

ARQUITECTURA DE SOFTWARE ORIENTADA A LA CREACIÓN DE MICROMUNDOS PARA LA ENSEÑANZA Y EL APRENDIZAJE

Lugo Manuel Barbosa Guerrero¹

RESUMEN

Quando se inicia el proyecto de la creación de una "Arquitectura de software orientada a la creación de micromundos para la enseñanza y el aprendizaje", se piensa diseñar una nueva arquitectura de software, tomando como base un patrón de software para que sirva como modelo en la creación de los diferentes desarrollos de software de micromundos en el programa de ingeniería de sistemas de la Universidad Libre - Seccional Bogotá.

Una arquitectura de software consiste en la estructura y comportamiento que debe tener desde el más pequeño programa hasta el más grande; esto igualmente contribuye a su clasificación; por lo tanto se hace posible el estudio de sistemas ya implementados, así como el desarrollo de nuevos, es cuando se trata el tema de los diferentes niveles de abstracción de la funcionalidad de los sistemas, los cuales están asociados con diferentes aspectos y componentes de su "arquitectura de software".

PALABRAS CLAVE

Arquitecturas de software, Framework, componentes, micromundos, Rational Unified Process, módulos.

ABSTRACT

When Initiating the project of the creation of a "Architecture of software oriented to the microworld creation for education and the learning", thought about designing a new architecture of software, taking as it bases a software pattern so that it serves like model in the creation of the different developments from microworld software in the program of engineering of systems of the University Libre - Seccional Bogotá.

A software architecture consists of the structure and behavior that must have from the smallest program to greatest; this also contributes to its classification; therefore it becomes the study of systems already implemented as well as the development of new possible, is when the subject is the different levels from abstraction of the functionality of the systems, which are associate with different aspects and components from his "architecture of software".

KEY WORDS

Architectures of software, Framework, components, microworlds, Rational Unified Process, modules.

Fecha de recepción del artículo: 8 de abril de 2008.

Fecha de aceptación del artículo: 9 de mayo de 2008.

¹ Docente Investigador. Universidad Libre – Seccional Bogotá. Facultad de Ingeniería. Departamento de Ingeniería de Sistemas.

INTRODUCCIÓN

En los procesos de enseñanza – aprendizaje se pueden llegar a presentar problemas como la falta de recursos y el desconocimiento del manejo de las tecnologías de información, lo cual puede obstaculizar el acceso a la educación de algunas personas que desean adquirir conocimiento.

Igualmente puede dificultar el aprendizaje en los estudiantes por las metodologías utilizadas dentro de los programas académicos; al ser el aprendizaje una actividad circunstancial para el ser humano, es necesario implementar herramientas que permitan tanto al estudiante desarrollar el autoaprendizaje como a los docentes brindarles un complemento en el proceso de enseñanza. Se pueden describir otros problemas como son:

- **Curriculos rígidos en contravía a la creatividad y manejo de las TICs:** Los avances en las tecnologías de la información y las comunicaciones – TICs, exigen cambios en los modelos educativos ya que nos encontramos ante una nueva generación de estudiantes que manejen las TICs de una buena manera.
- **Desaprovechamiento del potencial de los estudiantes:** Las instituciones educativas no cuentan en ocasiones con los recursos tecnológicos o no los aprovechan cuando los poseen, y esto conlleva a que los actores más afectados con este problema sean los estudiantes; es cierto que las instituciones que no incorporan las TICs en su currículo de los diferentes programas de pregrado y postgrado, desaprovechan el potencial que tienen las nuevas generaciones en su manejo.
- **Analfabetismo digital:** El analfabetismo digital es una de las características de los países en vía de desarrollo, lo cual genera una desigualdad científica y tecnológica que conlleva problemas de mayor escala en el contexto social, cultural, económico y educativo del país.

Preguntas de Investigación

- ¿Qué habilidades de pensamiento desarrollan los estudiantes al aprender con los micromundos?

- ¿Cómo se relacionan los estudiantes con los micromundos interactivos?
- ¿Cuáles son los efectos de los micromundos interactivos en la enseñanza / aprendizaje de los estudiantes de ingeniería de sistemas?
- ¿Cuál es el proceso para construir micromundos de tipo interactivos?
- ¿Qué tipos de aprendizaje se aplican al interactuar con micromundos de tipo interactivos?

Desde el principio se trabajó como un objetivo principal poder diseñar e implementar una nueva arquitectura de software que se utilice como modelo en el desarrollo de micromundos para la enseñanza – aprendizaje en el programa de ingeniería de sistemas de la Universidad Libre.

De allí se desprende que se tenía que trabajar inicialmente en:

- **Seleccionar el Design Patterns adecuado que contribuya en la creación de la arquitectura de software.**
- incrementar la articulación entre las asignaturas del programa.
- Promover el aprendizaje colaborativo.
- Promover procesos de enseñanza/aprendizaje y la relación con el medio.
- Investigar acerca de los factores que intervienen en la creación de este tipo de herramientas.

1. LA FUNDAMENTACIÓN TEÓRICA DE UN MICROMUNDO

- El micromundo es una herramienta de ambientes pedagógicos que facilita el aprendizaje mediante la simulación de un mundo real; por lo tanto se convierte en una realidad construida en un software educativo.
- El micromundo va a facilitar la creatividad, ya que desarrolla un pensamiento lógico y permite la creación de ambientes lúdicos, creativos, colaborativos e interactivos, entre otros basados en el desarrollo de educación constructivista.

- Es un ambiente en el cual los estudiantes pueden explorar y probar sus ideas ya que en algunos pueden crear simulaciones.
- Un concepto claro de lo que es un micromundo es el tomarlo como una herramienta de ambientes pedagógicos que facilita el aprendizaje mediante construcciones complejas hasta llegar a convertirse en una herramienta informática específica. El micromundo debe facilitar en el usuario la creatividad y el aprendizaje, debido a que debe ser implementado para desarrollar pensamientos críticos y pensamientos lógicos; permite la creación de ambientes colaborativos e interactivos, entre otros basados en el desarrollo de la educación constructivista.

2. ARQUITECTURA MICRO_LIBRE BASADA EN COMPONENTES

La arquitectura de un micromundo debe contener elementos que representen qué se pretende enseñar,

a quién se va a enseñar y cómo; por ello los elementos que forman parte de la arquitectura Micro_Libre se pueden clasificar en dos grandes grupos:

- Los que contienen conocimiento y los que utilizan ese conocimiento.

La arquitectura es necesaria para que se puedan tomar las decisiones pertinentes en cada momento de la instrucción; además esta arquitectura Micro_Libre estará basada en un núcleo que planificará las instrucciones sobre la base del conjunto de componentes externos (conocimiento pedagógico).

El modelo conceptual del micromundo es el que define los conocimientos y actitudes del alumno y el conocimiento del cómo se va a enseñar está repartido entre el planificador de actividades y los componentes de contenidos como se aprecia en la figura 2.

A continuación se describen algunos módulos de la arquitectura de software:

Figura 1. Modelo de un micromundo tomando como base una arquitectura de software basada en componentes (Autor)



- **Componentes de contenidos:** Contiene todos los módulos del micromundo junto con las actividades y competencias por adquirir.
- Un modelo parcial del dominio, modelo interno que se puede poner en correspondencia con el modelo de Micro_Libre.
- Una interfaz de desarrollo que será utilizada por los profesores para autorizar la introducción de los nuevos contenidos.
- Una interfaz del alumno que se encargará de la comunicación con el alumno (presentación de actividades, sus respuestas, etc.).
- La definición del componente que describe el funcionamiento del componente.

2.3 Modelo conceptual del dominio de la arquitectura propuesta

- El modelo conceptual del dominio representa el conocimiento de la asignatura que se pretende enseñar. Existen diferentes enfoques que dependen principalmente de la naturaleza del dominio por representar.
- El Micro_Libre se centra en la representación de dominios declarativos. El modelo más extendido para este tipo de dominios es de las redes semánticas de unidades de conocimiento, que

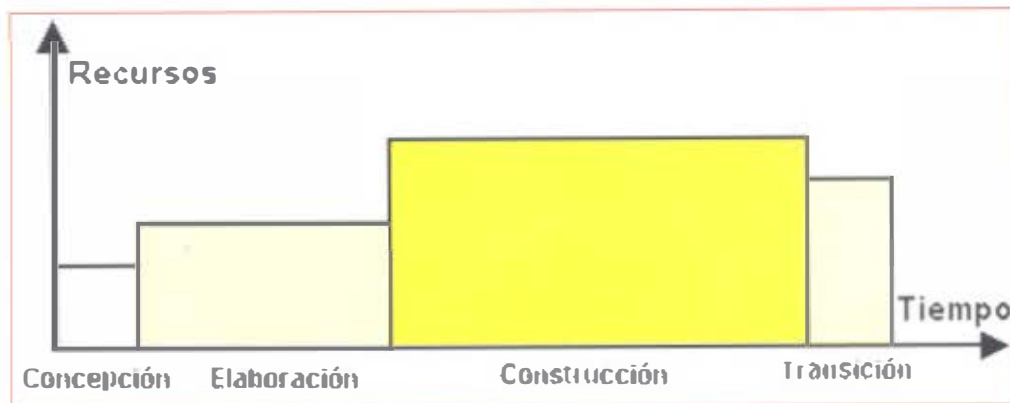
ha sido usado, entre otros, en los sistemas de autor.

- A la hora de modelar un área de conocimiento para un micromundo no se debe perder de vista la finalidad pedagógica del mismo.
- Esto influirá en dos sentidos: por un lado en la granularidad de la red, es decir, decidir hasta qué punto se debe descomponer las unidades de conocimiento en unidades más simples.
- Los elementos básicos del modelo del dominio en Micro_Libre son los conceptos y las relaciones. El dominio se define por:
 - Una red semántica de conceptos que describe las relaciones entre ellos y
 - Conocimiento pedagógico necesario para la instrucción que incluye una descripción informal de la semántica de las relaciones, y para cada concepto un tipo de evaluación y la nota mínima necesaria para considerar superada la actividad.

3. METODOLOGÍA UTILIZADA EN LA ARQUITECTURA PROPUESTA

Es necesario el uso de una metodología con el fin de proveer un enfoque estructurado para realizar tareas y responsabilidades en un desarrollo: es necesario con

Figura 3. RUP (Rational Unified Process)



el fin de asegurar la producción de software de alta calidad, que cumpla las necesidades de sus usuarios finales, que sea realizado en las fechas acordadas y con el presupuesto disponible; dentro de todas las metodologías se elige la metodología RUP (Rational Unified Process) la cual se divide en cuatro (4) fases que dan un estándar para el análisis, implementación, desarrollo y documentación de sistemas orientados a objetos junto con el lenguaje UML (Unified Modeling Language).

3.1 El Ciclo de Vida

- **Concepción:** aquí empieza el proyecto con el desarrollo de los casos de uso y la identificación de riesgos.
- **Elaboración:** Elaboración de planes; se completan los casos de uso y se eliminan los riesgos.
- **Construcción:** desarrollo del proyecto y elaboración del manual de usuario, el cual estará dividido en varias iteraciones.
- **Implementación:** Instalación, entrenamiento de usuarios.

Este ciclo de vida que se desarrolla por cada iteración, es llevado bajo dos disciplinas:

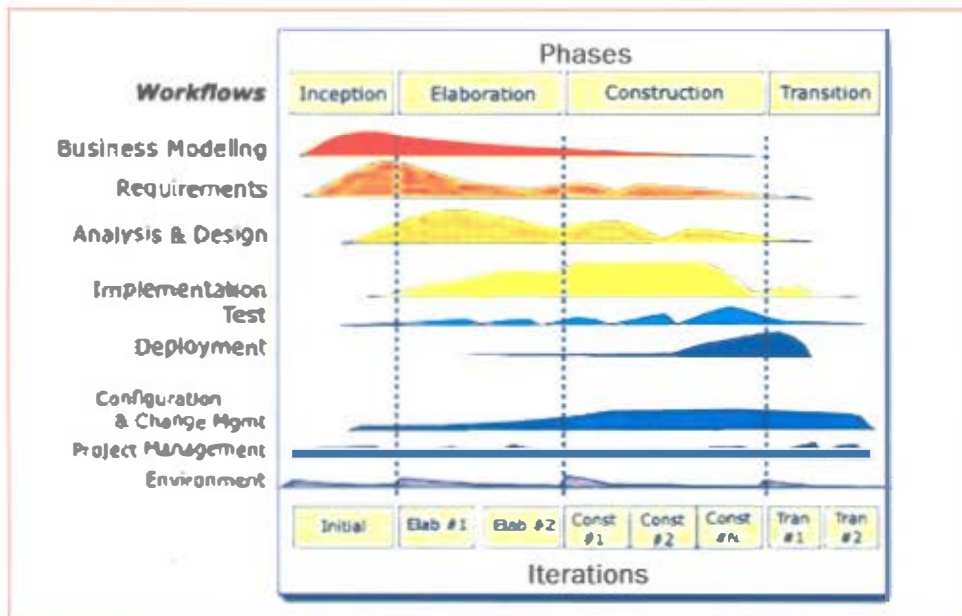
3.2 Disciplina de Desarrollo

- **Ingeniería de Negocios:** Entendiendo las necesidades del negocio.
- **Requerimientos:** Traslado de las necesidades del negocio a un sistema automatizado.
- **Análisis y Diseño:** Traslado de los requerimientos dentro de la arquitectura de software.
- **Implementación:** Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- **Pruebas:** Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

3.3 Disciplina de Soporte

- **Configuración y administración del cambio:** Guardando todas las versiones del proyecto.

Figura 4. Fases e Iteraciones de la Metodología RUP²



² Tomado de: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

- Administrando el proyecto: Administrando horarios y recursos.
- Ambiente: Administrando el ambiente de desarrollo.
- Distribución: Hacer todo lo necesario para la salida del proyecto.

Es recomendable que a cada una de estas iteraciones se la clasifique y ordene según su prioridad, y que cada una se convierta luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

Los elementos del RUP son:

- Actividades: Son los procesos que se llegan a determinar en cada iteración.
- Trabajadores: Vienen a ser las personas o entes involucrados en cada proceso.
- Artefactos: Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

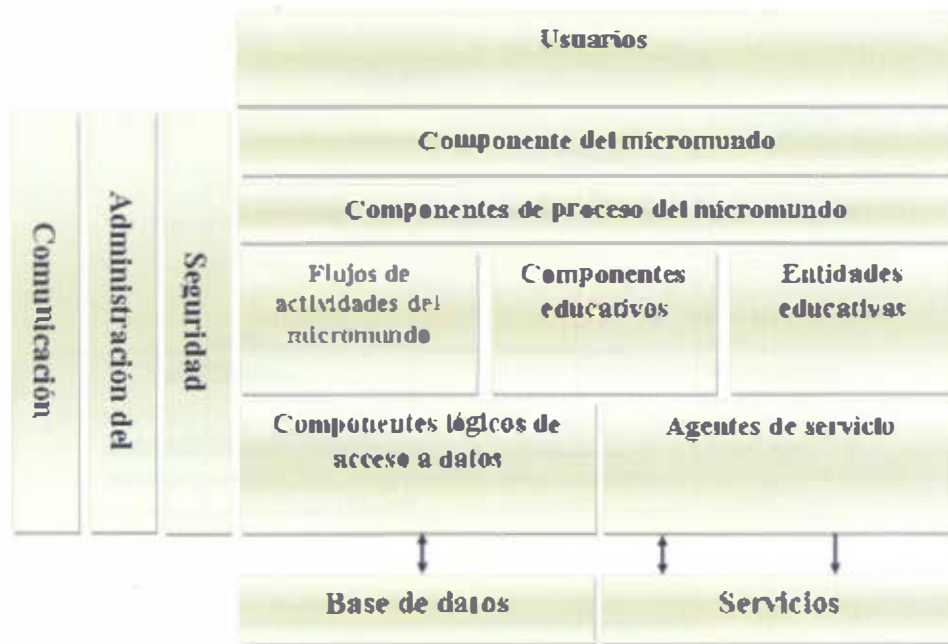
4. COMPONENTES

Los componentes de Software son todo aquel recurso desarrollado para un fin concreto y que puede formar solo o junto con otro/s un entorno funcional requerido por cualquier proceso predefinido. Son independientes entre ellos, y tienen su propia estructura e implementación³.

Un componente de software puede contener las siguientes características:

- Es una unidad ejecutable que puede ser instalada o implantada independientemente.
- Puede ser sujeto de composición por terceras partes; es decir, un desarrollador de software puede llegar y tomar el componente y agregarlo a

Figura 5. Componentes utilizados en el escenario de los micromundos



³ Tomado de: http://es.wikipedia.org/wiki/Componentes_de_software

lo que esté haciendo, o sea haría una composición de componentes.

- Un componente no tiene estado (al menos externamente visible).

Existen diversos modelos de componentes, los más conocidos son .NET, EJB (Enterprise Java Beans) y CCM (CORBA Component Model).

En la figura 5 se pueden apreciar los componentes sobre los cuales está soportada la nueva arquitectura de software.

CONCLUSIONES

La creación de nuevas arquitecturas de software durante los últimos años se ha dirigido a mejorar el proceso del software asociado a sistemas grandes y

muy complejos, ya que está comprobado que esas nuevas arquitecturas del software pueden desempeñar un papel vital en este proceso, pero que ha sido poco utilizada y subdesarrollada.

Utilizando componentes se obtienen ventajas ya que se puede explotar de una mejor manera la mediación entre el usuario y los desarrolladores de software y sobre todo en esta nueva arquitectura propuesta, ya que va orientada a la creación de micromundos para la enseñanza y el aprendizaje con módulos específicos.

Es muy importante la arquitectura de software ya que orienta al desarrollador y se pueden abstraer clases de componentes, métodos de composición, y esquemas de una variedad amplia de sistemas; estos corresponden algo a las nociones de los elementos del proceso y de datos, elementos que conectan, y del estilo arquitectónico, respectivamente.

BIBLIOGRAFÍA

DEWAYNE E., Perry, "Foundations for the Study of Software Architecture", AT&T Bell Laboratories / University of Colorado.

PHILIPPE, Kruchten, "Architectural Blueprints—The "4+1" View Model of Software Architecture", Rational Software Corp.

INFOGRAFÍA

María A. Mendoza Sánchez, "Metodologías De Desarrollo De Software", Consultada Febrero de 2008. http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html

Componentes de software: http://es.wikipedia.org/wiki/Componentes_de_software

Componentes de software: <http://jcastaneyra.wordpress.com/2006/07/26/componentes-de-software/>

Patrones de diseño: <http://www.thinkingInblue.com/?cat=53&paged=2>