Universidad Autónoma Metropolitana – Azcapotzalco
División de Ciencias Básicas e Ingeniería
Ingeniería en Computación

Proyecto Terminal
Aplicación web de administración de horarios para estudiantes

Reporte final

Alberto Manuel Velázquez Canales
Matrícula: 204206058

Asesora: Dra. Beatriz Adriana González Beltrán
Número económico: 30246

Índice de contenido

Introducción

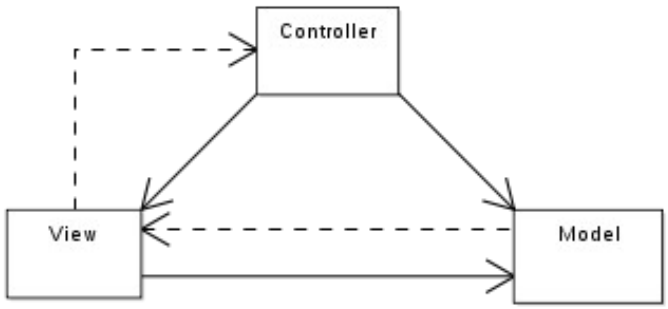
El objetivo general de este Proyecto Terminal fue desarrollar una aplicación web que programara las actividades de sus usuarios. Se pensó en los estudiantes de cualquier nivel como los usuarios objetivo de ésta.

La propuesta fue crear una aplicación en la que los usuarios pudieran especificar los períodos o ciclos escolares en los que estaban inscritos (por ejemplo, un usuario podría tener un par de períodos: uno para el trimestre que estuviera cursando en la Universidad y otro para las clases de alguna lengua extranjera que estuviera aprendiendo), las materias correspondientes a cada uno de los períodos y las actividades asignadas a cada materia. Basándose en ciertos criterios como tipo de actividad, la prioridad de la misma, la dificultad de la materia correspondiente, etc., la aplicación programaría de manera automática las actividades del usuario.

La aplicación contaría además con un módulo que enviaría por correo electrónico un mensaje de alarma cuando la fecha de fin de cada actividad se aproximara; además de un módulo que permitiría exportar las actividades como eventos en formato iCalendar.

Desarrollo del proyecto

Al tratarse de una aplicación web, se decidió que el desarrollo debería seguir el patrón *Model-View-Controller* (MVC), que establece una clara separación entre el código necesario para representar los modelos, las vistas y los controladores de la aplicación. Esto es ligeramente diferente a la arquitectura en tres capas que fue propuesta inicialmente.



Para tal efecto, se utilizó el *framework* CakePHP, el cual permite la creación de aplicaciones MVC para la web. La decisión de utilizar este *framework* se basó en el hecho de que en la propuesta de proyecto se había elegido el lenguaje PHP para desarrollar la aplicación. De manera consistente con la propuesta, la aplicación además hace uso del servidor de bases de datos MySQL y del servidor web Apache.

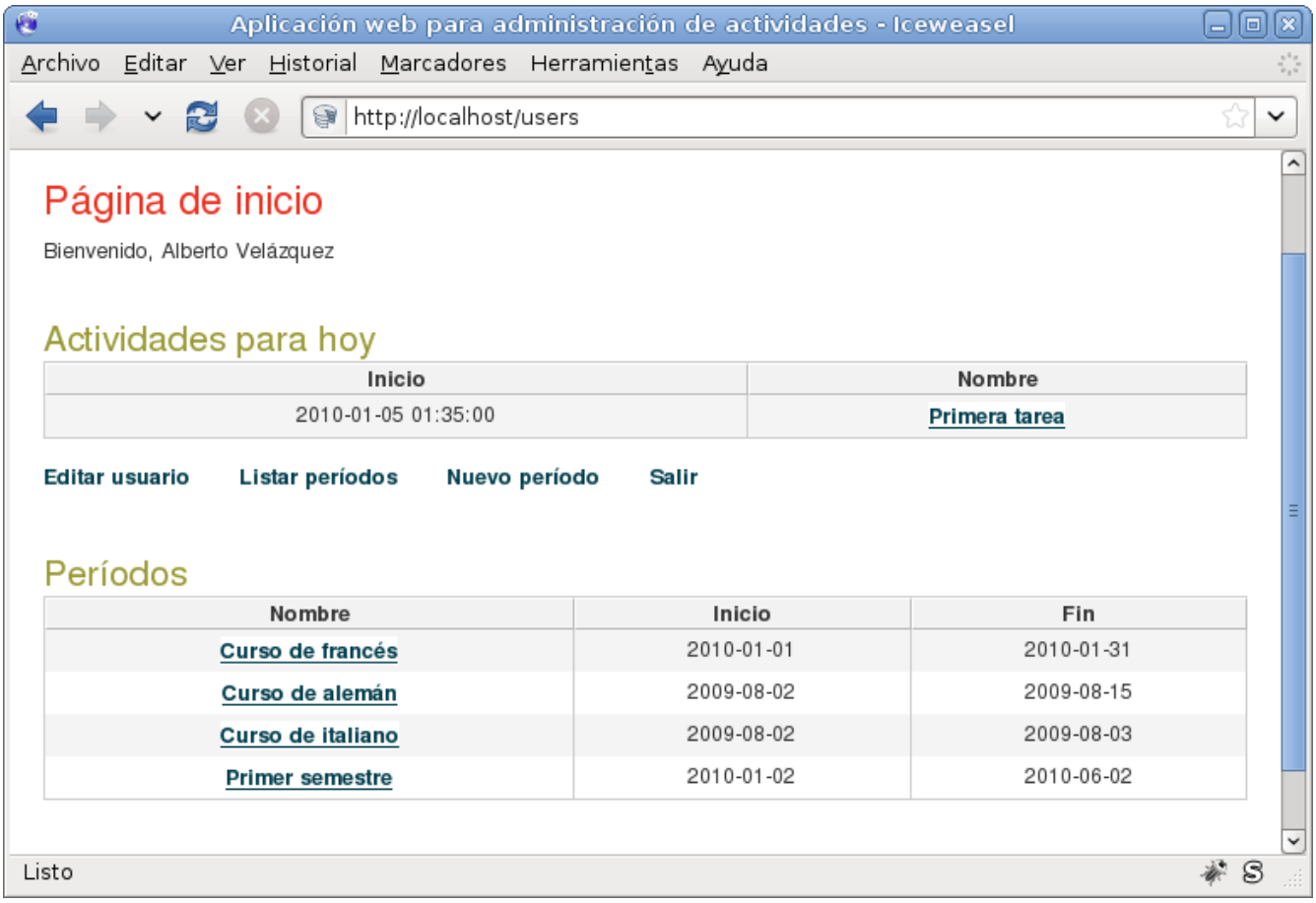
En CakePHP, los *modelos* se definen a partir de las tablas en la base de datos, los campos especificados en éstas y las relaciones con otras tablas. Los *controladores* son clases en las que se programa la lógica de la aplicación. Las *vistas* son plantillas que pueden leer datos enviados desde el controlador correspondiente y se encargan solamente de mostrarlos al usuario.

Durante el proyecto terminal no se desarrolló la totalidad de lo contenido en la propuesta. En particular, faltó desarrollar la programación automática de las actividades y los módulos de exportación a iCalendar. Sin embargo, la aplicación es capaz de manejar de manera robusta los períodos, las materias y las actividades, permite la programación manual de estas últimas y es multiusuario. Esto conforma una base sólida para extender la aplicación en el futuro.

Interfaz de usuario

La interfaz de la aplicación con el usuario es muy intuitiva y fácil de utilizar. Está conformada, básicamente, por una serie de páginas que permiten la administración de la cuenta del usuario, los períodos, las materias y las actividades.

La pantalla de inicio de la aplicación, una vez que el usuario ha ingresado a ésta, muestra una lista de actividades programadas para el día actual, además de una lista de los períodos registrados por el usuario.



A través de la página principal es posible acceder a una lista de los períodos definidos por el usuario, definir un período nuevo y ver, editar o borrar períodos. Para la creación de períodos se presentan formularios en los cuáles el usuario deberá introducir la información correspondiente. La aplicación muestra cuadros de diálogo (mediante JavaScript) que piden la confirmación del usuario al borrar cualquier período.

La aplicación muestra una serie de páginas equivalentes para la administración de las materias y actividades.

La aplicación cuenta con una validación robusta de los datos que se reciben por parte del usuario y, de ser el caso, muestra una serie de leyendas que informan sobre los errores cometidos al introducir la información. Un detalle importante de usabilidad es que la información introducida en los formularios vuelve a presentarse al usuario aunque no se hayan pasado todas las reglas de validación, de tal manera que sólo debe concentrarse en corregir el error cometido.

La explicación completa de la funcionalidad de la aplicación se encuentra en el manual de usuario.

Estructura de la base de datos

La base de datos de la aplicación está conformada por las siguientes tablas:

activities

Campo	Tipo	Comentario
`id`	`int`	Llave primaria
`title`	`varchar`	Título de la actividad
`priority`	`int`	Prioridad de la actividad
`type_id`	`int`	Llave foránea hacia la tabla `types`
`alarm`	`tinyint`	Booleano que indica si se desea registrar una alarma para la actividad
`duration_min`	`time`	Duración mínima de la actividad
`duration_max`	`time`	Duración máxima de la actividad
`start`	`datetime`	Fecha de inicio
`finish`	`datetime`	Fecha de fin
`course_id`	`int`	Llave foránea hacia la tabla `courses`
`sticky`	`tinyint`	Booleano que indica que la actividad se ha programado manualmente

courses

Campo	Tipo	Comentario
`id`	`int`	Llave primaria
`name`	`varchar`	Nombre de la materia
`credits`	`int`	Número de créditos
`location`	`varchar`	Lugar donde se imparte
`teacher`	`varchar`	Profesor que la imparte

difficulty	int	Dificultad de la materia
term_id	int	Llave foránea hacia la tabla terms

terms

Campo	Tipo	Comentario
id	int	Llave primaria
name	varchar	Nombre del período
start	date	Fecha de inicio
finish	date	Fecha de fin
user_id	int	Llave foránea hacia la tabla users

types

Campo	Tipo	Comentario
id	int	Llave primaria
title	varchar	Título
multiplier	int	Factor de multiplicación

users

Campo	Tipo	Comentario
id	int	Llave primaria
username	varchar	Nombre de usuario
password	varchar	Contraseña
email	varchar	Dirección de e-mail
name	varchar	Nombre

Además, se tienen tres tablas (acos, aros y aros_acos) que son utilizadas para guardar los permisos que cada usuario (*Access Request Object*) tiene sobre cada período, materia y actividad (*Access Control Objects*).

Como se puede apreciar, los campos de las tablas contemplan los criterios necesarios para realizar la programación automática (factor de multiplicación por tipo de actividad, dificultad de la materia, prioridad de la actividad, etc.), aunque ésta no se haya implementado.

Estructura de la aplicación

Al tratarse de una aplicación MVC, los archivos de código fuente de la aplicación se encuentran divididos en tres directorios principales.

- app/models/: Contiene los archivos en los que se implementan las clases que representan los modelos de la aplicación.

    - `activity.php`: Implementa la clase *Activity*, representa las actividades.
    - `course.php`: Implementa la clase *Course*, representa las materias.
    - `term.php`: Implementa la clase *Term*, representa los períodos.
    - `type.php`: Implementa la clase *Type*, representa los tipos de actividades.
    - `user.php`: Implementa la clase *User*, representa los usuarios de la aplicación.
- `app/controllers/`: Contiene los archivos en los que se implementan las clases que contiene la lógica de la aplicación.
    - `activities_controller.php`: Controlador para las actividades. Implementa la clase *ActivitiesController*.
    - `courses_controller.php`: Controlador para los cursos. Implementa la clase *CoursesController*.
    - `terms_controller.php`: Controlador para los períodos. Implementa la clase *TermsController*.
    - `users_controller.php`: Controlador para los usuarios de la aplicación. Implementa la clase *UsersController*.
- `app/views/`: Contiene las plantillas que conforman la interfaz de usuario de la aplicación: formularios, listas, etc.
    - `activities/`: Contiene las vistas de las actividades.
    - `courses/`: Contiene las vistas de las materias.
    - `terms/`: Contiene las vistas de los períodos.
    - `users/`: Contiene las vistas de los usuarios.

Información más detallada de la organización y contenido del código fuente de la aplicación puede encontrarse en el manual del programador.

Lista de entregables

La lista de documentos y archivos entregados al finalizar el Proyecto Terminal es la siguiente:

- Un manual de usuario.
- Un manual de instalación.
- Un manual del programador.
- Código fuente de la aplicación.

Los manuales fueron entregados en formato PDF. El código fuente se encuentra en un archivo TAR comprimido con Gzip.

Conclusión

Durante este Proyecto Terminal se desarrolló una aplicación web que sirve para ayudar a los estudiantes de todos los niveles a programar las actividades relacionadas con sus estudios. En ese sentido, se cubrió de manera general el objetivo presentado en la propuesta.

Sin embargo, al hacer un análisis más detallado de lo logrado, debe mencionarse que la programación automática de las actividades, una de las características más importantes de la aplicación propuesta, no fue desarrollada. En vez de eso, recae en el usuario de la aplicación la responsabilidad de programar las actividades por si mismo.

Aún así, el trabajo realizado en este Proyecto Terminal representa una base sólida que puede extenderse en el futuro. En particular, se ha visto que los modelos que conforman la aplicación, y su representación física en la base de datos utilizada, toman en cuenta la información necesaria para implementar algún algoritmo de programación automática.

Referencias

- Internet Engineering Task Force
  Internet Calendaring and Scheduling Core Object Specification (iCalendar)
  http://tools.ietf.org/html/rfc5545
- Wikipedia
  Model-View-Controller
  http://en.wikipedia.org/wiki/Model-view-controller
- Cake Software Foundation, Inc.
  CakePHP: the rapid development php framework
  http://cakephp.org/