

Implementation of an open source code framework using a new development methodology

Luis F. González Author, PhD (c) en Software Sistemas y Computación¹, Hernando Recamán, MSc en Ingeniería de Sistemas¹, and Gladis H. Vásquez, MBA¹
¹ Politécnico Jaime Isaza Cadavid, Colombia, lfgonzaleza@elpoli.edu.co, hrecaman@elpoli.edu.co, hgvasquez@elpoli.edu.co

Abstract– *Based on the theoretical concept of frameworks applied to the academic field, the MOON framework has been created, through the implementation of the Methodology for the Collaborative Development of Web Applications (MDCAW). This process allowed to represent a practical example in the teaching field, in which 100 students of a university institution participated, which acquired diverse benefits that are related in the work*
Keywords– *open source code, Collaborative development, Framework, Methodology*

Digital Object Identifier (DOI):<http://dx.doi.org/10.18687/LACCEI2018.1.1.84>
ISBN: 978-0-9993443-1-6
ISSN: 2414-6390

Implementación de un Framework de Código Abierto Utilizando una Nueva Metodología de Desarrollo

Luis F. González Author, PhD (c) en Software Sistemas y Computación¹, Hernando Recamán, MSc en Ingeniería de Sistemas¹, and Gladis H. Vásquez, MBA¹

¹ Politécnico Jaime Isaza Cadavid, Colombia, lfgonzaleza@elpoli.edu.co, hrecaman@elpoli.edu.co, hgvasquez@elpoli.edu.co

Resumen – A partir del concepto teórico de marcos de trabajo aplicado al campo académico, se ha creado el framework MOON, mediante la implementación de la Metodología para el Desarrollo Colaborativo de Aplicaciones Web (MDCAW). Dicho proceso permitió representar un ejemplo práctico en el campo docente, en el que participaron 100 estudiantes de una institución universitaria, los cuales adquirieron diversos beneficios que son relacionados en el trabajo. El proceso propuesto por la metodología a usar, requiere de la implementación de un Ambiente Virtual de Desarrollo (AVD) y de un método específico que consiste en el desarrollo de las actividades que se plantean a partir de un conjunto de iteraciones durante todo el ciclo de vida de un proyecto de software

Palabras claves -- Código abierto, desarrollo colaborativo, framework, metodología.

Abstract – From the theoretical concept of frameworks applied to the academic field, was created the framework MOON, by implementing of the methodology for the collaborative development of Web applications (MDCAW). This process allowed to represent a practical example in the teaching field, impacting 100 students from a university, which acquired many benefits that are related at work. The proposed process by the new methodology to use, requires the implementation of a Virtual Environment Development (AVD) and a specific method that is the development of activities that arise from a series of iterations during the entire life cycle of one software project.

Keywords-- Open source, collaborative development, framework, methodology

I. INTRODUCTION

En la actualidad, a la hora de desarrollar software se requiere de personal altamente calificado para participar en proyectos donde priman dos variables de calidad: bajo costo y tiempo empleado en todo el ciclo de vida del desarrollo. Lo anterior implica que se requieren profesionales con los conocimientos y habilidades para desarrollar software que cumpla con estas características. Las empresas de desarrollo han empezado a implementar técnicas y modelos que ofrece la ingeniería del software, estos definen y/o seleccionan la arquitectura de software acorde con el proyecto, utilizan un diseño basado en componentes, generan un desarrollo soportado en herramientas que aceleran su proceso y manejan pruebas transversales en toda la construcción del sistema. La debilidad del anterior modelo radica en la poca capacitación y entrenamiento, que limitan al desarrollador visualizar su complejidad.

Digital Object Identifier (DOI): <http://dx.doi.org/10.18687/LACCEI2018.1.1.84>
ISBN: 978-0-9993443-1-6
ISSN: 2414-6390

En el presente artículo se amplía la visión acerca de las herramientas disponibles para la construcción de aplicativos de software en ambientes virtuales, utilizando una metodología para el desarrollo colaborativo mediante la implementación de un marco de trabajo de código abierto.

Inicialmente se muestra una breve descripción de conceptos, para luego presentar los criterios definidos en la comparación de estos, los cuales son seleccionados con base en el análisis de un estado del arte sobre *frameworks* y con el aporte de expertos en el tema. Posteriormente, se ilustra el diseño de un instrumento que fue creado para la valoración de los marcos de trabajo seleccionados y se presenta un ejemplo del *framework* construido, el cual está formado por un conjunto de componentes físicos y lógicos, estructurados de tal forma que permiten ser reutilizados en el diseño y desarrollo de nuevos sistemas de información. Finalmente, se describe en forma breve la metodología a emplear en la implementación del *framework*.

II. FUNDAMENTACIÓN

Para una adecuada contextualización de la temática a abordar, se hace necesario revisar algunos conceptos que son de gran relevancia en el desarrollo de la investigación.

A. Framework o Marco de Trabajo

Conjunto de componentes físicos y lógicos estructurados de tal forma que permiten ser reutilizados en el diseño y desarrollo de nuevos sistemas de información [1].

B. Código Abierto (open source)

Es el término con el que se conoce al software distribuido y desarrollado libremente. Fue utilizado por primera vez en 1998 por algunos usuarios de la comunidad del software libre, tratando de usarlo como reemplazo al ambiguo nombre original en inglés del software libre (free software). No obstante, el término continúa siendo ambivalente, puesto que se usa en la actualidad por parte de programadores que no ofrecen software libre, pero en cambio, sí ofrecen las fuentes o código de los programas para su revisión o modificación previamente autorizada por parte de sus pares académicos [2].

C. Metodología para el Desarrollo Colaborativo de Aplicaciones Web (MDCAW)

Esta metodología, fue presentada como resultado de un trabajo de investigación realizado en la maestría en software libre de la universidad Autónoma de Bucaramanga, en convenio con la universidad Oberta de Catalunya (España) [3].

La metodología MDCAW, fue diseñada para ser tenida en cuenta en proyectos de software desarrollados por un grupo de usuarios ubicados en tiempo y espacios diferentes, quienes hacen uso de diversas herramientas de trabajo colaborativo. Ésta pretende a través de técnicas fácilmente aplicables, gestionar todo un proyecto, desde la selección del hardware y software a utilizar, hasta la aplicación de un método de desarrollo ya definido [4], haciendo uso de un Ambiente Virtual de Desarrollo (AVD), el cual obedece a un espacio de comunicación que integra un extenso grupo de recursos creados para facilitar y optimizar las tareas que se desean desarrollar [5].

III. SELECCIÓN DE MARCOS DE TRABAJO

Teniendo en cuenta que existen múltiples frameworks con diferentes características, se realiza un análisis detallado de los más conocidos, lo cual permite hacer el proceso de selección mediante las siguientes etapas.

Evitar la primera, es la caracterización de los marcos de trabajo más conocidos como son: Cake PHP [6], Akelos [7], Pear [8], Symfony [9], Zend [10], Ruby On Rails [11], Grails [12], Zoop [13], Codeigniter [14], Kumbia [15], Strust [16], Spring [17].

La segunda etapa, es la comparación de los doce marcos anteriores que se tomaron como base y a los cuales se les aplicaron los siguientes criterios:

- 1) *Lenguaje.*
- 2) *Documentación.*
- 3) *Curva de aprendizaje.*
- 4) *Compatibilidad con base de datos.*
- 5) *Validación de datos en el cliente.*
- 6) *SopORTE para AJAX.*
- 7) *Módulos para la presentación de datos y utilidades gráficas.*

La tercera etapa, obedece a la recopilación de información de los criterios de cada marco de trabajo, para lo cual fue necesario construir un instrumento que permita seleccionarlos, (ver fig.1).

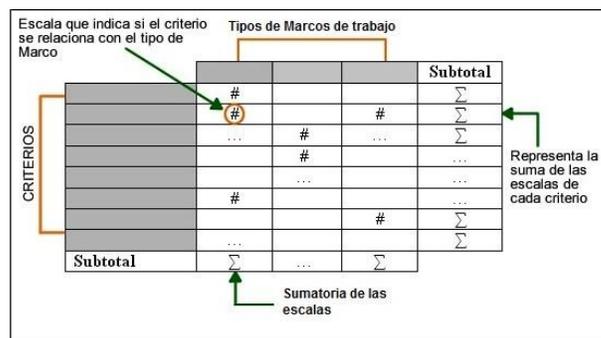


Fig.1 INSTRUMENTO PARA LA SELECCIÓN DE MARCOS DE TRABAJO.

Basado en el instrumento de selección, se procede a analizar cada criterio y a cualificar los marcos de trabajo. Esta cualificación se realiza haciendo uso de números para identificar cierta característica, como se muestra a continuación:

- 1) *El uno (1) representa el mínimo valor, indica ausencia del criterio.*
- 2) *El dos (2) representa un punto de control asignado, si y solo si el criterio evaluado no cumple con las calificaciones 1 y 3.*
- 3) *El tres (3) representa cumplimiento de todo lo especificado en el criterio.*

La aplicación del instrumento permite concluir que los marcos de trabajo que ofrecen demasiadas dificultades en el aprendizaje y que obtuvieron una baja evaluación, (por lo tanto no son recomendados) son: Grails, Codeigniter y Struts. De otro lado se observa que el de menor puntaje acumulativo en todos los criterios es Grails y que los más recomendados por cumplir con la totalidad de criterios en una forma satisfactoria son: Kumbia, Cake y Symfony. Finalmente, el marco de trabajo con el mayor puntaje acumulativo en todos los criterios es Kumbia. El resumen de los resultados arrojados en el estudio comparativo, según el instrumento aplicado, se presentan a continuación, (ver tabla I).

TABLA I.
RESUMEN DE LOS RESULTADOS OBTENIDOS, SEGÚN EL INSTRUMENTO APLICADO PARA SELECCIÓN DE LOS MARCOS DE TRABAJO.

Criterios	Cake	Akelos	Pear	Symfony	Zend	Ruby On Rails	Grails	Zoop	Codeigniter	Kumbia	Struts	Spring	Tot
Lenguaje	3	3		3	3	1	1	3	3	3	2	2	30
Documentación	3	2	2	3	2	2	1	2	2	3	1	3	26
Curva de aprendizaje	2	2	1	3	3	2	1	2	1	3	1	2	23
Base de Datos	2	2	3	2	2	2	2	2	2	2	2	2	25
Validación Integrada	3	2	3	2	2	2	2	2	1	3	2	2	26
Integración con AJAX	3	3	2	3	2	3	3	3	2	3	2	2	31
Módulos, Rss, feed, PDF	3	2	2	3	3	2	1	2	1	3	3	2	27
Utilidades gráficas	3	3	3	3	2	3	3	3	3	3	2	3	34
SUBTOTALES	22	19	19	22	19	17	14	19	15	23	15	18	222
Promedio de columnas 18.5													

Promedio filas = 27.75

Basados en el anterior análisis y partiendo de las bondades encontradas en el marco que obtuvo el mayor puntaje y las características encontradas en Symphony, se creó un nuevo framework de uso académico llamado MOON, que administra las tablas maestras sobre los programas, las asignaturas, los docentes y los estudiantes. Este podrá implementarse en un sistema de información integrado, que gestione los procesos de docencia en una institución universitaria, permitiendo optimizar los tiempos de desarrollo y facilitando la reusabilidad del software, la cual se refiere a "el proceso de creación de sistemas de software desde el software existente, en lugar de construir sistemas de software a partir de cero" [18]. En el numeral cinco se describen las características de MOON.

IV. METODOLOGÍA UTILIZADA EN LA IMPLEMENTACIÓN DEL MARCO DE TRABAJO

Partiendo de las características específicas requeridas en la implementación del framework MOON y teniendo en cuenta que su desarrollo debe ser realizado por múltiples actores pertenecientes a una institución universitaria, ubicados en tiempo y espacios diferentes, se empleó la metodología MDCAW, la cual plantea el uso de un AVD para la correcta gestión y seguimiento del proyecto.

El uso de MDCAW permite establecer claramente los diferentes procesos de administración y desarrollo del proyecto mediante la definición de cuáles son las fases y actividades que se involucran en cada una éstas, los roles y responsabilidades de los diferentes actores que interactúan en cada fase y las medidas que deben emplearse para contabilizar el desarrollo del proyecto.

MDCAW propone tres fases fundamentales: Composición, donde se realiza toda la planificación y se define el plan de gestión del proyecto, el personal a interactuar en el proceso y los componentes del AVD. Administración, en esta se coordinan todas las actividades relacionadas con la iteración que se va generando en la fase de desarrollo y en el uso de las herramientas de trabajo colaborativo. Desarrollo, en esta fase se realiza la asignación de roles y tareas y se cumple con el ciclo de vida del proceso de desarrollo de software mediante un plan de iteración compuesto por diferentes ciclos secuenciales, haciendo uso del modelo espiral, propuesto por la "Fábrica de Software Libre" [19]. Finalizada cada iteración se obtiene una versión de prueba del proyecto, permitiendo evaluar y controlar su calidad a través de las diversas herramientas configuradas en el AVD, (ver fig.2).

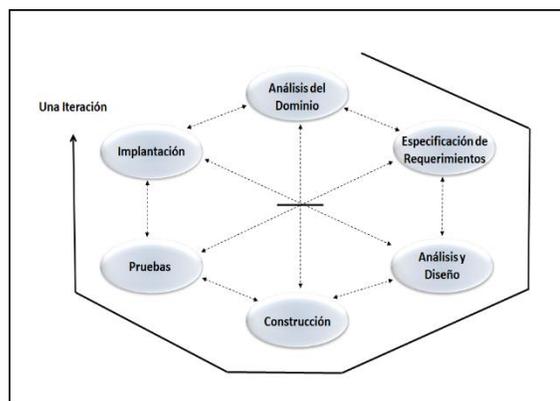


Fig.2 Proceso de desarrollo de aplicaciones de software. Modelo espiral [19].

V. DESCRIPCIÓN DEL FRAMEWORK IMPLEMENTADO

El *framework* académico MOON se creó con base en los resultados de la comparación de marcos de trabajo, registrados en el libro "*frameworks* para el desarrollo de aplicaciones Web de código abierto [20] y contiene un gran volumen de preguntas que se encuentran clasificadas para posteriormente ser evaluadas.

La interfaz Create, Retrieve, Update y Delete (CRUD) [21] del marco de trabajo MOON, genera una ventana que contiene los campos a registrar, (ver fig.3).



Fig.3 Interfaz CRUD del *framework* MOON.

- MOON se organiza a través de las siguientes carpetas:
- 1) *Config*: Guarda la información sobre el servidor, la base de datos, URL, perfiles y sesiones.
 - 2) *Images*: Directorio con las imágenes del aplicativo.
 - 3) *Modules*: Almacena el código para la creación de las listas de preguntas, evaluaciones y respuestas, aplicando el patrón de diseño facade [22] caracterizado por tener una interfaz unificada y sencilla, que es intermediaria entre el cliente y un grupo de interfaces. La arquitectura empleada en la codificación es el Modelo Vista Controlador (MVC), que "divide las aplicaciones Web en tres capas: MODEL la cual gestiona los datos de la aplicación, VIEW muestra los datos de los formularios a través del navegador, y CONTROLLER que proporciona enlace entre los anteriores componentes"[23].
 - 4) *Moon*: Contiene todas las clases agrupadas en carpetas, las más representativas son para bases de datos, formularios, parámetros, capas y seguridad.
 - 5) *Scripts*: Contiene los documentos teóricos para crear: la base de datos, tablas, registros, usuarios y permisos.

6) *Styles*: Contiene las diferentes hojas de estilos (.css) programadas para el *framework*.

La verificación de funcionalidad del marco de trabajo MOON, se realizó implementándolo en un sistema de evaluación de docentes, acorde con los lineamientos establecidos por la facultad de Ingenierías de la institución universitaria Politécnico Colombiano Jaime Isaza Cadavid de la ciudad de Medellín. Para su creación se seleccionaron las características de *Symfony* y se implementó bajo la metodología MDCAW.

MOON se ha utilizado en las siguientes instituciones: Unidades Tecnológicas de Santander de la ciudad de Bucaramanga y en el Politécnico Colombiano Jaime Isaza Cadavid de la ciudad de Medellín. Actualmente ha sido solicitado por la universidad Pontificia Bolivariana, seccional Bucaramanga.

VI. APLICACIÓN DE MDCAW EN MOON

MOON permite diseñar y desarrollar aplicaciones Web formadas por componentes reutilizables que requieren corto tiempo para su creación. Además cuenta con un ejemplo sobre la forma de diseñar, implementar y administrar sistemas de información de acuerdo con los requerimientos de las organizaciones. Las limitaciones de MOON se enfocan básicamente a las restricciones impuestas por las tecnologías disponibles.

Para el proceso de creación de MOON se establecieron las siguientes actividades apoyadas en la fase de desarrollo de la metodología MDCAW:

- 1) *Delimitación del problema.*
- 2) *Definición del alcance.*
- 3) *Levantamiento de requerimientos.*
- 4) *Análisis de requerimientos.*
- 5) *Diseño conceptual.*
- 6) *Diagrama entidad relación.*
- 7) *Versiones en codificación.*
- 8) *Pruebas por versiones.*
- 9) *Implantación por versiones.*

Lo anterior implica la implantación de una plataforma de trabajo que aproveche todas las funcionalidades del desarrollo colaborativo, el cual obedece a un modelo basado en la disponibilidad pública del código y la comunicación vía internet de diversos programadores y usuarios, ubicados en tiempo y espacios diferentes, con el fin de compartir cualquier problema en el desarrollo, de tal forma que este pueda ser caracterizado rápidamente y su solución sea obvia para alguno de los participantes[24]. Dicha plataforma se ha denominado un AVD, el cual permite la administración Web de cada proyecto a partir de diversas herramientas como: sistema de control de versiones, foros de discusión, herramientas de seguimiento de errores, soporte a peticiones y consultas, listas de correo para los miembros compartir documentación, gestión de tareas pendientes, envío de archivos y revisiones. Las anteriores herramientas se integran

bajo un servidor Web que realiza la coordinación de proyectos, agrupándolos en diversas categorías.

A. Selección de software

Teniendo en cuenta que el AVD se configura mediante el uso de programas y herramientas bajo software libre que permiten gestionar los recursos, se seleccionó el software a implantar, compuesto por: sistema operativo, herramientas de comunicación, de conferencia, de desarrollo y de gestión de proyectos (sistemas de control de versiones y de seguimiento de fallos), como se describen a continuación:

1) *Elección del sistema operativo.* Se seleccionó GNU/Linux en su distribución Debian [25] por considerar que ésta es una sólida plataforma para el manejo de servidores, por las excelentes herramientas de gestión y el amplio soporte de arquitecturas de procesadores.

2) *Elección de una herramienta de control de revision.* Dadas las características del proyecto y de acuerdo con las herramientas estudiadas, se ha seleccionado *Subversion* [26] como sistema de control de versión, ya que éste se ajusta a los requerimientos, pues se va a utilizar un repositorio centralizado de todo el código, del cual es responsable un único usuario.

3) *Elección de una herramienta de sistema de control de fallos.* Se ha seleccionado a Mantis Bug Tracker [27] por ser una herramienta sencilla y ligera. El acceso a ésta se realiza mediante un navegador para la comunicación y seguimiento de incidentes relacionados con el proyecto. No tiene ninguna restricción al tipo de navegador que debe usarse para trabajar como cliente y su objetivo es crear y mantener un sistema de control de bug basado en Web, el cual está diseñado de manera que sea fácilmente modificable, personalizable y actualizable.

4) *Selección de otro software necesario.* Para el correcto funcionamiento del AVD se hace necesario instalar básicamente un Sistema de Gestión de Bases de Datos (SGBD) [28], configurar un servidor Web y definir el lenguaje de programación a trabajar. El software seleccionado en cada categoría es: (ver tabla II).

TABLA II.
OTRO SOFTWARE SELECCIONADO PARA EL CORRECTO
FUNCIONAMIENTO DE UN AVD.

Descripción	Software seleccionado
Sistema de gestión de bases de datos	MySQL [29]
Servidor Web	Apache [30]
Lenguaje de programación	PHP [31]

B. Instalación del AVD

Como resultado de la aplicación de las dos primeras fases de la metodología propuesta, se instala el AVD, su descripción, características básicas y diferentes

funcionalidades se agrupan en un portal Web desarrollado bajo PHP el cual contiene:

- 1) *Página Web para cada proyecto.*
- 2) *Sistema de control de versiones.*
- 3) *Servidor Web para coordinación de proyectos*
- 4) *Agrupación de proyectos en diversas categorías.*
- 5) *Foros de discusión para miembros del proyecto.*
- 6) *Seguimiento de errores.*
- 7) *Soporte a peticiones*
- 8) *Soporte a consultas.*
- 9) *Listas de correo para que los miembros compartan documentación.*
- 10) *Gestión de tareas pendientes.*
- 11) *Envío de archivos y revisiones.*
- 12) *Noticias propias para cada proyecto.*

Luego de la puesta en marcha del AVD, se da inicio a la fase de administración a través de las siguientes actividades:

C. *Dar de alta al proyecto de creación del nuevo framework.*
 Para dar de alta al proyecto se realizan los siguientes pasos: diligenciar el formulario de registro, recepción de solicitud, trámite (si se considera que la solicitud cumple con los requerimientos mínimos y es pertinente, ésta es aprobada y el proyecto es accesible), dar de alta al proyecto y finalmente el usuario recibe una notificación de confirmación.

D. *Administración de los proyectos.*

Para poder obtener productos con calidad y alcanzar las metas propuestas, es fundamental hacer una buena administración y gestión de los proyectos, para lo cual se deben cumplirse las siguientes tareas:

- 1) *Gestión del proyecto*
- 2) *Planificación del trabajo*
- 3) *Asignar roles a los miembros.*
- 4) *Activar componentes del proyecto*
- 5) *Administrar los componentes del proyecto.*
- 6) *Crear la página Web del proyecto.*
- 7) *Establecer la clasificación del proyecto dentro del mapa de proyectos general.*
- 8) *Editar la información general.*
- 9) *Ver informe y estadísticas de evolución del proyecto.*
- 10) *Publicar solicitudes de ayuda para realizar un trabajo determinado.*

Para la implementación del *framework* MOON, en la fase de desarrollo se hace uso del modelo en espiral que plantea la metodología, el cual se fundamenta en una serie de iteraciones del ciclo de vida del *software* para irse ajustando a las necesidades del proyecto, hasta lograr la madurez del producto final. Su nivel de riesgo es alto debido a que se requiere la realización de un seguimiento estricto en cada iteración.

En la selección del lenguaje de programación se tiene en cuenta principalmente el contexto y los objetivos del proyecto, el nivel de conocimiento y la experiencia del equipo de desarrollo, el costo por el uso del lenguaje, los marcos de trabajos que se pueden usar, entre otros. MOON utiliza los

patrones de diseño que se orientan al desarrollo *Web*, los cuales requieren del uso de un lenguaje de programación orientado a objetos, por lo anterior el lenguaje seleccionado es PHP.

Con relación a la arquitectura de *software*, se deben tener en cuenta los objetivos y las limitaciones del sistema. Es recomendable optar por arquitecturas conocidas, analizando sus ventajas e inconvenientes, razón por la cual se seleccionó la arquitectura MVC.

El proyecto utiliza patrones de reutilización para los requisitos del sistema de información, permitiendo integrar todas las actividades necesarias, (ver tabla III).

Tabla III.
TIPOS DE REQUISITOS [32].

Tipo de requisitos	Descripción
Requisitos de información	Describen que información debe almacenar el sistema para satisfacer las necesidades de clientes y usuarios. Identifican los conceptos relevantes sobre los que se deben almacenar información y los datos específicos que son de interés.
Requisitos funcionales	Describen los casos de uso donde los diferentes actores utilizan los servicios proporcionados por el sistema. Cada requisito funcional identifica el evento de activación, las pre y post condiciones y los pasos que componen el caso de uso, así como las posibles excepciones.
Requisitos no funcionales	Describe aquellas características no funcionales que los clientes y usuarios desean que tenga el sistema a desarrollar.

El *framework* Moon representa un ejemplo práctico aplicado en la academia. Su desarrollo contiene un patrón de interfaz con componentes que apoyan el proceso de desarrollo de software, utilizando el patrón de diseño facade. El marco de trabajo emplea el lenguaje JavaScript [33], emplea hojas de estilo y su sistema manejador de base de datos es Posgress [34]. En la fig.4 se observa el diagrama entidad relación.

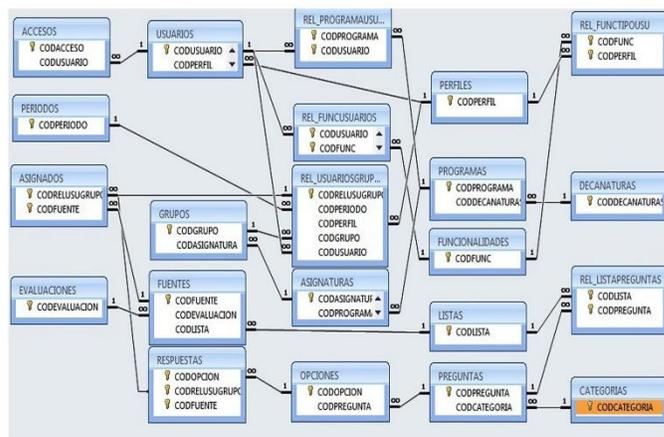


FIG.4. DIAGRAMA ENTIDAD RELACIÓN DEL *FRAMEWORK* MOON.

VII. CONCLUSIONES Y TRABAJOS FUTUROS

MDCAW, se evidencia como una herramienta de gran importancia que garantiza el curso normal de un proceso de desarrollo de software. Ésta por su sencillez y claridad puede convertirse rápidamente en una excelente herramienta metodológica de uso en los diferentes proyectos bajo el modelo colaborativo, tal como se evidenció en la implementación del framework MOON, dando como resultado un marco de trabajo competitivo, que contiene componentes reutilizables, ya que se ha creado tomando como referencia un marco existente que se adecuó de acuerdo con las necesidades del equipo de trabajo. Lo anterior permitió establecer una relación entre la teoría y la práctica, conduciendo a la innovación y generación de nuevo conocimiento, lo cual en términos de creación de software, casi siempre implica menos costos y tiempos de producción.

Se concluye que el desarrollo colaborativo a través de un AVD, se caracteriza como un conjunto de experiencias que permiten compartir comentarios, sugerencias, reflexiones y el aporte que cada integrante del equipo hace al proyecto, transformando el trabajo individual en un producto más rico que agrupa las observaciones realizadas por los integrantes.

Luego de aplicar el framework MOON, durante el desarrollo de la asignatura Taller de Programación Avanzada, en una muestra de 101 personas de una población total de 1010 estudiantes, perteneciente al quinto semestre del programa de ingeniería informática del Politécnico Colombiano Jaime Isaza Cadavid, se observó que estos obtuvieron los siguientes beneficios:

1) *Mejor fundamentación sobre el tema de marcos de trabajo.*

2) *Promoción en el uso correcto de la fase de documentación de un proyecto.*

3) *Optimización del tiempo en la puesta en marcha de un sistema de Información, ya que se proporciona un código fuente depurado que puede ser reutilizado.*

4) *Facilidad en el diseño y desarrollo de aplicaciones Web formadas por componentes reutilizables que requieren corto tiempo para su creación. Siendo el código fuente más reutilizado el perteneciente a la interfaz CRUD.*

5) *Las aplicaciones creadas a través del framework MOON, comparten una arquitectura común, debido a que se trabaja en un modelo basado por componentes.*

Además de los anteriores beneficios, se pudo observar un buen nivel de comprensión de los ejemplos prácticos en la realización de un trabajo de grado, en el cual se utilizó Java Active Record como un framework de persistencia objetual, basado en el patrón de diseño para la capa de acceso a datos [35].

Para futuros trabajos, se puede proponer un patrón de diseño que esté orientado a plataformas *OpenSource* y que permita independizar las tres capas del MVC, organizando la labor del desarrollador y permitiéndole agilizar su trabajo. Además se debe fortalecer el *framework* con ejemplos

codificados que contengan otros patrones de diseño que se encuentran en la literatura.

AGRADECIMIENTOS

Al Politécnico Colombiano Jaime Isaza Cadavid, por permitir aplicar el framework en una de sus asignaturas del programa de Ingeniería Informática y por facilitar los recursos de hardware para la implantación del AVD.

REFERENCES

- [1] Minetto, E., Frameworks para Desarrollo en PHP, Novatec, 2007.
- [2] Open Source Initiative, "The Open Source Definition", 1998. (Web). En: <http://www.opensource.org/docs/osd>. Último acceso: 25 de enero de 2012.
- [3] González, L. y Carrillo, L., Metodología para el desarrollo colaborativo de aplicaciones WEB, Editorial Académica Española, 2012.
- [4] González, L., Carrillo, L. y Rangel, J., "(MDCAW) Metodología para el Desarrollo Colaborativo de Aplicaciones. Web". En: Ediciones de la universidad Nacional de Colombia. (ed.), Memorias del Encuentro Nacional de Investigación en Posgrados ENIP 2009. Bogotá (Colombia), 2-4 de diciembre de 2010, pp. 20-30, 2010.
- [5] González, L. y Vásquez, G., "Implementación de la Metodología para el Desarrollo Colaborativo de Aplicaciones Web (MDCAW), basada en Arquitecturas Orientadas a Servicios (AOS)". En: Baralt, J. y Callaos, N. (eds), Memorias de la Novena Conferencia Iberoamericana en Sistemas, Cibernética e Informática. Orlando (Estados Unidos), 29 de junio al 2 de julio de 2010, pp. 183-188, 2010.
- [6] Syam, A. y Bari, H., CakePHP application development, Packt Publishing, 2008.
- [7] Hazem, M. y Nikos, M., "User Interface for Internet applications". En: Metin D. y Zoran B. (eds.), Proceedings of the 9th WSEAS international conference. Moscú (Rusia), 20-22 de agosto de 2009, pp. 383-398, 2009.
- [8] Schmidt, S., PHP programming with PEAR, Packt Publishing, 2006.
- [9] Craig, Z., Symphony start to finish, Wrox Press, 2011.
- [10] Pope, K., Zend Framework Web application development, Packt Publishing, 2009.
- [11] Holzner, S., Beginning Ruby on Rails, Wrox press, 2007.
- [12] Dickinson, J., Grails 1.1 Web application development, Packt Publishing, 2009.
- [13] Brampton, M., PHP5 CMS framework development, Pack publishing, 2010.
- [14] Griffith, A., Codeigniter 1.7 professional development, Packt Publishing, 2010.
- [15] Grupo Kumbia PHP, "Manual de Kumbia PHP framework", 2011. (Web). En: <http://www.kumbiaphp.com/blog/manuales-y-descargas/>. Último acceso: 1 de febrero de 2012.
- [16] Newton, D., Apache Struts 2 Web application development, Packt Publishing, 2008.
- [17] Lüppken, S. y Stäuble, M., Spring Web flow 2 Web development, Packt Publishing, 2009.
- [18] Krueger, Ch., "Software Reuse". ACM Computing Surveys, vol. 24, no 2, pp. 131-183, 1992.
- [19] Aguilar, J., Terán, O., Álvarez, J. y Abraham, B., Sirviendo al bienestar colectivo, Fábrica de Software Libre Fundacite, 2006.
- [20] Guerrero A. y Recaman H., Frameworks para el desarrollo de aplicaciones Web de código abierto, Editorial SIC, 2009.
- [21] Merkek, D., Expert PHP 5 tools, Packt Publishing, 2010.
- [22] Gamma, E. y Helmm, R., Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Longman Publishing Co., 1995.
- [23] Stratmann, E., Ousterhout, J., and Madan, S., "Integrating long polling with an MVC framework". En: USENIX Association. (ed),

Proceedings of the 2nd USENIX conference on Web application development (WebApps'11). Portlan (Estados Unidos), 15-16 de junio de 2011, pp. 121-132, 2011.

- [25] Raymond, E., *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, O'Reilly & Associates, Inc., 2001.
- [26] Trademark, "Official site Debian", 1997. (Web). En:
- [27] <http://www.debian.org/intro/about.en.html>. Último acceso: 20 de Enero de 2012.
- [28] Pilato, C., Collins-Sussman, B. and Fitzpatrick, B., *Version Control with Subversion*, O'Reilly Media, 2008.
- [29] Seqlegal, "Official site Mantis Bug Tracker", 2012. (Web). En:
- [30] <http://www.mantisbt.org/index.php>. Último acceso: 25 de Enero de 2012.
- [31] Silberschatz, A., Korth, H. y Sudarshan, S., *Fundamentos de bases de datos*, McGraw-Hill, 2001.
- [32] DuBois, P., *MySQL Cookbook*, O'Reilly & Associates, 2002.
- [33] Apache Software Foundation, *Apache HTTP Server Reference Manual - for Apache Version 2.2.17*. Network Theory Ltd., 2010.
- [34] Moulding, P., *PHP Black Book*, Paraglyph Press, 2002.
- [35] Durán, A., Ruiz, A., Corchuelo, R. y Toro, M., "Identificación de Patrones de Reutilización de Requisitos de Sistemas de Información". 2001. (Web). En: http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER00/toro.pdf. Último acceso: 1 de febrero de 2012.
- [36] Thau, D., *The Book of Javascript*, Starch Press, 2006.
- [37] Lerner, R., "At the Forge: PostgreSQL 9.0", *Linux Journal*, vol. no 9, pp. 204, 2011.
- [38] Montoya, D., *Desarrollo de un framework de persistencia objetual basado en el patrón de diseño Active Record*. Trabajo de grado, Politécnico Colombiano Jaime Isaza Cadavid, Colombia, 2011.
- [39] M. King, B. Zhu, and S. Tang, "Optimal path planning," *Mobile Robots*, vol. 8, no. 2, pp. 520-531, March 2001.
- [40] H. Simpson, *Dumb Robots*, 3rd ed., Springfield: UOS Press, 2004, pp.6-9.
- [41] M. King and B. Zhu, "Gaming strategies," in *Path Planning to the West*, vol. II, S. Tang and M. King, Eds. Xian: Jiaoda Press, 1998, pp. 158-176.
- [42] B. Simpson, et al, "Title of paper goes here if known," unpublished.
- [43] J.-G. Lu, "Title of paper with only the first word capitalized," *J. Name Stand. Abbrev.*, in press.
- [44] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," *IEEE Translated J. Magn. Japan*, vol. 2, pp. 740-741, August 1987 [Digest 9th Annual Conf. Magnetics Japan, p. 301, 1982].
- [45] M. Young, *The Technical Writer's Handbook*, Mill Valley, CA: University Science, 1989.