

Nueva arquitectura de software micro_libre para la creación de micromundos para la enseñanza y el aprendizaje

*Ingeniero Lugo Manuel Barbosa Guerrero - Docente Investigador
Universidad Libre, Bogotá –Colombia*

Resumen

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: Currículos rígidos en contravía a la creatividad y manejo de las TICs, desaprovechamiento del potencial de los estudiantes y analfabetismo digital.

Objetivo general

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.

Objetivos específicos

- 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.

Desarrollo

A pesar de la existencia de diversas metodologías y plataformas de desarrollo de software educativo, el trabajo del programador se ve incrementado por una falta de arquitectura de software que oriente los procesos que debe integrar un software educativo (micromundo), actualmente las arquitecturas de software existentes no se enfocan en una metodología flexible para desarrollar software educativo. Esta ponencia socializa todo un trabajo de investigación que se ha venido llevando a cabo, en la que se muestra una serie de componentes que orientan el diseño para que sea funcional y se simplifique su desarrollo.

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 y por lo tanto 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 la educación constructivista. 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. Por ello se debe parametrizar los pasos para su construcción y contenido, y es por eso que debe existir una nueva arquitectura de software que oriente al diseñador y/o programador en este desarrollo.

Arquitectura Micro_Libre basada en componentes

La arquitectura de software para la creación de micromundos 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 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 estudiante, el conocimiento del cómo se va a enseñar está repartido entre el planificador de actividades y el componente de contenidos.

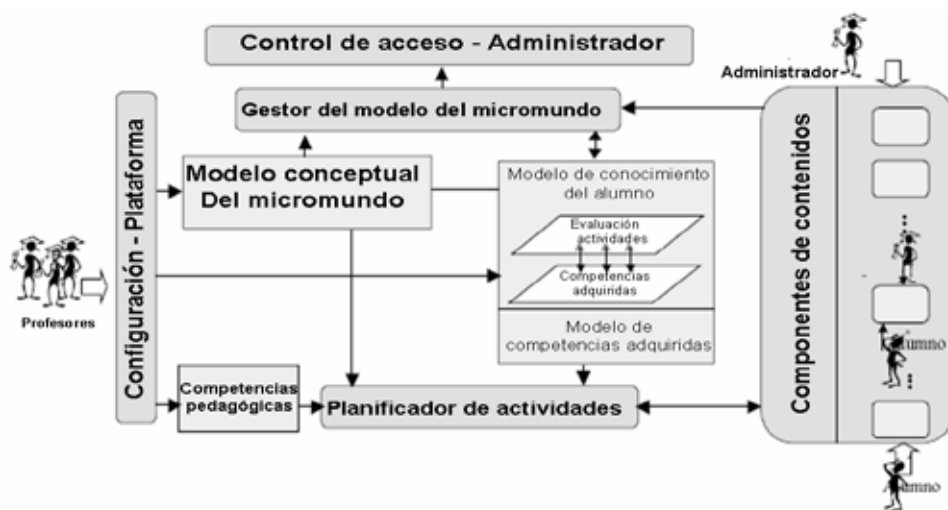


Figura 1. Prototipo de la arquitectura Micro_Libre basada en componentes

1. Módulo de conocimiento

Este módulo viene a estar comprendido por:

Modelo conceptual del dominio: contiene el conocimiento sobre la asignatura que el micromundo pretende enseñar.

Modelo de conocimiento del alumno: contiene información sobre los conocimientos que el alumno va adquiriendo durante el proceso de aprendizaje en el micromundo. Y está formado por dos capas: conocimiento evaluado y conocimiento adquirido.

Modelo de competencias adquiridas por el alumno: contiene información sobre las capacidades que el alumno va obteniendo que es fundamental para diseñar estrategias de enseñanza adecuadas.

Competencias pedagógicas: Son competencias pedagógicas asociadas a relaciones y conceptos propios de cada asignatura y/o profesor.

2. Módulo funcional

Este módulo viene a estar comprendido por:

Configuración / plataforma: este módulo permite a los profesores configurar y dotar de contenidos, con la ayuda del administrador, a los módulos de conocimientos del micromundo.

Gestor del modelo de micromundo: se encarga de manejar el modelo del micromundo.

Cada vez que se ejecuta un módulo, el gestor del modelo del micromundo se actualizará con la información de dicho módulo y basándose en los

resultados obtenidos por el alumno en la realización de la actividad encomendada.

Planificador de actividades: es el núcleo de la arquitectura. Se encarga de diseñar y componer las sesiones de aprendizaje, teniendo en cuenta las actividades. Para realizar su tarea se basa en los módulos de conocimiento del sistema.

Componentes de contenidos: Contiene todos los módulos del micromundo junto con las actividades y competencias a adquirir.

Arquitectura de un componente desde el punto de vista de Micro_Libre

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 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.

3. Modelo conceptual del dominio

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 a 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:

a) Una red semántica de conceptos que describe las relaciones entre ellos.

b) 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.

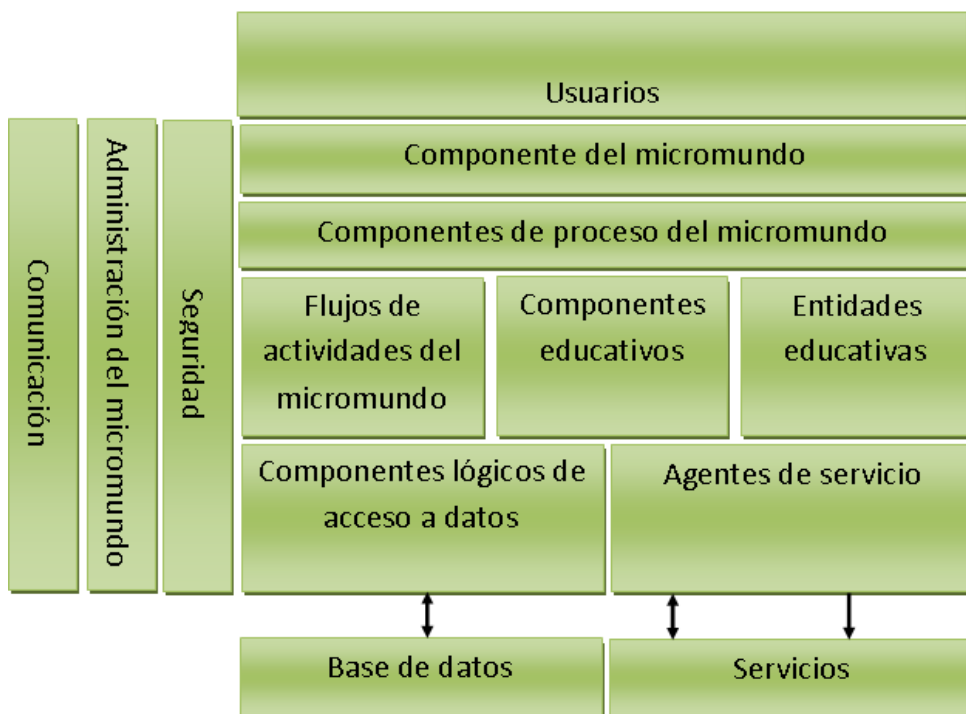


Figura 2. Componentes utilizados en el escenario de los micromundos

Los componente 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 implementación⁹⁹.

Un componente de software puede contener las siguientes características:

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

4. Desarrollo

El diagrama de despliegue del micromundo visualiza las relaciones físicas entre los componentes del hardware y el software en el software final, los cuales estarán formados por instancias de los componentes de la nueva arquitectura de software.

⁹⁹ Tomado de: http://es.wikipedia.org/wiki/Componentes_de_software

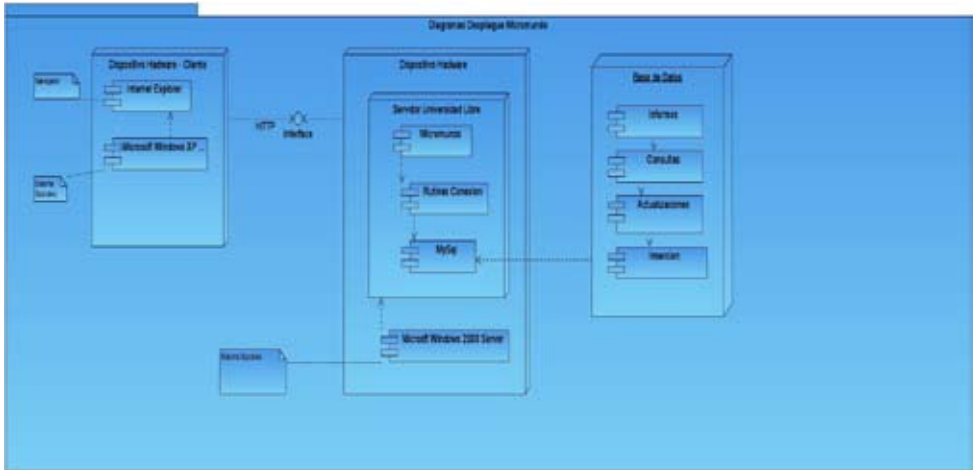


Figura 3. Diagramas Despliegue Micromundo

Rational Unified Process (RUP)

En la creación de esta nueva arquitectura de software se tiene en cuenta el Rational Unified Process ya que es una infraestructura muy flexible para el desarrollo de software, la cual proporciona una serie de prácticas recomendadas probadas y además una arquitectura configurable. Además permite seleccionar fácilmente el conjunto de componentes de la arquitectura.

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.

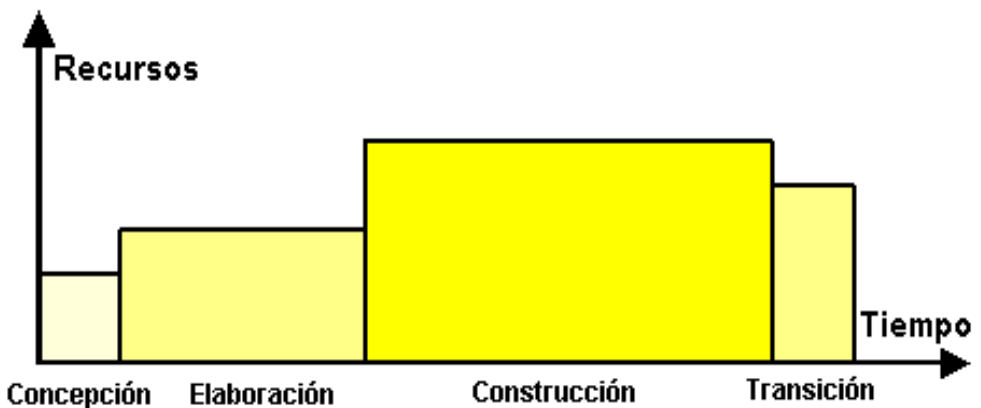


Figura 4. Fases de la infraestructura ó metodología Rational Unified Process (RUP)

5. Resultados

Desarrollar software educativo puede resultar una actividad muy difícil sobre todo cuando se orienta a competencias. Resolver los problemas que se presentan en las diferentes asignaturas, y más cuando se trabaja con un conjunto de métodos y técnicas que permitan medio defenderse manejando una complejidad en cada una de esas actividades, quedan solucionadas con esta nueva arquitectura de software para el desarrollo de micromundos llamada MICRO_LIBRE, la cual ya se probó en el desarrollo de un micromundo para el aprendizaje de redes Lan y sus resultados fueron muy buenos, ya que se minimizo el tiempo de análisis, el tiempo de programación, y cuando se llevo a la práctica, la muestra de estudiantes que se tomo como prueba más del 82% concluyó que se les hizo más fácil aprender todo sobre redes Lan por medio del micromundo.

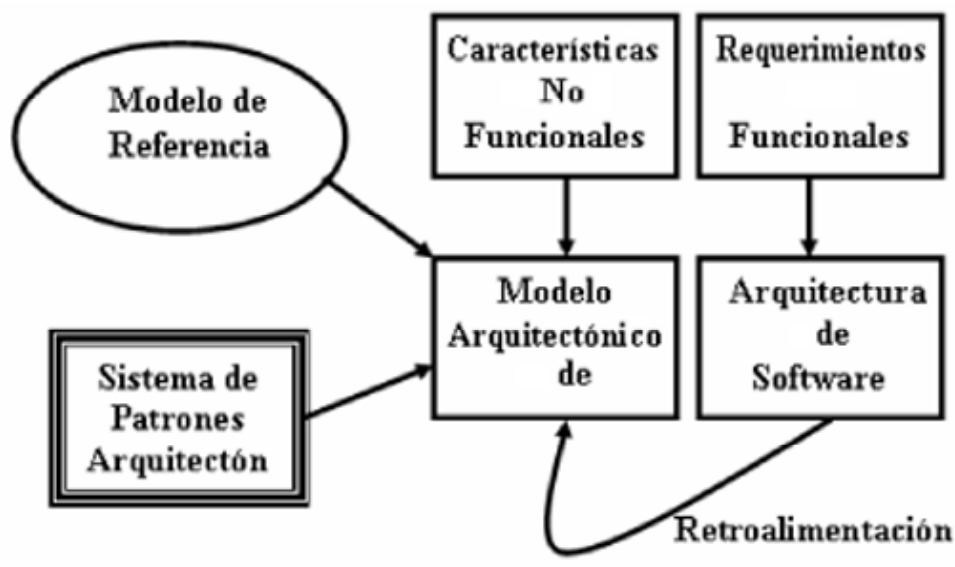


Figura 5. Elementos para Generar una Arquitectura de Software

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 esta 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 de datos, elementos que conectan, y del estilo arquitectónico, respectivamente.

BIBLIOGRAFÍA

- GALVIS, AH. (1992, Marzo), Materiales educativos computarizados: Ocasión para repensar los ambientes educativos?.
Primer Congreso Colombiano de Informática Educativa (Santafé de Bogotá: RIBIE-COL, Marzo 11-14 de 1992)
- GALVIS, AH. Mejoramiento educativo apoyado con informática: enfoque estratégico. *Informática educativa* 7 (1), 1994, pp. 49-91.
- Dewayne E. P., (2007, Junio). *Foundations for the Study of Software Architecture*, AT&T Bell Laboratories / University of Colorado.
- Philippe K., (2007, Agosto). *Architectural Blueprints—The “4+1” View Model of Software Architecture*, Rational Software Corp.
- Mendoza S. A. (2008, Febrero). *Metodologías de Desarrollo de Software*, consultada en: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html
- Gutierrez P., J. (2008, Febrero). *Un acercamiento al Software Educativo*, consultada en: <http://cnx.org/content/m16318/latest/>