

PROYECTO DE SOFTWARE 2014  
TRABAJO DE PROMOCIÓN  
FRAMEWORK PHP – CAKEPHP  
TIBALDO JUAN AGUSTÍN

Para el trabajo de promoción utilicé el framework CakePHP. El trabajo utilizando dicho framework fue bastante simple y fluido en general. Probé las versiones 2.6.x y 3.0 que salió el 22 de febrero. El trabajo final lo hice sobre 2.6.3. Tuve una gran pérdida de tiempo al implementar el modelo de autorización, siendo que al principio iba a implementar un sistema Acl, pero después de días decidí trabajar con un modelo más simple.

Cabe señalar que el trabajo lo realicé solo, dado que mi grupo no aprobó la cursada.

A continuación se detallan los puntos solicitados por la cátedra

## Fundamentación sobre la elección del framework

### Elegido:

*El framework me lo había recomendado un compañero el año pasado, me había comentado que era muy sencillo de utilizar y que disponía de una consola muy potente que hacía gran parte del trabajo. Al mirar varios frameworks (phalcon, yii, symphony php) y leer que era lo que opinaba la gente me decidí por utilizar CakePHP.*

## Las referencias de donde sacaron los materiales (libros, tutoriales, cursos, etc) :

*Básicamente me guíe por el cookbook : <http://book.cakephp.org/2.0/en/index.html> el cual es la documentación oficial y está bastante completo (aunque en algunas cosas están pobremente documentadas como ser Acl).*

*También me guíe por ejemplos dentro de la página oficial de cakephp.*

*Youtube : <https://www.youtube.com/watch?v=zvwQGZ1BxdM> para login y autorización.*

## -La descripción de los módulos desarrollados en el trabajo de la cursada que pudieron ser aprovechados para ser usados por el framework:

*los módulos que reutilicé (con modificaciones) fueron : consultas para los reportes y vistas de los reportes.*

*Las consultas las tuve que pasar de PDO al sistema de consultas SQL de cakephp (poco esfuerzo de no haber cambiado el nombre de las tablas en la DB para cumplir con el estándar de nombres de cake.).*

*Las vistas tuvieron que ser cambiadas dado que las anteriores estaban construidas utilizando plantillas twig y cakephp utiliza plantillas ctp (básicamente html con php embebido).*

## El mecanismo provisto para el manejo de seguridad y routing (si es que lo tiene):

**Manejo de seguridad:** *Consultas preparadas, formularios autogenerados, filtros de métodos HTTP, autorización centralizada con posibilidad de deshabilitar todo salvo lo explícitamente habilitado para determinados usuarios, validaciones en el modelo (estos fueron los que usé)*

**Routing:** se puede redefinir el root y varias otras rutas en básicas en Config/routes.

En el resto de la aplicación se utiliza el patrón controlador/acción para direccionarse. Toda acción o petición o vista siempre está en el ámbito de un controlador. Dichos controladores administran la aplicación íntegramente.

Además se puede definir ciertas redirecciones basadas en el estado de la sesión. Por ejemplo a donde redireccionar ante login o logouts, fallos de autorización.

## El mecanismo provisto para operaciones CRUD (si es que lo tiene):

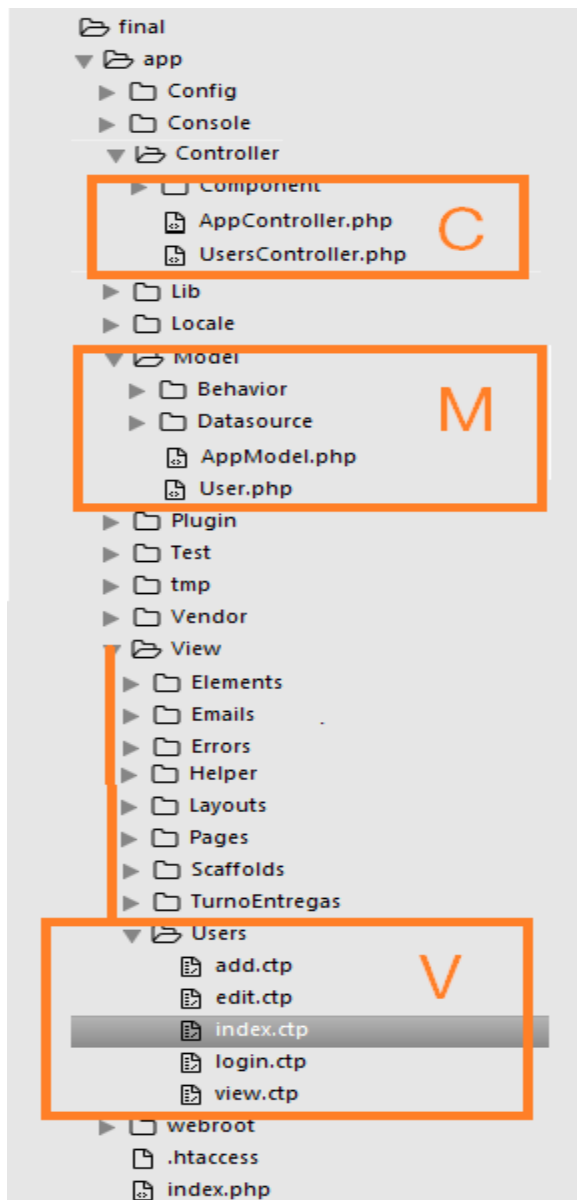
La generación de la aplicación que permite realizar operaciones CRUD se hace a través de la consola mediante interacción de interfaz de comandos con el usuario. Cake bake detecta nuestro modelo a partir de la base de datos una vez que ésta ha sido configurada. Luego se puede construir todo el sistema MVC a partir de estos de forma automática o interviniendo en dicha creación.

Siempre que se sigan las convenciones, además detecta de manera automática las relaciones entre tablas, campo descriptor de la tabla, etc.

Luego todo el sistema de operaciones CRUD está seteado. Aun así posee métodos delete, save, list, find, query para aumentar los controladores.

## La forma de manejar el MVC: explicando detalladamente el árbol de directorios

La aplicación es generada a partir de la consola de cake. MVC está manejado a partir de un árbol de directorios como sigue:



Los directorios que intervienen al MVC son los siguientes:

*App/Controller/Component* : son los controladores de la aplicación, la convención es camelCase del nombre de la tabla de la base de datos(si la hay) seguida por Controller.php. Además se dispone de la AppController.php, superclase que permite setear comportamiento general de todos los controladores.

*App/Model/Datasource/*: son los modelos de la aplicación, representan un modelo siendo la convención utilizar el nombre del objeto en singular y camelCase. AppModel es un modelo general que permite dar comportamiento general para todos los modelos.

*App/View/*: acá están todas las carpetas que poseen las vistas para un controlador específico. Dentro de estas carpetas se pueden poner las vistas que corresponden a los métodos que maneja un controlador, aunque esto no siempre se dará de esta forma. Por ejemplo, nuestro UsersController posee el método logout que no se corresponde a ninguna vista (ya que redirecciona al root).

La comunicación entre los tres está manejada automáticamente siguiendo ciertos métodos que nos da Cake para esto.

Por ejemplo el controlador usa `$this->set('mi_variable',$mi_variable)` para asignar

`$mi_variable` del controlador a `$mi_variable` de la vista.

O `$this->User->método` para invocar métodos del modelo desde el controlador.

Además cada método del controlador es asignado a una vista de forma automática, dicha vista tiene el mismo nombre que el método, como ser public function edit con la vista edit.ctp.