

Ingeniería de software aplicada a la educación en el área de matemáticas

José Antonio González López

Instituto Tecnológico de la Laguna

jose_ant_gzz@hotmail.com

Dr. Enrique Cuan Duron

Instituto Tecnológico de la Laguna

kcuan@gmail.com

Dra. Elisa Urquizo Barraza

Instituto Tecnológico de la Laguna

elisaurquizo@gmail.com

M.C. José Ruiz Ayala

Instituto Tecnológico de la Laguna

jjruizad@yahoo.com.mx

Dr. Diego Uribe Agundis

Instituto Tecnológico de la Laguna

diegouribe@acm.org

Resumen

En este trabajo se presenta el uso de *Unity* como Ambiente de desarrollo integrado. Aplicando el Proceso Unificado se está desarrollando un software que ayudará a los niños en el área de matemáticas a nivel primaria para que por medio de actividades escolares como: juegos, puzzles, ejercicios, etc., estos se muestran de forma atractiva para captar y mantener su atención, estas actividades son acordes al plan de estudios vigente por la Secretaria de Educación Pública de México, dicho software consta de dos

fases, una demostrativa donde se mostrará el proceso para solucionar algún problema y el de ejercicios donde el alumno realizará y verificará su respuesta.

Palabras Clave: matemáticas, proceso unificado, software.

Abstract

In this assignment we present the using of Unity as Integrated Development Environment and apply Unified Process, we are developing a software, this will help children in the elementary school in mathematics to learn by games, puzzles, exercises, etc. this will be displayed in attractive way to keep their attention, these activities are according to lesson plan used actually by Secretaría de Educación Pública in México, this software are in two phases, the first one shows the process that how the problem is solved and in the second the student will solve the problem by itself then check its answer.

Keywords: *mathematics, software, unified process.*

1. Introducción

La Secretaria de Educación Pública (SEP) aplica anualmente un examen estandarizado llamado ENLACE en las escuelas primarias y secundarias de sostenimiento público y privado en las 32 entidades federativas del país.

Esta prueba tiene como objetivo evaluar los conocimientos en las asignaturas de español, matemáticas, formación cívica y ética, historia y geografía siendo las materias de español y de matemáticas las materias bases que se evaluarán cada año y las demás se estarán rotando un año cada una.

La siguiente tabla es un extracto de las estadísticas a nivel nacional del año 2006 al 2013 en el que muestran los resultados globales que incluye al CONAFE, escuelas

generales, indígenas y particulares, se aprecia que la mayor parte de la población de los alumnos a nivel primaria se encuentra en el rango de elemental, esto significa que el alumno requiere fortalecer la mayoría de los conocimientos y desarrollar las habilidades de la asignatura evaluada. (SEP, 2015)

Tabla 1. Estadísticas de la prueba ENLACE del área de matemáticas.

PRIMARIA MATEMÁTICAS			GLOBAL				
GRADO	ENTIDAD	AÑO	INSUFICIENTE	ELEMENTAL	BUENO	EXCELENTE	ALUMNOS
GLOBAL	NACIONAL	2006	21.0	61.4	16.0	1.6	7,506,255
		2007	20.2	57.5	19.0	3.3	7,962,825
		2008	22.8	49.5	23.0	4.7	8,108,694
		2009	20.3	48.6	24.9	6.1	7,810,073
		2010	19.7	46.4	25.8	8.1	8,323,728
		2011	16.5	46.5	25.9	11.0	8,631,091
		2012	13.0	42.7	27.1	17.3	8,141,643
		2013	12.4	38.8	29.0	19.8	8,463,445

El propósito de desarrollar software educativo es mejorar el estatus de los alumnos en la prueba ENLACE, ya que la mayor parte de la población se encuentra en el nivel elemental.

El alumno al usar el software con actividades específicas para cada tema del área de matemáticas y al practicarlo constantemente reforzara su conocimiento obtenido en clase e incrementara la velocidad en que resuelve estos problemas y por lo tanto mejorará sus calificaciones.

Este software educativo será elaborado siguiendo el proceso unificado de desarrollo de software que es un marco de desarrollo que nos muestra las etapas y tareas agrupadas en etapas que en conjunto se le nombra ciclo de vida.

2. Métodos

El desarrollo de los módulos del software se basó en el plan de estudios vigente por la SEP, se presenta en dos partes; la demostración y ejercicios, como se muestra en la siguiente figura.

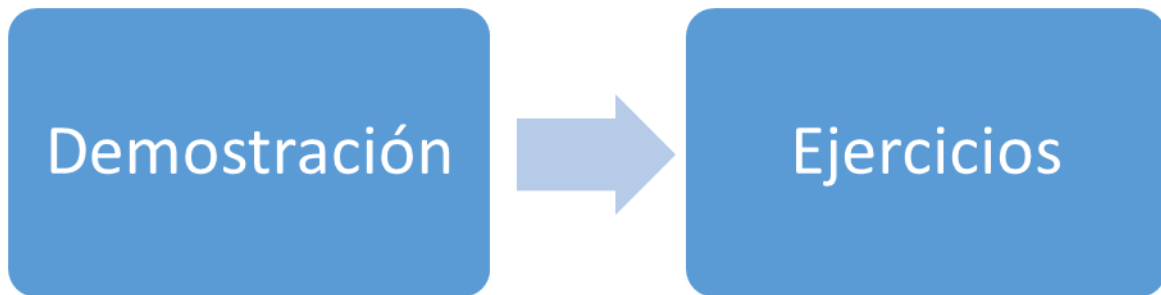


Figura 1. Diagrama de demostración de módulos genérico de trabajo.

En la primer parte de demostración se exhibe la ejecución practica de cómo se realiza cada tema especificado en el plan de estudios vigente. En la segunda parte de ejercicios se mostrarán problemas con cifras aleatorias y el alumno realizará las operaciones pertinentes de forma manual para posteriormente ingresar la información necesaria que será evaluada por el software dando un mensaje positivo o negativo respecto a la respuesta.

Para el desarrollo del software se utilizó el proceso unificado que a continuación se describe.

El proceso unificado es un proceso de desarrollo de software que comprende un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Sin embargo, además el proceso unificado es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema de software en construcción está conformado por componentes software interconectado 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 de software. De hecho, UML es una parte esencial del proceso unificado, sus desarrollos fueron paralelos.

Los verdaderos aspectos definitorios del Proceso Unificado se resumen en tres fases claves, dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental. Esto es lo que hace único al Proceso Unificado. (Jacobson Ivar, 2000)

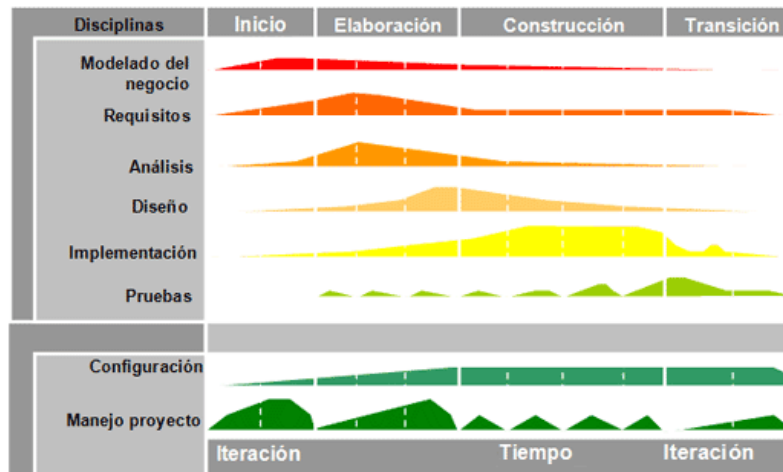


Figura 2. Proceso Unificado.

El proceso unificado va iterando en una serie de fases, conforme van creciendo los periodos se va creando el ciclo de vida de un sistema, en cada fase se va generando una versión del sistema, esta característica es la de iterativa e incremental, ya que cada vez que se va agregando una iteración más el sistema va adquiriendo más características hasta llegar al fin y concluir el proceso de creación de software.

En las diferentes fases se mostrarán solo algunos de los artefactos utilizados en cada una de ellas.

Fase de Inicio: La primera fase es la de inicio, en esta fase es cuando el proyecto nace, aquí es donde se plantean las siguientes preguntas: ¿Cuál es la visión y el análisis del negocio para este proyecto?, ¿Es viable?, ¿Comprar o construir?, Estimación aproximada del costo ¿Cuesta 10K o 100K millones de pesos?, ¿Deberíamos abordar o no seguir? (Craig, 2003)

Los artefactos que se usaron en esta fase son: Visión y Análisis del Negocio, Modelo de Casos de Uso, Especificaciones Complementarias, Glosario, Lista de Riesgos y Plan de Gestión de Riesgos para darle respuesta, Plan de Iteración, Plan de desarrollo de Software, Marco de Desarrollo, etc. Estos artefactos han sido elaborados parcialmente en esta fase y en especial en la primera iteración, conforme se dieron las demás iteraciones se fueron completando y corrigiendo.

Enseguida se muestran tres artefactos típicos de la fase de inicio en la disciplina de requisitos.

Primer artefacto “Visión”: Es una descripción positiva y breve de lo que el sistema de software espera que haga y pueda alcanzar de manera exitosa.

Visión: Se emplea el software educativo en las escuelas primarias, dicho software ha sido un apoyo para que los alumnos mejoren sus calificaciones.

Segundo artefacto “Glosario”: Es la terminología del dominio, en las palabras del cliente, esto es fundamental al momento de realizar el software, ya que una palabra puede tener diferentes significados y este servirá para el correcto entendimiento entre el cliente y el equipo de desarrollo.

Glosario

- Material concreto: Son objetos reales que representan físicamente a un algoritmo.
- Algoritmo: Es una o varias operaciones que se realizan para resolver un problema matemático.
-etc.

Tercer artefacto “Caso de Uso de Alto nivel”: Tiene como objetivo representar una unidad funcional de un sistema, en este artefacto pueden interactuar uno o más Actores. Para realizar los casos de uso es necesario conocer a los Actores y sus objetivos o necesidades dentro del negocio. Dentro de la fase de inicio se usan en mayor medida los casos de uso de alto nivel.

Casos de uso de alto nivel.

Caso de Uso: CU 01-Situar Fracción en la recta numérica.

Actores: Alumno.

El alumno elige “Situar Fracciones en la recta numérica”, el sistema le da un numero fraccionario y una recta numérica, el Alumno sitúa la fracción en la recta numérica, indica que ha finalizado, el sistema verifica su respuesta, el sistema muestra el mensaje de que ha finalizado satisfactoriamente

Fase de Elaboración

Durante la fase de elaboración se especifican detalladamente la mayoría de los casos y de diseña la arquitectura del sistema. La relación entre la arquitectura del sistema y el propio sistema es primordial. Haciendo una analogía la arquitectura es el esqueleto del sistema, en cuanto al manejo del proyecto se planificaron las actividades y realizó la estimación de recursos necesarios para terminar el proyecto.

La arquitectura son los elementos fundamentales del sistema: Subsistemas, Dependencias, Interfaces, Colaboraciones, Nodos, etc.

Los artefactos creados en la fase de inicio son refinados, Modelo de Caso de Uso, Visión, Especificaciones Complementarias, Glosario.

Otras son creadas: Modelado de Negocios, Modelo de Diseño, Modelo de Datos, Pruebas, etc.

Enseguida se muestra un artefacto típico de la fase de elaboración en la disciplina de requisitos.

Artefacto “Casos de Uso en su forma extensa”

Este artefacto fue creado basado es el mismo caso de uso que fue creado en la fase de inicio, la diferencia de este con el anterior es que este fue refinado, extendiendo sus características como a continuación se muestra.

Tabla 1. Detalle de casos de uso.

CFaD	Situar Fracción en Recta Numérica		
Versión	0.01		
Autores	José Antonio González López		
Fuentes			
Objetivos asociados			
Interesados	Estudiantes: Quieren que sea fácil de usar e intuitivo, llamativo, entretenido.		
Descripción	Permite al Estudiante situar una fracción en la recta numérica.		
Precondición	Estar autenticado en el sistema.		
Secuencia Normal	Paso	Acción del Actor (o intención)	Responsabilidad del Sistema
	1	El Estudiante elige “Situación Fracción en Recta Numérica”	
	2		El sistema muestra un número fraccionario.
	3		El sistema muestra una recta numérica.
	4	El Alumno sitúa la fracción en la recta numérica.	
	5		El sistema verifica la respuesta.
	6		El sistema muestra el mensaje de ejercicio concluido satisfactoriamente.
Postcondición	El Alumno respondió correctamente y se mostró un mensaje que indica que se concluyó correctamente.		

Tabla 1. Detalle de casos de uso (continuación).

Camino Alternativo	Paso	Acción
	1	
	2	
	3	
	4	a) El alumno no sitúa la fracción en la recta.
	5	
6		a) El sistema muestra un mensaje que indica que no se situó en la parte correcta. Volver al 4.
Rendimiento	Paso	Cota de tiempo
Frecuencia esperada		
Importancia		
Urgencia		
Comentarios	Los campos dejados en blanco se consultaran con un especialista, para una versión posterior se ingresen.	

Fase de Construcción.

Durante esta fase es donde se creó el producto de software, la arquitectura propuesta crece hasta llegar al sistema completo, al esqueleto que se construyó en la fase pasada se le añadió músculos que es el software, al finalizar esta fase los casos de uso son implementados (codificados), sin embargo estos pueden tener defectos. En esta fase los requisitos y la arquitectura son estables.

En esta fase de desarrollo se refinaron las siguientes disciplinas: Modelado del Diseño, Documento de Arquitectura de Software, Modelo de Datos, Modelo de Implementación, Plan de Desarrollo de Software y Modelo de pruebas.

Enseguida se muestra un artefacto típico de la fase de implementación en la disciplina de requisitos.

Codificación.

La codificación es escribir el código fuente de cada módulo.

```
class Rectanumerica
{
    float puntoReal;

    float puntoAlumno;

    public void setpuntoReal(float)...

    public float getpuntoReal()...

    public void setpuntoAlumno(float)...

    public float getpuntoAlumno()...

    public bool verificar()...
}
```

Figura 3. Ejemplo de Codificación.

Aquí en esta parte es donde se utiliza Unity es una plataforma de desarrollo flexible y poderosa para crear juegos y experiencias interactivos 3D y 2D multiplataforma.



Figura 4. Ambiente Integrado de Desarrollo de Unity.

En la figura 4 se muestra el IDE de desarrollo de Unity, este es amigable, ya tiene diseñado e implementadas algunas funciones como la física, y movimiento, solo es necesario personalizarlas. Igual como otros IDEs de desarrollo Unity tiene los elementos básicos como cuadros de texto, listas y botones. A continuación se muestra un pequeño bloque de código en un botón, en dicho botón.

```
var btnTexture : Texture;
function OnGUI() {
    if (!btnTexture) {
        Debug.LogError("Please assign a texture on the inspector");
        return;
    }
    if (GUI.Button(Rect(10,10,50,50),btnTexture))
        Debug.Log("Clicked the button with an image");
    if (GUI.Button(Rect(10,70,50,30),"Click"))
        Debug.Log("Clicked the button with text");
}
```

Figura 5. Ejemplo de código en C# en Unity.

Fase de transición

En esta fase es en la que el producto se convierte en la versión beta. En esta versión un número pequeño de usuarios prueban el producto hasta este punto pensado que es el final, este grupo informa al equipo de desarrollo los defectos y deficiencias que han detectado, el equipo de desarrollo corrige estos defectos y deficiencias e implementan mejoras.

Una parte importante durante esta fase es la capacitación de los clientes en el uso del software, este software al ser hecho para niños de primaria que van de los 7 a los 11 años de edad debe de ser muy intuitivo y no por ello se debe de realizar la ayuda y asistencia.

3. Resultados

Este software es un complemento para la comprensión de la materia de matemáticas a nivel primaria. Con las clases del docente y el software este les ayudará para elevar sus

promedios, sus marcas en la prueba enlace, también con el conocimiento obtenido podrá ser usado en su vida cotidiana y en su futuro académico.

4. Bibliografía

Craig, L. (2003). *UML y Patrones Una introducción al análisis y diseño orientado a objetos y al proceso unificado* (Segunda Edición ed.). Madrid: Pearson Education.

Jacobson Ivar, B. G. (2000). *El Proceso Unificado de Desarrollo de Software*. Madrid: Pearson Educacion.

SEP. (20 de Septiembre de 2015). *ENLACE*. Recuperado el 20 de Septiembre de 2015, de ENLACE: <http://www.enlace.sep.gob.mx/>