

# Una propuesta de Marco para la Construcción de Contenidos Digitales de Aprendizaje con prácticas ágiles

## MARCODA

Carlos Arturo Castro  
Castro  
USB  
Medellín - Colombia  
carlos.castro@usbmed.edu.co

Eucario Parra  
Castrillón  
USB  
Medellín - Colombia  
ingenieria.sistemas@usbmed.edu.co

Wilder Perdomo  
Charry  
USB  
Medellín - Colombia  
wilder.perdomo@usbmed.edu.co

Agustín Lagunes  
Domínguez  
UV  
Veracruz - México  
aglagunes@uv.mx

### ABSTRACT

This article presents a proposed framework for the construction of Learning Digital Contents with agile practices, using defined stages as the cycle of life and activities that express the philosophy of agile.

The Framework for Construction of Learning Digital Contents MARCODA has seven principles, included a software product that can be deployed in WEB environment as a learning content, presented as a problem whose solution facilitates learning, where the presence of users (teachers, students and thematic experts) is essential on every phases of the project construction.

### Resumen

Este artículo presenta una propuesta de Marco para la Construcción de Contenidos Digitales de Aprendizaje con prácticas ágiles con fases bien definidas en cuanto al ciclo de vida y actividades que expresan la filosofía del agilísimo.

Marco para la Construcción de Contenidos Digitales de Aprendizaje Denominado MARCODA, parte de 7 principios que incluyen el hecho de considerar un contenido de aprendizaje como un producto de software que puede ser desplegado en ambiente WEB, presentado como un problema cuya solución facilita el aprendizaje, en el que la presencia de los usuarios (profesores, estudiantes y expertos temáticos), es esencial en todas las fases del proyecto de construcción.

### Palabras clave

Ciclo de vida, Contenidos de Aprendizaje, Diseño software, Ingeniería de Software Educativo, Agilismo

### Categories and Subject Descriptors

D.2.4 [Software Engineering]: software design.

### General Terms

Verification, Management, Reliability, Software Design

### Keywords

Life cycle, learning content, software design, engineering educational software, software agile.

### INTRODUCCIÓN

MARCODA es un marco para desarrollar contenidos digitales para el aprendizaje y esencialmente recomienda unas fases específicas. Pero además la filosofía general de trabajo se fundamenta en buenas practicas agiles. Es de anotar que aunque el agilismo tiene una amplia aplicación en la industria del software, su aplicación es viable en otro tipo de proyectos.

El término Agilismo fue introducido por la industria del software como metodología para alcanzar desarrollos económicos, de calidad y simplicidad, y a través de equipos de trabajo muy organizados y con buena comunicación con los usuarios. En 2001 un grupo de agilitas liderados por Kent Beck expusieron el “Agile Manifesto and Agile Principles” [1], en el que se incluyó el término “Métodos Ágiles” en oposición a los métodos muy documentados, normatizados y muy formales que hacen pesados los desarrollos. Es de anotar que los métodos ágiles no implican ausencia de controles ni se planificación de eventos.

Las buenas practicas agiles pueden aplicarse a la ingeniería de software educativo y con esto se puede lograr que los contenidos para el aprendizaje adquieran las dimensiones estructurales del software como producto, la metodología se optimice en cuanto recursos de tiempo y esfuerzo, se alcance mayor efectividad en términos de logros pedagógicos y se organice la participación continua de los usuarios en las distintas fases y no solamente al principio y al final del proyecto, como suele ocurrir. Además, el agilismo aporta practicas simples que pueden entrar a corregir tendencias en el desarrollo de contenidos digitales para el aprendizaje que se enfocan en los guiones de multimedia y en la integración de aplicaciones, pero sin criterios de calidad en los procesos.

En cuanto a las prácticas ágiles aceptas y aplicadas con gran éxito en la industria del software [2] y que pueden ser acogidas para la construcción de contenidos digitales de aprendizaje se tienen:

Definir plan de entregables por iteraciones de corta duración (2 a tres semanas), priorización de requerimientos para cada iteración, estimar tiempo y recursos requeridos para cada iteración, reuniones informativas en períodos cortos y corto tiempo (eje: 15 minutos diarios), presencia de usuarios por iteración (profesores,

estudiantes, expertos temáticos), verificación, validación y evaluación por iteración, codificación con estándares.

Un contenido digital didáctico, debe tener una connotación de interactivo, dado que esta característica permite la comunicación recíproca entre el ser humano que aprende y el “software que enseña”.

Un contenido interactivo ofrece información personalizable en función de las necesidades y exigencias de los usuarios [3], y conlleva las siguientes características que aumentan los niveles cognitivos o posibilidades de aprendizaje y comprensión de los usuarios:

El flujo de información en múltiples sentidos (información en varias direcciones).

El rol activo del usuario en la selección de la información requerida.

Una interfaz enriquecida con elementos multimedia que mejora el nivel de comunicación.

Para que la información final sea de calidad, práctica y transparente, un departamento de contenidos interactivos debe ofrecer una gama de servicios añadidos tales como:

Búsqueda, clasificación y digitalización de la información; documentación y referencias para proyectos; mantenimiento y actualización de contenidos; traducción; estos servicios se deben conjugar con características esenciales que le aporten calidad a los contenidos interactivos: actualidad, profundidad, originalidad y fiabilidad.

La Ingeniería de software Educativo se define como el estudio de los conceptos, métodos, modelos, técnicas y herramientas para facilitar el análisis, diseño, producción, implementación, evaluación y prueba de productos de software destinado a mejorar procesos de aprendizaje [4]. Según se cita en [5], se han estructurado diferentes propuestas de Ingeniería del Software para desarrollar contenidos de aprendizaje, expresadas en fases, actividades, acciones, estrategias y prácticas: Gagné [6], Bianchini [7], Dorrego [8], Daziel [9], Marina Polo [10], Boyle [11], Strijker [12], Azpeitia et al [13], C. Galvis, Gómez y Mariño [14], Greune et al [15], Peláez y López [16].

Orjuela y Rojas [17] proponen prácticas extraídas de metodologías como XP, SCRUM y CRYSTAL para adaptarlas al contexto de la ingeniería del software educativo. En la fase de planeación, por ejemplo, proponen agregar un nuevo rol llamado "Un experto en Informática educativa: Es la persona encargada de emitir conceptos relacionados con los modelos pedagógicos apropiados para la construcción de software educativo" y como política agregan los aspectos educativos se deben contemplar ítems como el modelo pedagógico que debe soportar el software educativo articulado con el uso de las **nuevas tecnologías**". En la fase de análisis no incluyen elementos diferenciadores con los tradicionales y hace énfasis en el refinamiento de los casos de uso para especificar los requisitos funcionales. La fase de diseño integra el diseño educativo, diseño de comunicación y diseño computacional. El modelo propuesto finaliza con las fases de implementación (traducción de los diseños en código fuente), pruebas (pruebas piloto y pruebas de campo con potenciales usuarios del sistema) y evaluación (medir el nivel de cumplimiento de los objetivos).

Vilca y otros [18] proponen un enfoque metodológico que incluye cuatro etapas; a) Identificación de requerimientos (análisis de necesidades educativas y culturales); b) construcción del prototipo; c) evaluación del prototipo y d) Entrega del producto.

Benigni y otros [19] proponen modelos mentales y conceptuales en el análisis de requerimientos para desarrollar software educativo como modelos mentales de actores, mapas mentales, diagramas entidad relación y diagramas de casos de uso. Las autoras hacen énfasis en que es necesario emplear el uso de una metodología de desarrollo de software para cualquier tipo de proyecto informático que se desee llevar a cabo.

En cuanto a las prácticas ágiles aceptas y aplicadas con gran éxito en la industria del software [2] y que pueden ser acogidas para la construcción de contenidos digitales de aprendizaje se tienen:

Definir plan de entregables por iteraciones de corta duración (2 a tres semanas), priorización de requerimientos para cada iteración, estimar tiempo y recursos requeridos para cada iteración, reuniones informativas en períodos cortos y corto tiempo (eje: 15 minutos diarios), presencia de usuarios por iteración (profesores, estudiantes, expertos temáticos), verificación, validación y evaluación por iteración, codificación con estándares.

En la investigación presentada por Serna, Castro y Botero [20], la cual se basa en la Ingeniería de Software para Desarrollar Objetos de Aprendizaje (ISDOA), incluye 4 fases : a) Análisis e ingeniería de requisitos; b) Diseño, c) Desarrollo e implementación y d) evaluación de vida útil, las cuales están soportadas en procesos de mejora continua mediante verificación y validación (plan de pruebas) y evaluación de la calidad, como se observa en la Figura 1

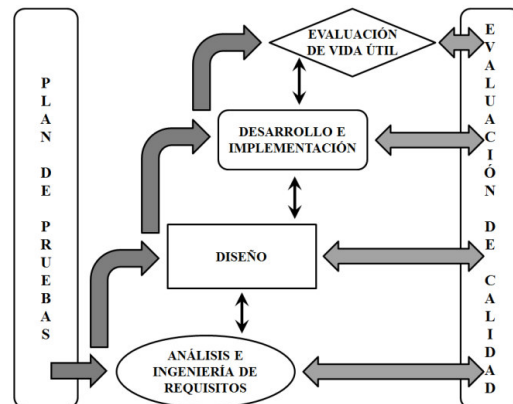


Figura 1. Ciclo de vida ISDOA [20]

A su vez, Parra [4] ha desarrollado y aplicado la Metodología de Desarrollo de Software para Objetos Virtuales de Aprendizaje (MESOVA). El ciclo de vida de esta metodología propone un flujo del ciclo de vida que incluye conceptos de los modelos en espiral, incremental y evolutivo, dándose especial importancia a la construcción ágil de prototipos. Esta investigación parte de diez principios, que incluyen elementos y prácticas en torno a la temática de los contenidos, pedagogía y didáctica, funcionalidad, documentación, diseño modular, usabilidad, calidad y pruebas de aprendizaje (evaluación del aprendizaje). MESOVA se compone

de cinco fases: a) Concepción del objeto; b) Diseño y desarrollo modular evolutivo; c) Integración y despliegue; y d) Pruebas de aprendizaje y consolidación. Adicionalmente propone unos puntos de control para evaluar y validar cada una de las fases, incluyendo la evaluación experimental del aprendizaje. En la figura No 2 se resume MESOVA

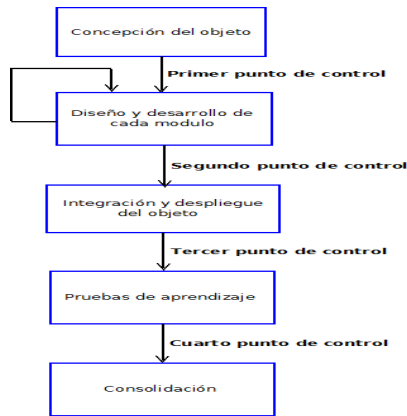


Figura 2. Fases de la metodología MESOVA [4]

Las metodologías MESOVA e ISDOA coinciden en incluir el aprendizaje basado en problemas (ABP) como estrategia didáctica para lograr los objetivos de aprendizaje y en la aplicación rigurosa de un ciclo de vida de desarrollo software (ingeniería de requisitos, diseño, desarrollo, verificación, validación y evaluación) para lograr contenidos manejados como componentes de software con calidad medible tanto desde los aspectos técnicos como didácticos y pedagógicos. No obstante ambas propuestas coinciden en que la rigurosidad de sus fases puede bloquear el desarrollo ágil y flexible que en momentos dados pueden ser puntos críticos de los proyectos.

En MESOVA se resalta un proceso de evaluación iterativa tanto de los contenidos como de los logros de aprendizaje, mediante la selección de grupos experimentales y de control, diseño de instrumentos de aplicación, elección de los ambientes adecuados y aplicación de pruebas de aprendizaje. En ISDOA se resalta la definición y evaluación de la vida útil del objeto de aprendizaje, así como la decisión de actualización o retiro, igualmente se resalta el proceso iterativo y evolutivo transversalizado por actividades de verificación y validación y evaluación de la calidad. Pero en ambos casos es carente la participación continua de los usuarios en el desarrollo del proyecto, tal como lo plantean las metodologías ágiles.

## 1. PROPUESTA DE MARCO PARA CONTENIDOS DIGITALES DE APRENDIZAJE

En este trabajo se presenta una propuesta el Marco para la Construcción de Contenidos Digitales de Aprendizaje con práctica ágiles (MARCODA) . El objetivo es proponer un conjunto de fases con actividades y prácticas ágiles que integran los elementos fundamentales de metodologías MESOVA e ISDOA dentro una estructura que contempla cuatro fases, cada una de las cuales está soportada mediante prácticas de verificación,

validación y evaluación experimental. Lo fundamental de esta integración es que se logra un proceso más delgado, ágil y adaptable al diseño y construcción de cualquier contenido digital de aprendizaje,

Como referente importante, MARCODA asume conceptualmente la interacción de los contenidos digitales como un elemento crucial para los efectos pedagógicos. Un contenido digital no necesariamente ha de ser interactivo. Un contenido digital didáctico, valga decir con propósitos educativos, debe tener una connotación de interactivo, dado que esta característica permite la comunicación recíproca entre el ser humano que aprende y el “software que enseña”.

Según Moya [21], los contenidos digitales son materiales multimedia que permiten a los participantes del proceso de aprendizaje buscar, manipular y contrastar la información, apoyados en la colaboración, la participación, la cooperación y la creatividad que proporciona el aprendizaje en los entornos digitales o en la red.

Estos servicios se deben conjugar con características esenciales que le aporten calidad a los contenidos interactivos: actualidad, profundidad, originalidad y fiabilidad.

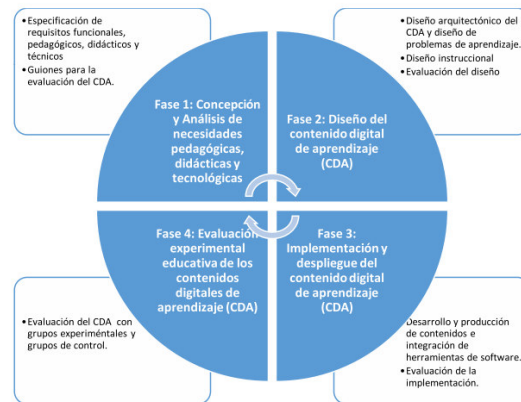


Figura No 3. Fases MARCODA (Elaboración propia)

Es de anotar en la segunda fase de MARCODA se realizan las actividades de lo que se conoce como el diseño instruccional en los proyectos recursos tecnológico y software para la educación. Se entiende el concepto de diseño instruccional, como la planeación de las secuencias de interacciones y actividades que orientaran los aprendizajes [22].

Considerando como base fundamental los conceptos anteriores sobre agilidad en la metodología e interactividad en los productos, MARCODA se ha aplicado en proyectos de la Universidad de San Buenaventura Medellín, de acuerdo con las siguientes fases:

### Fase 1: Concepción y análisis de necesidades pedagógicas, didácticas y tecnológicas.

En esta fase se hace un análisis de los requisitos del Contenido Digital de Aprendizaje (CDA) en relación con su funcionalidad y navegabilidad (Aspectos relacionados con la calidad del

software), su intencionalidad pedagógica y el contexto de ejecución y uso. Se deben considerar las siguientes actividades:

**a) Definición de nivel educativo, temáticas y competencias**

**El nivel educativo hace referencia a la posición formativa de los usuarios CDA.** Puede ser preescolar, básica, media, superior, empresarial, adultos u otras. Las temáticas hacen referencia a los temas de interés en los que se han identificado necesidades u oportunidades de aprendizaje. A su vez, en la definición de las competencias deben hacerse especificaciones en las categorías interpretativas, argumentativas y propositivas.

**b) Definición de la intencionalidad pedagógica y la didáctica con base en estrategias ABP [23]**

En este punto el equipo de desarrollo del CDA, debe definir la propuesta pedagógica que orientara los aprendizajes esperados en los estudiantes, dentro del marco del Aprendizaje Basado en Problemas. Un punto esencial es la identificación de los estilos de aprendizaje que se consideran en el diseño posterior de los problemas de aprendizaje. [24]; [25].

**c) Definición de actores y roles** (para análisis pedagógico, producción multimedial, desarrollo de software, actores para el grupo experimental y de control, equipo evaluador y equipo evaluador del contenido digital). La selección del equipo de actores depende de la naturaleza de los contenidos a desarrollar y de la modalidad del ambiente de aprendizaje (virtual totalmente, bimodal o presencial totalmente).

**d) Selección de la tipología de contenido.**

Dependiendo de aspectos como el nivel educativo, los estilos de aprendizaje y la temática a desarrollar, se debe tomar la decisión sobre el formato del contenido (por ejemplo, videos, páginas Web, simuladores, juegos, avatares, entre otros).

Aspectos fundamentales en esta decisión son los relacionados con la disponibilidad y accesibilidad de las herramientas de producción y desarrollo para los constructores, así como la facilidad para la navegación y acceso remoto por parte de los estudiantes.

**e) Definición del contexto de la vida útil del CDA.**

Es fundamental hacer análisis sobre la vigencia del contenido en cuanto a actualidad de las temáticas, periodos estimados de uso, naturaleza del grupo de usuarios que utilizaran el CDA, obsolescencia de la tecnología o la llegada de nuevas tecnologías para

repotenciar la capacidad del contenido. El equipo de desarrollo ha de definir cuál será la vida útil esperada del CDA.

**f) Definición de guiones para la evaluación del CDA**

Estos guiones son los instrumentos con los cuales se evaluará el proceso de construcción del CDA. En la fase del análisis inicial, deben considerarse las características, atributos y métricas que integraran los guiones con los cuales se evaluará: la calidad del diseño, la calidad de la producción y la calidad del contenido del CDA en uso. Es imprescindible incluir en estos guiones, conceptos sobre pedagogía y didáctica y desde el punto de vista tecnológico, lo relacionado con usabilidad, funcionalidad, fiabilidad, eficiencia, mantenibilidad y portabilidad (Estándar internacional de calidad del software, ISO 9126).

**g) Elaboración y actualización del plan de desarrollo del software.**

En este punto se describen los principales requerimientos funcionales y no funcionales, con base en reuniones cortas entre usuarios e interesados y documentación asociada a las actividades anteriores.

Inicialmente, el plan de desarrollo de software se estructura de tal manera que quede expresada una vista general del proyecto (Propósito, Alcance y Objetivos, Suposiciones y Restricciones, Entregables del proyecto, Glosario), estructura organizacional del proyecto (Participantes en el Proyecto, Interfaces Externas, Roles y Responsabilidades) y gestión del proceso (estimaciones, gestión de cambios, gestión de riesgos, plan de iteraciones, cronograma y acuerdos sobre la forma de seguimiento y control del proyecto). Una herramienta fundamental para la gestión del proyecto de desarrollo del CDA, es el cronograma. Este debe ser claro y simple, presentando las iteraciones y los tiempos estimados para obtener el producto en cada iteración. Si bien el cronograma debe visualizar la fecha de inicio y la fecha de fin del proyecto y todas las actividades a desarrollar, los tiempos estimados para cada iteración deben ser incluidos cada vez que termine la iteración anterior, de manera adaptativa y flexible. El plan de desarrollo de software se va actualizando en la medida que se cumplan los entregables en cada iteración, hasta convertirse en un documento visión en el que los requisitos funcionales y no funcionales son suficientemente claros, aunque se presenten cambios en cada una de las iteraciones. En cuanto a los requisitos no funcionales debe haber claridad en cuanto a estándares aplicables, requisitos de sistema, requisitos de desempeño, requisitos de entorno y requisitos de documentación (manual de usuario, ayudas en línea, guías de instalación y configuración) entre otros.

Los requisitos funcionales pueden ser descritos y especificados mediante historias de usuario y/o casos de uso.

En esta fase el desarrollo ágil implica que las iteraciones (espacios temporales) sean cortas, sostenidas, con entregables concretos y con la participación programada del líder del proyecto, el experto en ABP, el experto temático, el experto tecnológico y el docente que representa a los usuarios finales. Una buena práctica ágil es que los guiones de evaluación del CDA sirvan de mecanismo interno de validación del análisis de requisitos que se vaya haciendo.

## Fase 2. Diseño del contenido digital de aprendizaje (CDA)

En esta fase se cristalizan dos procesos fundamentales: el diseño tecnológico y el diseño instruccional del contenido. En ambos casos la planeación debe estar en coherencia con los conceptos y contextos definidos en la Fase 1. Según esto, el diseño instruccional está concebido como la guía de

los aprendizajes utilizando del CDA, pero no como un proceso aislado sino por el contrario, ubicado dentro de un marco de análisis de requisitos pedagógicos y tecnológicos y en coherencia con el ambiente de aprendizaje de los usuarios.

Las actividades que se consideran en esta fase son las siguientes:

### a) Elaboración de un mapa global del CDA

Se ha de presentar una propuesta esquemática general del diseño de interfaces, sin entrar en detalles funcionales. El mapa debe mostrar de manera global, como será la ruta de navegación y la interacción entre los usuarios y el CDA. Este mapa debe ser validado por los distintos roles de equipo de desarrollo, de acuerdo con los guiones de evaluación definidos en la fase 1.

### b) Diseño del problema o problemas que orientarán los aprendizajes (Diseño instruccional).

Una vez definidos los contenidos, los objetivos de aprendizaje, las competencias, los elementos pedagógicos y didácticos, así como los estilos de aprendizaje (fase 1), se estructuran los problemas (puede ser uno solo) que permitan acercamiento a los contenidos y que permita la adquisición de aprendizajes significativos. Es de anotar que el problema o problemas como tal, ameritan de unos protocolos de diseño que los debe definir el equipo de trabajo.

### c) Diseño de prototipos y arquitectura específica modular

Con base en el mapa del diseño global del contenido, se planean las diferentes versiones que se llaman prototipos (en ingeniería del software, un prototipo es una versión incompleta de un programa). Cada uno de estas versiones contempla aspectos didácticos, pedagógicos y arquitectónicos (estructura modular), que serán mejorados en la medida en que los guiones de evaluación avancen. Es decir, el proceso es evolutivo e

iterativo (cíclico), donde cada iteración corresponde a una versión del prototipo, hasta la versión final del CDA

### d) Aplicación de guiones para la evaluación del diseño

Se debe documentar de manera ágil, los procesos de aplicación de los guiones del diseño, hasta llegar a la versión finalmente aprobada. Las plantillas para la documentación deben ser definidas por el equipo de desarrollo. La evaluación debe garantizar que el diseño cumple con los propósitos planteados en la fase 1.

El desarrollo ágil implica unas iteraciones con las mismas características de la fase 1. Además, deben organizarse controles sobre la viabilidad del diseño teniendo como base los guiones de la fase 1 y los guiones del diseño. Una buena práctica ágil para darle fluidez al proyecto, es la participación en el proyecto de expertos en temas de arquitectura y calidad del software.

## Fase 3. Implementación y despliegue del contenido digital de aprendizaje (CDA)

Esta es una fase de producción, construcción e integración de herramientas de software, lenguajes y productos de multimedia, todo con el cuidado del fácil acceso y la navegabilidad para el estudiante usuario del contenido. Esto es, la producción debe considerar el ambiente de uso del estudiante en cuanto a la tecnología disponible. En esto es de especial importancia tener en cuenta:

- a) Selección de herramientas para construcción de los contenidos.
- b) Aplicación de herramientas para la integración de los contenidos.
- c) Despliegue del CDA en los ambientes de los usuarios (estudiantes y profesores).
- d) Aplicación de guiones para la evaluación del contenido.
- e) El contenido que se vaya construyendo de acuerdo con un plan de versiones de prototipos, debe irse evaluando de acuerdo con los guiones definidos en la fase 1.

En esta parte una buena práctica ágil consiste en considerar los resultados de las fases 1 y 2, las demandas de los guiones de evaluación del contenido y la participación activa del experto tecnológico que proceda bajo los principios siguientes (hasta donde lo permita la naturaleza del proyecto): mejor lo más fácil; aplicar lo que ya existe; integrar en interfaces antes que crear interfaces. Igualmente, deben conservarse las características de las interacciones descritas en la fase 1.

## Fase 4. Evaluación experimental educativa del CDA)

El contenido ya terminado debe evaluarse desde dos criterios: adaptando técnicas de validación y verificación para la calidad del software como usabilidad, funcionalidad o portabilidad (características externas) y experimentando en el ambiente natural de los usuarios finales, básicamente profesores y estudiantes (efectividad del contenido).

El modelo clásico de la investigación experimental comprende la selección de dos grupos de sujetos, según Rodríguez [26]: a) El grupo experimental (GE), constituidos por quienes están sometidos al tratamiento experimental; b) El grupo de control (GC), es un conjunto equivalente al anterior, el cual está también sujeto a observación durante el experimento, pero sin la manipulación de variables independientes. La comparación con el grupo de control permite establecer hasta donde la manipulación alteró las variables dependientes del grupo experimental.

Los estudios no aleatorizados o cuasi-experimentales, se caracterizan en que la muestra no se escoge en forma aleatorizada. Algunas de sus características son las siguientes: a) Intervención en grupos humanos en los cuales la presencia (las condiciones de la exposición del sujeto) no se puede manipular plenamente; b) Hay variables dependientes e independientes e hipótesis a contrastar, pero no hay aleatorización de los sujetos ni de los grupos experimentales, ni de los grupos de control; c) La validez interna de la experimentación está comprometida, ya que no se puede garantizar que los grupos comparados (experimental y de control) sean equivalentes [26] y [27]. En concreto, en los estudios cuasi-experimentales las poblaciones no son aleatorias.

En MARCODA los grupos son cuasi-experimentales, debido a que se requiere seleccionar estudiantes con igualdad de condiciones en aspectos como: nivel cognitivo, conocimientos previos, ambiente educativo, edad, entre otros (es responsabilidad del equipo de desarrollo el análisis y definición de esas variables. Para ser más específicos, el marco de trabajo considera como parte del ciclo de vida, crear grupos externos al proyecto (experimentales y de control) para evaluar el contenido digital.

La selección de los grupos de control y experimentales pueden ser:

- a) Ambos en ambientes virtuales.
- b) Unos presenciales y otros virtuales.
- c) Ambos en ambientes presenciales.

Es de anotar que el grupo experimental es el que utiliza el contenido digital y el grupo de control es el que se somete a los mismos aprendizajes, pero sin utilizar el CDA.

Es importante el registro estadístico de los resultados para validar el CDA y tomar la decisión sobre su uso en actividades educativas. Incluso, los metadatos asociados a los CDA's deberían informar sobre los resultados de la evaluación. En esta fase experimental, el equipo de trabajo debe realizar las siguientes actividades:

Selección de grupos experimentales y de control  
Diseño de experimentos integrando verificación, validación y evaluación de aprendizajes.

Los experimentos han de posibilitar el análisis de variables cuyas comparaciones en los grupos experimentales y de control, determinen la calidad del CDA en aspectos técnicos (usabilidad, funcionalidad, fiabilidad, eficiencia, mantenibilidad y portabilidad) y pedagógicos (intencionalidad, motivación, aprendizaje significativo).

Muy importante es el análisis de resultados y documentación de resultados de la experimentación. Esta última actividad es un insumo valioso para el proyecto de contenidos digitales, ya que genera información acerca las distancias entre los requisitos pedagógicos y tecnológicos planteados en la fase 1 y los resultados observados en la interacción de los estudiantes en la fase experimental. Si las diferencias son muy amplias y los resultados acerca de los aprendizajes en los grupos de control superan a los resultados del grupo experimental, entonces el equipo desarrollador del CDA decidirá entre una de las siguientes acciones: a) Repetir la fase de experimentación con actores distintos; b) Revisar cada una de las actividades constitutivas de las fases 1, 2 y 3 e intervenir las que provocaron puntos de ruptura; c) Abortar el proyecto definitivamente. En cualquiera de estas determinaciones resulta oportuno el proceder sobre información documentada y analizada y en consideración de los guiones de evaluación que se construyeron en la fase 1. Aunque los propios guiones pueden ser también motivo de revisión.

Unas buenas prácticas ágiles en ambas evaluaciones (de características del producto y de efectividad) es el diseño de escenarios basados en los guiones de evaluación de las tres fases anteriores.

Además, los actores en la evaluación deben tener total independencia del equipo que analizó, desarrollo e implemento el contenido digital. Es decir, los estudiantes y docentes que participen en la evaluación final, han de ser ajenos al equipo de desarrollo y a los procesos intermedios de evaluación aplicada en las fases 1, 2 y 3. Aparentemente esta práctica puede causar lentitudes en el proceso de evaluación, pero es de anotar que el agilidad se posibilita es porque más rápidamente se llega a la conclusión objetiva sobre la efectividad del contenido. Es decir, si es el mismo equipo el que evalúa al carecerse probablemente de objetividad, puede ser que tengan que incrementarse episodio de evaluación y de retroalimentación para llegar de todos modos a la invitación de personas ajenas al proyecto.

## 2. Ejemplo de aplicación de MARCODA

MARCODA se ha aplicado exitosamente en los siguientes proyectos, con el aval de la Universidad de San Buenaventura Medellín

1) Desarrollo de la Competencia Investigadora en Estudiantes Universitarios Mediante el Empleo Didáctico de Materiales Multimedia.

Este proyecto cuenta con el aval de la Universidad de San Buenaventura Medellín y Universidad Veracruzana de México con el apoyo de la Red de Investigación Multidisciplinar para la Cultura Investigadora (RIMCI). Uno de los objetivos es la creación de un contenido digital hospedado en la Web, para facilitar el aprendizaje sobre planteamiento y evaluación de hipótesis de investigación.

2) Sistema electrónico para la interpretación de la lengua de señas colombiana: Segunda fase.

Uno de los objetivos de este proyecto es la creación de contenidos digitales para facilitar el aprendizaje de lengua de señas a niños Colombianos con deficiencia auditiva.

En cuanto al proyecto para crear un contenido didáctico de aprendizaje sobre el planteamiento y evaluación de hipótesis de investigación, se presentan a continuación algunas aspectos que sirven de ejemplo sobre el proceso de desarrollo que implementa MARCODA en cuanto a: temática y competencias, definición de roles actores y funciones, cronograma de actividades, historias de usuario, mapa global de navegación del CDA y prototipo para la iteración 1:

Temática: formulación y tipos de hipótesis  
 Nivel educativo: educación superior en ingeniería

a) Interpretativas

Comprende la diferencia entre hipótesis, objetivos y preguntas de una investigación  
 Entiende el concepto de hipótesis en el marco de un proyecto de investigación

b) Argumentativa

Explica las diferencias entre los distintos tipos de hipótesis

c) Propositiva

Escribe hipótesis en coherencia con los objetivos y las preguntas de investigación.  
 Construye hipótesis de acuerdo con la estructura causa – efecto entre variables.

Definición de roles/actores /y funciones

Asesor temático / Eucario Parra Castrillón/ Validar la pertinencia del tema. Delimitar el alcance del tema. Definir del plan de contenidos.

Asesor pedagógico y didáctico/ Carlos Arturo Castro Castro; Eucario Parra Castrillón; Agustín Lagunes/ Definir y evaluar los problemas de aprendizaje. Definir y evaluar los canales de aprendizaje. Definir y evaluar los estilos de aprendizaje. Definir y evaluar el diseño instruccional.

Asesor en ingeniería del software/ Carlos Arturo Castro Castro, Wilder Perdomo Charry/ Elaborar el plan de desarrollo de software.

Verificar la ejecución del plan de software. Definir los instrumentos para especificación de requisitos. Definir el diseño arquitectónico.

Seleccionar las herramientas de desarrollo. Seleccionar las plataformas de despliegue del contenido. Construir y aplicar pruebas de validación y verificación de las fases de análisis, diseño, implementación e integración y de calidad del producto en uso.

Analistas programadores/ Alex Ortiz/ Crear programas. Integrar herramientas, programas y objetos digitales. Documentar los procesos del ciclo de vida del software. Actualizar el plan de desarrollo de software.

Usuarios/ Profesores y Estudiantes/Seleccionar las herramientas de desarrollo. Participar en las experimentaciones educativas. Participar en los procesos de validación y evaluación finales e intermedias.

En la figura 4, se presenta una vista del cronograma de actividades para el proyecto, incluyendo las iteraciones cada tres semanas (5 iteraciones):

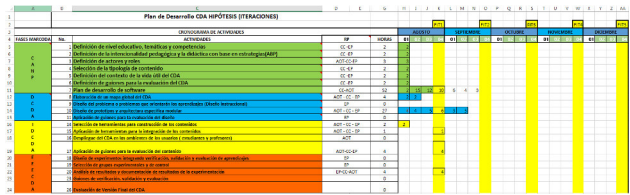


Figura 4. Cronograma proyecto Hipot

En la figura 5, se presenta una vista de dos de las historias de usuario, con las cuales se elaboró el mapa de navegación figura 6

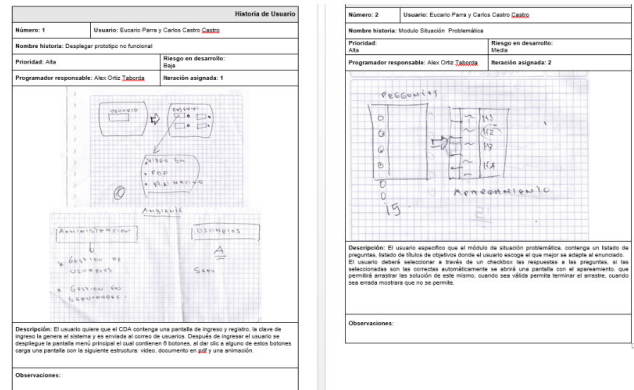


Figura 5 Historias de usuario proyecto Hipot

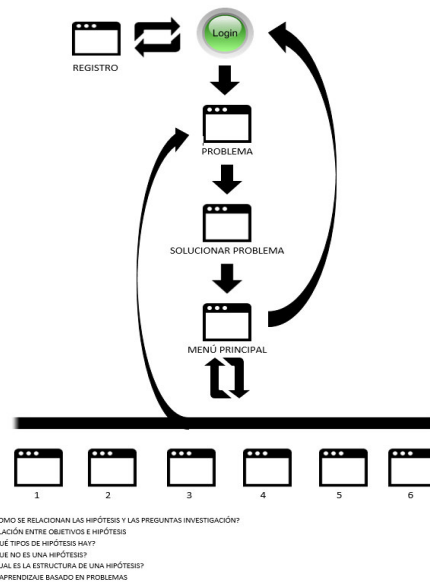


Figura 6 Mapa de Navegación proyecto Hipot.

En las figuras 7, 8, 9 y 10, se presenta una vista del prototipo del CDA para la primera Iteración:



Figura 7 pantalla de ingreso.



Figura 8 situación problemática.

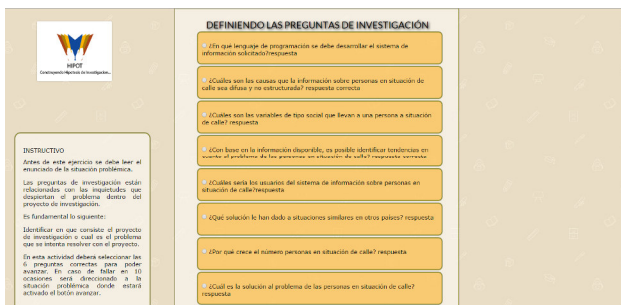


Figura 9 situación preguntas de investigación



Figura 10 explorando aprendizajes

## 4. Conclusiones

El desarrollo de contenidos digitales para el aprendizaje implica por sus características, de la adopción de metodologías y técnicas de ingeniería del software, ya que como se sabe, su objeto principal está relacionado con la calidad tanto de los procesos como de los productos de software. Con ello se asegura la planeación y el control sobre los flujos de información que van

emergiendo en el desarrollo del proyecto. Es decir, la ingeniería del software brinda la posibilidad de un desarrollo fielmente estructurado.

Es de anotar que en la construcción de contenidos digitales para el aprendizaje, se requiere de una ingeniería del software conectada con dos instancias fundamentales del ciclo de vida: la concepción y análisis de necesidades pedagógicas, didácticas y tecnológicas; y la experimentación en ambientes de usuario final. Esto es especialmente trascendente porque está en relación con la efectividad del contenido en escenarios de aprendizaje.

El otro asunto importante es el del agilismo. Debe comprenderse que los desarrollos de contenidos digitales deben estar a tono con las necesidades de su creación rápida, pero sin menoscabar aspectos relacionados con su efectividad. Para esto es fundamental la participación activa del usuario final desde la fase de concepción del contenido, la planeación de iteraciones, la coherencia con los guiones y la formalidad de las validaciones y experimentaciones finales.

MARCODA, es un proceso ágil para el desarrollo de contenidos de aprendizaje, los cuales son considerados como productos de software. MARCODA tiene un ciclo de vida bien definido para asegurar la planeación y el control en los proyectos de creación de contenidos de aprendizaje, con principios incluyen: accesibilidad a través de protocolos web, con evaluación continua de los aspectos didácticos, pedagógicos y tecnológicos, incluye la presencia de los usuarios (profesores, estudiantes y expertos temáticos en todas las fases del proyecto de construcción, con diseño y experimentación educativa y con el aprendizaje Baso en Problemas (ABP) como metódica propuesta. Adicionalmente MARCODA se basa en prácticas ágiles aceptadas y comprobadas en la industria del software como: Elaborar Historias de usuario, Definir plan de entregables por iteraciones de corta duración (2 a tres semanas), Priorización de requerimientos para cada iteración, Estimar tiempo y recursos requeridos para cada iteración, Reuniones informativas en períodos cortos y corto tiempo (eje: 15 minutos diarios), Verificación, validación y evaluación por iteración y codificación con estándares entre otras.

MARCODA está siendo aplicada con éxito, en varios proyectos de creación de contenidos digitales de aprendizaje, cuyos resultados serán expuestos en futuros trabajos.

## 4. REFERENCIAS

- [1] Kent Beck and et al. (2007) Principles behind the agile manifesto. Online: <http://agilemanifesto.org/principles.html> [Apr. 2015].
- [2] VersionOne, Inc. (2015) State of Agile. Online: <https://articulosit.files.wordpress.com/2015/04/state-of-agile-development-survey-ninth.pdf> [Sept. 2015].
- [3] Lloret, N. & Canet, F. (2008). Nuevos escenarios, nuevas formas de expresión narrativa: La Web 2.0 y el lenguaje audiovisual. Anuario Hipertext.net, número 6. Universitat Pompeu Fabra, Barcelona. [En línea]. <http://www.upf.edu/hipertextnet/numero-6/lenguaje-audiovisual.html> [Último acceso: 08 09 2015].
- [4] Parra Castrillón, Eucario. (2011) Propuesta de metodología de desarrollo de software para objetos virtuales de

- aprendizaje -MESOVA- Revista Virtual Universidad Católica del Norte. No. 34, online: <http://revistavirtual.ucn.edu.co/> [Último acceso: 08 09 2015].
- [5] Castro, C. Serna, E. and Botero, R. (2012). "A proposed design of Learning Objects". Proceedings of the 6th Euro American Conference on Telematics and Information Systems, EATIS 2012, Valencia, Spain, pp. 347-353.
- [6] Gagné, R. M. (1965). The conditions of learning. Holt, Rinehart and Winston.
- [7] Bianchini, A. (1992). Metodología para el desarrollo de aplicaciones educativas en ambientes multimedia. Trabajo de ascenso a la categoría de Profesor Asociado en la Escuela de Ingeniería de Sistemas, Cap. IV. Universidad Metropolitana de Caracas.
- [8] Dorrego, E. (1994). Modelo para la producción y evaluación formativa de medios instruccionales, aplicado al video y al software. Revista de Tecnología Educativa, 12, 3, 313-327.
- [9] Daziel, J. (2002). Reflections on the COLIS (Collaborative Online Learning and Information Systems) Demonstrator Project and the "Learning Object Lifecycle". In Proceedings of the ASCILITE 2002 Conference. Auckland, New Zealand, Dec. 8-11.
- [10] Polo, M. (2003). Aproximación a un Modelo de Diseño: ADITE. Docencia Universitaria, 4, 1, 67-83.
- [11] Boyle, T. (2003). Design principles for authoring dynamic, reusable learning objects. Australian Journal of Educational Technology, 19, 1, 46-58.
- [12] Strijker, A. (2004). Reuse of Learning Objects in Context: Human and Technical Aspects. PrintPartners Ipskamp.
- [13] Azpeitia, I; Monge, S & Ovelar, R. 2005. Una Aproximación al Diseño de una Guía de Buenas Prácticas en torno al Paradigma de los Learning Objects. En II Simposio Pluridisciplinar sobre Diseño, Evaluación y Descripción de Contenidos Educativos Reutilizables. Barcelona, España, Oct. 19-21.
- [14] Galvis, Gómez y Mariño (1998). Ingeniería de Software Educativo con Modelaje Orientado por Objetos: Un Medio para Desarrollar Micromundos Interactivos. Informática Educativa Vol. 11, No 1, 1998, p-30.
- [15] Gruene, M., Lenz, K. & Oberweis, A. (2005). Pricing of Learning Objects in a Workflow-Based E-Learning Scenario. In Proceedings of the 38th Annual Hawaii International Conference on System Sciences, HICSS '05. Big Island, Hawaii, Jan. 3-6.
- [16] Peláez, S. G. & López, B. 2006. Metodología para el Desarrollo de Software Educativo (DESED). UPIICSA, 14, 6, 41-42.
- [17] Orjuela A. y Rojas M. (2008). La metodología de desarrollo ágil como una oportunidad para la ingeniería del software educativo. Revista Avances en Sistemas e Informática. Vol. 5 No 2. Junio de 2008. Universidad Nacional de Colombia
- [18] Vilca, E. et Al. (2012). Propuesta de un enfoque para el desarrollo de software educativo intercultural. 3er Simposio Internacional de Innovación y Tecnología ISIT2012. [En línea]. <http://es.slideshare.net/ecler/ecler-software-educativo-intercultural> [Último acceso: 08 09 2015].
- [19] Benigni, G. et Al. (2014). Incorporando modelos mentales y conceptuales en el análisis de requerimientos para desarrollar software educativo. Espacios. Vol. 35 (Nº 3) Año 2014. Pág. 17. [En línea]. <http://www.revistaespacios.com/a14v35n03/14350318.html> [Último acceso: 08 09 2015].
- [20] Serna, M.E., Castro, C.C. and Botero, T.R. (2012). "SEDLO: Software Engineering for Developing Learning Objects." Proceedings of the 6th Euro American Conference on Telematics and Information Systems, Editor: Chagas, do N.R.P., Universidad Politécnica de Valencia, Spain, pp. 347-353.
- [21] Moya, M. (2013). From ICTs to the Learning Technologies Knowledge (LTK): the importance of digital educational content. Didáctica, Innovación y Multimedia (DIM). Online: <http://www.pangea.org/dim/revista.htm> [Sept. 2015].
- [22] Jardines, F.J. (2011). Revisión de los principales modelos de diseño instruccional. Innovaciones de Negocios. (págs. 357-389). México.
- [23] Parra, J., Castro, C. y Amariles, M. (2014). Casos de éxito de la aplicación de la metodología de aprendizaje basado en problemas ABP. IngEam (1), 12-23.
- [24] Gonzalez M. (2011). Estilos de aprendizaje: su influencia para aprender a aprender. Revista Estilos de Aprendizaje, nº7, Vol 7, abril de 2011.
- [25] Witham K., Mora O. y Sanchez T. (2008). Revista Electrónica de Trabajo Social. Universidad de Concepción. No 7.
- [26] Rodríguez, N. (2011). Diseños Experimentales en Educación. Revista de Pedagogía, vol. XXXII, núm. 91, julio-diciembre, pp. 147-158.
- [27] Ascheri E. y otros ( 2014). Software educativo en línea para la enseñanza y el aprendizaje de temas de cálculo numérico. Revista digital Matemática, Educación e Internet (<http://www.tecdigital.itcr.ac.cr/revistamatematica/>). Vol 14, No 2. Marzo - agosto 2014.