Aplicaciones Web - Cakephp - Aplicación Académica

Realizado por Julio Agüero Sánchez

II Semestre 2012

Tabla de contenido

[Descripción del problema 2](#_Toc335741560)

[Diseño del programa 3](#_Toc335741561)

[Decisiones de diseño 3](#_Toc335741562)

[Diagrama de clases 4](#_Toc335741563)

[Diagrama de ER de la base de datos 5](#_Toc335741564)

[Diagrama de arquitectura 6](#_Toc335741565)

[Wireframes 7](#_Toc335741566)

[Uso de CakePHP 9](#_Toc335741567)

[Manejo de modelos 9](#_Toc335741568)

[Manejo de controladores 9](#_Toc335741569)

[Manejo de vistas 10](#_Toc335741570)

[Manejo de rutas 10](#_Toc335741571)

[Archivos de configuración 10](#_Toc335741572)

[Análisis de resultado 12](#_Toc335741573)

[Manual de usuario 13](#_Toc335741574)

[Configuración 13](#_Toc335741575)

[Instalación 13](#_Toc335741576)

[Uso 13](#_Toc335741577)

[Conclusiones 14](#_Toc335741578)

[Evaluación 15](#_Toc335741579)

# Descripción del problema

Se tiene la necesidad de lograr gestionar cursos universitarios, las asignaciones, archivos, noticias y usuarios relacionados con estos. Es decir, se requiere manejar la creación de usuarios, creación de cursos, la "matricula" de estudiantes, subir archivos al espacio de un curso, agregar asignaciones al curso para que los estudiantes las realicen.

Este problema es algo común en centros académicos en los cuales ya no es factible el uso de registros análogos, sino que se debe buscar una opción digital, que brinde un mejor rendimiento, integración y hasta un incremento en la seguridad.

Como solución se propone la programación de una aplicación en PHP, utilizando el framework de desarrollo CakePHP. En esta aplicación se cubrirán los requerimientos indicados por el cliente, brindando así una plataforma tecnológica para el uso en las actividades relacionadas con el manejo de usuarios de una entidad académica y los cursos con todos sus elementos, i.e. asignaciones, archivos, noticias.

# Diseño del programa

## Decisiones de diseño

Como se mencionó anteriormente, se utiliza el framework CakePHP, por lo que se sigue un patrón arquitectónico de Model View Controller. Esto principalmente para aprovechar la agilidad de desarrollo que brinda el framework. Además por la familiaridad con el patrón, lo cual permite un mejor desarrollo con mayor calidad.

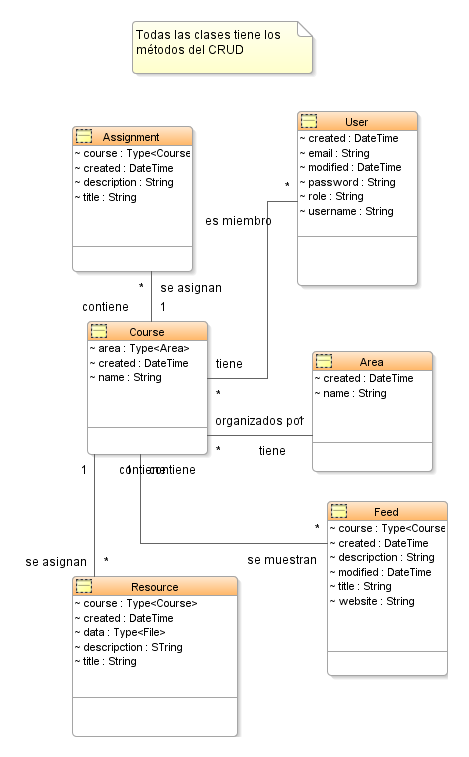
Se utilizará MySQL como base de datos, Apache como servidor web y Ubuntu como sistema operativo para el desarrollo.

CakePHP, tiene una serie de convenciones para el nombramiento de las tablas de la base de datos, los controladores, los modelos y las vistas. Todo esto para lograr ligar las diferentes tiers entre sí. Se eligió seguir estas convenciones para aprovecharse de todas las ventajas que provee el framework y facilitar el desarrollo de la aplicación, además de brindar un estándar en el nombramiento de clases, variables, vistas y en la base de datos. Y es más, ciertos nombres tuvieron que ser cambiados para que se lograra la función requerida.

Para lograr que ciertas autorizaciones fueran efectivas se tuvo que crear vistas similares pero con información diferente, dependiendo del rol que tuviera el usuario que estuviera haciendo la solicitud.

Para esta versión beta, se mantuvo una interfaz de usuario que fuera sencilla, usable y que proveyera de lo necesario.

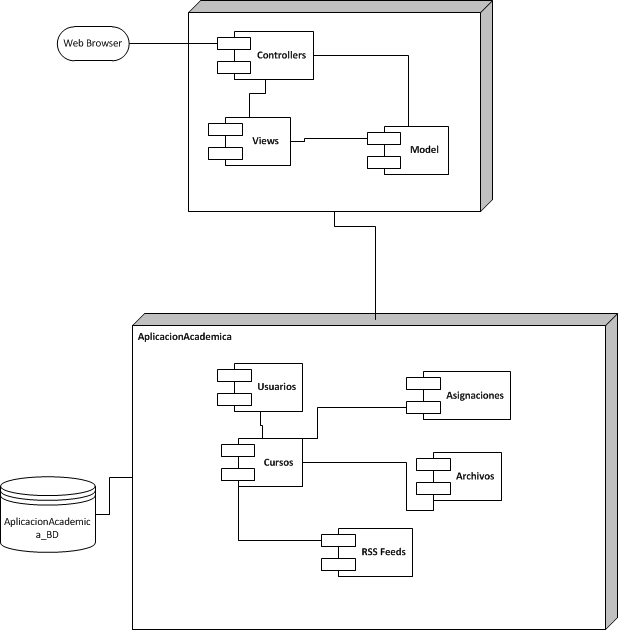
## Diagrama de clases



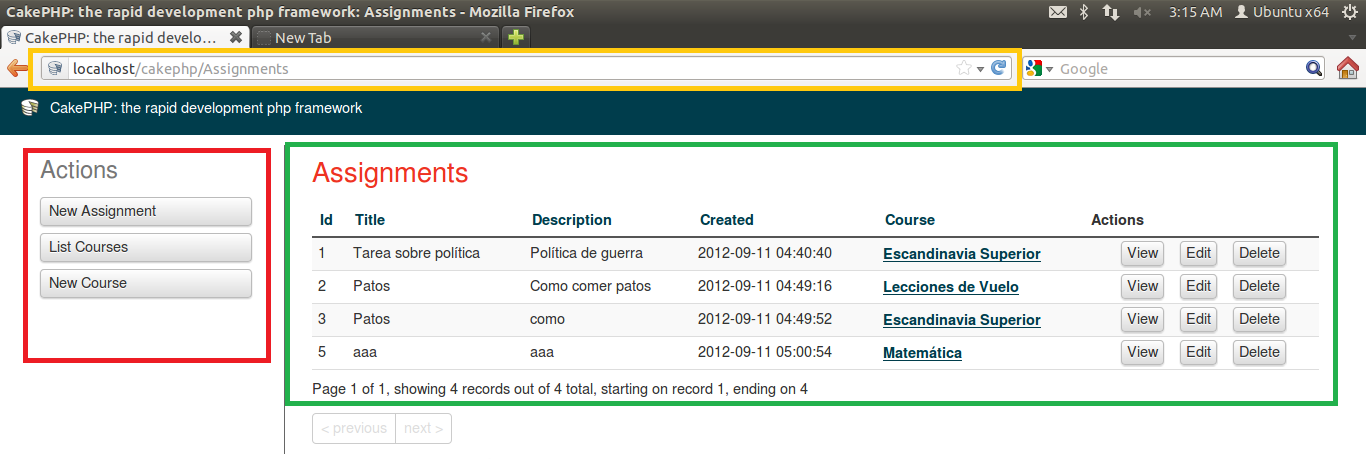
## Database_Diagram_AA.pngDiagrama de ER de la base de datos

# Diagrama de arquitectura

Se detallan los componentes principales y sus relaciones:



# Wireframes

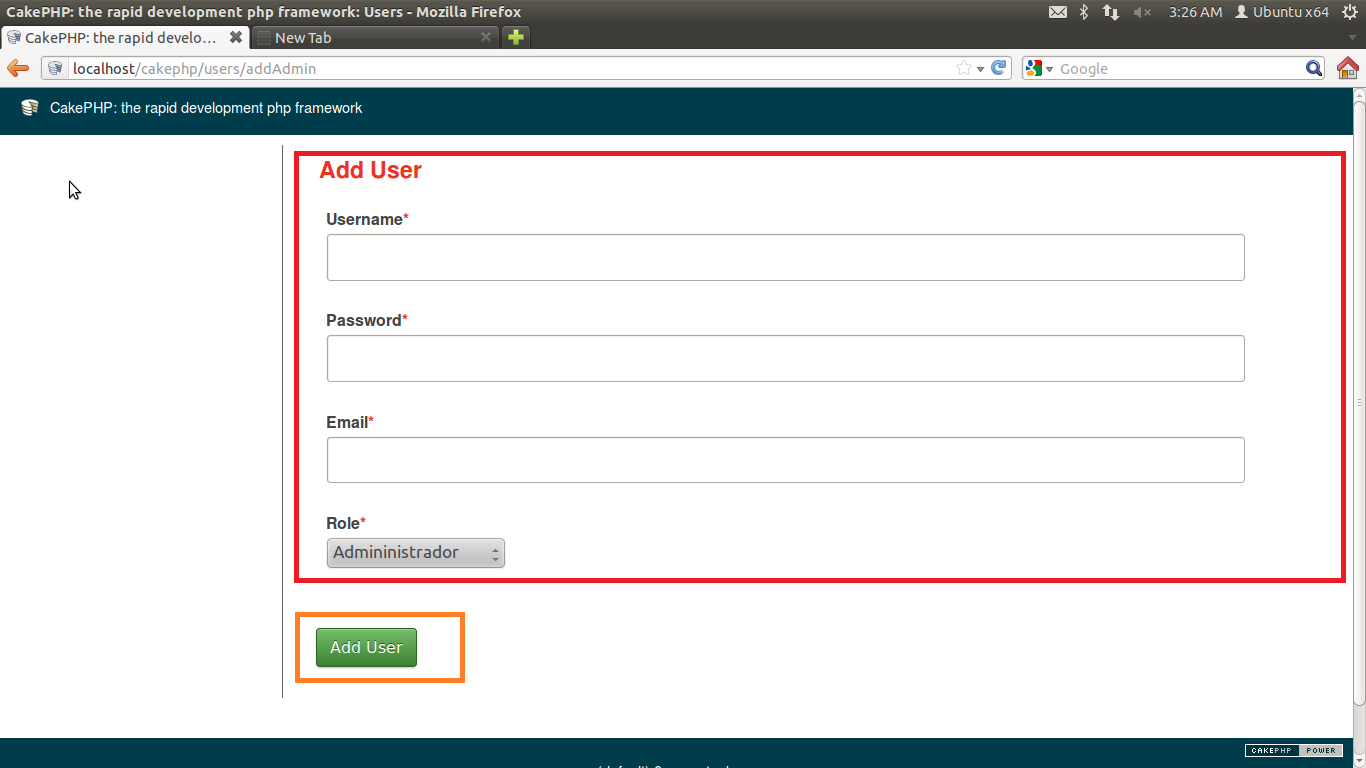
Muestra de la interfaz general de aplicación. En específico del controlador Assignments en la función index()

El cuadro amarillo es la barra de direcciones, parte básica para la navegabilidad de la aplicación. En ella es donde se pueden escribir las distintas direcciones, se escriben los nombres de los controladores, las acciones y los parámetros del request de tipo get.

El cuadro verde es donde se despliega el contenido principal de la aplicación. Que por lo general son resultados traídos desde la base de datos y algunos hipervínculos para realizar acciones sobre cada tupla. O también puede ser algún formulario para el ingreso de datos.

El cuadro es un menú de navegabilidad que se provee para desplazarse dentro de la aplicación con hipervínculos que llevan hacia distintos controladores con respectivas acciones.

Esta es la vista de AddAdmin para que el usuario administrador pueda agregar un usuario.



El cuadro rojo es donde se despliega el contenido principal de la aplicación, en este caso un formulario para ingreso de información de un nuevo usuario.

El cuadro naranja es el botón de submit del formulario.

# Uso de CakePHP

## Manejo de modelos

Se crearon modelos basándose en las tablas de la base de datos. Esto siguiendo las convenciones que tiene CakePHP. Los modelos tienen los mismos nombres de las tablas, pero en singular. e.g. en la base de datos: users, en el modelo: user.

En estos se contienen las validaciones para asegurarse que los datos que se quieran insertar en algún modelo sean correctas con relación al tipo de datos, estructura, tamaño, etcétera. e.g. que un campo no pueda ser vacío, que no sobrepase una cantidad de caracteres específica, que tenga cierta estructura como un correo electrónico.

Modelos utilizados en la solución:

* User.php
* Course.php
* CoursesUser.php
* Assignment.php
* Resourse.php
* Feed.php
* Area.php

Todos estos están en una carpeta llamada *Model*

## Manejo de controladores

Los controladores que son los que, como su nombre dice, controlan el flujo de la aplicación. Reciben los *requests* del web browser y aplican ciertas operaciones y los redirigen hacia la capa de acceso a datos, el modelo, o la vista.

Para lo que es CakePHP la convención es usar el nombre del modelo en plural y agregar la palabra "Controller". e.g. User*sController* . Por lo general se crea un controlador por cada modelo, para manejar los requests que se hagan que lo involucren.

Los controladores utilizados:

* UsersController.php
* CoursesController.php
* CoursesUserController.php
* AssignmentsController.php
* ResoursesController.php
* FeedsController.php
* AreasController.php

Estos están almacenados en una carpeta llamada *Controller*.

## Manejo de vistas

Las vistas son de formato .ctp que es el template de cakephp. En ellas se mezcla HTML y PHP para lograr mostrar el contenido que les brindan los controladores o para lograr ciertas validaciones.

Se crearon vistas CRUD para todos los modelos, excepto para CourseUser. Adicional a esto, se crearon las vistas: add\_admin.ctp, para tener una vista administrativa para añadir usuarios y edit\_normal.ctp, para lograr editar un usuario sin ser administrador.

Todas están almacenadas en View/*nombreDeModelo*

## Manejo de rutas

Para el ruteo se utilizó lo que brinda CakePHP por defecto, por ejemplo:

www.aplicacion.com/NombreControlador/Accion/Parametro1/Parametro2...

Entonces, simplemente se tuvo que apegar a las convenciones para que CakePHP se encargara del resto.

## Archivos de configuración

Se tuvo que crear dos archivos de configuración en la carpeta *Config*:

* database.php: en el cual se especifican los parámetros necesarios para conectarse al servidor donde se encuentra la base de datos. Por ejemplo, el nombre del servidor, el nombre de la base de datos, el usuario y la contraseña.
* email.php: en este se especifica la configuración para enviar correos. Colocando un protocolo para el envío, un puerto, un host para el envío de correos, un username y una contraseña.

# Análisis de resultado

|  |  |
| --- | --- |
| **Aspecto** | **Estado** |
| Gestión de Usuarios | Implementado con éxito |
| Autorización y Autenticación | Implementado con éxito |
| Gestión de Cursos | Implementado con éxito |
| Gestión de Áreas | Implementado con éxito |
| Gestión de Recursos de Cursos | Implementado con éxito |
| Gestión de RSS Feeds y lector de RSS | Implementado con éxito |
| Envío de correo a la hora de crear usuario | Implementado con éxito |
| Envío de correo a la hora de crear una asignación | No implementado por falta de tiempo |
| Envío de SMS | No implementado. |

El envío a correos a todos los usuarios de un curso al crear la asignación es una funcionalidad que se tiene pensada y se tiene una idea clara de cómo desarrollarla. Por cuestiones de tiempo no fue implementada, sin embargo sí se puede enviar un correo a una persona, por lo cual es aplicar este concepto pero iterando por los usuarios de un curso específico.

El envío de SMS no fue implementado por falta de tiempo e investigación. No se ha podido adentrarse en el tema, pero la opción principal era el uso de un API de alguna aplicación web que envíe SMS de manera gratuita.

# Manual de usuario

## Configuración

Para lograr hacer uso de esta aplicación, se necesitan ciertos aspectos:

* Servidor Web que soporte PHP, recomendación: Apache.
* Instalar y configurar PHP y MySQL en el servidor web.

Se recomienda hacer uso de una herramienta como LAMP. O subir el código fuente a un servidor que ya esté configurado y con todos los componentes previamente mencionados.

Para instalar un servidor LAMP en Ubuntu y PHPMyAdmin (para proveer una interfaz para trabajar con la base de datos que se instala) se deben seguir los pasos denotados en el siguiente URL: <http://community.linuxmint.com/tutorial/view/486>

## Instalación

Luego de hacer la configuración, se debe tomar la carpeta llamada cakephp y pegarla en el directorio *root* del servidor Apache (/var/www, por omisión).

## Uso

Para empezar a usar la aplicación se debe abrir un navegador de internet y escribir en la barra de direcciones: http://localhost/home. Esto los llevará a la página principal de la aplicación. Se debe crear un usuario para poder tener acceso a las funcionalidades de la aplicación. Al crearse se tendrá el rol de *estudiante*. Si se desea cambiar de rol, se debe contactar a algún administrador del sistema.

Para generar un URL y acceder al index de un controlador:

localhost/nombrecontrolador/

ejemplo para acceder al index de coursesController:

localhost/users

# Conclusiones

* La incursión en el uso de un nuevo framework, siempre representa una curva de aprendizaje exponencial. Sin embargo, al comparar CakePHP con otros frameworks (ASP MVC), se nota que CakePHP es mucho más sencillo de utilizar, mucho más intuitivo, haciendo que la curva de aprendizaje no se salga de las manos.
* El uso de un framework como CakePHP facilita mucho el desarrollo de una aplicación. Abstrae muchas cosas, lo que hace que se vea muy sencillo. Es confuso al principio justamente por eso, porque uno espera tener que escribir más líneas de código para realizar cosas que el framework hace "por debajo".
* También, el uso de un framework permite tener estandarización hasta cierto punto, por las convenciones sobre nombres, la estructura de la aplicación y hasta la estructura de las carpetas en las que se contiene el código fuente.
* Las aplicaciones web tienen una gran ventaja ya que no están ligadas a una plataforma específica. Por lo que se puede hacer uso de la aplicación con sólo tener un web browser (suponiendo que la aplicación está *hosteada* correctamente.
* Conocer sobre nuevos lenguajes de programación como PHP es algo muy bueno para el desarrollo del espectro de pensamiento abstracto. Aparte de ser un lenguaje muy utilizado en la actualidad por todas las aplicaciones web que están surgiendo.
* Hay que tener claro qué versiones de tecnologías se están usando ya que los comandos y funciones se van depreciando y paran de ser usadas, por lo que se puede estar revisando una fuente desactualizada y probando funciones que ya no existen en el lenguaje o tecnología.
* La configuración de software en Ubuntu es algo complejo y toma tiempo. Por lo que es una buena práctica realizarlo en adelantado para que no influya en el tiempo del proyecto. O bien, utilizar soluciones como LAMP que eliminan ciertas dificultades de la configuración.

# E5ttvaluación

|  |  |  |
| --- | --- | --- |
| Rubros | Puntos | Puntos Obtenidos |
| Documentación interna | 2% |  |
| Documentación externa | 18% |  |
| Manejo de correos | 5% |  |
| Manejo usuarios | 15% |  |
| Manejo cursos | 25% |  |
| Manejo noticias con rss | 5% |  |
| Revisión de tarea | 30 % |  |
| Total |  |  |