

# IMPLEMENTATION OF MODEL DEVELOPMENT OF WEB ORIENTED SOFTWARE TO UNIVERSIDAD SIMÓN BOLÍVAR SEDE CUCUTA

## IMPLEMENTACION DE UN MODELO DE DESARROLLO DE SOFTWARE ORIENTADO A LA WEB PARA LA UNIVERSIDAD SIMÓN BOLÍVAR SEDE CÚCUTA

Recibido: 25 de marzo 2015- aceptado: 26 de mayo 2015

Carlos. Angarita Sanguino.<sup>1</sup>  
O. Gallardo.<sup>2</sup>

### Keywords:

Academic management system, software development, Universidad Simon Bolívar, Headquarter, cucuta

### Abstract

This document describe the general aspects of design new model development information systems oriented web of University Simon Bolivar headquarters Cucuta, as well as implementation of first modules of Educational Information Management and Academic Management. The new development model began defining various design and security policies, which originated a structure software development for headquarters, which allows a process agile and identity institutional in systems. The system of Docent Management Information and Academic Management are two of the first modules which were realized in headquarters supported in the new development model, whose new characteristics are explained in the following document. Finally observed that the model meets the initially defined characteristics and structure ensures a good level of security, agility, dynamism and understanding, but it is clear that maybe evolve over time.

### Palabras clave:

Sistema de gestión académica, modelo de desarrollo de software, Universidad Simón Bolívar, Sede Cúcuta.

### Resumen

El presente documento describe los aspectos generales del diseño del nuevo modelo de desarrollo de sistemas de información orientados a la web de la Universidad Simón Bolívar sede Cúcuta, así como la puesta en marcha de los primeros módulos de Gestión de Información Docente y Gestión Académica. El nuevo modelo de desarrollo inició definiendo diversas políticas de diseño y seguridad, que originaron una estructura de desarrollo de software para la sede, la cual permitió un proceso ágil y una identidad institucional en los sistemas. El sistema de Gestión de Información Docente y Gestión Académica son dos de los primeros módulos que se realizaron en la sede, soportados en el nuevo modelo de desarrollo, cuyas características generales se explicaran en el presente trabajo. Finalmente se observa que el modelo cumple con las características definidas inicialmente y su estructura garantiza un buen nivel de seguridad, agilidad, dinamismo y comprensión, sin embargo es claro que podrá evolucionar con el tiempo.

1. Ingeniero de Sistemas en la Universidad Francisco de Paula Santander de Cúcuta. Jefe del Departamento de Sistemas, Universidad Simón Bolívar, Sede Cúcuta.
2. Ingeniero de Sistemas y Computación, Universidad de los Andes, Bogotá. Docente Universidad Simón Bolívar, Sede Cúcuta.

\*Este artículo es asociado al proyecto de investigación: Modelo de desarrollo de software orientado a la web para la universidad simón bolívar sede Cúcuta

## 1. Introducción

La Universidad Simón Bolívar, debe su existencia al espíritu creador del doctor José Consuegra Higgins, científico social, especializado en economía quien fundó la institución el 15 de octubre del 1972, iniciando sus labores académicas el 23 de marzo del 1973. Desde entonces ha sido su propósito, promover un ambiente pedagógico que favorezca el desarrollo de la capacidad de aprendizaje y comprensión del estudiante y por eso sus profesores poseen una alta calidad intelectual, ética y moral, así como un dominio de diversas metodologías para la enseñanza moderna. (1)

En 1999 la Universidad Simón Bolívar realiza una alianza con el Instituto Inpro sistemas del Norte, que es una Institución de Educación para el Trabajo y Desarrollo Humano fundada en 1985 por el doctor Tomas Wilches Bonilla, dando nacimiento a la sede Cúcuta, que tiene una estructura académico – administrativa propia y que actualmente cuenta con 6 carreras de pregrado, 4 de posgrado y un poco más de 4000 estudiantes, de los cuales un pequeño porcentaje son extranjeros, debido a su ubicación fronteriza.

El Departamento de Sistemas de la Universidad desarrolló el SIA - Sistema de Información Académica como una herramienta informática que da soporte a las actividades académicas y administrativas de la Universidad y que permite la interacción de toda la comunidad académica. La sede Cúcuta como parte de la principal de Barranquilla ha adoptado los distintos módulos que componen el sistema, realizando algunos ajustes según la dinámica propia de la sede y de la zona donde se ubica.

La constante evolución de las diversas tecnologías y metodologías en el área de la informática, así como la pronunciada necesidad de la sede y su constante variación de requerimientos, ha llevado al Departamento de Sistemas junto a las subdirecciones académica y administrativa al diseño, construcción y puesta en marcha de un nuevo sistema académico mediante el uso de nuevas tecnologías y con una tendencia al uso del software libre.

## 2. Marco general

El plan estratégico de la Universidad Simón Bolívar para el 2013-2017 establece un ítem relacionado con el fortalecimiento institucional entre las que se encuentra el objetivo “Consolidar el aporte del Sistema de Información SIAA al desarrollo de los procesos académicos y administrativos del sistema de gestión institucional” (2)

donde se evidencia la importancia que los sistemas de información en la institución y el respaldo que las directivas dan a su constante evolución.

La Universidad Simón Bolívar cuenta con un conjunto de sistemas informáticos que fueron desarrollados en la sede principal en la ciudad de Barranquilla, y que se implementaron en la sede Cúcuta. Estos sistemas adoptan diversos procesos académicos y administrativos de la Universidad. El departamento de sistemas de la sede ha ajustado los sistemas de información a los diversos requerimientos de la misma, pero igualmente hay diversas funcionalidades en el sistema que no son necesarias pero que vienen adicionadas. El departamento ha desarrollado algunos módulos paralelos al sistema SIA de la sede principal para poder dar soporte a los procesos institucionales, como son el ingreso de notas y el control de asistencia.

Una de las principales ventajas del sistema es el hecho de utilizar un sistema de base de datos unificados para todos los módulos, lo cual permite que la integración se produzca a través de las tablas de datos.

## 3. Evolución del sistema actual

El Sistema de Información Académico se refiere a un sistema central creado en la sede principal de Barranquilla el cual comprende un conjunto de aplicativos con arquitectura Cliente-Servidor (3) desarrollados en Visual Basic (4), con persistencia en una base de datos DB2 (5) (6) y que se comunican a través de la red de datos de la institución. Dichos aplicativos componen una solución a las diversas áreas administrativas y académicas de la Universidad y fueron implementados en la sede Cúcuta como una alternativa a la gestión de toda la sede.

Posteriormente se desarrolló en Barranquilla un aplicativo web para que los estudiantes realicen desde la impresión de los recibos de matrícula hasta la escogencia de horarios de los respectivos grupos, dicho aplicativo fue desarrollado en PHP (7); inicialmente fue implementado en Barranquilla pero después de un tiempo se implementó en Cúcuta, donde se le realizaron algunos ajustes propios de la sede y ha sido utilizado por dos semestres. Al momento de implementar el sistema se encontró que diversos datos que eran necesarios para el completo funcionamiento del proceso, debían ser gestionados con otro aplicativo, el cual no se había implementado completamente en la sede.

## 4. Desarrollo del nuevo modelo

La sede de Cúcuta ha implementado gran parte del

software del Sistema de Información Académico que se desarrolla en Barranquilla, como son el Sistema Académico, el Sistema Financiero, los Sistemas de Créditos y la Matricula Web, modificándolos y adicionando diversas funcionalidades propias de la sede, pero con el paso del tiempo ha realizado sus propios desarrollos, como soluciones a los inconvenientes generados o a las solicitudes realizadas.

Dentro de los diversos desarrollos del departamento de sistemas de Cúcuta se tienen el sistema de registro de notas, el cual era orientado a la web, pero con un motor de base de datos distinto al institucional, un módulo de registro de asistencia docente, así como diversos sistemas de encuestas estudiantiles o de aplicación de pruebas psicológicas.

Los principales inconvenientes que se podían evidenciar era la falta de homogeneidad en el desarrollo de los sistemas o módulos, debido a que cada desarrollador interpretaba la necesidad y daba una solución que no era completa, ya que la misma dependía en gran parte de la gestión que este realizaba al momento de ser utilizado por el usuario final; adicional a esto iniciar un desarrollo era extenso, ya que no se contaba con una base y hacer un módulo casi que obligaba a comenzar de cero.

Los principales desafíos que presenta el nuevo modelo de desarrollo se encuentra en proporcionar acceso en todo momento, así como mantener una relación con los datos y las estructuras actuales, mejorar el rendimiento y la seguridad. Algo que pretende este nuevo sistema es brindar una interfaz limpia y sencilla que contenga un alto grado de usabilidad (8) y que permita consolidar una imagen institucional de los sistemas.

El proyecto está orientado a crear una base tecnológica que permita agrupar todos los futuros desarrollos de la sede para gestionar los procesos académicos de la Universidad, y que permita un desarrollo de otros módulos de forma rápida y sencilla.

## 5. Aspectos destacados del sistema

El nuevo modelo de desarrollo debe cumplir con diversos criterios que permitan mantener una imagen corporativa, un rápido desarrollo de modulo, una alta seguridad pero que sea ajustable, etc. Para todo lo

anterior se definen un conjunto de elementos que utiliza el nuevo modelo:

- Orientado a la Web: se desarrolla el sistema orientado a la Web para que permita el acceso desde cualquier lugar.

- Orientado a Objetos: se desarrolla un sistema orientado a objetos que inicialmente demora el desarrollo pero que a la larga brinda mayor facilidad para el mantenimiento del sistema. (9)

- Interfaz: se desarrolla una interfaz con imagen corporativa, que mantiene los colores institucionales y que se convierte en un plantilla para el desarrollo de todas las aplicaciones de la sede. Adicional a esto, la capa de vista cumple con ciertos criterios de adaptabilidad y una dinámica de uso idéntica para todos los módulos.

- Sistema de navegación dinámica: los menús principales y menús de opciones se generan desde la base de datos, y se muestran en la interfaz principal. Los mismos son cargados según los roles asignados por lo tanto pueden ser activados y desactivados. Igualmente el pie de página se genera de forma dinámica según las opciones que tenga disponible el usuario.

- Seguridad a través de roles y perfiles: la seguridad del sistema se maneja a través del uso de usuarios y claves encriptados. Una parte importante de la seguridad es que se maneja la gestión por perfiles y roles, los cuales pueden ser asignados a los usuarios, y permiten mantener seguridad tanto a nivel de vista, como a nivel de controlador o modelo. Los roles pueden ser agrupados en perfiles para que puedan ser asignados a un conjunto de usuarios. Al momento de iniciar sesión se consultan los roles de los perfiles asignados y los cargados individualmente.

- Sistema de gestión de reportes: para la gestión de reportes se incorporaron dos elementos que hay que resaltar. El primero es el uso de Jasper Report<sup>1</sup> para la elaboración de reportes en formato PDF, ya que su diseño es fácil de realizar con la herramienta Ireport<sup>2</sup>. Para poder generar los reportes se utiliza JavaBridge<sup>3</sup> como medio para poder ejecutar los archivos Java que generan el PDF. Adicional se implementó una interfaz que puede ser

<sup>1</sup> Jasper Report es una herramienta gratuita y open source que se compone de un conjunto de librerías java para facilitar la generación de informes en nuestras aplicaciones tanto web como de escritorio. (13)

<sup>2</sup> Ireport es un constructor / diseñador de informes visual, poderoso, intuitivo y fácil de usar para JasperReports escrito en Java. (18) (19)

Investig. Innov. Ing. 2 (3) pp.38-45. Julio – Diciembre 2015 .Universidad Simón Bolívar. Barranquilla, Colombia. ISSN: 2344-8652

<sup>3</sup> PHP / JavaBridge es una implementación de un streaming, que puede ser usado para conectar un script de una maquina nativa con una máquina Virtual de Java. (14)

utilizada para generar todos los reportes, ya que la misma los gestiona de la base de datos de forma dinámica y le asigna un código que identifica los componentes de la interfaz que deben ser renderizados y enviados como parámetros para generar el reporte.

### 6. Visión general de la arquitectura del sistema

La arquitectura del sistema se basa en un patrón MVC (Modelo Vista Controlador) la cual permite separar la interfaz gráfica del usuario de los datos y de la lógica, basándose en tres componentes (10) (11) (12):

- Modelo: representa los datos y reglas del negocio.
- Vista: muestra la información al usuario y con otros sistemas.
- Controlador: reacciona a los eventos generados por el cliente.

Este modelo permite separar la lógica del negocio de la interfaz de usuario lo cual da mucha flexibilidad al desarrollador.

### 7. Componentes principales

La nueva arquitectura plantea los siguientes componentes principales:

- 1) Base de Datos (Modelo): permite realizar la persistencia de la información:
  - Tablas: estructuras de datos que almacenan la información.
  - Funciones: contiene lógica del negocio.
  - Disparadores: funciones que se ejecutan ante cualquier cambio en los datos de las tablas.
  - Clases de Entidad: objetos simples que representan las estructuras de los datos y son gestionados como objetos desde la aplicación.
- 2) Aplicaciones Académicas (Vista): corresponden a todas las interfaces que permiten el ingreso y consulta de información.
- 3) Reportes: dentro del sistema se plantea un modelo de extracción de información a través de reportes PDF que serán diseñados y generados con la ayuda de Jasper Report.

### 8. Arquitectura del sistema

La arquitectura general implementada para el sistema puede verse en la Fig. 1:

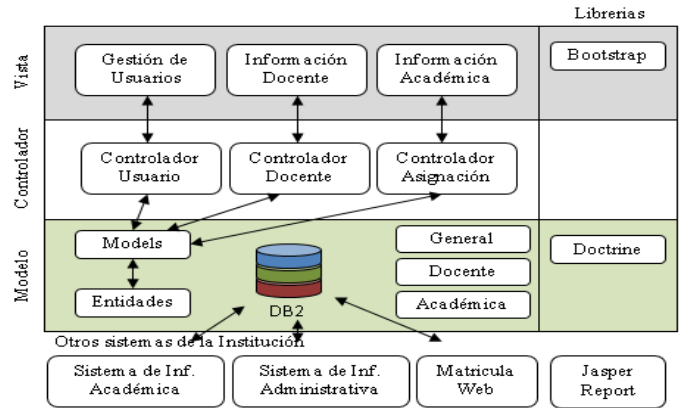


Fig. 1. Arquitectura del sistema del sistema esta basa en el patrón MVC y la interacción con otros sistemas la realizan a través de la base de datos. Fuente: Elaboración propia de los autores.

El sistema académico está dividido en varios módulos con las siguientes capas: Modelo, Controlador y Vista.

En la capa modelo se distingue que el sistema cuenta con un sistema de base de datos unificados, dividido en esquemas que corresponden a cada proceso, pero que igualmente comparten esquemas de configuración y control.

La capa controlador se organiza en módulos según las funcionalidades que desempeña. La organización de los controladores se relaciona directamente con el conjunto de requerimientos a los que debe dar solución.

La capa de vista indica la aplicación cliente encargada de desplegar la información necesaria, que para este caso son una única aplicación, que es gestionada a través de permisos para poder acceder a las diversas funcionalidades. La capa de vista se organiza por módulos según el requerimiento que se resuelva.

La capa de vista se subdivide funcionalmente en los diversos módulos, y la misma se puede clasificar según el modulo que desarrolla. Dentro de los módulos se pueden relacionar dos módulos importantes el de Gestión Académica (Asignación, Horarios, Aulas, etc.) y el de Gestión de Información Docente (Registro de hoja de vida del docente). Igualmente los perfiles se gestionan a través del uso de roles para cada módulo, tanto en la vista como en el modelo.

La relación con otros sistemas institucionales (Matriculas Web, SIA Académico y Financiero) se realiza a través de la

base de datos, debido a que se cuenta con un único motor de datos, que integra todas las aplicaciones. Así por ejemplo la gestión académica que se realiza se convierte en una entrada para el sistema de matrícula web.

## 9. Implementación y prueba del modelo

Este proyecto se evidencia a través del desarrollo de un aplicativo que debe cumplir con diversos requerimientos referentes a la parte académica de la sede, pero con el objetivo principal de crear un modelo de desarrollo de aplicaciones.

### A. Funcionalidades del Sistema

Es una aplicación Web orientada a la gestión de la información académica de la sede Cúcuta de la Universidad Simón Bolívar. La versión actual que se encuentra implementada y funcional en todos los programas académicos. Entre sus principales funcionalidades se encuentran las concernientes a la gestión académica:

- Administración de carga académica: permite la creación de cursos de asignaturas y horarios de los mismos. Igualmente permite la edición de los horarios, validando el cruce de salones y el cruce de horarios de docentes que se encuentren adscritos a los grupos. Permite la asignación de la carga académica de los docentes, validando el cruce de aulas y los topes de asignación de horas, según el tipo de contrato del docente.
- Administración de infraestructura: permite la gestión de las aulas y laboratorios, juntos con sus respectivos elementos audiovisuales. Genera reportes de ocupación de aulas para determinar la disponibilidad de salones.
- Administración de actividades complementarias: permite la asignación de actividades a los docentes por actividades externas al aula de clase, como son semilleros de investigación, coordinación, etc. Igualmente se permite la respectiva validación de las horas asignadas para temas de pagos en nómina.

Adicional a los módulos de gestión académica, se cuenta con el módulo de gestión de información docente, cuya finalidad principal es que los docentes realicen el registro y actualización de su hoja de vida, según un formato definido por la sede, así de esta forma se pueden registrar estudios, experiencia, producción académica, etc. Dentro de las principales funcionalidades se encuentra:

- Administración de formación académica y continuada: permite realizar el registro de los diversos títulos del

docente, tanto en formación pregrado y posgrado, así como aquella formación referente a cursos y diplomados.

- Administración de experiencia laboral: permite el registro de la información laboral de los docentes.
- Administración de producción académica: permite el registro de la información de la producción de libros, artículos, investigación, etc. que realiza el docente. Igualmente permite el registro de otra información que se solicita en la institución.
- Generar hoja de vida: permite a los docentes o funcionarios de los distintos programas, el generar la hoja de vida del docente según la plantilla definida por la institución.
- Validación de información: el sistema cuenta con una funcionalidad que permite la validación de la información registrada por los docentes. Dicha validación la puede realizar el departamento de Talento Humano, el cual determina la veracidad de la información o realizan las respectivas correcciones, para que la misma sea publicada.

En la se puede apreciar la interfaz del sistema, que consiste en un menú horizontal el cual provee las opciones principales que generalmente pertenecen a un módulo. En la sección izquierda se encuentra el menú de opciones secundarias que corresponden al submenú de cada módulo. La sección central está destinada a la visualización de información y el pie de página muestra un desglose de todas las opciones disponibles en el sistema, según el perfil del usuario. La interfaz se desarrolló pensando en que fuera adaptable a cualquier dispositivo, ya sea equipos de escritorio, tabletas o móviles.

The screenshot shows the 'Administración Carga de Docentes' interface. It features a top navigation bar with 'Inicio', 'Docentes', 'Actividades', 'Asignación Académica', and 'Reportes'. A left sidebar contains 'Opciones de Menú' with items like 'Carga Académica', 'Administrar Curso', 'Registrar Curso', and 'Consultar Horarios'. The main content area includes a 'Formulario de Búsqueda de Cursos de Asignatura' with dropdowns for 'Programa' (TRABAJO SOCIAL), 'Asignatura' (Debe seleccionar un programa), 'Semestre', 'Jornada', 'Docente', and 'Grupo'. Below the form is a 'Listado de Cursos de Asignaturas (2)' table.

Programa	Código	Nombre	Sem	Grupo	Jornada	Cap	Docente	Nombre	Acciones
TRABAJO SOCIAL	8548	ELECTIVA EN TRABAJO SOCIAL II	6	A	DIURNO	30	1000	DRADONARI	🔍 🗑️ 🔄
TRABAJO SOCIAL	8548	ELECTIVA EN TRABAJO SOCIAL II	6	A	NOCTURNO	25	5000	COORDINADOR CONTINUA	🔍 🗑️ 🔄

At the bottom, there are four main menu categories: 'Docentes' (Importar Docentes, Listar Docentes), 'Actividades' (Actividades Académicas), 'Asignación Académica' (Carga Académica, Administrar Curso, Registrar Curso, Consultar Horarios), and 'Reportes' (Sin Navegación). A footer contains copyright information and a navigation path: 'Inicio | Sistema Académico | Aula Estándar | Docentes | Egresados | Funcionarios'.



Fig. 1 La interfaz del sistema se diseñó pensando en mantener la imagen institucional, que la navegación sea lo más sencilla posible y adaptable a cualquier dispositivo. Fuente: Elaboración propia de los autores.

El sistema adicionalmente cuenta con un módulo de gestión de usuario, roles, perfiles y menús, para la gestión de los permisos y privilegios de acceso al sistema.

### B. Implementación

Toda la arquitectura se pone a prueba con la creación de los primeros módulos de gestión académica y gestión de información docente. Todo el desarrollo se realizó en tecnología orientada a la Web, basado en el lenguaje PHP y HTML<sup>4</sup> y se especifica de la siguiente forma:

- **Vista:** la vista se desarrolló con el uso de Twitter Bootstrap<sup>5</sup>, con lo cual se logra un desarrollo consistente en la vista y que es adaptable a todos los navegadores y dispositivos. La comunicación a la siguiente capa se realiza a través de la interacción del usuario, que envía una acción al controlador.
- **Controlador:** se implementa utilizando clases POO<sup>6</sup> en PHP que reciben las peticiones desde la vista, a través del uso de direcciones amigables, y permiten realizar la invocación o llamado de métodos que se encuentran dentro de las clases y las mismas inyectan los objetos a las clases del modelo. Igualmente los POO reciben objetos mapeados del modelo y son inyectados hacia la vista para que la información sea desplegada.

<sup>4</sup> HTML es el lenguaje de marcado para páginas web (20). Este estándar está a cargo de la W3C – Consorcio World Wide Web (21)

<sup>5</sup> Colección de herramientas de software libre para la creación de sitios y aplicaciones web (15)

<sup>6</sup> Programación Orientada a Objetos: referencia los objetos realizados en PHP.

- **Modelo:** El modelo es la base de datos, la cual se encuentra en DB2. Esta base de datos se utiliza a través de clases POO en PHP que fueron mapeadas a través de un ORM<sup>7</sup> llamado Doctrine, el cual proporciona una capa de persistencia basada en anotaciones y que adicional cuenta con un lenguaje DQL<sup>8</sup>. Doctrine puede generar las clases POO correspondientes a las entidades o tablas de las bases de datos, pero debido a la versión de la base de datos que se utiliza no fue posible, así que las estructuras fueron mapeadas manualmente, pero las mismas permiten gestionar las tablas como si fuesen objetos.

### C. Metodología de desarrollo

Para el desarrollo del proyecto se realizaron un conjunto de fases dentro del desarrollo de cada módulo. Estas fases fueron definidas para todos los desarrollos:

- **Captura de requerimiento:** los requerimientos son dados por los funcionarios que harán uso del sistema, programas, dependencias, etc.
- **Prueba de tecnología y ajuste del modelo de desarrollo:** se realizan las pruebas del funcionamiento del modelo y se hacen los respectivos ajustes.
- **Desarrollo del módulo:** se desarrolla el modulo, comenzando con la definición de perfiles, menú, roles y terminando con el desarrollo del requerimiento.
- **Prueba del módulo y documentación:** se debe realizar pruebas con los usuarios finales del sistema y se deben crear los respectivos manuales del sistema.
- **Puesta en producción y capacitación.**

Debido a la premura de los resultados, se realizan prototipos iniciales que son sometidos a pruebas por los funcionarios, los cuales realizan recomendaciones u observaciones.

### D. Puesta en producción

Para la puesta en producción del sistema, se realizaron dos estrategias paralelas, una para el módulo de carga académica y el otro para el módulo de información docente. Inicialmente se procedió a crear los usuarios de los distintos programas académicos y de los departamentos, los cuales realizaron el registro y creación de los cursos y horarios de cada programa. Para esta implementación se realizaron capacitaciones con los usuarios, en las cuales se explicó el funcionamiento del sistema y los funcionarios retroalimentaban con

<sup>7</sup> Mapeador de Objetos Relacional: técnica utilizada para convertir datos de un modelo relacional a un modelo orientado a objetos. (16)

<sup>8</sup> Doctrine Query Language: lenguaje de consultas inspirado en Hibernate que utiliza un lenguaje de SQL (17)

observaciones que daban origen a ajustes. En total se ingresaron 1332 registros correspondientes a los cursos y 1974 registros correspondientes a los horarios de los cursos. Esta información se convirtió en el insumo de la matrícula web, y debido a inconvenientes de disponibilidad de docentes, era necesario cambiar horarios, para lo cual el sistema debió validar cruces de salón, de horarios de docentes y horarios de estudiantes.

Para implementar el módulo de formación docente se realizó un proceso de revisión de hojas de vidas en recursos humanos. Se recolectaron y cargaron al sistema una cantidad inicial de 450 títulos de formación y los docentes ya han actualizado sus hojas de vida, registrando 96 títulos más. En lo referente a educación continuada, los docentes han registrado 256 registros. Inicialmente el proceso con los docentes fue complicado, pero hubo varios factores que apoyaron el proceso de registro de los datos, entre los que se encuentra:

- Manual en línea del uso del sistema de registro de información docente.
- Gestión de los programas para realizar un seguimiento de los docentes que habían actualizado sus hojas de vida, para lo que se elaboró un reporte que mostrara la fecha de actualización o inclusión del último registro.

A los largo del proceso se vio la necesidad de definir nuevos roles y asociarlos al sistema, así como de crear nuevos perfiles.

#### 10. Conclusiones y trabajos futuros

Este proyecto generó un modelo de desarrollo de aplicaciones, que se convierte en la base de todos los futuros sistemas que se realicen en la sede, y después de diversas reuniones, se evidencia que hay bastante trabajo aun por realizar y diversos sistemas que desarrollar, tanto para la gestión académica como para la gestión administrativa. Después de reuniones iniciales con los jefes de procesos, ya se estudia la posibilidad de desarrollar diversos módulos y sistemas:

- Sistema de Gestión de Congresos
- Sistema de Gestión y Seguimientos de Clientes (Estudiantes)
- Sistema de Gestión de Grupos de Investigación y Semilleros

Al cierre de este artículo ya se encontraba en desarrollo un módulo para la gestión de proyectos de investigación formativa y un sistema de bitácora de docentes. Igualmente se pudo evidenciar que el proceso de

desarrollo de nuevos módulos es muy rápido, y los procesos y documentación realizados en el modelo inicial son muy útiles y permiten una apropiación de la tecnología de forma rápida.

#### Agradecimientos

Los autores desean agradecer a todo el equipo del trabajo del departamento de sistemas de la sede Cúcuta, especialmente el trabajo del Ing. Miguel Roper y el Ing. Yan Angarita que aportaron parte del desarrollo de este proyecto.

Igualmente expresar un agradecimiento a las directivas de la sede Cúcuta, encabezadas por el Dr. Tomás Wilches Bonilla, por la confianza recibida para la realización de este proyecto.

Y para finalizar un agradecimiento al Ing. Gustavo Chacón y su equipo de trabajo de la sede principal, ya que ellos desarrollaron todo lo referente al sistema de información académica.

#### Referencias

- [1] Universidad Simón Bolívar extensión Cúcuta. [En línea] <http://www.unisimoncucuta.edu.co/es/la-universidad/antecedentes-historicos>.
- [2] Plan de Gestión 2013-2017 Universidad Simón Bolívar.
- [3] Universidad de las Américas Puebla. [En línea] [Citado el: 2014 de 10 de 10.] [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/marquez\\_a\\_bm/capitulo5.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf).
- [4] Microsoft Developer Network. [En línea] [Citado el: 2014 de 11 de 01.] <https://msdn.microsoft.com/es-es/library/2x7h1hfk.aspx>.
- [5] IBM. [En línea] 09 de 09 de 2014. <http://www-01.ibm.com/software/data/db2/>.
- [6] Wikipedia - DB2. [En línea] 09 de 09 de 2014. <http://es.wikipedia.org/wiki/DB2>.
- [7] Sitio Oficial PHP. [En línea] [Citado el: 2014 de 10 de 10.] <http://php.net/>.
- [8] Yusef Hassan, Francisco J. Martín Fernández. Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información. [En línea] 2004. [Citado el: 2014 de 10 de 15.] [http://www.upf.edu/hipertextnet/numero-2/disenio\\_web.html](http://www.upf.edu/hipertextnet/numero-2/disenio_web.html).
- [9] Construyendo aplicaciones web con una metodología de diseño orientada a objetos. Dario Andres Silva, Barbara Mercerat.
- [10] Pitt, Chris. PRO PHP MVC. 2012: s.n.
- [11] Cómo mantener el patrón modelo-vista-controlador en una aplicación orientada a la WEB. López, Carlos

- Armando. Diciembre de 2009, Tecnología en Informática, Vol. 7.
- [12] Patrón Modelo-Vista-Controlador. Yenisleidy Fernández Romero, Yanette Díaz González. 1, 2012, Telemática, Vol. 11.
- [13] Sanroman, Javi. <http://jsanroman.net/2007/11/%C2%BFque-es-jasper-reports-2/>. [En línea] [Citado el: 2013 de 11 de 20.] <http://jsanroman.net/2007/11/%C2%BFque-es-jasper-reports-2/>.
- [14] PHP / Java Bridge. [En línea] <http://php-java-bridge.sourceforge.net/pjb/>.
- [15] Wikipedia - Twitter Bootstrap. [En línea] [http://es.wikipedia.org/wiki/Twitter\\_Bootstrap](http://es.wikipedia.org/wiki/Twitter_Bootstrap).
- [16] Object Role Modeling. [En línea] <http://www.orm.net/>.
- [17] Doctrine. [En línea] <http://www.doctrine-project.org/>.
- [18] JasperSoft Community. [En línea] <http://community.jaspersoft.com/project/ireport-designer>.
- [19] Adictos al trabajo. [En línea] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=ireport>.
- [20] HTML - Wikipedia. [En línea] 2014 de 10 de 10. <http://es.wikipedia.org/wiki/HTML>.
- [21] W3C. [En línea] 10 de 10 de 2014. <http://www.w3.org/>.