

Proposal of an Assistant for the Automation of the Design and Creation Process of Learning Objects.

Jorge Bermeo-Conto¹, Miguel Zúñiga-Prieto¹, Boris Cabrera¹, Jorge Maldonado-Mahauad^{1,2}

¹Department of Computer Science, University of Cuenca, Cuenca, Ecuador

²Department of Computer Science, Pontificia Universidad Católica de Chile, Santiago, Chile

jorge.bermeo@ucuenca.edu.ec, miguel.zunigap@ucuenca.edu.ec, jjmaldonado@uc.cl, boris.cabrered@gmail.com

Abstract—Learning Objects (LOs) are digital educational materials that enable the teaching and learning process to be mediated. However, their design and creation often exceeds the possibilities that a teacher has for their production. Teachers are experienced users in the educational field and have little or no experience in programming tools, so it is expected that the production of an OA can be scalable. The aim of this paper is to present an intermediate software solution that supports the systematic application and documentation of the phases proposed in the DICREVOA methodology for the design, analysis and implementation of OA. For this a case study with the proposed graphic editors is presented. As a result, this wizard for the automation of the design and creation process of OA will allow teachers to exploit and improve the productivity, reliability, maintainability and portability of this type of educational resource from their experience in the domain of the educational context.

Keywords— Learning Objects, Digital Educational Materials, Model Driven Engineering, MDE, Domain Specific Languages, DSL

I. INTRODUCCIÓN

Los Materiales Educativos Digitales (MED) son recursos digitales que son utilizados como mediadores en el proceso de enseñanza-aprendizaje. Para esto, su diseño y construcción se basa en criterios didácticos, pedagógicos y tecnológicos, orquestados por el diseño instruccional [1]. Un ejemplo de MED son los Objetos de Aprendizaje (OA).

En 1994, Wayne Hodgins, utiliza por primera vez el término “Objeto de Aprendizaje”. El nombre fue inspirado al observar a su hijo jugar con bloques de LEGO [3]. La idea de los OA es que fueran recursos digitales auto-contenidos y reutilizables, lo que permitiría que varios OA puedan ser combinados y crear recursos digitales de mayor granularidad, como por ejemplo lecciones, módulos, etc. Para este trabajo un OA es una unidad didáctica digital independiente, cuya estructura está formada por un objetivo de aprendizaje específico, contenidos, actividades y una autoevaluación. Puede ser reutilizado en diferentes contextos tecnológicos (p. ej., Repositorios, Entornos Virtuales de Enseñanza y Aprendizaje) y educativos (p.ej., clases presenciales, clases virtuales), además cuentan con unos metadatos que propician su localización dentro de repositorios [2].

Para la creación de MED a través del paradigma de OA, el docente hace uso de distintas estrategias que dependen directamente de su conocimiento (p. ej., didáctica, pedagogía y tecnología), y cuyo propósito es lograr resultados de aprendizaje concretos en los estudiantes. Sin embargo, la naturaleza compleja del diseño, creación de los OA sigue siendo un reto vigente para

los docentes. Más aún, si estos carecen de un equipo multidisciplinario institucional que le proporcionen el soporte necesario durante la creación de estos materiales. Esto da como resultado, que sea el docente quien debe encargarse de todo el proceso de producción [3]–[8].

Actualmente existen diversas metodologías para el diseño y creación de OA [9] que han sido reportadas en los últimos 18 años. Cada una de estas han contribuido de diferentes maneras dependiendo del contexto para el cual fueron concebidas. Por ejemplo, las metodologías venezolanas para la creación de OA consideran en sus propuestas el uso del diseño instruccional, el estándar de metadatos LOM, y como estándar de empaquetamiento SCORM. Las metodologías mexicanas hacen uso de patrones pedagógicos, plantillas, ciclos evolutivos de prototipos, UML y herramientas que guían el diseño del OA. Es de destacar, que la mayoría de metodologías no proponen para el diseño niveles de granularidad específicos que luego sirvan para la reutilización del OA (sobre todo la reutilización educativa), de manera tal que se puedan combinar los OA con el fin de crear colecciones de OA tales como módulos, lecciones, etc. [9].

A partir de lo anterior, y tomando los aspectos más relevantes desde el punto de vista didáctico, pedagógico y tecnológico, los autores en [10] han propuesto una metodología denominada DICREVOA 2.0. Esta metodología está pensada para docentes que no son del contexto de la informática y que no cuentan con un equipo institucional multidisciplinario que les proporcionen el soporte necesario en la creación OA. Las bondades y utilidad de esta metodología han sido validadas en distintos estudios realizados sobre distintos contextos [11] lo que ha demostrado su factibilidad de uso al momento de crear, diseñar, publicar y evaluar OA [9]. DICREVOA 2.0 promueve 5 fases para el proceso de producción de OA y son: Análisis, Diseño, Implementación, Evaluación y Publicación. Como soporte al proceso de Análisis y Diseño se han construido una serie de plantillas manuales que sirven para documentar estas fases y facilitar el proceso de abstracción de las necesidades que originan la creación del OA. Para la implementación del OA la metodología hace uso de herramientas de autor que deben ser programadas (en algunos casos), (p. ej., eXeLearning, Ardora, Hot-potatoes, etc.), para la publicación en un repositorio de OA (ROA) propone el empaquetado con SCORM y para la Evaluación del OA utiliza una rúbrica creada sobre distintas herramientas existentes. Sin embargo, los docentes son usuarios con experiencia en el dominio educativo y tienen poca o nula experiencia en la programación de herramientas. La producción de OA debe poder ser escalable para un docente, quienes a partir del conocimiento del dominio puedan

explotar y mejorar la productividad, confiabilidad, mantenibilidad y portabilidad de los mismos. Por este motivo se requiere de soluciones de software intermedias que apoyen la aplicación sistemática y documentación de las fases propuestas en la metodología DICREVOA.

Para dar respuesta a lo anterior, la Ingeniería Dirigida por Modelos (en inglés Model-Driven Engineering – MDE) es un paradigma de ingeniería de software, que se centra en la creación y explotación de modelos de dominio como principales artefactos del proceso de desarrollo, más que en conceptos informáticos. Además su propósito [12] busca minimizar los costes y tiempo de desarrollo de las aplicaciones software, en la búsqueda de mejorar la calidad de las aplicaciones, independiente de la plataforma donde el software se ejecuta [13].

Por lo expuesto anteriormente, en este trabajo se propone un asistente de software que facilitará a los docentes la ejecución sistemática y documentación de las fases del proceso de análisis, diseño e implementación del OA dejando de lado los aspectos tecnológicos de las herramientas involucradas para su implementación. Para esto la solución propuesta contará con editores gráficos que faciliten la documentación de las fases de análisis y diseño (p. ej., diseño instruccional) de manera independiente de la tecnología que se empleará para la implementación del OA (p. ej., eXeLearning). Además, los editores gráficos asistirán al docente en las decisiones tecnológicas. Como resultado, la documentación capturada en estos editores gráficos podrá servir de insumo para generar diferentes versiones del código que implementa el OA, de acuerdo al estilo de aprendizaje y al Sistema de Gestión de Aprendizaje (en inglés Learning Management System - LMS) a utilizar para el despliegue del OA.

El artículo se organiza de la siguiente forma: la sección II presenta los trabajos relacionados, la sección III presenta la propuesta de automatización de un asistente, la sección IV presenta la propuesta desarrollada, la sección V presenta un caso de estudio ilustrativo de la aplicación del editor; y finalmente, la sección VI presenta las conclusiones y líneas futuras.

II. TRABAJO RELACIONADO

A partir de la bibliografía existente se han identificado trabajos que han utilizado MDE para la creación de OA. Autores como Souza [14], consideran que los OA no se adaptan fácilmente a los perfiles de los estudiantes por lo que propone su personalización utilizando MDE y describe una arquitectura de los OA personalizables. Una de las limitaciones de este trabajo es que no está soportado por un proceso sistematizado que guíe al docente en la creación de los OA. Por otro lado Wang [15] propone un Modelo de Plataforma Independiente (PIM) expresado a través de UML, y el Modelo Específico de Plataforma (PSM) expresado a través un perfil UML para la interface EJB (Enterprise JavaBeans). No obstante, esta propuesta se centra en la descripción general de Model Driven Arquitectura (MDA) y no precisa el soporte al docente.

Bizonova [16] define una solución basada en modelos para el desarrollo abierto y colaborativo de OA presentando un enfoque al problema de la integración de los LMS basado en mapas conceptuales. Como limitación se ha identificado que no se involucra en el proceso de creación de OA.

Montenegro [17], propone la construcción de un Modelado

Específico de Dominio (DSM) para la creación de módulos en los LMS, concluyen que reducen significativamente el tiempo y esfuerzo en la construcción y despliegue de cursos sobre plataformas LMS. Sin embargo, esta propuesta no se centra directamente en OA y menos en las fases para su creación.

Dodero por su parte [18], propone un enfoque basado en modelos y una herramienta para editar diseños de aprendizaje llamada FlexoLD. Esta herramienta abstrae a los diseñadores instruccionales y se enfocada en las características del diseño instruccional, pero dejando de lado las fases de análisis y diseño multimedial importantes para la generación de OA.

Malek [19] propone la herramienta de autor ContAct-Me (CONText y ACTivity Adaptive Modeler para entornos de aprendizaje maleable) basada en el lenguaje CAAML (Context-aware) Adaptive Activities Modeling Language), el cual es un lenguaje de modelado visual específico del dominio. Este permite a los diseñadores pedagógicos modelar el contexto a través de actividades de aprendizaje adaptativo y reglas de co-adaptabilidad. Sin embargo, deja de lado la fase de análisis.

Finalmente, Graciotto [20] define un enfoque basado en modelos para el desarrollo de OA que va desde la definición de objetivos de aprendizaje hasta la generación de contenido. Esta propuesta se centra en el diseño de OA, pero deja de lado las fases de análisis que es importante, más aún, dado a que de esta se deriva la necesidad de un OA.

Por lo expuesto anteriormente, es evidente la falta de propuestas que relacionen aspectos metodológicos que guíen en la creación del OA con herramientas de software que asistan al docente durante su implementación. Propuestas que permitan al docente centrarse en aspectos didácticos, pedagógicos y de la temática que domina; en donde, los aspectos tecnológicos requeridos para la implementación sean soportados por un enfoque de automatización.

III. PROPUESTA DE AUTOMATIZACIÓN DE UN ASISTENTE

A. Contextualización

La metodología DICREVOA 2.0 es el resultado del estudio de 29 propuestas, heurísticas y metodologías [21] utilizadas en el contexto Iberoamericano y que han sido utilizadas para el diseño y creación de OA. DICREVOA 2.0 está compuesta de 5 fases: Análisis, Diseño, Implementación, Evaluación y Publicación. Estas fases tienen como propósito guiar al docente en la generación de un OA. En la fase de Análisis, el docente realiza el levantamiento de los requerimientos de un OA, documentándolos en una serie de plantillas que son llenadas a mano. Estas plantillas sugieren identificar necesidades del OA tales como: el tema, la descripción textual del contenido, el nivel educativo de la población destinataria, estilo(s) de aprendizajes que deberá favorecer tiempo estimado en el que se podría completarlo, el contexto educativo en el que será utilizado, el tipo de licencia a utilizar, y requerimientos no funcionales. En la fase de Diseño, el docente también es provisto de plantillas que le guían en la toma de decisiones desde dos perspectivas: (1) del diseño instruccional y (2) del diseño multimedial. Desde la perspectiva instruccional, toma decisiones acerca de: la descripción textual del contenido, el objetivo de aprendizaje, los contenidos, las actividades y autoevaluación. Desde la perspectiva multimedial, toma decisiones acerca de: el diseño de la interfaz, las estructuras de las

pantallas y definición la navegación para el OA. En la fase de Implementación, el docente construye la estructura general del OA planteado en la fase de Diseño. Para esto, utiliza herramientas de autor que permitan integrar cada uno de los elementos contemplados en la fase anterior. Esta integración se realiza mediante los instrumentos disponibles en la herramienta de autor seleccionada. Ya en la fase de Evaluación, se plantea la valoración del OA desde dos perspectivas: la del usuario y la del docente. Para la evaluación desde la perspectiva del usuario, se utiliza la rúbrica para la evaluación de la calidad denominada CUSEOA [22]. Mientras que para la evaluación desde la perspectiva del docente se propone aplicar la rúbrica denominada CODA [23]. Finalmente, en la fase de Publicación se pone el OA a disposición de los estudiantes por medio de un ROA.

B. Enfoque de Automatización

El enfoque de automatización propuesto en este trabajo plantea un asistente de software, consistente en DSLs (lenguajes), que soporte la aplicación de la metodología DICREVOA 2.0, facilitando la documentación de las decisiones tomadas durante el análisis, diseño e implementación de OA, así como generadores de código que automatizan la implementación de OAs.

MDE proporciona las bases para construir DSLs mediante meta-modelos, los cuales proveen un mecanismo unificado y expresivo para definir los conceptos del dominio. Los lenguajes están compuestos por: i) sintaxis abstracta (meta-modelo) que define el conjunto de conceptos (meta-classes) del lenguaje y sus relaciones, conjuntamente con reglas para combinarlos; conceptos utilizados para describir las decisiones tomadas en el dominio y documentadas en modelos; ii) sintaxis concreta, que define la notación que el usuario final utilizará para especificar programas o modelos conforme a la sintaxis abstracta (p. ej., una notación gráfica). Adicionalmente, el enfoque MDE hace uso de los modelos creados en un determinado lenguaje para mediante transformaciones de modelo a texto (M2T) generar código fuente.

En este contexto, en la siguiente sección, en primer lugar, se

define la sintaxis abstracta del lenguaje que permite describir las decisiones tomadas durante las fases de Análisis y Diseño de OA siguiendo la metodología DICREVOA 2.0, y de manera independiente de la herramienta de autor en que se implementará. En términos de MDE estas decisiones se documentarán en Modelos Independientes de la Plataforma (Platform Independent Model - PIM). Luego, se define la sintaxis abstracta del lenguaje que permitirá describir las decisiones tomadas durante la fase de Implementación, decisiones acerca de los instrumentos de las herramientas de autor utilizadas para implementar el OA, que en términos de MDE se documentarán en Modelos Específicos de Plataforma (Platform Specific Model - PSM).

Finalmente, se define la sintaxis concreta que serán las representaciones gráficas utilizadas en los editores gráficos del asistente. Con respecto a la generación de código, las decisiones tomadas durante el Análisis, Diseño e Implementación, y documentadas con el asistente, son utilizadas para automatizar la fase de implementación en la que se construye el OA. Sin embargo, este último punto no es cubierto en este trabajo.

IV. ASISTENTE PARA LA CONSTRUCCIÓN DE UN OA

DSLs ofrecen mediante abstracciones y notaciones apropiadas la expresividad necesaria para especificar un problema de dominio específico. Para identificar los conceptos que forman parte de la sintaxis abstracta del lenguaje, se utilizó la terminología que emplea DICREVOA 2.0.

A. Lenguaje para la Especificación del Análisis y Diseño de OA

La Figura 1, presenta la sintaxis abstracta del *Lenguaje para la Especificación del Análisis y Diseño de OA (PIM en términos de MDE)*, cuyos conceptos (escritos con letra cursiva) brindan la expresividad para especificar las decisiones tomadas durante las fases de Análisis y Diseño de manera independiente a la herramienta de autor en el que se implemente el OA.

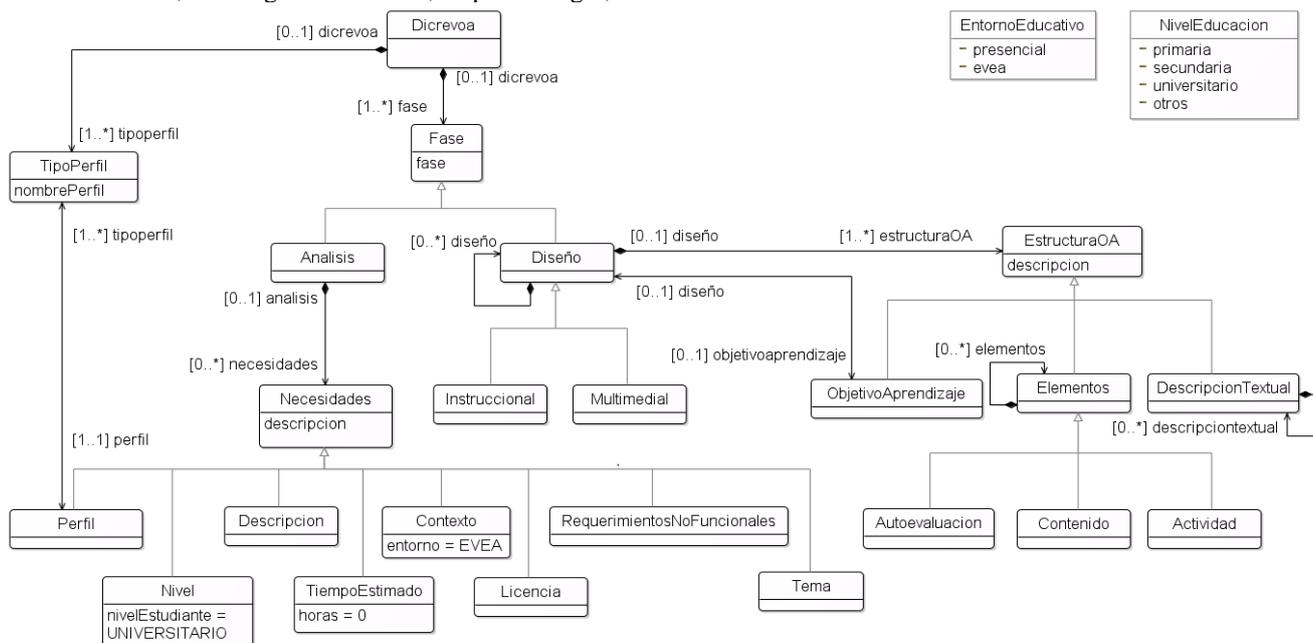


Figura 1. Extracto del Lenguaje para la Especificación del Análisis y Diseño de OA

Tal como propone DICREVOA 2.0 (ver sección III.A) el docente tendrá que proveer información acerca de las *Fases de Análisis y Diseño*. En donde, durante el *Análisis* especifica las *Necesidades* o requerimientos iniciales del OA; necesidades que se clasifican en: *Tema*, *Descripción* del contenido, *Nivel* definido por su *Nivel de Educación* (*PRIMARIA*, *SECUNDARIA*, *UNIVERSITARIO* y *OTROS*), *Perfil* del estudiante considerando su *Tipo de Perfil* definido por su estilo de aprendizaje y otras necesidades. Durante el *Diseño*, el docente especifica la estructura interna del OA (*Estructura del OA*). Debido a las limitaciones de espacio, en esta sección, solo se presentan los conceptos y atributos más relevantes.

B. Lenguaje para la Especificación de la Implementación de OA

La Figura 2 presenta la sintaxis abstracta del *Lenguaje para la Especificación de la Implementación de OA* (PSM en términos de MDE), el cual facilita la especificación de los instrumentos de visualización que se utilizarán para exponer los contenidos del OA. Los conceptos de este lenguaje se basan en la terminología utilizada en eXeLearning, esto debido a que DICREVOA 2.0 recomienda el uso de esta herramienta de autor. eXeLearning dispone de instrumentos llamados *iDevices*, que permiten introducir diferentes recursos didácticos, como, por ejemplo: texto libre, galería de imágenes, ficheros adjuntos, actividades de lectura, entre otros instrumentos que permitirán implementar el OA.

En la fase de Implementación el docente utiliza el *Lenguaje para la Especificación de la Implementación de OA* (ver Figura 2) para especificar los instrumentos (*IDevice*) que utilizará para exponer el contenido de los elementos de la estructura interna del OA agrupados en un contenedor (*Pagina*), descritos por el *Lenguaje para la Especificación del Análisis y Diseño de OA* (ver Figura 1). El concepto *EstructuraOA*, común en ambos lenguajes, permite establecer el mapeo entre elementos de la estructura interna de OA con los instrumentos que exponen su contenido.

Con este lenguaje, el docente podrá especificar los instrumentos que utilizará para exponer el contenido de un elemento de la *Estructura del OA*, pudiendo ser: *Actividades Interactivas*, *Actividades No Interactivas*, *Información Textual* e *Información Textual*. Por ejemplo, en caso de *Información Textual* podrá especificar el *Conocimiento Previo* que se requiere para comprender el contenido de ese elemento, una *Nota aclaratoria* de ese contenido y/o expresarlo en *Texto Libre*. Otro ejemplo se podría identificar cuando el docente, desee incluir preguntas para la *Autoevaluación* de la *Estructura del OA*, y podría elegir *Actividades Interactivas* para exponer las preguntas requeridas en la autoevaluación del OA, pudiendo escoger entre: *Preguntas Verdadero Falso*, *Pregunta de selección Múltiple*, *Cuestionario Scorm*, *Pregunta de Elección Múltiple*, *Rellenar Huecos* o *Actividad Desplegable*. En esta sección, solo se presentan los conceptos y atributos más relevantes.

C. Editor Gráfico

Con el propósito de definir la sintaxis concreta de los lenguajes

propuestos en este trabajo, se utilizó la herramienta Obeo Designer [24], un plug-in de Eclipse Basado en GMF (Graphical Modeling Framework) [25] para crear un editor gráfico. El editor gráfico del asistente se implementó siguiendo un diseño basado en múltiples pestañas organizadas de tal manera que permite al docente aplicar de manera sistemática las fases de la metodología DICREVOA 2.0. En donde, cada una de las pestañas, contiene un editor dedicado a describir las decisiones tomadas durante la ejecución de las fases de Análisis, Diseño e Implementación. El editor ayuda al docente a conocer la información que tiene que proveer durante cada una de las fases por lo que la barra de herramientas de cada pestaña incluye elementos correspondientes a cada paso de una fase. Por ejemplo, en la Figura 3 se puede apreciar las paletas de herramientas correspondientes al editor gráfico para el Lenguaje para la Especificación del Análisis y Diseño de OA. Cada herramienta de la figura permite especificar al docente las decisiones tomadas en cada paso del Análisis o Diseño.

V. CASO DE ESTUDIO ILUSTRATIVO

Con el objetivo de ilustrar la aplicabilidad del asistente para la automatización de la creación de OA utilizamos un ejemplo en el cual un docente requiere construir un OA que cubra la necesidad de transmitir conocimientos relacionados a la temática de Web Quest. Por lo que, el docente ha planteado crear un curso cuyo tema es “¿Cómo diseñar una WebQuest efectiva?”. Este ejemplo fue tomado de la documentación de la metodología DICREVOA 2.0¹ (Pag. 66 - 70); en donde, las decisiones tomadas por el docente durante las fases de Análisis, Diseño e Implementación fueron documentadas en herramientas ofimáticas. Hemos utilizado esa documentación para como información de especificación para ilustrar cómo el asistente soporta la aplicación sistemática y toma de decisiones en las diferentes fases.

Durante la Fase de Análisis el docente hace uso de la pestaña Análisis del editor gráfico del Lenguaje para la Especificación del Análisis y Diseño de OA (ver Figura 4) para documentar la ejecución de cada paso de la fase (ver Figura 4a), utilizando elementos de la paleta de herramientas correspondiente (ver Figura 4b). El docente especifica: i) El Tema que va a tratar el OA, “¿Cómo diseñar una WebQuest efectiva?”. ii) La Descripción Textual del OA, “El OA presenta información detallada de cómo planificar cada una de las etapas de una Webquest (Introducción, Tarea, Proceso, Recursos, Evaluación y Conclusión) ...”. iii) La descripción de los estilos de aprendizaje en los que se enfocará el OA, “El OA está dirigido a todos los docentes que deseen conocer y aprender a diseñar una WebQuest efectiva, por lo tanto ...”; además, toma decisiones acerca del Perfil del Estudiante al que favorecerá el OA, seleccionando el estilo de aprendizaje “ACTIVO o REFLEXIVO” (ver Figura 4C). iv) Nivel de educación al que está dirigido el OA, “Educación General Básica, Secundaria y Educación Superior”. v) Tiempo requerido para completar el OA, “El tiempo requerido para recorrer e interactuar con el Objeto de Aprendizaje es de 4 horas”. vi) Contexto

¹ Documentación de la Metodología DICREVOA 2.0:
<https://www.cedia.edu.ec/es/libro/objetos-de-aprendizaje>

educativo en el cual debería utilizarse el OA, “Los WebQuest al ser recursos didácticos atractivos, motivadores e interactivos, son de uso frecuente que pueden ser reutilizados en diferentes asignaturas...”. vii) Licencia que utilizará para su OA, “Licencia Creative Commons Atribución Compartir Igual 3.0 Ecuador”.

Finalmente, viii) Requerimientos no Funcionales de acuerdo a las necesidades que tenga el OA, “Requerimientos técnicos: Sistema operativo Windows o Linux, Navegador Internet ...”.

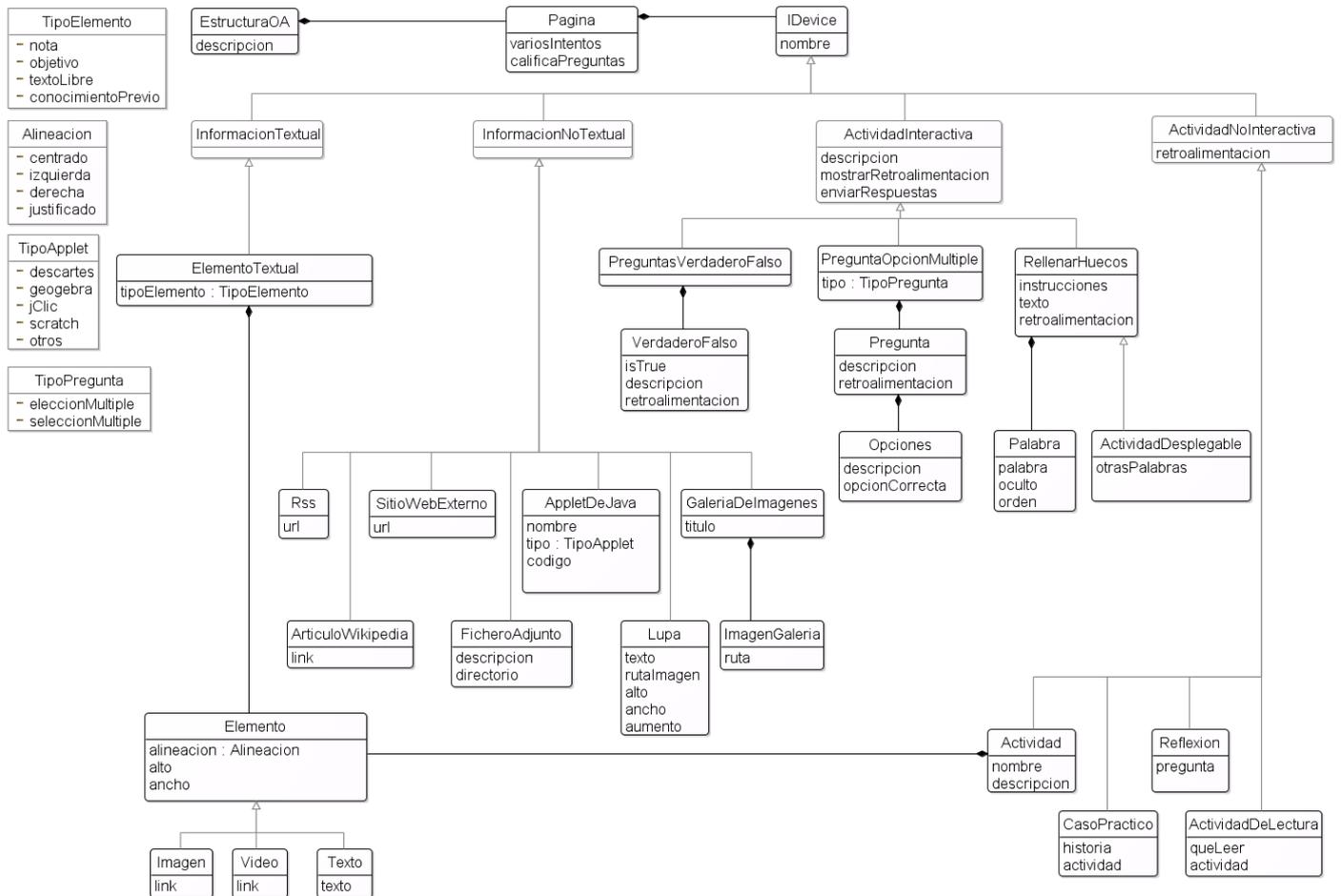


Figura 2. Extracto del Lenguaje para la Especificación de la Implementación de OA

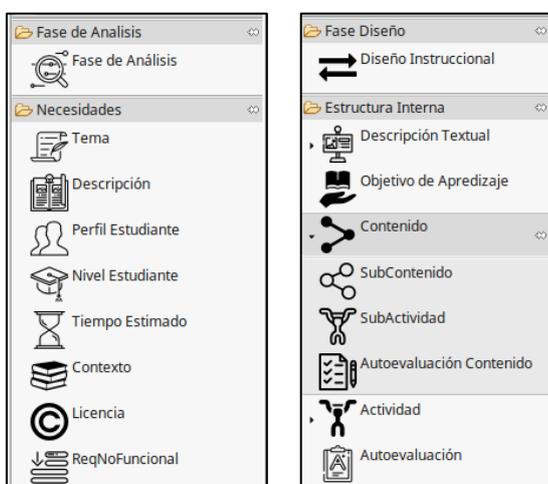


Figura 3. Paletas de herramientas - Análisis y Diseño

Durante la fase de Diseño el docente inicia por el Diseño Instruccional, haciendo uso de la pestaña Diseño Instruccional del editor gráfico del Lenguaje para la Especificación del Análisis y Diseño de OA para documentar la ejecución de cada paso de la fase (ver Figura 5a) utilizando elementos de la paleta de herramientas correspondiente que le permiten especificar la estructura interna del OA (ver Figura 5b). El docente especifica entre otros: La Descripción Textual de cómo se estructurará el contenido del OA, el Objetivo de Aprendizaje que cubrirá con el desarrollo del OA, la descripción global del Contenido a tratar, Actividades y Autoevaluación que incluirá el OA para cumplir con el Objetivo de Aprendizaje planteado.

Una vez finalizado el Diseño Instruccional, el docente hace uso de la pestaña de Diseño Multimedia del editor gráfico del Lenguaje para la Especificación del Análisis y Diseño de OA para continuar con las tareas propias del Diseño Multimedia.

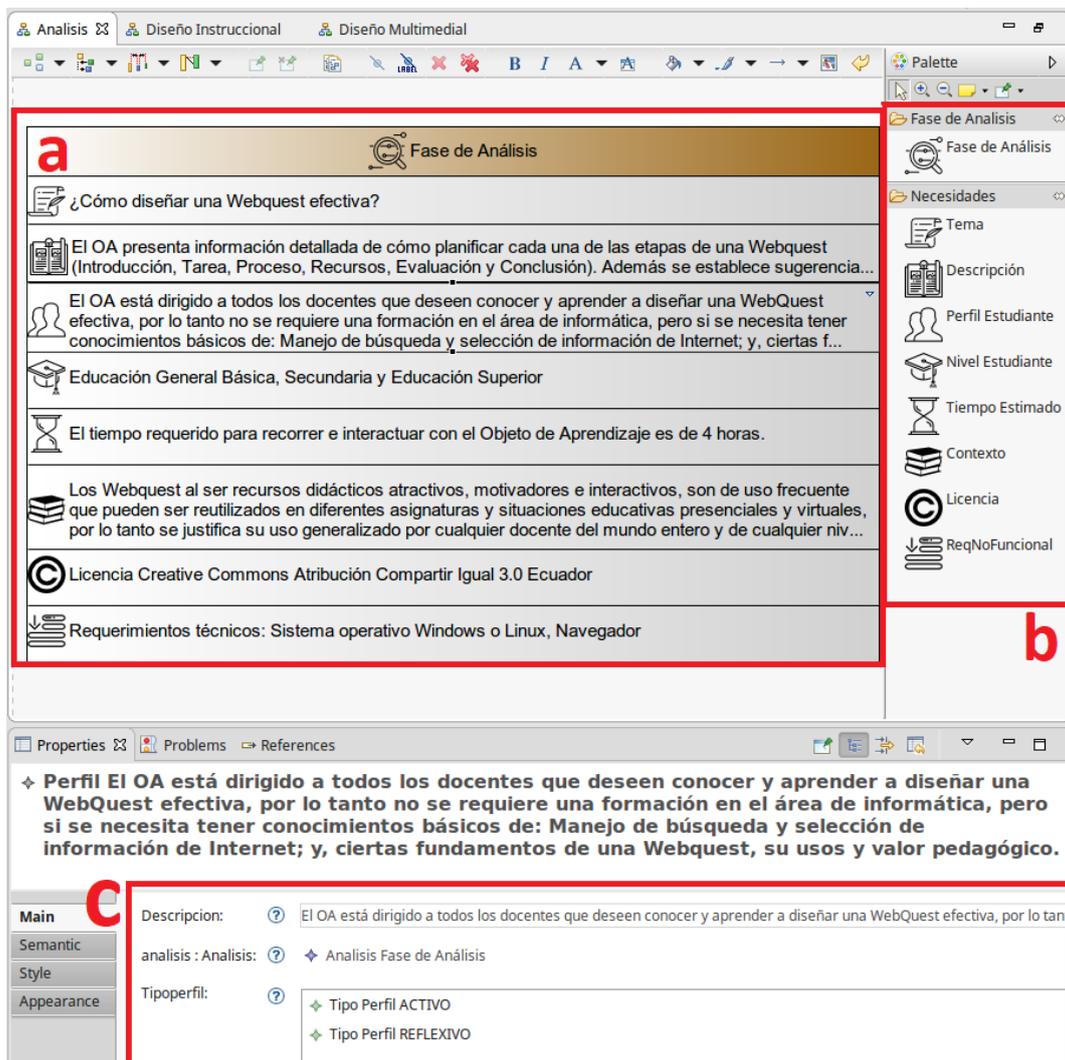


Figura 4. Editor gráfico del asistente – Pestaña Fase de Análisis.

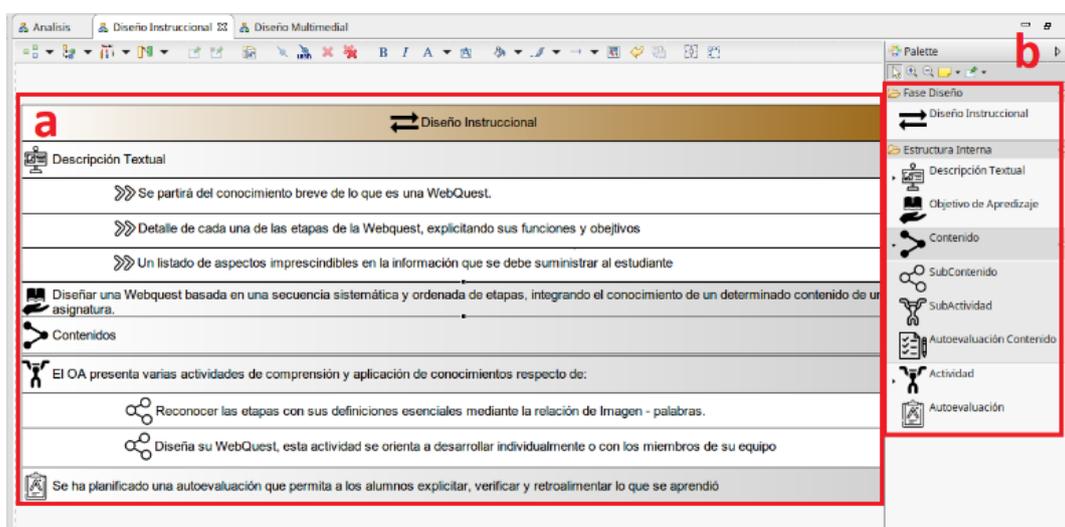


Figura 5. Editor gráfico del asistente – Pestaña Fase de Diseño

Finalmente, durante la fase de Implementación, el docente hace uso del editor gráfico del Lenguaje para la Especificación de la Implementación de OA que le permite especificar los instrumentos que utilizará para exponer el contenido de la estructura interna del OA. Además, ejecuta transformaciones M2T para generar un esqueleto de la OA en eXeLearning.

VI. CONCLUSIONES

En este trabajo se ha presentado un asistente de software que facilitará a los docentes la ejecución sistemática y documentación de las fases del proceso de análisis, diseño e implementación del OA dejando de lado los aspectos tecnológicos de las herramientas involucradas para su implementación. Para esto la solución propuesta cuenta con editores gráficos que facilitan la documentación de las fases de análisis y diseño (de manera independiente de la tecnología. Además, los editores gráficos asistirán al docente en las decisiones tecnológicas. La documentación capturada en estos editores gráficos servirá de insumo para generar el código que implementa el OA, de acuerdo al estilo de aprendizaje y al Sistema de Gestión de Aprendizaje.

La factibilidad del asistente se ha mostrado mediante un caso de estudio en el que un docente tiene que ejecutar las fases de análisis, diseño e implementación; todas ellas soportadas por el asistente. Además, en base a las decisiones documentadas con los lenguajes propuestos en el asistente se generó el código correspondiente a la implementación (esqueleto del OA en eXeLearning). La solución presentada en este trabajo puede ser utilizada para soportar otros escenarios de creación de OA mediante la creación de generadores de código acordes al escenario. Por ejemplo, generadores de código que produzcan diferentes versiones del código que implementa OA accesibles; en donde cada versión depende de la discapacidad descrita en el perfil de los estudiantes.

Como trabajo futuro se espera validar empíricamente esta propuesta y extenderla con generadores de código que supere la limitación actual de generar esqueletos del código implementación.

AGRADECIMIENTOS

Este trabajo es soportado por el proyecto DIUC_XIV_2016_038. Además, se agradece a Franklin Bernal y Gabriel Loja, estudiantes de la Escuela de informática de la Universidad de Cuenca quienes participaron en la creación del Lenguaje para la Especificación de la Implementación de OA (ver Figura 2).

REFERENCIAS

[1] I. Pianucci, M. Chiarani, y M. Tapia, «Elaboración de materiales educativos digitales», *1er Congr. Int. Punta Este*, 2010.

[2] Maldonado, J., & Astudillo, G. (2014). Los Objetos de Aprendizaje: Un estado del arte en Iberoamérica. In VI Conferencia Iberoamericana sobre Tecnologías y Aprendizaje, Miami, Estados Unidos.

[3] L. B. Mora, B. S. Vásquez, y G. E. Santos, «Objetos de Aprendizaje en la enseñanza de las Matemáticas», vol. 4, n.º 10, pp. 550–558, 2017.

[4] E. Urrutia, F. Urrutia, A. Larrea, y T. S. Antonio, «Construcción de objetos virtuales de aprendizaje aplicando ingeniería de software», *MASKANA CEDIA 2015*, pp. 209–217, 2015.

[5] J. M. Prado, *Impacto de la incorporación de objetos de aprendizaje en el desarrollo de competencias tecnológicas en secundaria*. Mexico: Instituto Tecnológico y de Estudios Superiores de Monterrey, 2015.

[6] E. Parra, «Methodology Proposal of Software Development for Virtual Learning Objects MESOVA», *Rev. Virtual Univ. Católica Norte*, vol. 1, pp. 113–137, 2011.

[7] N. Bilorou, *Guía para la evaluación de impacto de la formación*. Montevideo, UR: OIT/Cintefor, 2011.

[8] J. Maldonado, J. Bermeo, y F. Vélez, *Diseño, Creación y Evaluación de Objetos de Aprendizajes, Metodología DICREVOA 2.0*. Cuenca: CEDIA, 2016.

[9] J. Maldonado, J. P. Carvallo, y J. Sigüencia, «Metodologías y Propuestas Metodológicas Para el Diseño de Objetos de Aprendizaje: Un Estado del Arte en Iberoamérica», *An. Temporários LACLO 2015*, vol. 10, n.º 1, p. 36, oct. 2015.

[10] J. J. Maldonado, J. L. Bermeo, y G. Pacheco, «Assessing a Methodological Proposal for the Design, Creation and Evaluation of Learning Objects Oriented to Educators with Diverse Educational and Technological Competencies», *CLEI Electron. J.*, vol. 19, n.º 1, pp. 3-3, abr. 2016.

[11] Maldonado, J. J., Fernández-Pampillón, A. M., & Sanz, C. V. (2015, June). Analysis framework for tailored selection of learning objects methodologies. In Collaboration Technologies and Systems (CTS), 2015 International Conference on (pp. 148-158). IEEE.

[12] Booch, G., InFrankel, D. y Parodi J, «The MDA Journal: Model Driven Architecture Straight from the Masters», *MDA Manif.*, 2004.

[13] «Desarrollo Dirigido por Modelos (MDD) en el Contexto Educativo | Pascuas Rengifo | Scientia et technica». [En línea]. Disponible en: <http://revistas.utp.edu.co/index.php/revistaciencia/article/view/9321>. [Accedido: 17-abr-2018].

[14] F. Souza, J. Castro, y R. Andrade, *Model-Driven Development in the Production of Customizable Learning Objects. IEEE International Conference. Advanced Learning Technologies (ICALT)*, 2010.

[15] H. Wang, D. Zhang, y J. Zhou, «MDA-based development of elearning system», *Comput. Softw. Appl. Conf. Proc. 27th Annu. Int.*, 2003.

[16] Z. Bizonova, D. Ranc, y M. Drozdova, *Model driven e-learning platform integration*. 2008.

[17] C. Montenegro, P. Gaona, J. Cueva, y O. Sanjuan, «Aplicación de ingeniería dirigida por modelos (MDA), para la construcción de una herramienta de modelado de dominio específico (DSM) y la creación de módulos en sistemas de gestión de aprendizaje (LMS) independientes de la plataforma», *DYNA Univ. Nac. Colomb.*, vol. 78, n.º 179, pp. 43–52, 2011.

- [18] J. Doderó, A. Martínez, I. Ruiz, C. Cornejo, y A. León, «FlexoLD: Un Lenguaje Específico de Dominio para Diseños de Aprendizaje», *Actas Los Talleres Las Jorn. Ing. Softw. Bases Datos*, vol. 4, n.º 3, 2010.
- [19] J. Malek, M. Laroussi, A. Derycke, y H. B. Ghezala, «Model-Driven Development of Context-aware Adaptive Learning Systems», *10th IEEE Int. Conf. Adv. Learn. Technol.*, 2010.
- [20] M. G. Silva, E. F. Barbosa, y J. Maldonado, «Model-Driven Development of Learning Objects», *41st ASE/IEEE Front. Educ. Conf.*, 2011.
- [21] J. Maldonado, Z. Sanz, y Fernández-Pampillón, «Desarrollo de un marco de análisis para la selección de metodologías de diseño de objetos de aprendizaje (oa) basado en criterios de calidad para contextos educativos específicos», *Univ. Nac. Plata*, pp. 79–110, 2015.
- [22] S. Massa, M. DeGiusti, y Pesado, *Métodos de evaluación de usabilidad: una propuesta de aplicación en Objetos de Aprendizaje*. In XIV Workshop de Investigadores en Ciencias de la Computación, 2012.
- [23] C. Fernández-Pampillón, A. D. Romero, y I. A. Ranero, *Herramienta de Evaluación de la Calidad de Objetos de Aprendizaje Universitarios (COdA): Guía para la producción y evaluación de materiales didácticos digitales. v. 1.1*. 2011.
- [24] Juliot, E., Benois, J.B., «Viewpoints Creation using Obeo Designer or how to Build Eclipse DSM without being an Expert Developer?», *Obeo Whitepaper*, 2010.
- [25] «Graphical Modeling Framework | The Eclipse Foundation». [En línea]. Disponible en: <http://www.eclipse.org/modeling/gmp/>. [Accedido: 23-abr-2018].