



Scientia Et Technica

ISSN: 0122-1701

scientia@utp.edu.co

Universidad Tecnológica de Pereira

Colombia

MARTÍNEZ VILLALOBOS, GUSTAVO; CAMACHO SÁNCHEZ, GERMÁN DARÍO; BIANCHA
GUTIÉRREZ, DANIEL ALBERTO

DISEÑO DE FRAMEWORK WEB PARA EL DESARROLLO DINÁMICO DE APLICACIONES

Scientia Et Technica, vol. XVI, núm. 44, abril, 2010, pp. 178-183

Universidad Tecnológica de Pereira

Pereira, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=84917316032>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

DISEÑO DE FRAMEWORK WEB PARA EL DESARROLLO DINÁMICO DE APLICACIONES

Design Framework for the Development Dynamic Web Applications

RESUMEN

El uso de Frameworks en la construcción de aplicaciones Web es un tema de reciente desarrollo e investigación en el área del software, y está muy relacionado con el concepto “Web 2.0” y los sistemas modernos de información. Este artículo presenta avances del proyecto que plantea el diseño y creación de un Framework web robusto y eficiente, implementado con herramientas de software libre, que facilite el desarrollo de aplicaciones web corporativas a través de asistentes, con su estructura general (sesiones, seguridad y navegación) y las diferentes páginas de gestión, a partir del modelo de datos de la aplicación.

PALABRAS CLAVES: SGBD, Framework, Ingeniería Web, LDAP, ORM, Página Web dinámica, Patrón de diseño, WebApps.

ABSTRACT

Using Frameworks in building Web applications is a topic of recent developments and research in the area of software, and is related to the concept “Web 2.0” and modern information systems. This paper presents progress of the project involving the design and creation of a robust and efficient Web Framework, implemented with free software tools that facilitate the development of corporate web applications through wizards, with its general structure (session, security and navigation) and the various pages of management, from the model of application data.

KEYWORDS: DBMS, Design Pattern, Dynamic Website, Engineering Web, Framework, LDAP, ORM, WebApps.

1. INTRODUCCIÓN

En el desarrollo tradicional de un sistema de información es indispensable la normalización de datos; no importa cómo se manipule la información de una empresa u organización lo ideal es que esté estructurada de un modo conocido para poder manejarla, almacenarla, recuperarla. Para este proceso se definen modelos de datos con una determinada estructura (que habitualmente se convierten en tablas de una base de datos). Anteriormente, la única información estructurada en un sistema era la referente a “los datos”. El problema subsecuente a este es que un sistema es mucho más que datos (Código fuente, librerías, archivos de configuración, etc.). Todo este código generado y su orden mismo dependían directamente de los encargados de desarrollar el sistema y, desde la existencia de los primeros compiladores hasta la aparición de los primeros generadores de código comercial u orientado a “usuarios finales”, la generación de código era exclusividad de programas compiladores especializados.

Ante esta situación problemática surgieron los llamados “Frameworks”, [1] con el propósito de normalizar y

estructurar el código del sistema, facilitando un esquema (un patrón, un esqueleto) para el desarrollo y/o la implementación de aplicaciones. El uso de frameworks para cualquier tipo de desarrollo reduce el tiempo de elaboración e implementación y ayuda a hacer un trabajo mantenible y escalable, según las características del mismo [2].

Un framework agrega funcionalidad extendida a un lenguaje de programación, automatiza muchos de los patrones de programación para orientarlos a un determinado propósito, proporcionando una estructura al código, mejorándolo y haciéndolo más entendible y sostenible, y permite separar en capas la aplicación. En general, divide la aplicación en tres capas:

- La lógica de presentación que administra las interacciones entre el usuario y el software.
- La Lógica de datos que permite el acceso a un agente de almacenamiento persistente u otros.
- La lógica de dominio o de negocio, que manipula los modelos de datos de acuerdo a los comandos recibidos desde la presentación.

GUSTAVO

VILLALOBOS

Ingeniero de Sistemas, Esp.

Profesor Asistente

Programa Ingeniería de Sistemas

Universidad de Ibagué

gustavo.martinez@unibague.edu.co

MARTÍNEZ

GERMÁN DARÍO CAMACHO SÁNCHEZ

Estudiante último semestre

Ingeniería de Sistemas

Universidad de Ibagué

germancamacho@gmail.com

DANIEL ALBERTO BIANCHA GUTIÉRREZ

Estudiante último semestre

Ingeniería de Sistemas

Universidad de Ibagué

dabiang@bundenet.com

Los Web Framework [3] pretenden facilitar el desarrollo de Aplicaciones web (Sitios web, intranets, etc). Actualmente, PHP es conocido por su simplicidad y es ampliamente usado en este campo del desarrollo de software web. También es común el uso de otros lenguajes “de scripting” o interpretados como Perl, Asp ó JSP que permiten mayor facilidad de uso y flexibilidad en el rápido desarrollo de aplicaciones, pero al mismo tiempo tienden a producir código de difícil sostenimiento y poco eficiente. En este ámbito, los patrones de diseño más utilizados son aquellos que se centran en separar la presentación (páginas HTML, CSS) de la lógica o backend.

También es relevante la técnica de programación ORM [4] para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos; es el mapeo objeto-relacional (más conocido por su nombre en inglés, Object-Relational mapping). En la práctica, crea una base de datos orientada a objetos virtual sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo). Hay paquetes comerciales y de uso libre disponibles que desarrollan el mapeo relacional de objetos, aunque algunos programadores prefieren crear sus propias herramientas ORM.

El desarrollo de sistemas de información es una de las tareas que más tiempo requiere en las empresas de desarrollo de software, debido a que, además del tiempo que se necesita dedicar al levantamiento de la información y los requerimientos, la implementación de la solución en determinado lenguaje o herramienta es un proceso técnico complejo.

El presente trabajo plantea el diseño y creación de una estructura Framework bien definida, reusable, en la que sus componentes faciliten la creación de aplicaciones web, provea una capa de abstracción sobre la arquitectura original, ocultándola o adaptándola para no tener que utilizar el protocolo http de manera nativa y así acelerar los tiempos de desarrollo y mantenimiento de software. Esta solución es propuesta con base en herramientas de software libre, aprovechando los beneficios de las licencias GPL, en pro de reducir los costos de licenciamiento, buscando la calidad y rapidez de desarrollo que este tipo de framework brinda, facilitando de esta manera también la adquisición de sistemas como estos por empresas pequeñas. Los frameworks ofrecen un alto grado de reutilización –mucho más que con clases individuales-. En consecuencia, si una organización está interesada (¿y quién no lo está?) en incrementar su grado de reutilización del software, entonces debería enfatizar la creación de frameworks¹.

Para el desarrollo se adoptó un proceso secuencial con orientación IWeb; por su estructura, en este tipo de proyectos se puede sacar provecho a su organización. El proceso genérico del marco de trabajo – comunicación, planeación, modelado, construcción y despliegue- es aplicable a la IWeb [5]. Estas cinco actividades del marco de trabajo se refinan en un conjunto de tareas del IWeb que se adaptan a las necesidades del Framework Web. Se aplicaron empleando un flujo del proceso incremental, como se muestra en la siguiente figura.²

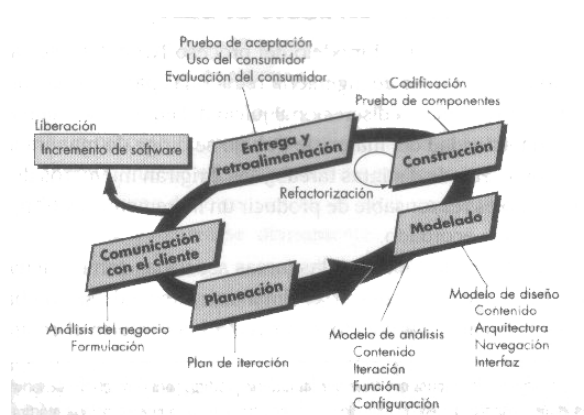


Figura 1. Proceso IWeb

2. PROCESO DE DESARROLLO WEB

La ingeniería Web (IWeb) es el proceso con el que se crean WebApps de alta calidad. La IWeb no es un clon perfecto de la ingeniería de software, pero toma prestados muchos conceptos y principios fundamentales de ella. Además, el proceso IWeb acentúa actividades técnicas y administrativas similares. Existen sutiles diferencias en la manera como se dirigen dichas actividades, pero el método primordial dicta un enfoque disciplinado para el desarrollo de un sistema software.

En el desarrollo de software, un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. En programación, es un set de funciones o código genérico que realiza tareas comunes y frecuentes en todo tipo de aplicaciones (creación de objetos, conexión a base de datos, etc.). Esto brinda una base sólida sobre la cual desarrollar aplicaciones concretas y permite obviar los componentes más triviales y genéricos del desarrollo y abre camino a que diseñadores y programadores puedan pasar más tiempo identificando requerimientos de software.

¹ Larman, Craig, UML y patrones – Análisis y diseño orientado a objetos. 2da Edición. Pág. 503.

² Pressman, Roger, Ingeniería de Software – Un enfoque práctico. 6ta Edición. Pág. 511.

En general, los frameworks son contruidos con base en lenguajes orientados a objetos. Esto permite una mejor modularización de los componentes y óptima reutilización de código. Además, en la mayoría de los casos un framework implementará uno o más patrones de diseño de software que aseguren la escalabilidad del producto. El uso de frameworks web para cualquier tipo de desarrollo reduce el tiempo de elaboración e implementación y ayuda a hacer un trabajo mantenible y escalable, según las características del mismo.

2.1 Análisis y diseño del sistema

El plan trazado para el proyecto Framework Web consistió de una definición de tareas y un calendario de plazos respecto al periodo proyectado para el desarrollo del incremento de la WebApp. Se establecieron los casos de uso de requerimientos funcionales³ y no funcionales para que el sistema cumpliera con los objetivos propuestos, como la generación de formularios simples, maestro detalle, formularios de búsqueda, pantallas de resultados y a su vez la administración de toda la Metadata que se requiere para crear una interfaz de usuario amigable, tanto para el desarrollo como para el ambiente de producción e implementar un esquema de manejo de usuarios, y roles que se ajuste a cualquier tipo de aplicación en general y que sea compatible con protocolo LDAP.

Nombre	Configuración de listas de valores
Requerimiento:	
Clasificación:	Funcional
Descripción del Requerimiento:	
Se requiere que el software provea al usuario una funcionalidad que permita configurar las distintas listas de valores a utilizar en todos los formularios desarrollados.	
Consideraciones:	
Las listas de valores podrán ser estáticas o dinámicas basadas en datos que sean ingresados en otras estructuras, a manera de consultas de otras tablas, estas mismas deberán considerar el uso de relaciones Padre-Hijo para utilizar listas dependientes.	

Figura 2. Ejemplo de requerimiento funcional del proyecto

LDAP ("Lightweight Directory Access Protocol", en español, Protocolo Ligero de Acceso a Directorios) es un protocolo de tipo cliente-servidor para acceder a un

servicio de directorio. Un directorio es una base de datos, pero en general contiene información más descriptiva [6].

Las labores convencionales de análisis y diseño de ingeniería del software se adaptaron al desarrollo del Framework Web. El intento era desarrollar análisis "rápidos", modelos y patrones de diseño que definieran los requisitos y al mismo tiempo representaran una WebApp que cumpliera con las necesidades del proyecto.

Se realizó el modelo Entidad-Relación, encargado de manejar la metadata de las aplicaciones, esquema general de usuarios con permisos y autorizaciones, configuración dinámica de apariencia visual, definición del motor para la aplicación de plantillas.

2.2 Estudio de reutilización

A partir de los modelos obtenidos se revisaron las secciones que mostraban funcionalidades similares o iguales, para determinar si se podían crear componentes que homologaran la funcionalidad entre sí.

Se realizó la búsqueda de herramientas *open-source* que facilitaran la implementación de las funciones específicas, como por ejemplo la aplicación de plantillas, o manejo general de bases de datos mediante interfaces en PHP [7].

2.3 Ámbito del sistema

El Framework Web para el Desarrollo Dinámico de aplicaciones se encargará de generar la infraestructura esquemática de formularios, campos, relaciones, y operaciones CRUD, a partir del modelo de base de datos de un sistema previamente contemplado.

A partir de la base de datos generada para un sistema, el Framework permitirá la configuración de todas las tablas del mismo que requieran ser usadas y/o administradas a través de un entorno web, facilitando la personalización y validación de cada atributo (campo).

Cada formulario podrá ser configurado independientemente de acuerdo a las especificaciones que necesite en cuanto a permisos de acceso y operaciones permitidas.

La administración del Framework tiene la funcionalidad de ser modificable, ajustándose a las necesidades del sistema para el que se esté adaptando.

3. ARQUITECTURA DEL SISTEMA

³ Fuente: los autores

La arquitectura del Framework es inherentemente Web para poder lograr los requerimientos propuestos, pero una arquitectura Cliente/Servidor Web [8] no le da el valor agregado que sí da la correcta definición de componentes y delegación de responsabilidades a estos mismos dentro de la aplicación.

Esta arquitectura deberá aportar a la calidad del desarrollo, teniendo como objetivo la escalabilidad, fiabilidad, desempeño y seguridad así que los componentes a diseñar deben cumplir o permitir obtener estos puntos.

La Información de funcionamiento del Framework se guarda en una estructura de tablas de base de datos, por lo que el acceso a los datos es primordial, por esto para completar la arquitectura es necesario indicar que el servidor debe permitir la comunicación con el respectivo DBMS.

Establecidas las necesidades, se definió la arquitectura Cliente/Servidor⁴ mediante la cual el acceso al Framework se realiza a través de un Navegador de Internet, teniendo claro que utilizará http sobre TCP/IP como protocolo de comunicación. Es importante indicar que estas características permiten que el Framework sea utilizado como aplicativo tanto en ambiente Intranet, como en Internet [9].

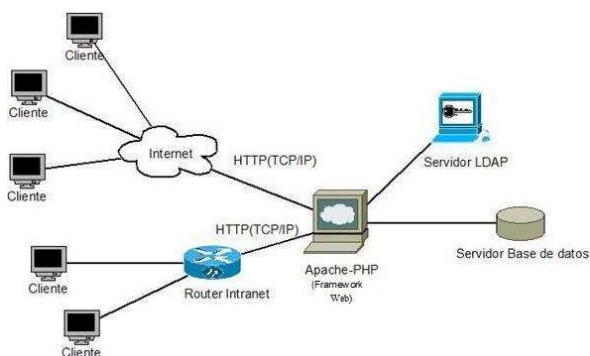


Figura 3. Arquitectura del sistema

Las herramientas y la tecnología IWeb aplicadas para construir el Framework Web modelado, dieron como resultado el incremento del WebApp, se dirigió una serie de pruebas rápidas para asegurar que se descubrieran los errores en el diseño (es decir: contenido, arquitectura, interfaz, navegación).

3.1 Componentes

⁴ <http://blog.buhoz.net/blog1.php/2008/03/06/frameworks-mvc-de-desarrollo-web> (Frameworks MVC de desarrollo Web, sitio consultado en marzo de 2009).

Para complementar la arquitectura escogida y enfrentar diferentes problemas de interoperabilidad, seguridad, facilidad de acceso y desempeño, se definieron los siguientes componentes⁵:

- **Controlador:** Se encarga de procesar las solicitudes que se realizan desde la interfaz gráfica y las re-direcciona a los procesos especializados que devuelven su respuesta mediante el mismo controlador.
- **Gestor de Seguridad:** Encargado de validar los permisos del usuario en sesión para realizar la autenticación y autorización del mismo, este componente es accedido por el controlador en el momento de requerir una acción determinada así se logra en el momento que la sesión del usuario tenga cualquier cambio, que este se vea reflejado al usuario.
- **Gestor de formularios:** Realiza todo el trabajo relacionado con la carga, validación y operaciones CRUD sobre los formularios, de esta manera se desacopla la operación central del Framework de lo relacionado con acceso a datos y gestión de seguridad.
- **Gestor de acceso a datos:** Este componente se encarga de realizar el acceso a las Bases de datos que se hayan configurado, así se centraliza el acceso y se permite el uso de diferentes tipos de base de datos. Este componente se implementará para cumplir el rol de capa de abstracción de datos.

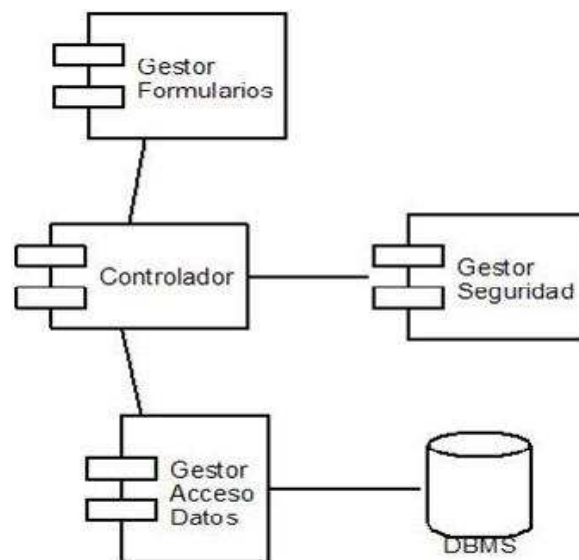


Figura 4. Componentes del sistema

3.2 Interfaz gráfica

⁵ Fuente: los autores.

Se eligió la maquetación por cascadas de estilo CSS para ser la encargada de ubicar los componentes de interfaz gráfica, y para dar todas las características correspondientes a tamaño de letra, espaciado, tamaños de títulos, colores, imágenes e íconos.

Figura 5. Ejemplo de lista con base en tabla (Fuente: autores)

La cascada de estilos CSS generada para el Framework [10] a manera de base contempla diferentes conceptos del diseño de interfaces, para facilitar su manejo a los usuarios. Entre estos se encuentran, los tamaños del texto, que permiten en las resoluciones óptimas de trabajo la lectura de los textos generados por el sistema.

3.3 Un resultado: selección de tablas

Uno de los resultados, a partir de la implementación del proyecto, es la posibilidad de seleccionar las tablas que aún no son gestionadas por el sistema para realizar la carga de su correspondiente información.

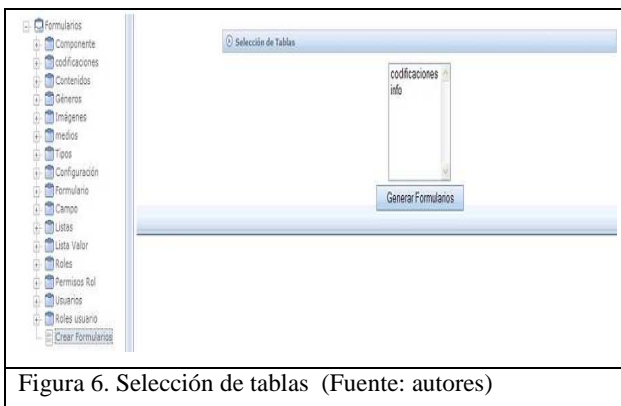


Figura 6. Selección de tablas (Fuente: autores)

Cuando se han seleccionado las tablas y después se elige el botón “Generar Formularios”, se realiza el proceso de carga, y se muestran los mensajes correspondientes a la carga tanto de los campos como de la tabla en general.

3.4 Herramientas de desarrollo

Para el desarrollo del Framework Web se dejaron a disposición las siguientes herramientas de software libre:

Komodo Edit 4.4 (www.openkomodo.com): Editor de texto con herramientas de marcado de sintaxis PHP, carga dinámica de variables en tiempo de desarrollo y proposición de funciones propias y de usuario para autocompletar.

MySQL versión 5.0.51b (www.mysql.com): Para el desarrollo y pruebas se utiliza este DBMS en su versión community; este motor de base de datos permite cargar la información teniendo siempre como objetivo las pruebas del Framework en un ambiente cercano al de producción.

Apache (www.apache.org): Servidor de red para el protocolo http. Una vez se inicializa crea unos subprocesos para poder gestionar las solicitudes. Apache se encarga de resolver las peticiones de páginas de Internet de los clientes utilizando el protocolo de Internet http. Es la herramienta libre más utilizada como servidor Web a nivel mundial, y es confiable para dar al Framework disponibilidad 7/24.

PHP (www.php.net): Lenguaje de scripting seleccionado por su versatilidad, rapidez, escalabilidad. Se configura como módulo de Apache y permite de una manera sencilla configurar un balanceador de carga o un sistema espejo para sobrellevar cualquier problema de no disponibilidad [11].

AdoDB (www.adodb.sourceforge.net): Es una librería de abstracción para PHP, mediante la cual se puede realizar acceso a diferentes fuentes de datos de manera transparente y estandarizada. AdoDB permite realizar tareas de una manera única solamente cambiando el motor de base de datos al que se accede, así que los desarrollos pueden quedar desacoplados de la implementación del DBMS.

Subversion 1.5.5 (www.subversion.tigris.org): Sistema de control de versionamiento o CVS, permite conexión HTTP, en el cual se crea un proyecto gestionado por el CVS, que se encarga de asignar un código a cada archivo para poder llevar control sobre el mismo y facilitar a los desarrolladores llevar una traza sobre los cambios sobre diferentes tipos de archivo.

TortoiseSVN 1.5.5 (www.tortoissvn.net): Herramienta sobre Windows que realiza las tareas de cliente CVS, el cual se conecta a un servidor externo para acceder a un repositorio de archivos versionado. Esta herramienta permite crear repositorios locales, también posee herramientas que permiten hacer “merge”, unión de dos archivos editados por diferentes desarrolladores para lograr unificar los cambios realizados.

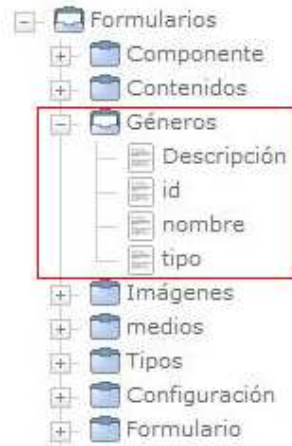


Figura 7. Menú modo Administrador (Fuente: autores)

4. CONCLUSIONES Y RECOMENDACIONES

El Framework Web dispone de las siguientes funcionalidades: Proceso de autenticación, administración de roles y de usuarios, creación y administración de formularios, generación de formularios a partir de tablas, manejo de listas de valores y gestión de registros.

Debido a la naturaleza del producto desarrollado, se realizó la selección de un modelo pequeño de dominio para solucionar mediante la herramienta y de esta forma lograr explorar y comprobar que se habían alcanzado los requerimientos.

Fue evidente la importancia de definir estándares de codificación y de administración del código fuente para lograr el nivel de calidad requerido en la implementación del Framework.

La comunicación entre la base de datos y el usuario en cuanto a entradas y salidas es facilitada por medio de una interfaz web apoyada en una arquitectura Cliente/Servidor.

Es necesario efectuar la revisión de ciertos detalles de compatibilidad entre las partes principales del sistema, las diferentes librerías que permiten realizar AJAX de una manera Cross-browser, manejo de gestores de plantillas, y acceso dinámico a la Metadata de las tablas en diferentes motores de Base de Datos.

Aunque se realizaron pruebas de las funcionalidades del sistema, es recomendable aplicar otros modelos de prueba y continuar investigando para el proceso de evolución de este producto software.

5. BIBLIOGRAFÍA

- [1] Qué es un framework? Referencia web [en línea].
<http://www.asembla.com/wiki/show/dX01x-pRGr3le5abIIDkbG/7-2-queesunframework>.
- [2] Larman, Craig. UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. Madrid: Pearson Educación, 2003.
- [3] Frameworks MVC de desarrollo web. Referencia web [en línea].
<http://www.blog.buhoz.net/blog1.php/2008/03/06/frameworks-mvc-de-desarrollo-web>.
- [4] Ceballos, Francisco. JAVA2. Interfaces gráficas y aplicaciones para Internet. México: Alfaomega-RaMa, 2008.
- [5] Pressman Roger S, Ingeniería del Software. Un enfoque práctico. México: McGraw-Hill, 2005.
- [6] Kroenke, David. Procesamiento de Bases de Datos. México: Pearson Educación, 2003.
- [7] Bravo, Ginés. PHP5. México: Alfaomega-RaMa, 2005.
- [8] Sommerville Ian, Ingeniería del Software. Madrid: Pearson, 2005.
- [9] Generación de código en la programación web avanzada. Referencia web [en línea].
http://www.geothesis.com/index.php?option=com_content&task=view&id=346.
- [10] Frameworks para PHP. Referencia web [en línea].
<http://www.slideshare.net/andreskarp/frameworks-para-php-adwa>
- [11] López Quijado, José. PHP y MySQL Programación dinámica en el lado del servidor. México: Alfaomega-RaMa, 2007.