento utilizado indican que:

- El 100% de los estudiantes dice estar de acuerdo con el uso de una herramienta computacional *online* para la evaluación bajo el enfoque de procesos, en el área de Matemáticas I.
- El 100% manifiesta que desea conocer su proceso formativo y reflexionar sobre sus propias dificultades y aciertos, para ser consciente del proceso que ha seguido durante el aprendizaje.
- El 99% de personas encuestadas está de acuerdo en realizar preguntas al profesor a través de esta herramienta.
- Al momento de realizar operaciones algebraicas, con lo cual deberían identificar las partes positivas y negativas de las curvas operando, 50% de los estudiantes realizó correctamente esta acción dejando el otro 50% a estudiantes que presentaron fallas.



2. Propuesta metodológica y su aplicación: Para el desarrollo del *software* de evaluación *online* con enfoque de procesos se siguió la metodología propuesta en el capítulo anterior. Por cada una de las fases que componen la metodología se obtuvieron objetos (artefactos), los cuales serán mostrados en la Tabla N° 1; aquí se detallan los objetos generados en la Fase Inicial, Fase de Planificación, Fase de Diseño, Fase de Desarrollo y Fase de Prueba.

TABLA N° 1
ARTEFACTOS RESULTANTES

Fases	Artefactos	Definir necesidades de alto nivel	Reuniones	Def. términos del dominio	Facilidad económica	Facilidad operativa	Facilidad técnica	Adm. de riesgos	Monitoreo del progreso	Requerimientos func. y no func.	Iteraciones	Plan de iteraciones	Plan de entrega	Estudio de necesidades	Descripción de los usuarios	Propósito y objetivos referidos al proy.	Análisis estructural	Esp. de los conocimientos previos	Formulación de objetivos específicos	Selección de estrategias instruccionales	Contenido	Selección de estrategias de evaluación	Determinación de variables técnica	Guión didáctico	Guión técnico	Guión de contenido	Verificar correcto funcionamiento	Prototipo y corrección	Construcción del software	Verificar el funcionamiento del sistema
Inicial	Doc. visión	X	X						X				X	X																
	Análisis de factibilidad	X		X	X	X																								
	Lista inicial de riesgo			X	X	X	X																							
	Des, inicial de la arq.	X				X																								
	Plan del proyecto								X																					
Planificación	Glosario	X	X																											
	Historias de usuarios	X	X						X																					
	Planificación	X							X	X	X	X	X																	
	Lista de riesgo			X	X	X	X																							
Diseño	Diseño educativo															X	X	X	X	X	X	X	X							
	Tarjetas CRC		X						X																					
	Def. de la arquitectura	X				X																								
	Modelo de datos		X						X																					
Desarrollo	Producción																					X	X	X						
	Unidades de Prueba											X														X				
	Realización																										X			
Prueba	Implementación																											X		
	Pruebas de aceptación																												X	

Eje de evaluación

- Artefactos de la Rational Unified Process
- Artefactos de la Metodología de Diseño Instruccional de Software Educativos
- Artefactos de la Metodología eXtreme Programming
- Artefacto de Scrum



A continuación se detallan algunos artefactos indicados en la Tabla N° 1.

1. En la fase de Inicio, el documento Visión contiene:

a. Funcionalidades del Sistema:

- El Módulo de Evaluación del Proceso Formativo debe permitir determinar los logros del estudiante dentro del proceso de enseñanza-aprendizaje.
- La evaluación debe estar diseñada bajo un enfoque de procesos.
- El módulo de evaluación deberá contar con una serie de actividades que el alumno deberá resolver.
- Registrar la participación y las conexiones de forma automática, de manera que los profesores puedan acceder a tal información.
- El tema que se utilizará para lograr la evaluación del proceso formativo serán las operaciones de suma y producto de curvas en el plano cartesiano.
- Cualquier estudiante de Matemáticas I de la FACyT, puede hacer uso de este *software*.
- Los datos importantes de los usuarios serán sus nombres, apellidos, cédula, sección, año, carrera, *login* y *password*.

b. Usuarios del Sistema. Se distinguen dos tipos de usuarios:

Estudiante: Realiza las actividades prácticas planteadas en el *software* y revisa y analiza su proceso formativo.

Profesor: Revisa y analiza la evaluación formativa de los estudiantes, supervisa y actualiza el material práctico que se encuentra en el *software* educativo.



c. Involucrados en el Sistema (Stakeholders). Se tienen los siguientes:

Experto en Contenido: Profesional representado por el docente de la Universidad.

Decide lo que el alumno debe saber, construye las actividades que se han de realizar para construir el conocimiento, analiza las capacidades de los estudiantes, diseña las estrategias para comunicar al alumno la información, elabora las pruebas necesarias para conocer si el proceso de comunicación ha sido satisfactorio. Establece qué criterios desea conocer en la evaluación formativa.

WebMaster: Persona encargada de un sitio *web*. Decide las tecnologías que se van a usar, los servidores y la estructura a ser empleada en el sitio donde se presenta el producto.

Programador: Persona encargada del desarrollo y creación del *software*. Creación e implementación de código fuente para el desarrollo del *software*.

Diseñador Instruccional: Persona que hace accesible el conocimiento a los participantes, a través de las estrategias y los medios óptimos para lograr un proceso de instrucción satisfactorio y significativo para los alumnos. Estructuración de los contenidos, estrategias de enseñanza y evaluación, objetivos, secuencias instruccionales y distribución de actividades, que deben ser incluidos en el producto.

d. Necesidades de Usuarios: El estudiante de FACYT necesita una herramienta computacional que le permita conocer su proceso formativo y reflexionar sobre sus propias dificultades y aciertos, para procurar ser consciente del proceso que ha seguido durante su aprendizaje, facilitando la autoevaluación. Los profesores por su parte necesitan un instrumento que facilite la evaluación de su práctica docente, para la incorporación de novedades adecuadas a la ayuda pedagógica.

e. Características del Sistema:

- Presentación de *applet* para la graficación a mano alzada.



- Emisión de reportes de evaluación al estudiante.
- Emisión de reportes al profesor.
- Interfaz para la elaboración de preguntas al profesor.
- Interfaz para consultar las preguntas frecuentes.

g. Restricciones Principales:

- Restricciones de la plataforma de *software*
- Restricciones de la plataforma *hardware*

2. En la fase de planificación se ilustra una historia de usuario:

Historia del Cliente: *Marlene Arias*

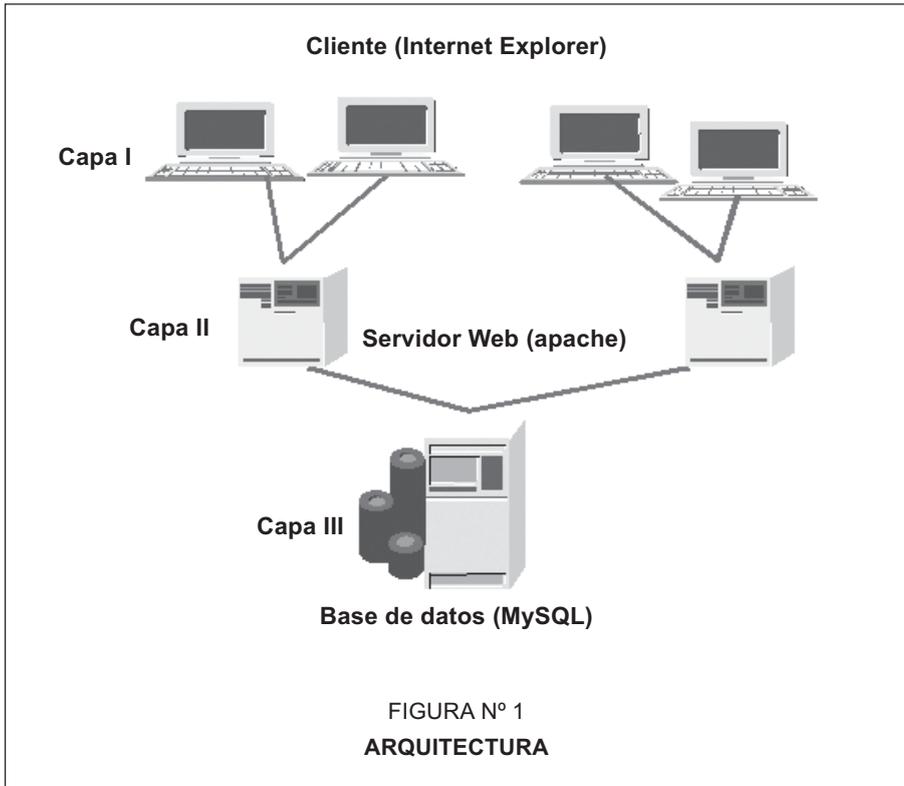
Fecha: <u>29/05/04</u>	Tipo de Actividad:	Prioridad:
Número de Historia: <u>3</u>	Nueva: <u>X</u>	Alta: <u>X</u>
Referencia Anterior: _____	Arreglar: _____	Media: _____
	Mejorar: _____	Baja: _____

Descripción de la Tarea:

Posibilidad de desplegar varios objetos simultáneamente, posibilidad de dibujar curvas a mano alzada, capacidad de modificar curvas a mano alzada.

3. En la fase de diseño se tiene la definición de la arquitectura mostrada en la figura N° 1, se detalla la arquitectura cliente-servidor de tres capas. La capa I representa el *software* cliente que recibe los modelos de presentación desde el servidor y los presenta al usuario, para que éste pueda manipular la información que hay en el sistema y tomar acciones

sobre la misma. La capa II utiliza un patrón de diseño conocido como MVC (*Model-View-Controller*). La capa III tiene como misión la administración de la información que maneja el sistema.



A continuación, en las figuras Nos. 2 y 3 se muestran algunas de las interfaces del *software*:

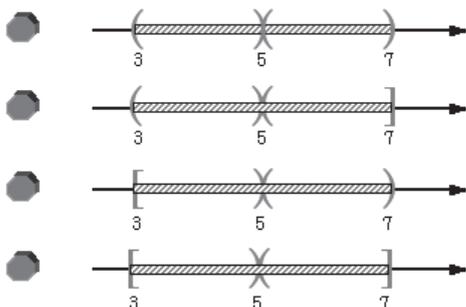


INTERVALOS, PARES ORDENADOS Y PUNTOS DEL PLANO

A continuación se presenta un conjunto de preguntas.

Cada una tiene cuatro opciones, haz click sobre la que consideres correcta.

La representación gráfica del intervalo $[3,7] \setminus \{5\}$ en la recta es:



Siguiente

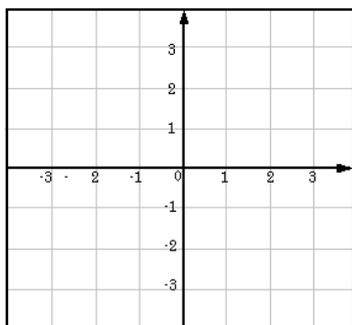
FIGURA N° 2

EJEMPLO 1 DE INTERFAZ

INTERVALOS, PARES ORDENADOS Y PUNTOS DEL PLANO

A continuación se presenta el plano cartesiano.

Haz clic en la ubicación en el plano que corresponde al punto señalado.



Punto $(-1,1)$

Siguiente

FIGURA N° 3

EJEMPLO 2 DE INTERFAZ



Conclusiones

Las conclusiones de esta investigación indican que al utilizar el *software* de evaluación *online* bajo el enfoque de procesos, el alumno conoce sus fallas y aciertos durante la construcción de su conocimiento. La herramienta computacional, permite al alumno realizar un seguimiento de su aprendizaje de forma autónoma, libre y continua; y a su vez le posibilita aprender de sus errores, facilitando el proceso de aprendizaje.

Adicionalmente, el profesor puede evaluar su práctica docente, así como también obtener estadísticas con respecto al proceso de aprendizaje de sus estudiantes.

La propuesta metodológica creada en este trabajo combina varias metodologías, entre ellas una para diseñar materiales educativos y otra para el desarrollo de *software*, y se unen a ellas artefactos del proceso de desarrollo de *software* de *Rational*, garantizando de esta manera la producción de *software* educativos de alta calidad.

El desarrollo de este *software*, es el comienzo para la elaboración de otros módulos de evaluación en otras áreas o unidades académicas.

Se sentaron las bases teóricas de la evaluación, de servir de soporte a futuros trabajos de investigación en esta área.

Finalmente, este trabajo representa un aporte a la comunidad científica porque permite combinar metodologías de dos áreas de conocimiento: computación y educación, para la producción de un *software* de calidad en el área de la evaluación. Además, contribuirá a desarrollar más eficientemente el proceso de enseñanza-aprendizaje, con la utilización del *software* para la evaluación bajo enfoque de procesos.



Referencias bibliográficas

- AHUMADA, P. (2001). *La evaluación en una concepción de aprendizaje significativo* (pp. 24-25). Chile: Ediciones Universitarias de Valparaíso de la Universidad Católica de Valparaíso.
- AHUMADA, P. Estrategias y procedimientos para una evaluación auténtica de los aprendizajes en la enseñanza universitaria. Extraído el 2 de noviembre de 2006 desde www.depcuadernos.net/interface/asp/web/leer_articulo.asp?ArticleID=104
- ALCALDE, L. y SALAS, A. (2003). *Software Educativo en la Web para la Enseñanza – Aprendizaje de Curvas en el Plano Cartesiano en la Asignatura Matemáticas I de la Facultad Experimental de Ciencias y Tecnología*. Valencia: Universidad de Carabobo.
- ARIAS, M. (2001). *Desarrollo de Lineamientos Generales para el Diseño Instruccional de Software Educativos*. Valencia: Universidad de Carabobo.
- BLANCO, O. (2004). "Tendencias en la evaluación de los aprendizajes". *Revista de Teoría y Didáctica de las Ciencias Sociales*. 9, 111-130.
- BOURAS, C.; LAMPASAS, P. and SPIRAKIS, P. (2002). "Steps: Supporting Traditional Education Procedures – A TCP/IP Multimedia Networks-Based Model". *Multimedia Tools and Applications*. 16, 251-276.
- CHARTE, F. (2002). Programación extrema. Extraído el 7 de febrero de 2006. http://www.windowstimag.com/atrasados/2002/63_abr02/articulos/observa_extrema.asp
- DUFFY, T. & CUNNINGHAM, D. (1996). "Constructivismo: implications for the design and delivery of instruction". En D. Jonassen (Eds.). *Handbook of Research for Educational Communications and Technology* (pp. 171). New York, USA: Macmillan Library Reference USA.
- LETELIER, P. y SÁNCHEZ, E. (2003). Metodologías Ágiles en el Desarrollo de Software. Extraído el 31 de mayo de 2005. <http://issi.dsic.upv.es/tallerma/actas.pdf>
- KRUCHTEN, F. (2001). *The rational unified process, an introduction* (pp. 139-206), segunda edición. United States of America: Addison Wesley.
- PARZINGER, M.; NATH, R. and LEMONS, M. (2001). "Examining the Effect of the Transformational Leader on Software Quality". *Software Quality Journal*. 9, 253-267.



- PUJOL, L. (1999). "Los hipermedios como herramienta para facilitar el aprendizaje significativo: una perspectiva constructivista". *Revista Agenda Académica* 6 (2), 125-134.
- SALCEDO, P. (2002). "Ingeniería de *Software* educativo, teorías y metodologías que la sustentan". Extraído el 25 de septiembre de 2005.
<http://www.inf.udec.cl/revista/ediciones/edicion6/isetm.PDF>
- SHIH-CHUNG, L. (2001). "Development of instructional strategy of computer application software for group instruction". *Computers & Education*. 37, 1-9.
- SOSA, R.; GARCÍA, A.; SÁNCHEZ, J.; MORENO, P. y REINOSO, A. (2005). "B-Learning y Teoría del Aprendizaje constructivista en las Disciplinas Informáticas: Un esquema de ejemplo a aplicar". *Recent Research Developments in Learning Technologies*. 2, 647-648.