

Una Estrategia Para El Apoyo De Los Procesos De Enseñanza - Aprendizaje De La Programación En Ingeniería De Sistemas Utilizando Objetos Virtuales De Aprendizaje. Un caso de estudio en el Programa de Ingeniería de Sistemas de la UCC Bucaramanga

Marlene Lucila Guerrero Julio, Sergio Arturo Medina Castillo
Programa de Ingeniería de Sistemas, Facultad de Ingenierías, Universidad Cooperativa de Colombia
Bucaramanga – Santander, Colombia

Resumen— Los objetos virtuales de aprendizaje se han convertido en materia de estudio desde hace algunos años por parte del Ministerio de Educación Nacional y las universidades de Colombia y el Mundo. Las principales razones de esto radican en que son herramientas esenciales para potenciar los procesos de educación, ya que propician de una manera más sencilla la conexión de estos con las Tecnologías de la Información y la Comunicación (TIC's).

La Universidad Cooperativa de Colombia seccional Bucaramanga, acorde con esta tendencia, adelanta desde hace tres años un proceso de desarrollo de herramientas de tecnología de información que permitan realizar acompañamiento virtual a las actividades presenciales del pensum académico. De la mano con esta iniciativa, se desarrolló el proyecto de investigación denominado “Objetos Virtuales de Aprendizaje para apoyar las Actividades Formativas de los Componentes del Plan de Estudios del Programa de Ingeniería de Sistemas”, el cual tiene como fin diseñar y construir herramientas que permitan fortalecer los procesos de enseñanza aprendizaje de los componentes del programa y que propendan a la estandarización SCORM, con el fin de lograr su reutilización, accesibilidad, interoperabilidad y duración. De igual manera, este proyecto pretende fortalecer el desarrollo en términos de la utilización de bibliotecas digitales que permitan el acceso a la información en el momento que se necesite y a la utilización de ambientes inteligentes de enseñanza aprendizaje, que potencien el aprendizaje centrado en el estudiante y en el trabajo en equipo.

Este artículo pretende proporcionar una visión general de las estrategias diseñadas para apoyar los procesos de enseñanza – aprendizaje del componente de programación en Ingeniería de Sistemas, utilizando objetos virtuales de aprendizaje (OVA), centrándose en los siguientes cuestionamientos: ¿Cuáles deberían ser los objetos informativos y digitales pertenecientes a un OVA que apoyen las actividades presenciales del componente de programación de un programa de Ingeniería de Sistemas?,

¿Cuáles son las actividades virtuales que permiten apoyar las necesidades de aprendizaje de la lógica de programación y del aprendizaje y aplicación de algoritmos de programación en contextos organizacionales reales? y ¿Cómo deben crearse los objetos SCORM de los OVAS para el componente de programación de manera que se logre la reutilización, accesibilidad, interoperabilidad y duración en dicho programa?

Palabras Claves: Educación, Ingeniería de Sistemas, Objetos Virtuales de Aprendizaje, Programación.

I. INTRODUCCIÓN

En Colombia, si bien hay diversas definiciones para lo que se entiende por un “Objeto de Aprendizaje” (Learning Object en inglés), el Ministerio de Educación Colombiano [1] los define como: “todo material estructurado de una forma significativa, asociado a un propósito educativo y que corresponda a un recurso de carácter digital que pueda ser distribuido y consultado a través de la Internet. El objeto de aprendizaje debe contar además con una ficha de registro o metadato consistente en un listado de atributos que además de describir el uso posible del objeto, permita la catalogación y el intercambio del mismo”. De esta manera, el OVA se convierte en una herramienta de tecnología de información cuyo principal objetivo es el de convertirse en apoyo del proceso de enseñanza aprendizaje. Por tanto será entonces un mecanismo de interacción e influencia en la educación, ya que no solo se limita a las actuaciones sincrónicas y presenciales, sino al conjunto de todas aquellas actuaciones que profesores y estudiantes desarrollan sin estar ambos presentes, en un mismo espacio o coincidir en el tiempo.

Si enfocamos esta definición en el ámbito de programación, se abre camino para que se reflexione sobre las actividades virtuales que apoyarían el proceso de enseñanza aprendizaje de la lógica de programación y del aprendizaje y aplicación de algoritmos de programación en contextos organizacionales reales y sobre cuáles serían los mecanismos que permitirían lograr la reutilización, accesibilidad, interoperabilidad y duración de dichas

actividades.

II. HACIA UNA COMPRESIÓN DE LOS OBJETOS VIRTUALES DE APRENDIZAJE

La National Learning Infrastructure Initiative [2], define los objetos de aprendizaje como aquellos módulos de recursos digitales únicos, identificados y meta-etiquetados que pueden ser usados para el soporte del aprendizaje.

La anterior definición nos lleva a la identificación de los cuatro componentes claves en el diseño de objetos virtuales de aprendizaje, los cuales son: contenidos, actividades de aprendizaje, elementos de contextualización y metadatos. Estos elementos también pueden ser identificados en las definiciones que se muestran en el cuadro 1.

CUADRO 1
IDENTIFICACIÓN DE LOS COMPONENTES CLAVE EN EL DISEÑO DE OBJETOS VIRTUALES DE APRENDIZAJE

| Autor | Definición |
|--------------|--|
| Davila, 2008 | Cualquier entidad, digital o no digital, que pueda ser usada en aprendizaje, educación o entrenamiento apoyado en tecnología”. |
| Downes, 2006 | Son colecciones de alto nivel tales como cursos y currículos. |
| Wiley, 2006 | Cualquier recurso digital que pueda ser reutilizado como soporte del aprendizaje. |
| IEEE, 2006 | Cualquier entidad digital que pueda ser utilizada para el aprendizaje, la educación y el entrenamiento. |

De acuerdo con lo anteriormente expuesto, se puede afirmar que los objetos virtuales cumplen una función de mediadores pedagógicos en el quehacer docente, los cuales han sido diseñados intencionalmente para un propósito de aprendizaje y sirven a los actores de las diversas modalidades educativas.

En tal sentido, los objetos virtuales de aprendizaje deben diseñarse a partir de criterios como:

1. Atemporalidad: Para que no pierdan vigencia en el tiempo y en los contextos utilizados.
2. Didáctica: Los objetos tácitamente responden a qué, para qué, con qué y quién aprende.
3. Usabilidad: Que faciliten el uso intuitivo del usuario interesado.
4. Interacción: Que motiven al usuario a promulgar inquietudes y retornar respuestas o experiencias sustantivas de aprendizaje.
5. Accesibilidad: Garantizada para el usuario interesado según los intereses que le asisten.

III. ENTENDIENDO LAS NECESIDADES DE APRENDIZAJE EN EL COMPONENTE DE PROGRAMACIÓN

La Asociación Colombiana de Ingeniería de Sistemas ACOFI y el ICFES, definen la “Programación” como uno de los componentes principales de la ingeniería de Sistema[3]. De acuerdo con ACOFI, el Ingeniero de Sistemas debe desarrollar la capacidad de analizar, diseñar e implementar soluciones de tecnología de información en los distintos problemas organizacionales teniendo en cuenta procesos y criterios de calidad de Ingeniería de Software. Lograr dicha competencia en los estudiantes, requiere del desarrollo de actividades y contenidos que propendan al desarrollo de la lógica de programación y al aprendizaje y aplicación de algoritmos de programación en contextos organizacionales reales.

De igual manera, la ACM¹, define los tópicos que componen el cuerpo de conocimiento de un programa de ciencias de la computación[4], los cuales son: Estructuras discretas, fundamentos de programación, interacción humano – computador, computación visual y gráfica, algoritmos y complejidad, sistemas inteligentes, arquitectura y organización, administración de la información, computación centrada en redes, configuración profesional y social, ingeniería del software, lenguajes de programación y ciencias computacionales.

Inmersos en estos tópicos se encuentran la mayoría de las asignaturas que hacen parte del componente de programación del programa de Ingeniería de Sistemas de la universidad Cooperativa de Colombia. El pensum en términos generales está conformado por cuatro componentes que abarcan la malla curricular, tal y como se muestra en la figura 1. En esta figura se pueden observar las asignaturas que hacen parte del componente de programación perteneciente a las áreas² de Ciencias Básicas de la Ingeniería e Ingeniería Aplicada.

Cada componente del programa centra su atención en una competencia general a la cual se desea llegar a partir de las competencias específicas de cada asignatura, las cuales se encuentran definidas en los formatos microcurriculares, los planes de aula y los syllabus de las asignaturas.

Ahora bien, si se desea comprender la competencia del componente de programación, es necesario que se desglosen las competencias específicas de cada una de las asignaturas que conforman dicho componente. De [5] obtenemos los 4 referentes claves para la descripción de una competencia específica: Criterios de desempeño, Saberes esenciales, Evidencias requeridas y Rango de aplicación.

¹ Association for Computing Machinery – ACM. Asociación internacional encargada de desarrollar guías curriculares para programas de grado en computación.

² Contenidos Referenciales del marco de fundamentación conceptual y especificaciones de prueba – ECAES. (ACOFI, 2005).

FIGURA 1
COMPONENTES DEL PROGRAMA DE INGENIERÍA DE SISTEMAS DE LA UNIVERSIDAD COOPRETAIVA DE COLOMBIA SECCIONAL BUCARAMANGA.



El análisis de las competencias específicas de las asignaturas del componente nos permite definir la competencia general del componente de Programación de la siguiente manera “Analizar, diseñar e implementar soluciones de tecnología de información a los distintos problemas organizacionales teniendo en cuenta procesos y criterios de calidad de Ingeniería de Software”. Lograr dicha competencia en los estudiantes, requiere del desarrollo de actividades y contenidos que propendan al desarrollo de la lógica de programación y al aprendizaje y aplicación de algoritmos de programación en contextos organizacionales reales.

IV. EL APOYO DE LAS ACTIVIDADES VIRTUALES

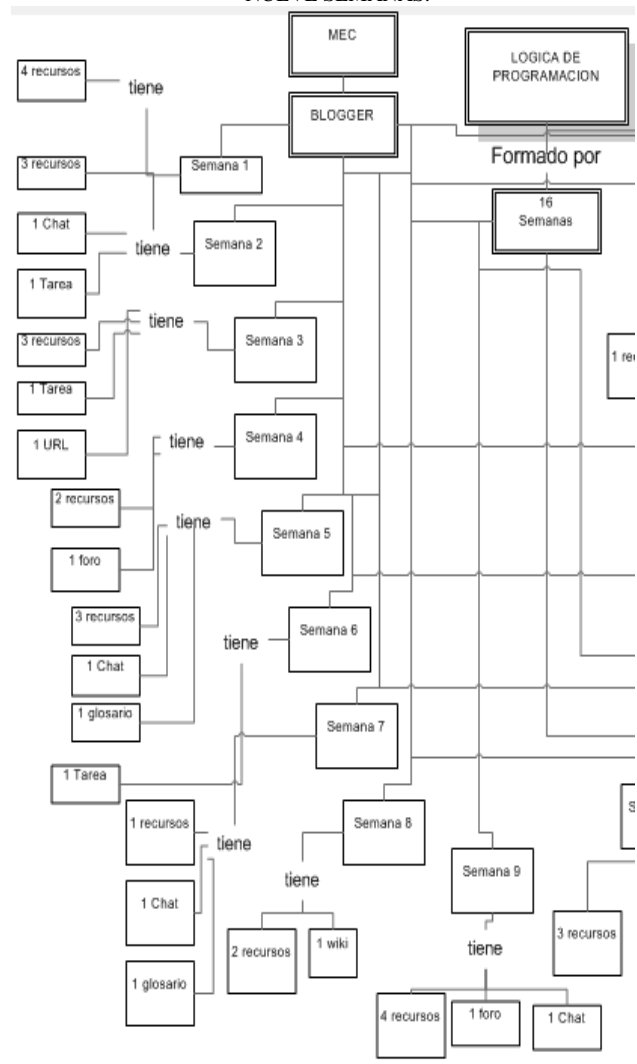
Utilizando las competencias definidas en cada asignatura del componente de programación y orientados por el concepto de diseño instruccional, en donde la enseñanza se centra en el proceso de aprendizaje y no en los contenidos específicos, permitiendo al docente combinar los diferentes materiales con las actividades propuestas, para conducir al estudiante a desarrollar habilidades en la creación de interpretaciones por sí mismo, y manipular situaciones hasta que las asuma como parte del aprendizaje (motivación no sólo como factor externo sino interno) [6], se diseña las actividades virtuales que permitirán apoyar los procesos de enseñanza aprendizaje de cada una de las asignaturas del componente.

Utilizando como caso específico la asignatura Lógica de Programación, se incorporan los elementos fundamentales

del proceso de diseño instruccional, el cual incluye el análisis de los participantes, la ratificación de metas y objetivos, el diseño e implantación de estrategias y la evaluación. En la figura 2 se aprecian los elementos y actividades virtuales utilizadas para apoyar nueve del total de semanas en las que se desarrolla comúnmente la asignatura durante un semestre regular³.

Posteriormente, se describen en detalle cada una de las actividades planteadas [7]. Dichas actividades se encuentran plasmadas en los planes de aula y los syllabus de cada asignatura, con el fin de hacerlas parte integral del proceso de enseñanza aprendizaje y se encuentran relacionadas directamente con la metodología MICEA (Metodología Interdisciplinaria Centrada en Equipos de Aprendizaje) [8].

FIGURA 2
DISEÑO INSTRUCCIONAL DE LÓGICA DE PROGRAMACIÓN PARA NUEVE SEMANAS.



La metodología MICEA centra su atención en cinco momentos específicos: Momento Presencial (que para el caso de los objetos virtuales de aprendizaje pasará a

³ La Universidad Cooperativa de Colombia seccional Bucaramanga, desarrolla semestres regulares de 16 semanas.

denominarse “momento de teorización del conocimiento [9]), momento de trabajo individual dirigido, momento de trabajo en equipo, momento de acompañamiento y momento de socialización.

V. DISEÑO DE LOS OBJETOS INFORMATIVOS Y DIGITALES DE LOS OVAS Y APLICACIÓN DE SCORM

Tomando como referencia la relación de competencias, diseño instruccional y actividades, se construyen los objetos virtuales de aprendizaje y se definen los objetos informativos y los objetos digitales que servirá de apoyo a cada asignatura, cada uno de ellos con sus respectivos metadatos, lo cual permite garantizar interoperabilidad y accesibilidad de los mismos.

Ahora bien, para garantizar su reutilización y duración, se empaquetan los objetos digitales e informativos utilizando cualquiera de las herramientas de libre distribución en la WEB. Para el caso específico se utilizó el programa Reload (Reusable eLearning Object Authoring and Delivering) [10], el cual puede ejecutarse sobre plataformas Windows, Mac OS X y Linux y soporta metadatos IMS, IEEE LOM, paquetes de contenido IMS 1.1.4, SCORM 1.2 y SCORM 2004. En la figura 3, se presenta un ejemplo del objeto virtual montado sobre Moodle para la asignatura Estructura de Datos.

FIGURA 3
EJEMPLO DEL OBJETO VIRTUAL PARA UNA ASIGNATURA DEL COMPONENTE DE PROGRAMACIÓN

The screenshot shows a Moodle course interface for 'UNIVERSIDAD VIRTUAL COOPERATIVA'. The course is titled 'INST ▶ Estructuras de Datos Bucaramanga 2009II'. The main content area is a 'Diagrama semanal' (weekly diagram) showing three weeks of content:

- 21 de agosto - 27 de agosto (SEMANA 1):** Includes 'Syllabus Estructura De Datos', 'Plan De Aula Estructura De Datos', 'ESTRUCTURAS DE DATOS', 'alg_ed_est_datos_basic', and 'Blogger de Estructuras de Datos (Ayuda)'. A 'Novedades' (News) section is also visible.
- 28 de agosto - 3 de septiembre (SEMANA 2):** Includes 'ESTRUCTURAS DE DATOS', 'APUNTADES', 'Apunadores', and 'Blogger Estructuras de Datos (Ayuda)'.
- 4 de septiembre - 10 de septiembre:** Includes 'Compiladores Carlos Alberto Acevedo Rey', 'Estructura de Datos', and 'Carlos Alberto Acevedo'.

The interface includes a left sidebar with navigation options like 'Personas', 'Actividades', 'Administración', and 'Mis cursos'. A right sidebar shows 'Novedades', 'Eventos próximos', and 'Actividad reciente'.

Apoyados en el desarrollo de instrumentos para la WEB disponibles en Internet, se diseñaron los bloggers que se enlazan con la plataforma Moodle mediante hipervínculos y

llamados especiales contemplados dentro de los metadatos, los cuales sirven como ayuda interactiva del proceso de enseñanza aprendizaje y complementan el diseño instruccional⁴.

Adicionalmente a esto, para generar un valor agregado que mejore el proceso de enseñanza – aprendizaje de la Programación, se desarrolla un objeto virtual de aprendizaje OVA, compuesto por un ejercitador de problemáticas bajo entorno WEB, utilizando una metodología que involucra pasos ó etapas específicos con resultados tangibles y verificables. La finalidad principal de esta OVA es el manejo paso a paso de soluciones a problemas que involucren información basados en el proceso de abstracción mediante la aplicación del paradigma de la programación orientada a objetos y la programación sobre la base de datos mediante el uso de funciones y procedimientos almacenados en la base de datos y los disparadores o trigger.

El ejercitador de problemáticas define tres actores o perfiles del software, que son el administrador, docente y estudiante. El administrador tiene asignadas actividades correspondientes a la seguridad, manejo de usuarios y respaldo o backups, (ver Figura 4 en la siguiente página, Casos de uso del administrador).

El docente se encargará de definir problemáticas y su solución a través de la metodología que se defina, cargando el paso a paso de la solución de cada problema. Además definirá el banco de preguntas y respuestas, junto con los logros y deficiencias generadas en cada situación problema, (ver Figura 5 en la siguiente página, Casos de uso del docente).

El estudiante tiene definido actividades correspondientes a la consulta de las temáticas específicas, problemáticas y su solución, junto con la presentación de las evaluaciones de cada una de las temáticas, (ver Figura 6 en la siguiente página, Casos de uso del estudiante).

Para el diseño de la base de datos, se tienen en cuenta tres (3) aspectos fundamentales que componen la estructura del ejercitador de problemáticas. Ellos son:

1. Perfiles o actores.
2. Situaciones Problemáticas.
3. Evaluación.

⁴ Un ejemplo de los bloggers diseñados se puede encontrar en <http://programacionucc.blogspot.com/>

FIGURA 4
CASOS DE USO DEL ADMINISTRADOR

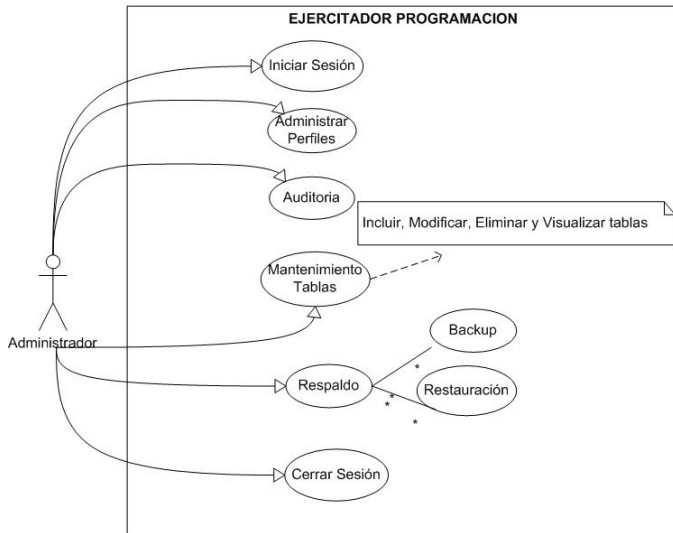
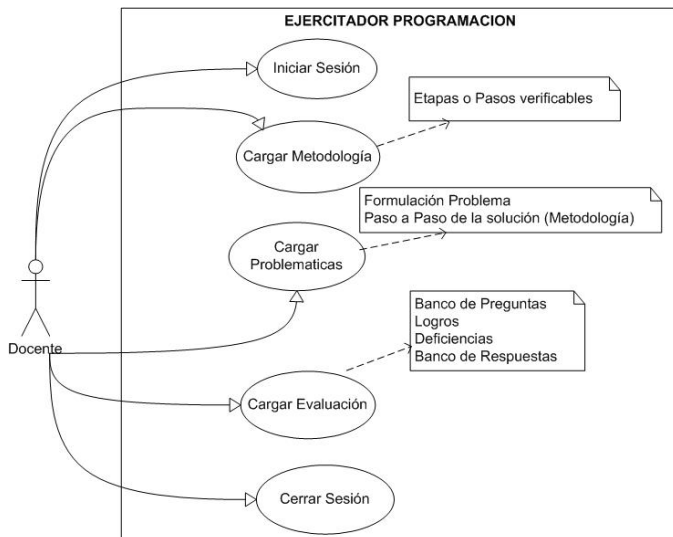


FIGURA 5
CASOS DE USO DEL DOCENTE

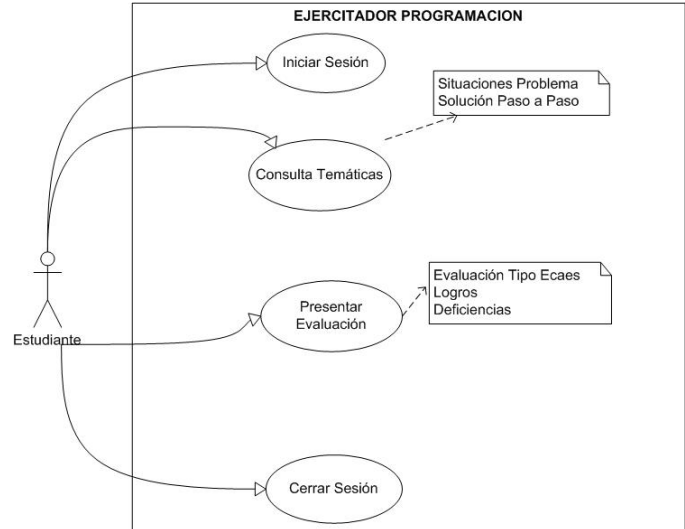


Los perfiles o actores involucran a los personajes que interactúan con el software y los elementos de control y registro anexos a ellos. Se componen de entidades, entre otras, como el perfil, que identifica los actores del software, el usuario, en la cual se almacena la información detallada de cada uno de los interactuantes con el programa, administradores, estudiantes y docentes, la auditoría para almacenar en ella la bitácora con las actividades que cada usuario desarrolla durante una sesión de trabajo y el permiso, en la cual se definen las opciones a las cuales tienen acceso cada uno de los perfiles.

La situación problemática involucra aquellos aspectos relacionados con la metodología y sus etapas o pasos, la formulación de las problemáticas y su solución paso a paso de dependiendo de cada metodología.

El aspecto evaluación involucra las preguntas, que son situaciones problemáticas, respuestas, logros y deficiencias que se alcanzarían en cada una de estas situaciones. Es importante anotar que docente será el encargado de generar toda esta estructura de evaluación, mientras el estudiante realizará su proceso de evaluación mediante el análisis de los casos presentados en el banco de situaciones problemáticas.

FIGURA 6
CASOS DE USO DEL ESTUDIANTE



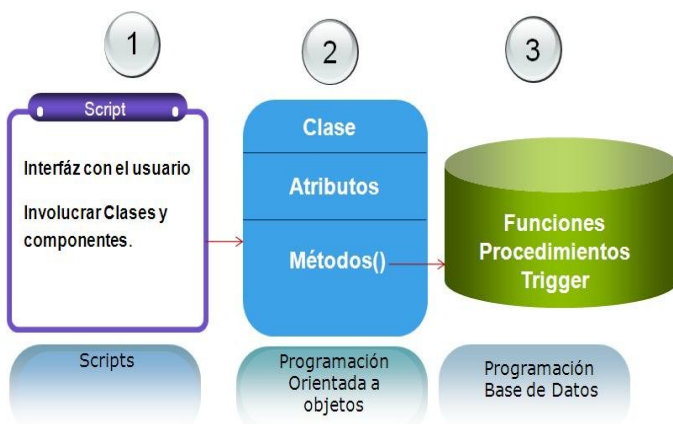
En el diagrama de clases definido para el ejercitador, se tiene en cuenta los tres actores que participan en el software, como son el administrador, estudiante y docente que heredan los atributos y métodos genéricos de una clase primaria llamada usuario. Los atributos genéricos de la clase usuario son el código, nombre y clave y los métodos genéricos definidos en la clase usuario son inicio_sesion(), donde se desarrolla lo concerniente al control de acceso, auditoria(), donde se genera el log de auditoría para el usuario y permisos(), donde se controla que los perfiles utilicen las opciones definidos para cada uno de ellos.

Estas actividades de cada una de estas clases son el detalle de lo previsto en los casos de uso para cada uno de los actores o personajes que participan en el ejercitador de problemáticas.

Con respecto al estándar de construcción a emplear, se utilizan dos tecnologías importantes, como lo son, la programación orientada a objetos y la programación sobre la base de datos mediante el uso de funciones, procedimientos y disparadores (trigger), que redundan en eficiencia y seguridad para el desarrollo en ambiente WEB y que facilita el mantenimiento futuro del ejercitador de problemáticas. Para la programación de cada uno de los procesos, la aplicación estructura tres grandes aspectos estándares de aplicación de las tecnologías mencionadas. El primero de ello es la función, procedimiento o disparador (trigger) de la base de datos, que contiene las operaciones en términos de SQL y PL-SQL que permiten la realización del proceso

específico, generando mayor seguridad y eficiencia al centralizar el almacenamiento y procesamiento sobre la base de datos. El segundo aspecto hace referencia a las clases involucradas en dicho proceso, a través de sus atributos y sobre todo, sus métodos. Es importante anotar que bajo el estándar definido, los métodos de las clases se encargaran de ejecutar la función ó procedimiento correspondiente al proceso, en la base de datos. Finalmente, el tercer aspecto hace referencia al script de código fuente que involucra la interfaz gráfica con el usuario, en el cual se implementarán todos estos aspectos de la GUI del proceso y la incorporación de las clases y componentes que se utilizarán dentro del mismo. Se puede apreciar este esquema en la Figura 7. Estándar de construcción.

FIGURA 7
ESTÁNDAR DE CONSTRUCCIÓN



Es importante recalcar que esta OVA, representado en el Ejercitador de Problemáticas, se aplicará para su prueba sobre las asignaturas del componente de programación, pero podría ser utilizado en cualquier área que involucre problemáticas que se resuelvan a través de una serie de pasos ó etapas definidos en una metodología.

VI. CONCLUSION

Los objetos virtuales de aprendizaje pueden ser utilizados para apoyar los procesos de enseñanza aprendizaje de la programación, siempre y cuando se acompañen de un diseño instruccional claro y detallado y se analicen las competencias que se desean formar en los estudiantes con el fin de generar las estrategias propicias para la vinculación de las actividades y los temas de estudio.

Los objetos virtuales diseñados para apoyar de manera virtual las asignaturas del componente de programación no deben centrarse únicamente en la sistematización de contenidos, sino que también deben contener asociaciones con diferentes herramientas de tecnología de información que permitan garantizar la interacción docente estudiante y que permitan, a través de soluciones informáticas apoyar las necesidades de aprendizaje de la lógica de programación y del aprendizaje y aplicación de algoritmos de programación

en contextos organizacionales reales.

REFERENCIAS

- [1] Leal D., Zea C. (2006). Iniciativa Nacional de Objetos de Aprendizaje. Ministerio de Educación Nacional. Programa. de Nacional de Medios y Nuevas Tecnologías.
- [2] NLII. (2008). Mapping the Learning Space: Deeper Learning and Learning Theories. Consultado el 21 de Febrero de 2009. URL: <http://www.west.asu.edu/nlii/learning.htm>.
- [3] ACOFI. (2005). Marco de Fundamentación Conceptual, Especificaciones de Prueba. Ecaes ingeniería de sistemas. Versión 6.0.
- [4] ACM / IEEE. (2001). Final Report of the Joint ACM/IEEE-CS Task Force on Computing Curricula 2001 for Computer Science. Consultado el 23 de Febrero de 2009. URL: <http://www.acm.org/education/education/education/curricvols/cc2001.pdf>.
- [5] Tobón S. (2006). Las Competencias y el Desarrollo Curricular. Diplomado formación por competencias. Universidad Cooperativa de Colombia. Barrancabermeja. Julio.
- [6] Tobón M. (2007). Diseño Instruccional en un entorno de aprendizaje abierto. Universidad Tecnológica de Pereira.
- [7] Guerrero M. (2009). Objetos Virtuales de Aprendizaje para Apoyar las Actividades Formativas del Programa de Ingeniería de Sistemas. Universidad Cooperativa de Colombia Seccional Bucaramanga.
- [8] Velandia C. (1997). Metodología Interdisciplinaria, ACSI-Pro. 1Ed, Colombia.
- [9] Gómez L., et al. Integración de la metodología pedagógica MICEA en un entorno virtual de Enseñanza aprendizaje, aprovechando las nuevas tecnologías de la Información y las comunicaciones. (2005). Universidad Cooperativa de Colombia.
- [10] University of Bolton. (2008) Reload - Reusable eLearning Object Authoring and Delivering. Consultado el 9 de Marzo de 2009. <http://www.reload.ac.uk/>