

**Universidad Autónoma Metropolitana
Unidad Azcapotzalco**

División de Ciencias Básicas e Ingeniería

Ingeniería en Computación

Proyecto Terminal:

“Aplicación para el sistema moodle que actualice las notas de un alumno con los cambios de un archivo del profesor”

Alumno:

*Vázquez Cabal Rodrigo Antonio
206306692*

Trimestre 11-I

Asesora del proyecto:

- *Dra. Silvia Beatriz González Brambila.*
 - *No. Económico 17204.*
 - *Departamento de Sistemas.*

Índice

Índice de imágenes	4
Índice de Tablas	6
Capítulo 1. Descripción del proyecto	7
1.1. Estructura del CD:.....	7
Capítulo 2. ¿Qué es Moodle? Funciones y Servicios	8
2.1. Introducción	8
2.1.2. Especificaciones del diseño de la base de datos.	9
2.2. Antecedentes	9
2.2.1. La plataforma Moodle.....	9
Capítulo 3. Propuesta de Solución	12
3.1. Justificación	12
3.2. Objetivos	16
3.2.1. Objetivo general.	16
3.2.2. Objetivos particulares.	16
Capítulo 4. Metodología de Desarrollo	17
4.1. Descripción técnica	17
4.2. Especificaciones técnicas	18
4.3. Recursos básicos para el desarrollo	20
Capítulo 5. Desarrollo	21
5.1. Arquitectura de la aplicación propuesta.....	21
5.2. Metodología de Análisis, Diseño e Implementación.....	22
5.3. Módulo de implementación.....	23
Capítulo 6. Herramientas de Comparación y Mezcla de Archivos xml	24
6.1. Comparación de las herramientas	29
Capítulo 7. Instalación de Moodle	30
7.1. Instalación de MySQL.....	30
7.2. Instalación de Apache	30
7.3. Instalación de PHP.....	31
7.3.1. Obtener soporte de MySQL para PHP.....	32

7.4. Instalación de phpMyAdmin	33
7.5. Instalación de Moodle 2.0.....	34
Capítulo 8. Modificación del Código	35
8.1. Diseño de la Base de Datos	35
8.1.1. Carga de archivos nuevos.....	35
8.1.2. Estructura de la Base de Datos.....	38
8.1.3. Diccionario de Datos	38
8.2. Implementación al código.....	39
Capítulo 9. Resultados	46
Conclusiones y crecimiento del proyecto	51
Glosario de Términos	52
Referencias Electrónicas y Bibliográficas.....	53

Índice de imágenes

- Figura 1. Recursos de Moodle. Pág. 4.
- Figura 2. Archivo original del profesor. Pág. 5.
- Figura 3. Modificaciones del profesor. Pág. 7.
- Figura 4. Notas del alumno. Pág. 8.
- Figura 5. Actualizar. Pág. 10.
- Figura 6. Menú actualizar Pág. 11.
- Figura 7. Generación del archivo final. Pág. 12.
- Figura 8. Arquitectura Moodle. Pág. 14.
- Figura 9. Diagrama de actividades del módulo. Pág. 16.
- Figura 10. Comparación de dos archivos XML con Oxygen XML Diff. Pág. 17.
- Figura 11. Comparación KDiff3. Pág. 17.
- Figura 12. Comparación de dos archivos XML con DiffDog. Pág. 18.
- Figura 13. Comparación de dos archivos XML Diff & Merge. Pág. 19.
- Figura 14. Funcionamiento de Apache. Pág. 23.
- Figura 15. Información de PHP. Pág. 24.
- Figura 16. Paquete de MySQL para PHP. Pág. 24.
- Figura 17. Interfaz phpMyAdmin. Pág. 25.
- Figura 18. Acceso a la interfaz de Actualizar. Pág. 31.
- Figura 19. Botón agregar. Pág. 32.
- Figura 20. BD mdl_files con el nuevo archivo. Pág. 33.
- Figura 21. BD Notificación de Actualización de Archivo. Pág. 35.
- Figura 22. Examinar archivo del alumno. Pág. 35.
- Figura 23. Botón Mezclar del Alumno. Pág. 36.
- Figura 24. Pop Actualización de Notas del Alumno. Pág. 36.

Figura 25. Descargar Archivo. Pág. 36.

Figura 26. Elegir ruta de descarga. Pág. 37.

Figura 27. Archivo viejo del profesor. Pág. 38.

Figura 28. Archivo nuevo del profesor. Pág. 39.

Figura 29. Archivo actualizado del profesor. Pág. 40.

Figura 30. Notas del alumno. Pág. 41.

Figura 31. Archivo actualizado del Alumno. Pág. 24.

Índice de Tablas

Tabla 1. Comparación de herramientas. Pág. 21.

Tabla 2. mdl_files de la BD de Moodle. Pág. 29.

Tabla 3. Diccionario de datos de los archivos de Moodle. Pág. 30.

Tabla 4. BD mdl_files con el nuevo archivo. Pág. 32.

Capítulo 1

Descripción del proyecto

En este capítulo se presenta una descripción general de la aplicación desarrollada para el sistema Moodle que permite actualizar las notas de un alumno con los cambios de un archivo del profesor, así como la estructura del CD con toda la documentación.

En el capítulo 1 se tiene lo que es la introducción y los antecedentes donde se describe lo que es Moodle.

En el capítulo 2 se analiza la problemática en la justificación, concluyendo así con los objetivos.

La estructura del proyecto que se va a realizar como las especificaciones técnicas, los recursos que se necesitaron, el diseño y la implementación se describen en los capítulos 4 y 5.

Las herramientas las cuales se investigaron, así como la comparación de estas se detallan en el capítulo 6.

Para poder realizar este proyecto fue necesario instalar el sistema Moodle, todos los pasos que se llevaron a cabo se encuentran en el capítulo 7.

En el octavo capítulo se describe como se modificó el código y la base de datos, cuyos resultados están en el capítulo 9.

Al final del documento se encuentra el glosario de términos y las referencias bibliográficas.

1.1. Estructura del CD:

En la raíz se encuentra la documentación del proyecto así como cuatro carpetas:

- Código fuente.
- Código documentado.
- Manual.
- Software utilizado.

Capítulo 2

¿Qué es Moodle? Funciones y Servicios

2.1. Introducción

En la vida estudiantil se presentan diversos tipos de aprendizaje, por mencionar uno de ellos tenemos la “educación en línea”, la cual tiene como patrón común utilizar una plataforma de “enseñanza-aprendizaje”. Cada vez más se usa más éste método en varias universidades, para cursar materias, incluso para estudiar carreras y maestrías completas en línea. Diversas universidades ocupan la plataforma Moodle para realizar este tipo de enseñanza. En particular la División de Ciencias Básicas e Ingeniería de la Unidad Azcapotzalco de la Universidad Autónoma Metropolitana ha puesto cursos y notas a disposición de los alumnos desde hace ya más de una década, implantando así lo que se conoce como b-learning, en donde se mezcla la educación a distancia y la tradicional.

Dentro del b-learning de la UAM-A cuyo nombre es “cbienlinea” estuvo el objetivo de desarrollar un espacio virtual para ayudar a desarrollar los procesos de enseñanza y aprendizaje a través del desarrollo y la implementación de un CMS¹ (course management system) para manejar los procesos de enseñanza y aprendizaje centrados en el estudiante. El CMS seleccionado fue Moodle y la mayoría de los maestros de Ciencias Básicas e ingeniería (DCBI) usan esta herramienta.

La mayoría de los cursos ofrecidos en la DCBI son a través de tradicionales salones de clases, pero algunos maestros prefieren usar la modalidad b-learning llamada Moodle, la cual es una aplicación diseñada para ayudar a los educadores a crear cursos amigables de calidad en línea.

¹ CMS (Course management system) es un conjunto de herramientas de software que de manera integral proveen un espacio en línea para facilitar el aprendizaje y la enseñanza [23].

Hoy en día se tienen alrededor de 50 cursos de b-learning siendo enseñados cada 4 meses. El CMS de “cbienlinea” está siendo accesado de 160 a 190 estudiantes diariamente por varios cursos.

Por esta razón este proyecto propone la comparación y actualización de archivos, una aplicación la cual se agregará a la plataforma Moodle, dándole al estudiante como resultado que siempre estén actualizados los archivos en los cambios que hace el profesor, así como también agilizar el trabajo de los profesores ahorrando tiempo al dar explicaciones.

2.1.2. Especificaciones del diseño de la base de datos.

Se hizo un estudio de cómo Moodle maneja las bases de datos, en su creación, almacenamiento, acceso y modificación de los elementos dentro de ella.

Con esto se realizaron pruebas sobre este tipo de manejo para los datos que se utilizaron en el desarrollo de este proyecto. Principalmente se ocupa lo que es la encriptación² para poder acceder a los diferentes tipos de archivos, con esto se implementaron algoritmos que hicieron este manejo de una manera más sencilla y así desarrollar este proyecto de actualización de archivos.

2.2. Antecedentes

2.2.1. La plataforma Moodle

Moodle es una plataforma de aprendizaje, implementada como una aplicación Web, que permite la gestión de cursos, y sirve de apoyo para los profesores y los alumnos durante el proceso de enseñanza-aprendizaje. Esta plataforma puede utilizarse para el aprendizaje a distancia (e-learning) o de manera complementaria a un curso presencial (b-learning).

El sistema Moodle está desarrollado en GNU/Linux y su uso se ha incrementado en los últimos años, ya que por sus características provee un gran número de módulos, casi igual que otras plataformas similares, siendo muy atractiva por el uso de la tecnología denominada LAMP [20], esto significa que para su instalación, codificación y uso solo requiere software libre como Linux, Apache, MySQL o Postgres y PHP, además de proveer el código fuente de la aplicación para poder adaptar los módulos[1].

² Encriptación es la codificación en la información de archivos o de un correo electrónico para que no pueda ser descifrado en caso de ser interceptado por alguien mientras esta información viaja por la red [22].

Algunas de las características más importantes de esta plataforma son:

- El código está escrito en el lenguaje de programación PHP, siendo fácil su modificación, ya que PHP tiene una curva de aprendizaje corta.
- Es posible instalarlo sin modificaciones bajo diferentes sistemas operativos, tales como Unix, Linux, Windows, Mac OS X, Netware, entre otros [2].
- Se puede personalizar para darle una imagen corporativa o de alguna institución educativa mediante el uso de hojas de estilo.

En el rol del profesor, Moodle tiene un conjunto de diferentes recursos que les permiten añadir cualquier contenido en un curso, los cuales se muestran en la Figura 1.

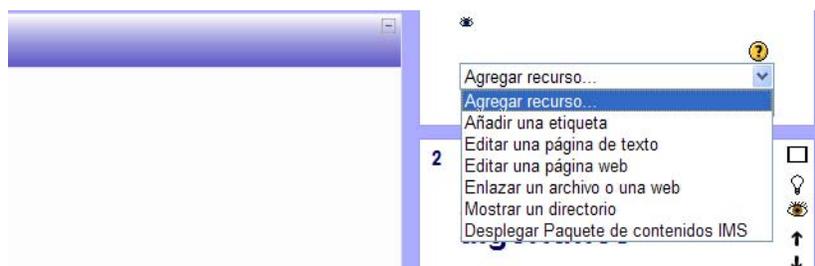


Figura 1. Recursos de Moodle.

Se distribuye gratuitamente como Software bajo la Licencia Pública GNU, esto significa que tiene derechos de autor, pero se tienen ciertas libertades como copiar, usar, modificarlo y también se puede proporcionar el código fuente a otros, aunque no se debe modificar o eliminar la licencia original y los derechos de autor, y aplicar esta misma licencia a cualquier trabajo derivado de él.

Actualmente en la Universidad Autónoma Metropolitana (UAM-A) se utiliza este sistema y uno de los módulos que tiene es el de subir las notas a la página y así el alumno puede descargar estos archivos, haciendo anotaciones sobre ellos, pero en ocasiones cuando un profesor cambia las notas originales, el alumno no logra darse cuenta de estos cambios ni en dónde fueron realizados, provocando que a este le haga falta información nueva o errores corregidos por el profesor, por lo que el alumno estará trabajando con dos o más archivos en lugar de uno solo.

Con este proyecto, se pretende crear una aplicación donde los alumnos inscritos a un curso puedan recibir notificaciones cuando haya cambios en los archivos, así como tener la opción de actualizar sus notas con las del archivo actualizado del profesor.

Existen algoritmos para la comparación de archivos de diferentes tipos, los cuales nos muestran los cambios y las diferencias entre estos.

Capítulo 3

Propuesta de Solución

3.1. Justificación

La UAM-A, así como otras instituciones educativas, para empezar a darle forma a un modelo de integración social y educativo ha decidido emplear, entre otras cosas, la plataforma Moodle la cual es usada para gestionar usuarios, recursos y actividades de formación, administrar el acceso, controlar y hacer seguimiento del proceso de aprendizaje, realizar evaluaciones, generar informes, gestionar servicios de comunicación como foros de discusión, entre otros.

Dentro de algunas universidades se utiliza el sistema moodle, para el intercambio de información entre el profesor y el alumno, por ejemplo el profesor sube las notas a la plataforma desde la cual se tiene acceso a través de internet y el alumno tiene que bajar estas notas.

Sin embargo, cuando un profesor desea modificar las notas o las tareas es complicado para el alumno ya que no tiene conocimientos de estas modificaciones ni qué o donde se realizaron los cambios.

La solución que se plantea en esta propuesta es crear una aplicación que permita al alumno enterarse de las modificaciones realizadas por el profesor y actualizar sus notas automáticamente, si así lo desea. La actualización de las notas contendrá los cambios realizados por el profesor, así como las anotaciones que posiblemente el alumno haya realizado durante la revisión del material.

Esta implementación beneficiaria en primera instancia a los alumnos de diversas universidades ya que contara con información actual que le permita conservar las anotaciones que había realizado con anterioridad en los archivos actualizados por el profesor. En segunda instancia el profesor tendrá mayor comodidad ya que podrá evitar decir exactamente en dónde ocurrieron los cambios y logrando un ahorro de tiempo y esfuerzo, que podría invertirse en partes más sustantivas de los temas propios del curso, es decir, automatizando la administración del curso.

El diseño y la implementación de este sistema requieren de conocimientos de programación, análisis y diseño de sistemas, diseño e implementación de

interfaces de usuario y manejo de algoritmos de cifrado, conocimientos que se van adquiriendo durante el transcurso de la carrera de Ingeniería en Computación en la Universidad Autónoma Metropolitana unidad Azcapotzalco.

A continuación en la Figura 2 se muestra un ejemplo de un archivo original tipo "xml" subido por el profesor, posteriormente las notificaciones de este (ver Figura 3) y las notas de un alumno en el antiguo archivo (ver Figura 4).

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.4"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <servlet>
    <servlet-name>action</servlet-name>
    <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
    <init-param>
      <param-name>action</param-name>
      <param-value>9</param-value>
    </init-param>
    <init-param>
      <param-name>config</param-name>
      <param-value>/WEB-INF/struts-config.xml</param-value>
    </init-param>
    <init-param>
      <param-name>debug</param-name>
      <param-value>3</param-value>
    </init-param>
    <init-param>
      <param-name>detail</param-name>
      <param-value>3</param-value>
    </init-param>
    <load-on-startup>0</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.do</url-pattern>
  </servlet-mapping>

  <context-param>
    <param-name>driver</param-name>
    <param-value>com.mysql.jdbc.Driver</param-value>
  </context-param>
  <context-param>
    <param-name>cadenaCon</param-name>
    <param-value>jdbc:mysql://localhost/practica03</param-value>
  </context-param>
</web-app>
```

Figura 2. Archivo original del profesor.

En los recuadros negros marcados en la Figura 3 se muestran las modificaciones del profesor al archivo original (ver Figura 2).

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.4"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <servlet>
    <servlet-name>Accion</servlet-name>
    <servlet-class>org.apache.struts.action.DataBase.ActionServlet</servlet-class>
    <init-param>
      <param-name>action</param-name>
      <param-value>9</param-value>
    </init-param>
    <init-param>
      <param-name>config</param-name>
      <param-value>WEB-INF/struts-config.xml</param-value>
    </init-param>
    <init-param>
      <form-beans>
        <form-bean name="ProcesaSeleccionForm" type="ts.struts.beans.ProcesaSeleccionForm"></form-bean>
        <form-bean name="ResultadoForm" type="ts.struts.beans.ResultadoForm"></form-bean>
      </form-beans>
      <param-name>debug</param-name>
      <param-value>5</param-value>
    </init-param>
    <init-param>
      <param-name>detail</param-name>
      <param-value>8</param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.do</url-pattern>
  </servlet-mapping>

  <action-mappings>
    <action path="/procesa" name="ProcesaSeleccionForm" scope="request" type="ts.struts.servlets.ProcesaSeleccionAction">
      <forward name="resultados" path="/paginas/resultado.jsp"></forward>
      <forward name="noResults" path="/paginas/noResultado.jsp"></forward>
    </action>
  </action-mappings>

  <context-param>
    <param-name>driver</param-name>
    <param-value>com.mysql.jdbc.Driver</param-value>
  </context-param>
  <context-param>
    <param-name>cadenaCon</param-name>
    <param-value>jdbc:mysql://localhost:practica04</param-value>
  </context-param>
</web-app>
```

Figura 3. Modificaciones del profesor.

Las notas del alumno resaltadas en recuadros negros en la Figura 4 son hechas del archivo original (Figura 2) del profesor.

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.4"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <servlet>
    <servlet-name>action</servlet-name>
    <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
    <init-param>
      <param-name>action</param-name>
      <param-value>9</param-value>
    </init-param>
    <init-param>
      <param-name>config</param-name>
      <param-value>/WEB-INF/struts-config.xml</param-value>
    </init-param>
    <init-param>
      <param-name>debug</param-name>
      <param-value>3</param-value>
    </init-param>
    <init-param>
      <param-name>detail</param-name>
      <param-value>3</param-value>
    </init-param>

    <form-validation>
      <global>
        <!-- opciones de validación-->
        <validator name="required"
          classname="org.apache.struts.validator.FieldChecks"
          method="validateRequired"
          methodParams="java.lang.Object,
            org.apache.commons.validator.ValidatorAction,
            org.apache.commons.validator.Field,
            org.apache.struts.action.ActionMessages,
            org.apache.commons.validator.Validator,
            javax.servlet.http.HttpServletRequest"
          msg="errors.required"/>
      </global>
    </form-validation>
    <load-on-startup>0</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.do</url-pattern>
  </servlet-mapping>

  <context-param>
    <param-name>driver</param-name>
    <param-value>com.mysql.jdbc.Driver</param-value>
  </context-param>
  <context-param>
    <!-- la practica que se ocupara esta guardada en el localhost en practica 03-->
    <param-name>cadenaCon</param-name>
    <param-value>jdbc:mysql://localhost/practica03</param-value>
  </context-param>
</web-app>

```

Figura 4. Notas del alumno.

Como se logró observar los archivos difieren y hay falta de información en las notas del alumno por lo que este tendrá la opción de actualizar sus notas con las modificaciones del profesor si este así lo desea.

3.2. Objetivos

3.2.1. *Objetivo general.*

- Diseñar e implementar una aplicación para la plataforma Moodle que compare dos archivos de notas de profesor, encontrando similitudes para incluir las anotaciones del alumno en un nuevo archivo.

3.2.2. *Objetivos particulares.*

- Elegir un programa ya existente para la comparación de archivos basado en la velocidad y exactitud de estos.
- Adecuar este programa a la plataforma Moodle.
- Diseñar e implementar un módulo para el aviso al alumno de los cambios realizados.
- Probar el funcionamiento de las aplicaciones desarrolladas.

Capítulo 4

Metodología de Desarrollo

4.1. Descripción técnica

Se buscó y utilizó una herramienta que compare y combine dos diferentes archivos tipo “xml”. La herramienta de actualización de archivos fue agregada al sistema Moodle, el cual permite que los alumnos estén siempre al día con las modificaciones que haga el profesor a las notas que el alumno descargó previamente.

Para la realización de esto se dio seguimiento a diferentes etapas:

- La primera etapa consistió en hacer una comparación entre distintas herramientas para la comparación y mezcla de archivos “xml”. La finalidad de las comparaciones y evaluaciones de estas herramientas es encontrar la más eficiente en cuanto a velocidad, resultados, exactitud y que cumpla con los requerimientos que se necesitan.
- Implementar los algoritmos elegidos e incorporarlos a la plataforma moodle, para esto es necesario conocer la manera en la que está funcionando el sistema, por lo que se deben estudiar las herramientas con que está hecho y conocer el diseño del código fuente.
- El archivo resultante se almacenará en una base de datos.
- El alumno tendrá la opción de actualizar sus notas si así lo desea. Para esto se hará una segunda comparación entre las notas actualizadas del profesor y el alumno generando otro archivo el cual se guardara en la ubicación dada por el alumno, si no desea actualizar podrá descargar por aparte el nuevo archivo del profesor.

4.2. Especificaciones técnicas

La aplicación construida comienza cuando el profesor desea actualizar un archivo en Moodle tal como se muestra en la Figura 5.

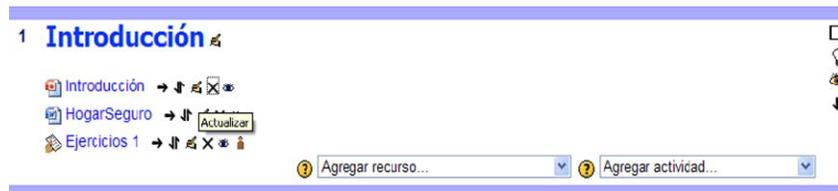


Figura 5. Actualizar

En la Figura 6 se observa la parte de actualizar archivos, cuando el profesor seleccione el botón de “Guardar cambios” en la parte inferior, el sistema se activa, empezando así con la comparación.

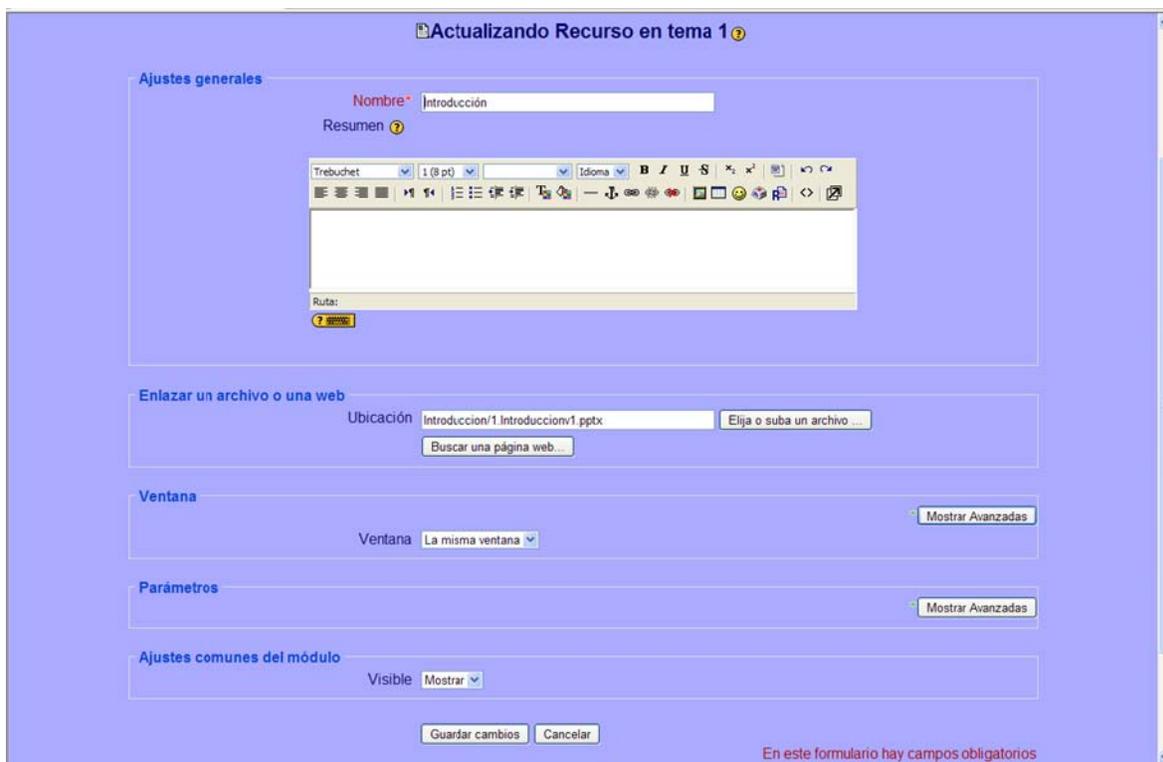


Figura 6. Menú actualizar

Primero se guarda el nuevo archivo del profesor y se aplica el algoritmo de comparación entre el anterior archivo y el más reciente que se agregó a Moodle. Posteriormente se genera un archivo con el contenido de los cambios que fueron

hechos. Existen diferentes formas de cambio, las cuales son la inserción, la eliminación y la actualización de datos como las principales funciones.

Por consiguiente, cuando el alumno entra al curso en el cual está inscrito, este puede ver un mensaje el cual muestra las notas que han sido actualizadas y así, podrá decidir si quiere que se le muestren los cambios; ya que en ocasiones dichos movimientos son mínimos, y al dar una afirmativa, se ejecutará una pantalla divulgando los datos con sus respectivas actualizaciones.

Si acepta la actualización, la comparativa es entre las notas, la creada automáticamente por la aplicación y la nota del alumno (las cuales el alumno tendrá que agregar), logrando la comparación nuevamente y generando así el nuevo archivo con las notas del alumno actualizadas que podrá guardar en la ruta que éste especifique (Figura 7).

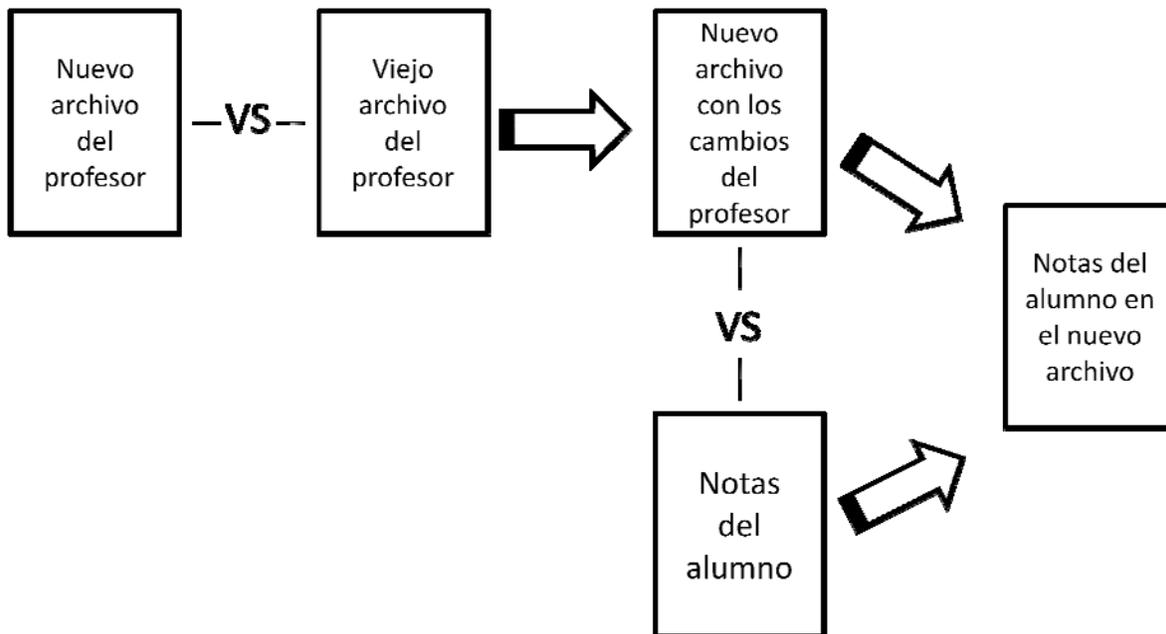


Figura 7. Generación del archivo final.

El proyecto se dio por concluido una vez que se implementaron y probaron todos los módulos a desarrollar. Para el manejo de archivos se considerará que estos son colocados en un directorio específico de donde son cargados por la aplicación desarrollada, además se limitará a usar archivos tipo “xml” ya que es de estándar abierto³ y de baja complejidad para su manipulación.

4.3. Recursos básicos para el desarrollo

- Se contó con todos los recursos para poder desarrollar el proyecto satisfactoriamente.
- Características de la PC
Se contó con una Laptop, la cual tiene como características técnicas las siguientes:
 - Procesador Intel® core™2 CPU 1.73 GHz.
 - Memoria RAM de 2.5 GB.
 - SATA 160GB.
 - Linux Ubuntu 10.04.
 - IDE Eclipse Helios (versión de Eclipse para desarrollar PHP).

³ Estándar abierto es cuando el modo de representación de los datos es transparente y/o su especificación está disponible públicamente objetivo. Los formatos abiertos son, ordinariamente, estándares determinados por autoridades públicas o instituciones internacionales cuyo objetivo es establecer normas para interoperabilidad de software. Un formato abierto puede ser codificado en una forma *transparente* (leible en cualquier editor de texto: este es el caso de lenguajes de marcado) o en forma *binaria* (no leible en un editor de texto pero enteramente decodificable una vez que las especificaciones del formato son conocidas). [3]

Capítulo 5

Desarrollo

En este capítulo se detalla cada uno de los módulos que componen al sistema. Describiremos sus características, dependencias, tareas específicas y salidas. Así mismo se hará un seguimiento de cada proceso que los módulos realicen.

5.1. Arquitectura de la aplicación propuesta

En la Figura 8 se muestra la arquitectura de la aplicación desarrollada para el sistema Moodle de la actualización de notas.

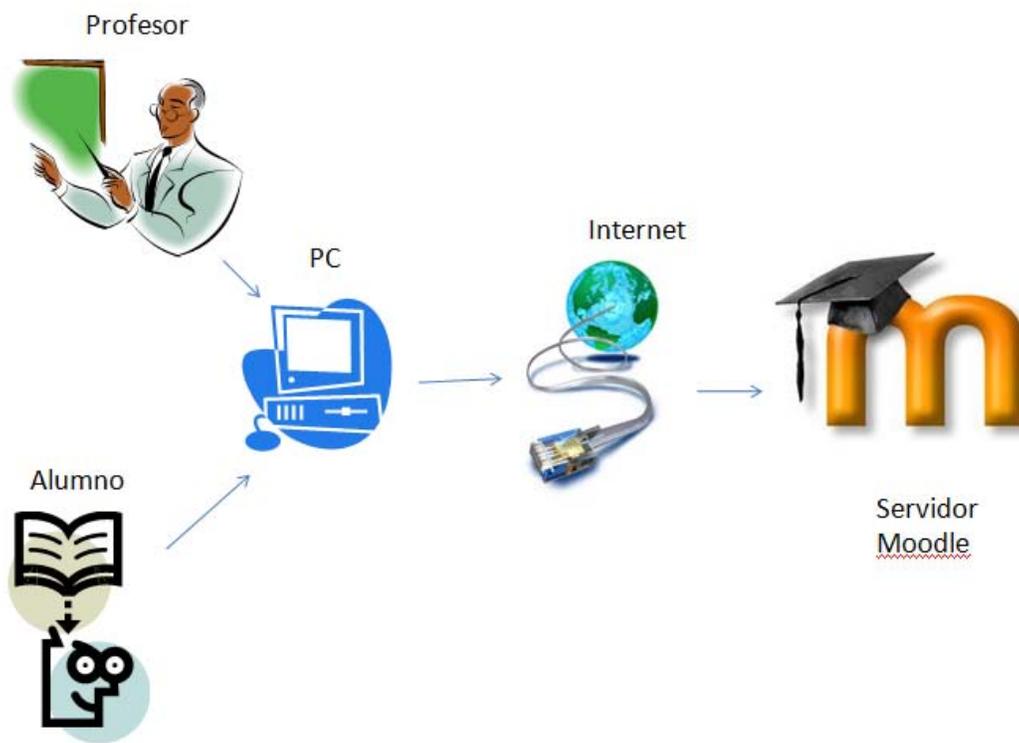


Figura 8. Arquitectura Moodle.

La arquitectura utilizada en esta aplicación está basada en el modelo cliente/servidor, siguiendo el estándar de la plataforma Moodle, la cual consiste en dos capas front/end y back/end [19].

- Font/end(Cliente): Es la parte de la aplicación con la que el usuario interactúa. Como se puede ver en la imagen los usuarios serían el profesor y el alumno que interactúan en la aplicación mediante las PC's.
- Back/end(Servidor): Es la parte no interactiva de la aplicación, en esta parte se encuentra la base de datos y el código de la aplicación.

5.2. Metodología de Análisis, Diseño e Implementación

Como primer paso, se analizó el módulo que se implementó, para ello se utilizó el lenguaje de modelado de sistemas UML (*Unified Modeling Language*) [21]. Para el desarrollo de este proyecto sólo se requirió el diagrama de actividades.

Como segundo paso, se consideraron las diferentes herramientas de comparación y mezcla de archivos.

En el tercer paso, se instaló la distribución estable más reciente de la plataforma Moodle.

Luego se estudió cierta parte del código de Moodle sobre el manejo de archivos, esto con el fin de poder implementar la herramienta elegida.

Por último se implementó la herramienta de mezcla para la posible actualización de las notas del profesor y las notas del alumno, donde el alumno cuando inicia sesión puede ver en una ventana de actualizaciones recientes que el archivo ha sido modificado.

5.3. Módulo de implementación

Se hizo el diagrama de actividades (ver Figura 9), el objetivo principal de este tipo de diagrama es mostrar el funcionamiento de uno o varios métodos mostrándolo paso a paso; como se puede observar hay una secuencia para el proceso que hace Moodle:

1. Se valida el curso.
2. Se verifica que el usuario esté registrado en el curso.
3. Se crean diferentes tablas, una de ellas es la de los temas del curso en el cual se agregan archivos y como opción se pueden actualizar accedendo a la base de datos.
4. Se aplica la herramienta para mezclar los archivos para así después el alumno se entere de dichos cambios y pueda actualizar sus notas.

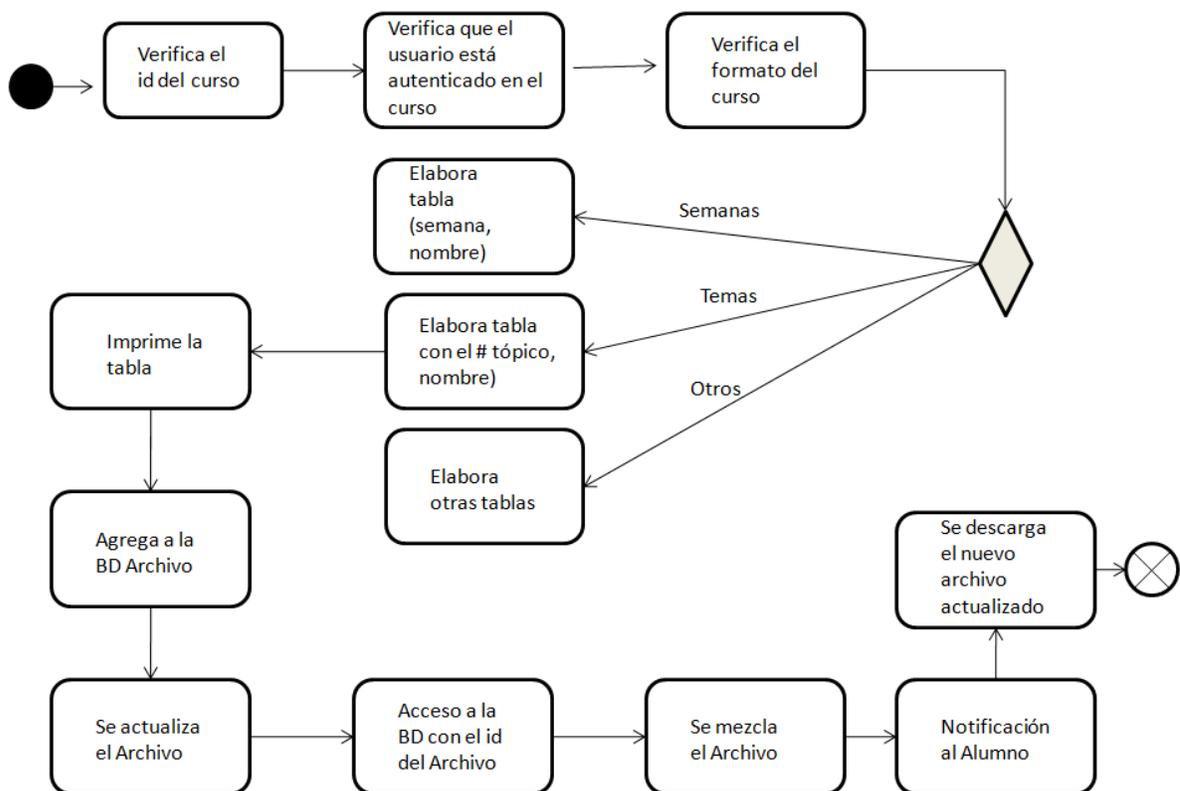


Figura 9. Diagrama de actividades del módulo.

Capítulo 6

Herramientas de Comparación y Mezcla de Archivos xml

En este capítulo se compararon las herramientas más comunes para la comparación y mezcla de archivos.

✓ *Oxygen XML Diff* [4]

XML Diff es una solución completa para comparar y combinar archivos XML, en esta se ofrece la comparación de directorios y archivos. La comparación también se realiza dentro de archivos ZIP. Cuando se realiza una diferenciación entre los archivos pueden ser comparados por varios métodos: por fecha y hora, por el contenido, comparación binaria (el contenido del archivo es comparado byte por byte, esto es más rápido que el método de contenido pero solo se utiliza cuando la diferencia es de importancia). En la Figura 10 vemos un ejemplo de cómo funciona esta herramienta, logrando notar los cambios que hay entre un archivo y otro.

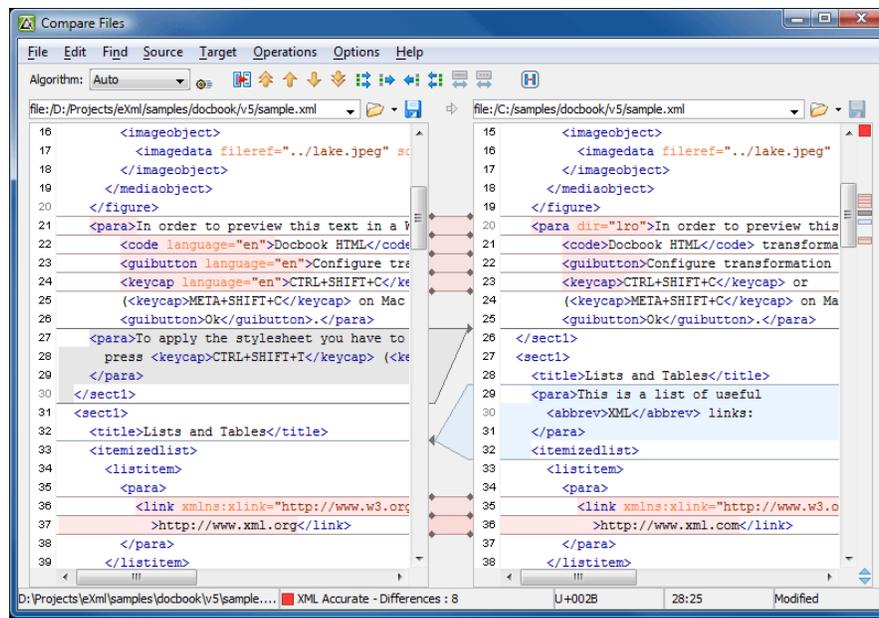


Figura 10. Comparación de dos archivos XML con Oxygen XML Diff.

Sistemas operativos:

- Windows XP.
- Windows Vista, 7, 2003, Server 2008.
- Mac OS X version 10.4 o más reciente.
- Unix/Linux.

✓ **KDiff3** [5]

Es una herramienta para ver diferencias y fusionar archivos y directorios que sean de texto, mostrando las diferencias línea a línea y carácter a carácter. También tiene un editor para resolver conflictos de fusión (ver Figura 11).

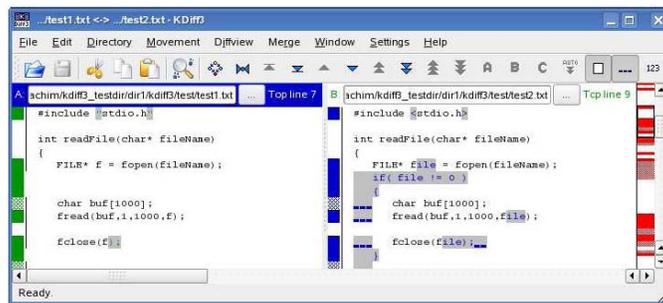


Figura 11. Comparación KDiff3.

Sistemas operativos:

- Linux.
- Unix.
- MS-Windows.
- Apple Mac OS X.

✓ **DiffDog** [6]

Incluye una funcionalidad de gran alcance para la comparación, la combinación y edición de archivos XML de una manera inteligente y consciente de XML. Incluye características que hacen la diferenciación de archivos de texto más fácil, tales como ayudas a la navegación de archivos y opciones de visualización de personalización. En la Figura 12 nos muestran las diferencias de ambos archivos con una línea de unión a las partes de código que son similares.

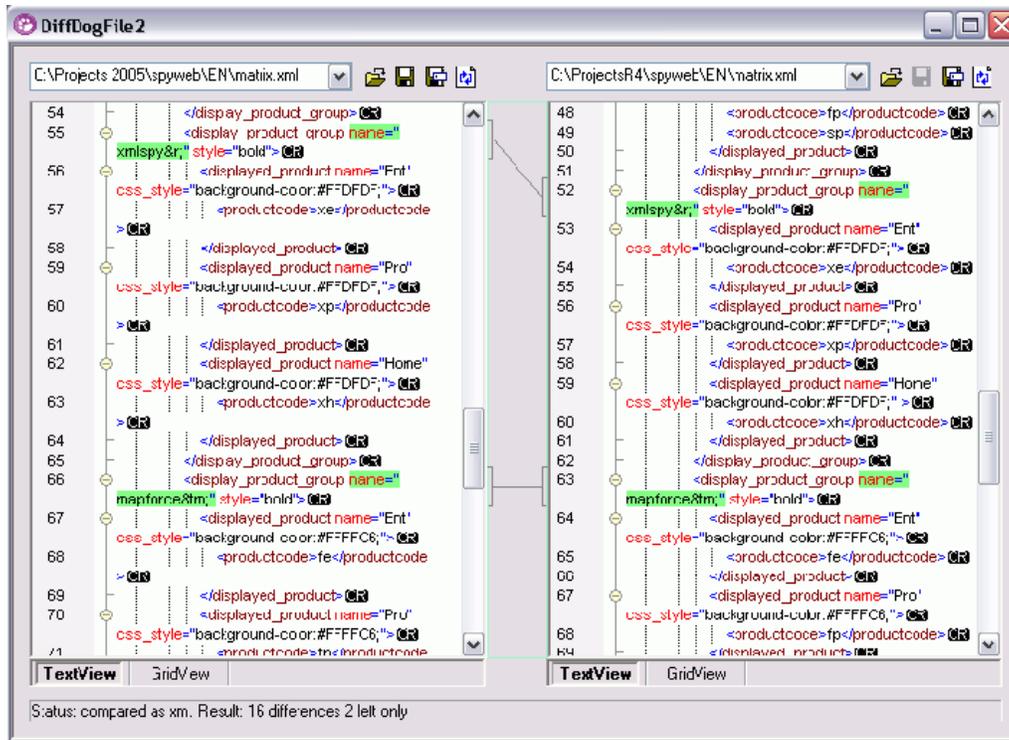


Figura 12. Comparación de dos archivos XML con DiffDog.

Sistemas operativos:

- Windows XP, Vista, 7.

✓ **Xml Diff & Merge [7]**

Es un programa escrito en Java que puede ser utilizado para comparar los documentos XML para conciliar o comprender las diferencias entre los documentos debido a los cambios realizados por los usuarios. La herramienta indica cada diferencia con el documento en base con los símbolos y códigos de color y ofrece la oportunidad de editar cada diferencia a medida que el paso por el documento. Incluye correcciones de errores. Se puede ver un ejemplo en la Figura 13 donde al combinarse los dos archivos, el resultante adquiere líneas de código que el primer archivo no tenía.

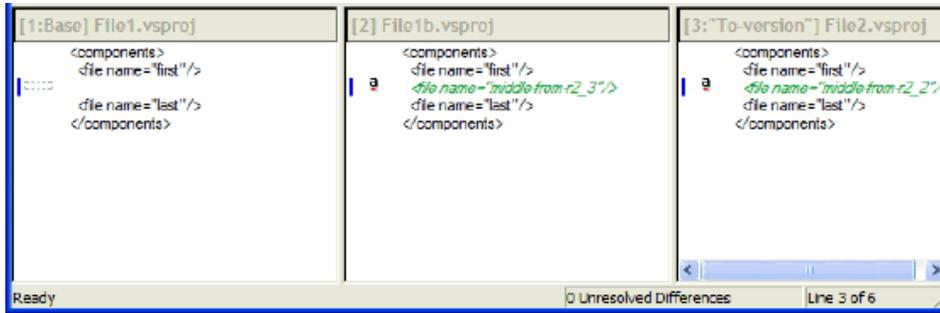


Figura 13. Comparación de dos archivos XML Diff & Merge.

Sistemas operativos:

- Windows NT, XP, 2000.
- Linux
- AIX

✓ **Herramienta Diff and Merge**

Diff y *Merge* son unas herramientas en Linux las cuales podemos llamar por comandos.

Diff nos muestra la diferencia entre dos archivos[10].

Modo de uso:

“**diff** [Opciones] Archivo1 Archivo2“

Si se utiliza este comando, se podrá ver la diferencia entre los archivos, como se muestra abajo:

```
# diff Archivo1 Archivo2
1d0
< In Archivo1 only
2a2
> In Archivo2 only
```

En el output, un "<" en la primera columna indica que la línea existe en el archivo mencionado primero, —es decir, archivo1. Un ">" en ese lugar indica que la línea existe en el segundo archivo (archivo2). Los caracteres 1d0 de la primera línea del output muestra lo que debe hacerse para modificar en el archivo1 para hacerlo idéntico al archivo 2.

Otra opción, -y, muestra el mismo output, pero uno al lado del otro:

```
# diff -y Archivo1 Archivo2 -W 120
In Archivo only          <
In Archivo and file2    > In Archivo1 and file2
                        > In Archivo2 only
```

Si solo se desea saber si los archivos son diferentes, no necesariamente en qué difieren, puede utilizar la opción -q.

```
# diff -q Archivo3 Archivo4
# diff -q Archivo3 Archivo2
Files Archivo3 and Archivo2 differ
```

Los archivos 3 y 4 son los mismos, con lo cual no hay output; caso contrario, se informa si los archivos difieren.

El output muestra contenidos de ambos archivos pero suprime los duplicados, los signos + y – de la primera columna indican las líneas de los archivos. Ningún carácter en la primera columna indica su presencia en ambos archivos.

El comando tiene en cuenta el espacio en blanco. Si quiere ignorar el espacio en blanco, use la opción -b. Use la opción -B para ignorar las líneas en blanco. Finalmente, use -i para ignorar el caso.

Merge Es la herramienta de Linux que mezcla dos archivos.

Modo de uso:

```
merge [opciones] archivo1 archivo2 archivo3
```

-A: Se mezclan todos los cambios que van desde archivo2 al archivo3 y lo guarda en el archivo1.

-p: Envía resultados a la salida estándar en lugar de sobrescribir el archivo1.

-q: No advierte acerca de los conflictos.

Merge incorpora todos los cambios que van desde el archivo2 al archivo3 en el archivo1. El resultado normalmente entra en el archivo1.

Supongamos archivo2 es el original, y ambos archivo1 y archivo3 son modificaciones de archivo2. Después merge combina todos los cambios.

Nota: Cuando la combinación de archivos todos los archivos debe existir.

Sistemas operativos:

- Linux
- Unix

6.1. Comparación de las herramientas

Se hizo una tabla para comparar herramientas arriba mencionadas respecto a las características que se consideraron las más importantes para la realización de este proyecto de actualización de archivos (ver Tabla 1):

Herramienta	Sistema Operativo	Lenguaje en el que está desarrollado	Complejidad de uso	Tipo
Oxygen Diff	Windows, Linux, Mac OS	Java	Alta	Comercial
KDiff3	Windows, Linux, Mac OS, Unix	Java	Media	Comercial
DiffDog	Windows	C#	Alta	Comercial
XML Diff & Merge	Windows, Linux, AIX	Java	Media	Comercial
Diff & Merge	Linux	C	Sencilla	Libre

Tabla 1. Comparación de herramientas de comparación y mezcla de archivos XML.

Conforme a esto se tomó la decisión de usar la herramienta “Diff & Merge” primeramente por el sistema operativo en el cual funciona (*Linux*), ya que es donde trabajamos para hacer el proyecto de actualización de archivos, en segundo lugar la adaptación al sistema Moodle, el cual se logra con solo un llamado al sistema, en cuanto a la complejidad de uso “Diff & Merge” es sencilla ya que no requiere intervención del usuario para ser ejecutada y por último pero no menos importante, la licencia es libre ya que se encuentra incluida dentro del sistema de Linux.

Capítulo 7

Instalación de Moodle

Se instaló la versión más reciente de moodle (2.02) en Ubuntu 10.04.

Para poder instalar Moodle 2.02 fue necesario instalar primero la base de datos de MySQL 5, el servidor Apache 2 y phpMyAdmin.

A continuación se detallarán todos los pasos a seguir para poder instalar la plataforma de Moodle, primeramente entramos con permisos de administrador desde la Terminal para poder realizar las instalaciones.

- *sudo su* ->Comando para entrar como administrador.

7.1. Instalación de MySQL

Posteriormente se instaló MySQL 5 con el siguiente comando:

```
apt-get install mysql-server mysql-client
```

Se nos pide que demos nuestra contraseña para el usuario root la cual es válida para el usuario root@localhost.

7.2. Instalación de Apache

Después de instalar MySQL se procedió a instalar el Apache 2, el cual está disponible como un paquete de Ubuntu package, así pues, se puede instalar con el siguiente comando:

```
aptitude install apache2
```

Ahora abrimos un navegador de internet con la dirección "http://localhost", y se logra observar en la Figura 14 que el Apache2 funciona correctamente.



Figura 14. Funcionamiento de Apache.

La documentación en Ubuntu de Apache se genera por default en la ruta:

`/var/www/`

7.3. Instalación de PHP

Instalamos PHP5 y el módulo de PHP para Apache como sigue:

```
apt-get install php5 libapache2-mod-php5
```

Posteriormente reiniciamos Apache2:

```
/etc/init.d/apache2 restart
```

Para probar el funcionamiento correcto de PHP5 creamos el archivo "info.php" en la siguiente ruta donde está el servidor de Apache:

```
vi /var/www/info.php
```

El archivo debe contener esta información:

```
<?php  
phpinfo();  
?>
```

Ahora en el navegador llamamos al archivo con la dirección "http://localhost/info.php" y se logra observar la información de php que se instaló (ver Figura 15).

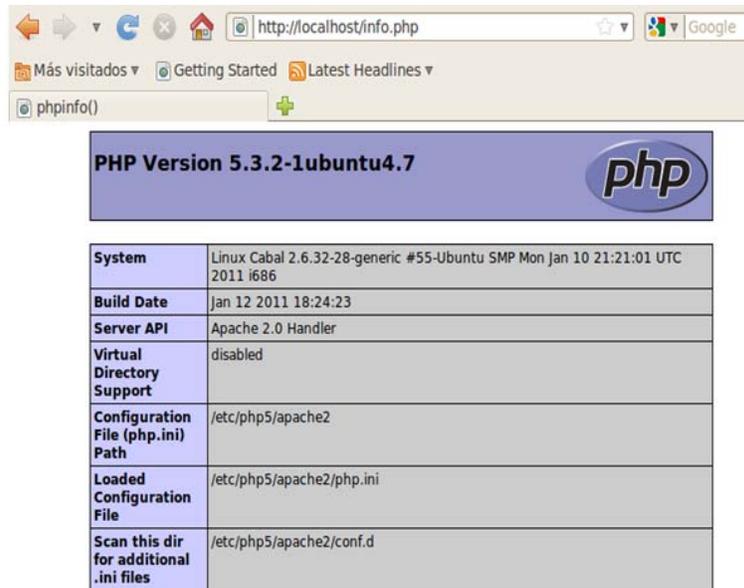


Figura 15. Información de PHP.

7.3.1. Obtener soporte de MySQL para PHP

Es necesario instalar varios paquetes para el correcto funcionamiento de MySQL en php, así que se instalaron los siguientes:

```
Apt-get install php5-mysql php5-curl php5-gd php5-idn php-pear php5-imagick php5-imap php5-mcrypt php5-memcache php5-mhash php5-ming php5-ps php5-pspell php5-recode php5-snmpphp5-sqlite php5-tidy php5-xmllrpc php5-xsl php5-json
```

Reiniciamos Apache y para verificar que se instalaron correctamente ejecutamos info.php y notamos que el módulo de MySQL ya aparece (ver Figura 16).

mysql

MySQL Support	enabled
Active Persistent Links	0
Active Links	0
Client API version	5.1.41
MYSQL_MODULE_TYPE	external
MYSQL_SOCKET	/var/run/mysqld/mysqld.sock
MYSQL_INCLUDE	-I/usr/include/mysql
MYSQL_LIBS	-L/usr/lib -lmysqclient_r

Figura 16. Paquete de MySQL para PHP.

7.4. Instalación de phpMyAdmin

PhpMyAdmin es una interface web por medio de la cual puedes manejar las bases de datos de MySQL, así que procedemos a instalarlo como una opción para darse una idea de lo que se va creando y/o modificando[11].

Utilizamos el siguiente comando:

```
sudo apt-get install phpmyadmin
```

Después de instalarlo hay que hacer un enlace simbólico desde “usr/share/phpmyadmin hacia /var/www/phpmyadmin” con la instrucción :

```
sudo ln -s /usr/share/phpmyadmin /var/www
```

Luego entonces podemos acceder a phpMyAdmin con “http://localhost/phpmyadmin/” en el navegador (ver Figura 17).

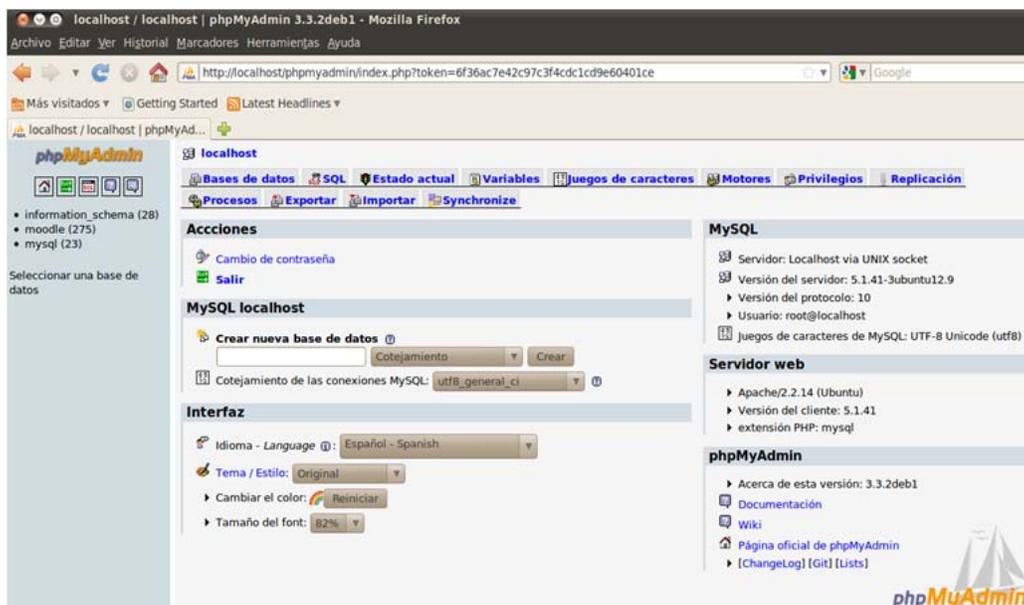


Figura 17. Interfaz phpMyAdmin.

7.5. Instalación de Moodle 2.0

Finalmente se prosigue con la instalación de Moodle descargándolo desde <http://download.moodle.org/>

Descargamos el archivo de Moodle para Linux, lo descomprimos y seguimos los pasos de instalación[12].

Al finalizar la instalación se puede acceder a Moodle con la dirección “<http://localhost/moodle>”.

Creamos la ruta de almacenamiento de archivos: `/var/moodledata/filedir/`

Capítulo 8

Modificación del Código

La modificación del código de Moodle es necesaria dadas las especificaciones de la implementación de la herramienta de mezcla de archivos. Sin embargo, se reconoce que esta opción presenta el siguiente inconveniente: la versión que se genere a partir de la modificación del código de Moodle, posiblemente tendrá problemas de compatibilidad con versiones posteriores y/o anteriores de la misma plataforma Moodle.

8.1. Diseño de la Base de Datos

8.1.1. Carga de archivos nuevos

Cuando los archivos son recibidos, ya sean de nuevos registros de usuarios o agregación de archivos, son almacenados en la carpeta de entrada, (destino definido por la aplicación de Moodle, la dirección por defecto que el Sistema da es: “/var/moodledata/”). Específicamente para esta aplicación de la herramienta de mezcla, que trata de los archivos “xml” subidos por el profesor se ubican en la ruta “/var/moodledata/filedir”.

Es importante que sólo se agreguen y eliminen archivos a través de la interfaz de Moodle, es decir, no entrar a las carpetas directamente, ya que este maneja ciertos procesos para el almacenamiento como es la encriptación de los nombres de los archivos, proceso para volver ilegible información, esta, una vez encriptada sólo puede leerse aplicándole una clave (“id”) manejada por Moodle para poder desencriptar y utilizar los archivos.

Por ejemplo:

Considere que el nombre del archivo es “*documento1.xml*” y después de que Moodle lo encripta sería “*fe8059fehi89d239ce3i3*”. Este se guarda dentro de dos carpetas anidadas en la ruta “/var/moodledata/filedir”, la primera carpeta tiene el nombre de los primeros dos caracteres del nombre del archivo (“fe”) y la segunda

subcarpeta los siguientes dos caracteres ("80"), por ende el sistema de almacenamiento de archivos dejaría el archivo en la ruta siguiente:

"/var/moodledata/filedir/fe/80/fe8059fehi89d239ce3i3".

El nombre de la tabla para almacenar estos archivos tiene el nombre de mdl_files, la cual contiene varios parámetros entre los que se encuentran el "id", "contenthash", "itemid", "filename" los cuales son los que ocuparemos para manejar la base de datos y que se muestran en la Tabla 2.

id	contenthash	pathnamehash	contentid	component	filename	itemid	filepath	filename	userid	filesize	mime-type	status	source	author	license
1003	da39a3ec-cfe6-4b0d-3230-0e9796018904680709	8041233e79c4ed728843e7f2d37c2b0c4d4f	13	user	draft	91292477	/	-	2	0	NULL	0	NULL	NULL	NULL
1004	fe806496d390d440d2c002366600aa70e6d979e	800121ae-0a1819e82c-ca1986a97c07e7d233	13	user	draft	539201045	/	2.xml	2	1110	application/xml	0	NULL	Rodrigo Cabal	rightsreserved
1005	da39a3ec-cfe6-4b0d-3230-0e9796018904680709	992798a5313c-05a4160450980d2c3d37c50a123	13	user	draft	539201045	/	-	2	0	NULL	0	NULL	NULL	NULL
1006	fe806496d390d440d2c002366600aa70e6d979e	0e8450bc-1415084428893170007c48-0a007893cd	13	user	draft	506793337	/	2.xml	2	1110	application/xml	0	NULL	Rodrigo Cabal	rightsreserved
1007	da39a3ec-cfe6-4b0d-3230-0e9796018904680709	000cc-889f9070bc-cbe8714cc9f04c78980cc475	13	user	draft	506793337	/	-	2	0	NULL	0	NULL	NULL	NULL
1008	fe806496d390d440d2c002366600aa70e6d979e	110c-08412a688459063c-36d79e0606c189a2	13	user	draft	571753767	/	2.xml	2	1110	application/xml	0	NULL	Rodrigo Cabal	rightsreserved
1009	da39a3ec-cfe6-4b0d-3230-0e9796018904680709	018079712c-343902400a1e99751c642ae00c044	13	user	draft	571753767	/	-	2	0	NULL	0	NULL	NULL	NULL
1010	fe806496d390d440d2c002366600aa70e6d979e	901c28d90c-b0c70501828a13c-0e7c83e925004	13	user	draft	070073706	/	2.xml	2	1110	application/xml	0	NULL	Rodrigo Cabal	rightsreserved
1011	da39a3ec-cfe6-4b0d-3230-0e9796018904680709	e6c9c-a873477810f05042c-843308433548700c22	13	user	draft	070073706	/	-	2	0	NULL	0	NULL	NULL	NULL
1012	fe806496d390d440d2c002366600aa70e6d979e	0e4c-8a88020e0d79eac20b0697173e71371000cd	13	user	draft	578542915	/	4.xml	2	1048	application/xml	0	NULL	Rodrigo Cabal	rightsreserved
1013	da39a3ec-cfe6-4b0d-3230-0e9796018904680709	8070aeb40c1c12c0c-00000e893104da02839378	13	user	draft	578542915	/	-	2	0	NULL	0	NULL	NULL	NULL
1014	fe806496d390d440d2c002366600aa70e6d979e	0c-0a0e20081c-0c-3a224320446027004e4e7	13	user	draft	301033970	/	4.xml	2	1048	application/xml	0	NULL	Rodrigo Cabal	rightsreserved
1015	da39a3ec-cfe6-4b0d-3230-0e9796018904680709	04c4d71038430e2c4e70c-006441007078c-c04	13	user	draft	301033970	/	-	2	0	NULL	0	NULL	NULL	NULL
1016	fe806496d390d440d2c002366600aa70e6d979e	e90c-d8d06078c70d0e0000000082c-775302c	13	user	draft	277592218	/	4.xml	2	1048	application/xml	0	NULL	Rodrigo Cabal	rightsreserved
1017	da39a3ec-cfe6-4b0d-3230-0e9796018904680709	2c-14d0312120c0e00001d1d138429e8254e9	13	user	draft	277592218	/	-	2	0	NULL	0	NULL	NULL	NULL
1018	fe806496d390d440d2c002366600aa70e6d979e	71c2c4402e201103c-0270d0d0748715460190	13	user	draft	414878946	/	4.xml	2	1048	application/xml	0	NULL	Rodrigo Cabal	rightsreserved
1019	da39a3ec-cfe6-4b0d-3230-0e9796018904680709	122071c73a01512ec-993844ac-8775a77140c-cf9	13	user	draft	414878946	/	-	2	0	NULL	0	NULL	NULL	NULL
1020	fe806496d390d440d2c002366600aa70e6d979e	111e9178030a2e07942076a040a009a2e65c-808	13	user	draft	520085268	/	4.xml	2	1048	application/xml	0	NULL	Rodrigo Cabal	rightsreserved
1021	da39a3ec-cfe6-4b0d-3230-0e9796018904680709	00a0d0dc-180003705d4d990c7e912e45200e	13	user	draft	520085268	/	-	2	0	NULL	0	NULL	NULL	NULL
1022	fe806496d390d440d2c002366600aa70e6d979e	43796274400811e-6000984121e25d2-07190	13	user	draft	509293460	/	4.xml	2	1048	application/xml	0	NULL	Rodrigo Cabal	rightsreserved
1023	da39a3ec-cfe6-4b0d-3230-0e9796018904680709	20edc20c-830388c-97cc7598c0c0c7094e0a30	13	user	draft	509293460	/	-	2	0	NULL	0	NULL	NULL	NULL

Tabla 2. mdl_files de la BD de Moodle.

8.1.2. Estructura de la Base de Datos

El equipo Servidor tiene como uno de sus módulos el almacenamiento de la información de los archivos agregados por el usuario. Esto no quiere decir que se reduzca solo a los archivos que son añadidos por el usuario sino que también se tiene información personal de cada uno de los usuarios registrados (Nombre, login, password, correo electrónico, curso, Imagen, etc.), así como también información de los cursos dados de alta en Moodle (Nombre, Fecha, Recursos, contraseña, temas, etc.).

8.1.3. Diccionario de Datos

Un Diccionario de Datos es una recopilación de información que contiene las características lógicas de los datos que se van a utilizar en el sistema, incluyendo el contenido y la organización. Identifica los lugares donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información. En un Diccionario de Datos se encuentra la lista de todos los elementos que forman parte del flujo de datos de todo el sistema. Los elementos más importantes son flujos de datos, almacenes de datos y procesos. El Diccionario de Datos que ocuparemos es el de la tabla llamada “mdl_files” (ver Tabla 3) donde nos muestra los campos de la tabla con sus respectivos atributos.

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
id	bigint(10)		UNSIGNED	No	None	auto_increment
contenthash	varchar(40)	utf8_unicode_ci		No		
pathnamehash	varchar(40)	utf8_unicode_ci		No		
contextid	bigint(10)		UNSIGNED	No	None	
component	varchar(100)	utf8_unicode_ci		No		
filearea	varchar(50)	utf8_unicode_ci		No		
itemid	bigint(10)		UNSIGNED	No	None	
filepath	varchar(255)	utf8_unicode_ci		No		
filename	varchar(255)	utf8_unicode_ci		No		
userid	bigint(10)		UNSIGNED	Sí	NULL	
filesize	bigint(10)		UNSIGNED	No	None	
mimetype	varchar(100)	utf8_unicode_ci		Sí	NULL	
status	bigint(10)		UNSIGNED	No	0	
source	text	utf8_unicode_ci		Sí	NULL	
author	varchar(255)	utf8_unicode_ci		Sí	NULL	
license	varchar(255)	utf8_unicode_ci		Sí	NULL	
timecreated	bigint(10)		UNSIGNED	No	None	
timemodified	bigint(10)		UNSIGNED	No	None	
sortorder	bigint(10)		UNSIGNED	No	0	

Tabla 3. Diccionario de datos de los archivos de Moodle.

8.2. Implementación al código

Con el apoyo de la página web que contiene el código documentado de Moodle [13], en formato HTML se pudo encontrar el archivo php a modificar para poder implementar la herramienta de mezcla.

El módulo que se presenta en esta sección es el “File manager”, el cual contiene las funciones de agregar los archivos, eliminarlos, modificarlos, así como también agregar directorios, entre otras cosas.

La ruta del archivo modificado es “/var/www/moodle/lib/form/filemanager.php” el cual tiene funciones para la obtención de ciertas características de los archivos y del cliente, la mayoría utilizando funciones de manejadores de archivos definidas en “/var/www/moodle/lib/filestorage/*.php”.

Debido a la gran complejidad de manipular el código y del gran número de clases y funciones anidadas, se optó por hacer una función con sentencias MySQL [14] para poder obtener la ruta encriptada del archivo por medio del “itemid” (ver tabla1) que la misma clase de filemanager.php proporciona.

En primer lugar se efectuó el diagrama de actividades (ver Figura 7) el cual nos muestra qué hace paso a paso un método o propiedad. Los diagramas de actividad se parecen mucho a la definición de un método o propiedad utilizando pseudo-código. Este diagrama nos permitió estudiar la mejor forma para modificar los métodos necesarios para la implementación de la herramienta de mezcla.

A continuación se procedió a modificar el código, primeramente al acceder a la opción de actualizar un archivo (ver Figura 18), se pasa a la interfaz modificada.



Figura 18. Acceso a la interfaz de Actualizar.

Posteriormente en la sección de “Contenido” se agrega el nuevo archivo para actualizar (ver Figura 19).



Figura 19. Botón agregar.

Una vez que se agrega el archivo podemos ver en la Base de Datos de “phpMyAdmin” que ha sido agregado y encriptado (ver Tabla 4).

id	contenthash	pathnamehash	contextid	component	filearea	itemid	filepath	filename
1033	fe80fa9496d5906d40d2c00256d6bba7de6d9fae	09e0cdef1252b53187ff1f69150b0851eb0f2ef1	13	user	draft	790099003	/	Viejo.xml
1034	da39a3ee5e6b4b0d3255bfe95601890afd80709	1792e3a76142200cfb2430ec58d2ec4019dc7e2f	13	user	draft	790099003	/	.
1037	fe80fa9496d5906d40d2c00256d6bba7de6d9fae	74df1874d1552588f44443b85153a9c6532be08e	13	user	draft	733511235	/	Viejo.xml
1038	da39a3ee5e6b4b0d3255bfe95601890afd80709	818dcaa9d14a43c8e5a954bbc4f4c70cdc862bd8	13	user	draft	733511235	/	.
1039	fe80fa9496d5906d40d2c00256d6bba7de6d9fae	3144cc18154fb14528e6ee39e122794a673b0e00	37	mod_resource	content	0	/	Viejo.xml
1040	da39a3ee5e6b4b0d3255bfe95601890afd80709	4b45d0de29a43a9680736186a102bc32795456b9	37	mod_resource	content	0	/	.
1041	fe80fa9496d5906d40d2c00256d6bba7de6d9fae	14e237ea9b06a92f0caf3b6e0b3541451184cfb2	13	user	draft	986986037	/	Viejo.xml
1042	da39a3ee5e6b4b0d3255bfe95601890afd80709	5398482b515bcc72e404ef11e242382552f7ea60	13	user	draft	986986037	/	.
1043	fe80fa9496d5906d40d2c00256d6bba7de6d9fae	b08177e4d7e67cdc1e8300c816be0c21b895b794	13	user	draft	319098956	/	Viejo.xml
1044	da39a3ee5e6b4b0d3255bfe95601890afd80709	87546cd3c939ce3d9e4993bbcc94e8a3dced299	13	user	draft	319098956	/	.
1045	68d666d2e3e63c35a931502a6080e98d8e8d75bd	953f7d1aac87f05cac141ca304dd25013b5daf1	13	user	draft	319098956	/	Nuevo.xml

Tabla 4. BD mdl_files con el nuevo archivo.

Como se logra observar el “itemid” (marcado con amarillo en la Tabla 4) que se utiliza para sacar el nombre del archivo es el mismo, en este caso definido con el número “319098956” (columna itemid de la Tabla 4).

Para poder obtener el valor “itemid” llamamos a la siguiente función anteriormente creada:

```
$itemid = $this->getValue();
```

Donde *getValue()* es una función la cual nos regresa el valor de itemid.

Una vez obtenido el valor “itemid” se utilizó la siguiente secuencia MySQL para poder obtener el “contenthash” y el “id” (ver Tabla 3) del viejo archivo:

```
SELECT contenthash, id FROM mdl_files WHERE itemid=$itemid
```

Así pues, obtenemos la ruta y el “id” del viejo archivo, con este podemos obtener la ruta del segundo archivo con la siguiente secuencia MySQL:

```
SELECT contenthash FROM mdl_files WHERE id=$id_nuevo
```

Por consiguiente ya tenemos ambas rutas, del viejo archivo a actualizar y del nuevo.

Una vez teniendo estos importantes datos, procedemos a hacer la actualización del archivo haciendo click en el botón “Mezcla” que se agregó para realizar dicha función (ver Figura 20).

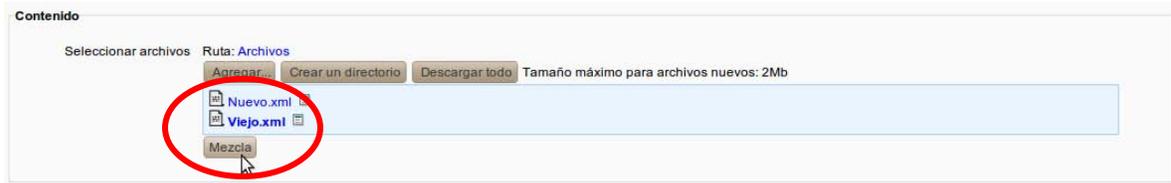


Figura 20. BD mdl_files con el nuevo archivo.

El botón “Mezcla” está creado de la siguiente manera:

```
<form onSubmit="file_manager($itemid)">
<input type="submit" name="mezclar" value = "Mezcla"></form>
```

Donde “file_manager(\$itemid)” es la función que se ejecuta para tener acceso a la base de datos junto con la herramienta de mezcla de archivos.

Se implementa el siguiente algoritmo para hacer la actualización del archivo:

1.- Se crea y se abre un archivo temporal para que se guarden los cambios.

```
$archivo="/var/moodledata/filedir/$carpeta_viejo1/$carpeta_viejo2/temporal.xml";
```

```
$id = fopen($archivo,'w+');
```

2.- Se define la siguiente sentencia con la ruta del viejo, nuevo y archivo temporal:

```
$e = "/var/moodledata/filedir/$carpeta_viejo1/$carpeta_viejo2/temporal.xml
/var/moodledata/filedir/$carpeta_viejo1/$carpeta_viejo2/$viejo
/var/moodledata/filedir/$carpeta_nuevo1/$carpeta_nuevo2/$nuevo";
```

Donde:

\$carpeta_viejo1 y \$carpeta_nuevo1 son las carpetas principales donde se encuentran las carpetas contenedoras del archivo viejo y el archivo nuevo respectivamente.

\$carpeta_viejo2 y \$carpeta_nuevo2 son las subcarpetas donde se encuentran los archivos a mezclar.

\$viejo y *\$nuevo* son los nombres de los archivos a mezclar.

3.- Ocupamos la función “system()” para hacer llamado a los comandos unix y escribimos el comando de actualización:

```
system('merge -q '.escapeshellcmd($e), $retval);
```

Donde:

merge es el comando que hace la mezcla.

escapeshellcmd() es un comando unix que se utiliza para omitir cualquier caracter en una cadena que pueda ser usado para engañar a un comando de shell a que ejecute comandos arbitrarios. Esta función puede ser usada para asegurarse de que cualquier dato viniendo de la entrada del usuario sea pasado desapercibido antes de que este dato sea pasado a la función system() [15];

\$e es la ruta anteriormente definida.

\$retval es donde se guarda la última línea de ejecución del comando “system()”, la cual para los fines de este proyecto no nos es de utilidad.

4.- Renombramos el archivo actualizado con el nombre del archivo original y borramos el archivo sobrante:

```
$f= "/var/moodledata/filedir/$carpeta_viejo1/$carpeta_viejo2/$viejo";
```

```
system('rm '.escapeshellcmd($f), $retval);
```

```
$g="/var/moodledata/filedir/$carpeta_viejo1/$carpeta_viejo2/temporal.xml  
/var/moodledata/filedir/$carpeta_viejo1/$carpeta_viejo2/$viejo";
```

```
system('mv '.escapeshellcmd($g), $retval);
```

Donde:

\$f es la ruta del viejo archivo, el cual vamos a eliminar.

rm- es un comando unix para borrar un archivo.

\$g secuencia que se usa para renombrar el archivo actualizado “*temporal.xml*” al nombre del archivo original “*\$viejo*”.

mv- comando unix para renombrar un archivo.

Una vez terminado este procedimiento, cuando el alumno entra al curso, ve la notificación que cierto archivo del curso ha sido actualizado (ver Figura 21).



Figura 21. BD Notificación de Actualización de Archivo.

Al acceder al link de la notificación del archivo actualizado se manda al alumno a una página la cual se modificó para agregar estas funciones (“/moodle/mod/resource/locallib.php”), la cual contiene la opción de descargar el nuevo archivo o subirlo para hacer la actualización de sus notas como se muestra en la Figura 22.

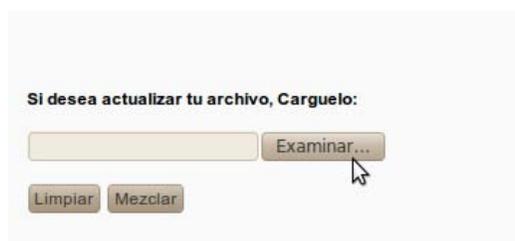


Figura 22. Examinar archivo del alumno.

En esta parte se implementaron las siguientes etiquetas, las cuales le dan la opción al alumno de examinar por su archivo (ver Figura 23) para así poder subirlo.

```
<form method="POST" action="cargando.php" enctype="multipart/form-data">
<input type="file" name="43rchive">
<input type="hidden" name="id_archivo" id="id_archivo" value="<?php echo $filename2;?>">
<br><br><input type="reset" value="Limpiar">
<input type="submit" name="continuar" value="Mezclar">
</form>
```

Donde:

cargando.php es donde se encuentra la función encargada de subir el archivo al servidor.

`$filename2` es una variable con el "id" del archivo modificado del profesor, la cual se manda a la función que está en "`cargando.php`", esta variable la obtenemos con una función llamada "`get_id()`".

Como siguiente paso, cuando se le da click al botón "`Mezclar`" (Figura 23) se sube el archivo al servidor, ejecutándose así, el algoritmo de comparación explicado en la sección anterior de la actualización del profesor.

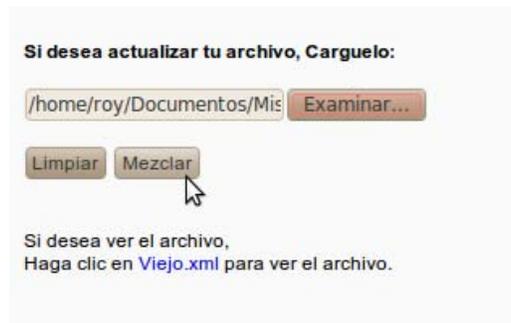


Figura 23. Botón Mezclar del alumno.

Al realizarse esto, se manda una notificación (ver Figura 24) de que la actualización de las notas del alumno se realizó satisfactoriamente.



Figura 24. Pop Actualización de Notas del alumno.

Como último paso el alumno puede descargar el archivo y guardarlo en la ruta que desee de su PC como se muestra en la Figura 25 y Figura 26.

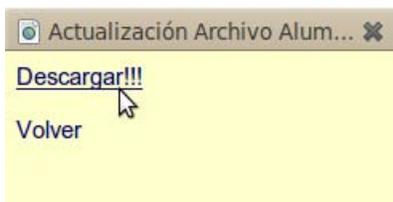


Figura 25. Descargar archivo.

En la parte de la descarga fue necesario implementar una función para forzar la descarga, ya que si no, simplemente el archivo se abre como una página de internet, la función es la siguiente:

```
<a href="descargar.php">Descargar!!!</a>
```

Donde “*descargar.php*” es el archivo donde se encuentra la función para hacer la descarga.

Función descarga:

```
$rutafinal = "../temporalalumno/alumno.xml";  
header("Content-type: application/force-download");  
header("Content-Disposition: attachment; filename=\"$rutafinal\\n");  
$fp=fopen("$rutafinal", "r");  
fpassthru($fp);
```

Donde “*\$rutafinal*” es la ruta donde se encuentra el archivo actualizado.



Figura 26. Elegir ruta de descarga.

Capítulo 9

Resultados

En este capítulo se ven los resultados de la herramienta de mezcla implementada sobre la plataforma de Moodle. A continuación se muestra la parte de la modificación del archivo del profesor.

En este ejemplo primero tenemos el archivo que el profesor subió originalmente (ver Figura 27).

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.4"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <servlet>
    <servlet-name>action</servlet-name>
    <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
    <init-param>
      <param-name>action</param-name>
      <param-value>9</param-value>
    </init-param>
    <init-param>
      <param-name>config</param-name>
      <param-value>/WEB-INF/struts-config.xml</param-value>
    </init-param>
    <init-param>
      <param-name>debug</param-name>
      <param-value>3</param-value>
    </init-param>
    <init-param>
      <param-name>detail</param-name>
      <param-value>3</param-value>
    </init-param>
    <load-on-startup>0</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.do</url-pattern>
  </servlet-mapping>
  <context-param>
    <param-name>driver</param-name>
    <param-value>com.mysql.jdbc.Driver</param-value>
  </context-param>
  <context-param>
    <param-name>cadenaCon</param-name>
    <param-value>jdbc:mysql://localhost/practica03</param-value>
  </context-param>
</web-app>
```

Figura 27. “Archivo viejo” del profesor

Como rectángulos negros tenemos marcadas las diferencias entre la Figura 27 y la Figura 28, que corresponden al “Archivo viejo” y al que contiene modificaciones del profesor respectivamente (Archivo nuevo).

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.4"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <servlet>
    <servlet-name>Accion</servlet-name>
    <servlet-class>org.apache.struts.action.DataBase.ActionServlet</servlet-class>
    <init-param>
      <param-name>config</param-name>
      <param-value>WEB-INF/struts-config.xml</param-value>
    </init-param>
    <init-param>
      <form-beans>
        <form-bean name="ProcesaSeleccionForm" type="ts.struts.beans.ProcesaSeleccionForm"></form-bean>
        <form-bean name="ResultadoForm" type="ts.struts.beans.ResultadoForm"></form-bean>
      </form-beans>
      <param-name>debug</param-name>
      <param-value>5</param-value>
    </init-param>
    <init-param>
      <param-name>detail</param-name>
      <param-value>8</param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.do</url-pattern>
  </servlet-mapping>

  <action-mappings>
    <action path="/procesa" name="ProcesaSeleccionForm" scope="request" type="ts.struts.servlets.ProcesaSeleccionAction
    <forward name="resultados" path="/paginas/resultado.jsp"></forward>
    <forward name="noResults" path="/paginas/noResultado.jsp"></forward>
  </action>
  </action-mappings>

  <context-param>
    <param-name>driver</param-name>
    <param-value>com.mysql.jdbc.Driver</param-value>
  </context-param>
  <context-param>
    <param-name>cadenaCon</param-name>
    <param-value>jdbc:mysql://localhost/practica04</param-value>
  </context-param>
</web-app>

```

Figura 28. Archivo nuevo del profesor.

Después cuando el profesor actualiza el archivo y se aplica el algoritmo que escogimos para la mezcla (*merge*), se tienen los siguientes cambios (ver Figura 29).

En los rectángulos de color rojo se pueden observar los cambios que fueron aplicados desde el “Archivo nuevo” y en los azules se observan los cambios desde el “Archivo viejo”.

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.4"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <servlet>
    <servlet-name>Accion</servlet-name>
    <servlet-class>org.apache.struts.action.DataBase.ActionServlet</servlet-class>
    <init-param>
      <param-name>action</param-name>
      <param-value>9</param-value>
    </init-param>
    <init-param>
      <param-name>config</param-name>
      <param-value>/WEB-INF/struts-config.xml</param-value>
    </init-param>
    <init-param>
      <form-beans>
        <form-bean name="ProcesaSeleccionForm" type="ts.struts.beans.ProcesaSeleccionForm"></form-bean>
        <form-bean name="ResultadoForm" type="ts.struts.beans.ResultadoForm"></form-bean>
      </form-beans>
      <param-name>debug</param-name>
      <param-value>5</param-value>
    </init-param>
    <init-param>
      <param-name>detail</param-name>
      <param-value>8</param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.do</url-pattern>
  </servlet-mapping>
  <action-mappings>
    <action path="/procesa" name="ProcesaSeleccionForm" scope="request" type="ts.struts.servlets.ProcesaSeleccionAction">
      <forward name="resultados" path="/paginas/resultado.jsp"></forward>
      <forward name="noResults" path="/paginas/noResultado.jsp"></forward>
    </action>
  </action-mappings>
  <context-param>
    <param-name>driver</param-name>
    <param-value>com.mysql.jdbc.Driver</param-value>
  </context-param>
  <context-param>
    <param-name>cadenaCon</param-name>
    <param-value>jdbc:mysql://localhost/practica04</param-value>
  </context-param>
</web-app>

```

Figura 29. Archivo actualizado del profesor.

En esta primera etapa se puede concluir que el programa funcionó correctamente así como la mezcla de ambos archivos, para así generar el “Archivo actualizado” (ver Figura 29).

En la segunda parte, tenemos la actualización de las notas del alumno, donde sus notas modificadas (ver Figura 30) desde el “Archivo viejo” del profesor (ver Figura 27) están marcadas con rectángulos negros.

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.4"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <servlet>
    <servlet-name>action</servlet-name>
    <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
    <init-param>
      <param-name>action</param-name>
      <param-value>9</param-value>
    </init-param>
    <init-param>
      <param-name>config</param-name>
      <param-value>/WEB-INF/struts-config.xml</param-value>
    </init-param>
    <init-param>
      <param-name>debug</param-name>
      <param-value>3</param-value>
    </init-param>
    <init-param>
      <param-name>detail</param-name>
      <param-value>3</param-value>
    </init-param>
  </servlet>

  <form-validation>
    <global>
      <!-- opciones de validación-->
      <validator name="required"
        classname="org.apache.struts.validator.FieldChecks"
        method="validateRequired"
        methodParams="java.lang.Object,
          org.apache.commons.validator.ValidatorAction,
          org.apache.commons.validator.Field,
          org.apache.struts.action.ActionMessages,
          org.apache.commons.validator.Validator,
          javax.servlet.http.HttpServletRequest"
        msg="errors.required"/>
    </global>
  </form-validation>
  <load-on-startup>0</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>action</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>

  <context-param>
<param-name>driver</param-name>
<param-value>com.mysql.jdbc.Driver</param-value>
</context-param>
<context-param>
<!-- la practica que se ocupara esta guardada en el localhost en practica 03-->
<param-name>cadenaCon</param-name>
<param-value>jdbc:mysql://localhost/practica03</param-value>
</context-param>
</web-app>

```

Figura 30. Notas del alumno.

Finalmente después de aplicar la mezcla en las notas del alumno con el “Archivo actualizado” del profesor, se obtiene el “Archivo actualizado del alumno” (ver Figura 31), donde los rectángulos rojos son las modificaciones del “Archivo actualizado” del profesor y los azules indican lo de las notas originales del alumno.

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.4"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <servlet>
    <servlet-name>Accion</servlet-name>
    <servlet-class>org.apache.struts.action.DataBase.ActionServlet</servlet-class>
    <init-param>
      <param-name>action</param-name>
      <param-value>9</param-value>
    </init-param>
    <init-param>
      <param-name>config</param-name>
      <param-value>/WEB-INF/struts-config.xml</param-value>
    </init-param>
    <init-param>
      <form-beans>
        <form-bean name="ProcesaSeleccionForm" type="ts.struts.beans.ProcesaSeleccionForm"></form-bean>
        <form-bean name="ResultadoForm" type="ts.struts.beans.ResultadoForm"></form-bean>
      </form-beans>
      <param-name>debug</param-name>
      <param-value>5</param-value>
    </init-param>
    <init-param>
      <param-name>detail</param-name>
      <param-value>8</param-value>
    </init-param>
  </servlet>
  <form-validation>
    <global>
      <!-- opciones de validación-->
      <validator name="required"
        classname="org.apache.struts.validator.FieldChecks"
        method="validateRequired"
        methodParams="java.lang.Object,
          org.apache.commons.validator.ValidatorAction,
          org.apache.commons.validator.Field,
          org.apache.struts.action.ActionMessages,
          org.apache.commons.validator.Validator,
          javax.servlet.http.HttpServletRequest"
        msg="errors.required"/>
    </global>
  </form-validation>
  <load-on-startup>0</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>action</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>
<action-mappings>
  <action path="/procesa" name="ProcesaSeleccionForm" scope="request" type="ts.struts.servlets.ProcesaSeleccionAction">
    <forward name="resultados" path="/paginas/resultado.jsp"></forward>
    <forward name="noResults" path="/paginas/noResultado.jsp"></forward>
  </action>
</action-mappings>
  <context-param>
    <param-name>driver</param-name>
    <param-value>com.mysql.jdbc.Driver</param-value>
  </context-param>
  <context-param>
    <!-- la practica que se ocupara esta guardada en el localhost en practica 03-->
    <param-name>cadenaCon</param-name>
    <param-value>jdbc:mysql://localhost/practica04</param-value>
  </context-param>
</web-app>

```

Figura 31. Archivo actualizado del Alumno.

Conclusiones y crecimiento del proyecto

De la presente investigación se desprenden una serie de conclusiones relevantes no ya tan sólo para entender el papel de la educación en línea. De hecho, si se revisa la evolución de los planteamientos sobre comparación y actualización de archivos, se podría ver una aplicación la cual se agregó a la plataforma Moodle, dándole al estudiante como resultado que siempre esté actualizado en los cambios que hace el profesor.

Se diseñó una aplicación para la plataforma Moodle que logró la comparación de dos archivos de notas de profesor, encontrando similitudes para incluir las anotaciones del alumno en un nuevo archivo. Propiciando el mejoramiento de las notas, y mayor aprovechamiento de este proyecto.

Se utilizó una herramienta unix llamada "Merge". Posteriormente se adaptó dicha herramienta a Moodle.

También se creó un módulo para dar a conocer a los alumnos los cambios y / o actualizaciones de los archivos, esto se considera que puede dar como resultado agilizar el proceso para el aprendizaje de los alumnos, propiciando una retroalimentación para el profesor. Así pudiendo crear un mejor ambiente en el cual el alumno se podrá desarrollar de una manera íntegra obteniendo conocimientos de otros compañeros y profesores.

Este proyecto abarcó diversos factores del aprendizaje como son tecnología, intercambio de ideas y agilidad en el proceso. Brindará facilidades para todas aquellas instituciones que usen este sistema, ya que se beneficiarán de todos los servicios de actualización y comparación de archivos.

El tipo de aplicación que se utilizó va en constante crecimiento, por lo que queda abierto el proyecto a cualquier tipo de modificación en un futuro, como por ejemplo agregar una aplicación que convierta archivos de "Word", "Power Point" y "PDF" (que son los más utilizados) a "XML" para poder hacer la respectiva actualización de archivos.

Glosario de Términos

XML: Por sus siglas en inglés de *eXtensible Markup Language* (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Permite definir la gramática de lenguajes específicos.

GNU: La Licencia Pública General (*General Public License*), está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Apache: Servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.)

MySQL: Sistema relacional, multihilo y multiusuario con más de seis millones de instalaciones.

PHP: Lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

LAMP: Se refiere a un conjunto de subsistemas de software necesarios para alcanzar una solución global, en este caso configurar sitios web o servidores dinámicos con un esfuerzo reducido.

Moodle: es un Ambiente Educativo Virtual, sistema de gestión de cursos, de distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea. Este tipo de plataformas tecnológicas también se conoce como LMS (Learning Management System).

UML: *es un conjunto de herramientas, que permite modelar (analizar y diseñar) sistemas orientados a objetos.*

E-learning: Aprendizaje asistido por tecnologías de la información y la comunicación (TIC). Fomenta el uso intensivo de las TIC facilitando la creación, adopción y distribución de contenidos, así como la adaptación del ritmo de aprendizaje y disponibilidad de las herramientas de aprendizaje independientemente de límites horarios o geográficos. Permitiendo al alumno intercambiar opiniones y aportes a través de las TIC.

B-learning: (Blended Learning), se traduce como “Formación combinada” o “Enseñanza mixta”. Se trata de la modalidad semipresencial de estudios que incluye tanto formación no presencial como presencial.

CMS (Course Management System): conjunto de herramientas de software que de manera integral proveen un espacio en línea para facilitar el aprendizaje y la enseñanza.

Referencias Electrónicas y Bibliográficas

[1] Tesina: "MOODLE como plataforma para la enseñanza virtual en el Instituto Politécnico Nacional" realizado en Junio del 2009 por *Barquera García Karen Melanny*.

[2] <http://docs.moodle.org/es/>

[3] <http://www.openformats.org/esShowAll>

[4] http://www.oxygenxml.com/xml_diff_and_merge.html

[5] <http://kdiff3.sourceforge.net/>

[6] <http://www.altova.com/diffdog/xml-diff.html>

[7] <http://www.alphaworks.ibm.com/tech/xmldiffmerge>

[8] <http://tdm.berlios.de/3dm/doc/index.html>

[9] <http://www.topologi.com/diffx/>

[10] http://linux.about.com/library/cmd/blcmdl1_diff.htm

[11] <http://www.guia-ubuntu.org/index.php?title=PhpMyAdmin>

[12] http://docs.moodle.org/es/Instalaci%C3%B3n_de_moodle

[13] http://72.15.209.135/nav.html?_functions/index.html

[14] "Creación de un portal con PHP y MySQL", 3ª Edición, Autor: Jacobo Pavón Puertas, Editorial: Alfaomega Ra-Ma.

[15] <http://www.php.net/manual/es/index.php>

[16] <http://docs.moodle.org/en/Development>

[17] <http://moodle.org/mod/forum/>

[18] <http://www.desarrolloweb.com/faq/89.php>

[19] <https://csimoodle.ucd.ie/moodle/login/index.php>

[20] http://www.dimagin.net/es/tec_lamp.php

[21] www.ibm.com/software/rational/uml

[22] <http://www.larevistainformatica.com/que-es-enciptacion-informatica.htm>

[23] http://www.itesm.mx/va/innovacioneducativa/01/1_4.htm

Manual de instalación y uso

“Aplicación para el sistema moodle que actualice las notas de un alumno con los cambios de un archivo del profesor”

Alumno:

Vázquez Cabal Rodrigo Antonio
206306692

Trimestre 11-I



Índice de contenido

Índice de contenido.....	2
Introducción	3
Requerimientos	3
Características de la PC	3
Instalación de Moodle.....	4
Instalación de MySQL.....	4
Instalación de Apache	4
Instalación de PHP.....	5
Obtener soporte de MySQL para PHP.....	6
Instalación de phpMyAdmin	7
Instalación de Moodle 2.0.....	7
Configuración de Moodle.....	8
Uso de la herramienta de actualización.....	9
Glosario de Términos	16
Referencias Electrónicas y Bibliográficas	17

Introducción

En la vida estudiantil se presentan diversos tipos de aprendizaje, por mencionar uno de ellos tenemos la “educación en línea”, la cual tiene como patrón común utilizar una plataforma de “enseñanza-aprendizaje”. Cada vez se usa más éste método en varias universidades, para cursar materias, incluso para estudiar carreras y maestrías completas en línea. Diversas universidades ocupan la plataforma Moodle para realizar este tipo de enseñanza.

Por esta razón este proyecto propone la comparación y actualización de archivos, una aplicación la cual se agregará a la plataforma Moodle, dándole al estudiante como resultado que siempre esté actualizado en los cambios que hace el profesor, así como también agilizar el trabajo de los profesores ahorrando tiempo.

Requerimientos

Características de la PC

Se contó con una Laptop, la cual tiene como características técnicas las siguientes:

- Procesador Intel® core™2 CPU 1.73 GHz.
- Memoria RAM de 512MB.
- SATA 4GB.
- Linux, Windows, Mac.

Instalación de Moodle

Se instaló la versión más reciente de moodle (2.02) en Ubuntu 10.04.

Para poder instalar Moodle 2.02 fue necesario instalar primero la base de datos de MySQL 5, el servidor Apache 2 y phpMyAdmin.

A continuación se detallarán todos los pasos a seguir para poder instalar la plataforma de Moodle, primeramente entramos con permisos de administrador desde la Terminal para poder realizar las instalaciones.

- *sudo su* ->Comando para entrar como administrador.

Instalación de MySQL

Posteriormente se instaló MySQL 5 con el siguiente comando:

```
apt-get install mysql-server mysql-client
```

Se nos pide que demos nuestra contraseña para el usuario root la cual es válida para el usuario root@localhost.

Instalación de Apache

Después de instalar MySQL se procedió a instalar el Apache 2, el cual está disponible como un paquete de Ubuntu package, así pues, se puede instalar con el siguiente comando:

```
aptitude install apache2
```

Ahora abrimos un navegador de internet con la dirección “http://localhost”, y se logra observar en la Figura 1 que el Apache2 funciona correctamente.



Figura 1. Funcionamiento de Apache.

La documentación en Ubuntu de Apache se genera por default en la ruta:

```
/var/www
```

Instalación de PHP

Instalamos PHP5 y el módulo de PHP para Apache como sigue:

```
apt-get install php5 libapache2-mod-php5
```

Posteriormente reiniciamos Apache2:

```
/etc/init.d/apache2 restart
```

Para probar el funcionamiento correcto de PHP5 creamos el archivo “info.php” en la siguiente ruta donde está el servidor de Apache:

```
vi /var/www/info.php
```

El archivo debe contener esta información:

```
<?php  
phpinfo();  
?>
```

Ahora en el navegador llamamos al archivo con la dirección “<http://localhost/info.php>” y se logra observar la información de php que se instaló (ver Figura 2).



Figura 2. Información de PHP.

Obtener soporte de MySQL para PHP

Es necesario instalar varios paquetes para el correcto funcionamiento de MySQL en php, así que se instalaron los siguientes:

```
Apt-get install php5-mysql php5-curl php5-gd php5-idn php-pear php5-imagick
php5-imap php5-mcrypt php5-memcache php5-mhash php5-ming php5-ps php5-
pspell php5-recode php5-snmp php5-sqlite php5-tidy php5-xmlrpc php5-xsl php5-
json
```

Reiniciamos Apache y para verificar que se instalaron correctamente ejecutamos info.php y notamos que el módulo de MySQL ya aparece (ver Figura 3).

mysql

MySQL Support	enabled
Active Persistent Links	0
Active Links	0
Client API version	5.1.41
MYSQL_MODULE_TYPE	external
MYSQL_SOCKET	/var/run/mysqld/mysqld.sock
MYSQL_INCLUDE	-I/usr/include/mysql
MYSQL_LIBS	-L/usr/lib -lmysqlclient_r

Figura 3. Paquete de MySQL para PHP.

Instalación de phpMyAdmin

PhpMyAdmin es una interface web por medio de la cual puedes manejar las bases de datos de MySQL, así que procedemos a instalarlo como una opción para darse una idea de lo que se va creando y/o modificando[11].

Utilizamos el siguiente comando:

```
sudo apt-get install phpmyadmin
```

Después de instalarlo hay que hacer un enlace simbólico desde `usr/share/phpmyadmin` hacia `/var/www/phpmyadmin` con la instrucción :

```
sudo ln -s /usr/share/phpmyadmin /var/www
```

Luego entonces podemos acceder a phpMyAdmin con “`http://localhost/phpmyadmin/`” en el navegador (ver Figura 4).

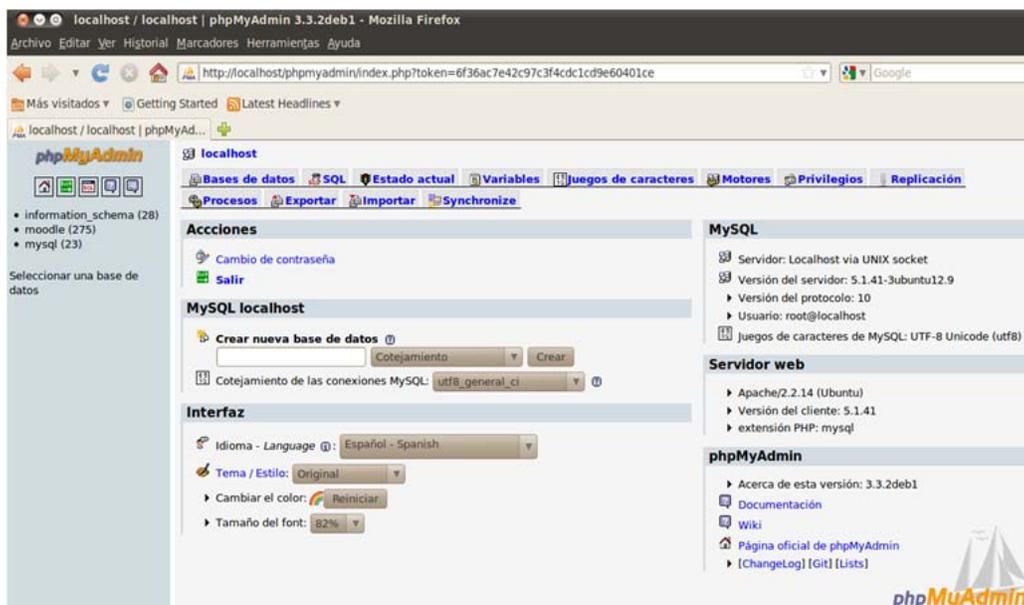


Figura 4. Interfaz phpMyAdmin.

Instalación de Moodle 2.0

Finalmente se prosigue con la instalación de Moodle descargándolo desde <http://download.moodle.org/>

Descargamos el archivo de Moodle para Linux, lo descomprimos y seguimos los pasos de instalación[12].

Copiamos la carpeta *moodle*, */var/www/* donde se encuentra instalado nuestro servidor Apache.

Configuración de Moodle

Moodle almacena los archivos subidos por los usuarios en una carpeta que debe llamarse *moodledata*. Dicha carpeta sólo debe ser accesible para el servidor y para ningún otro usuario de la red. Por esta razón vamos a crear dicha carpeta en, por ejemplo, */var/moodledata/filedir/*

En segundo lugar editaremos, con el bloc de notas o cualquier otro editor de texto plano, el archivo de muestra *config-dist.php* y lo guardaremos con el nombre *config.php*, después de haber introducido correctamente los siguientes parámetros de configuración de la base de datos:

```
$CFG->dbtype = 'mysql'; Tipo de base de datos: MySQL  
$CFG->dbhost = 'localhost'; Nombre del servidor de la base de datos  
$CFG->dbname = 'tecnologia'; Nombre de la base de datos  
$CFG->dbuser = 'root'; Usuario  
$CFG->dbpass = ''; Contraseña  
$CFG->dbpersist = true; ¿Deben reutilizarse las conexiones?  
$CFG->prefix = 'mdl_'; Prefijo para todos los datos
```

También debemos introducir, en el archivo *config.php*, las rutas correctas:

```
$CFG->wwwroot = 'http://localhost/moodle';  
$CFG->dirroot = 'C:\Archivos de programa\EasyPHP1-8\www\moodle';  
$CFG->dataroot = 'C:\moodledata';
```

Al finalizar la instalación se puede acceder a Moodle con la dirección "*http://localhost/moodle*".

Uso de la herramienta de actualización

Con el apoyo de la página web que contiene el código documentado de Moodle [13], en formato HTML se pudo encontrar el archivo php a modificar para poder implementar la herramienta de mezcla.

El módulo que se presenta en esta sección es el “File manager”, ea cual contiene las funciones de agregar los archivos, eliminarlos, modificarlos, así como también agregar directorios, entre otras cosas.

La ruta del archivo modificado es “/var/www/moodle/lib/form/filemanager.php” el cual tiene funciones para la obtención de ciertas características de los archivos y del cliente, la mayoría utilizando funciones de manejadores de archivos definidas en “/var/www/moodle/lib/filestorage/*.php”.

Debido a la gran complejidad de manipular el código y de las tantas clases y funciones anidadas, se optó por hacer una función con sentencias MySQL [14] para poder obtener la ruta encriptada del archivo por medio del “itemid” que la misma clase de filemanager.php proporciona.



Figura 5. Acceso a la interfaz de Actualizar.

Posteriormente en la sección de “Contenido” se agrega el nuevo archivo para actualizar (ver Figura 6).



Figura 6. Botón agregar.

Una vez que se agrega el archivo podemos ver en la Base de Datos de “phpMyAdmin” que ha sido agregado y encriptado (ver Tabla 1).

id	contenthash	pathnamehash	contextid	component	filearea	itemid	filepath	filename
1033	fe80fa9496d5906d40d2c00256d6bba7de6d9fae	09e0cdef1252b53187ff1f69150b0851eb0f2ef1	13	user	draft	790099003	/	Viejo.xml
1034	da39a3ee5e6b4b0d3255bfe95601890afd80709	1792e3a76142200cfb2430ec58d2ec4019dc7e2f	13	user	draft	790099003	/	.
1037	fe80fa9496d5906d40d2c00256d6bba7de6d9fae	74df1874d1552588f44443b85153a9c6532be08e	13	user	draft	733511235	/	Viejo.xml
1038	da39a3ee5e6b4b0d3255bfe95601890afd80709	818dcaa9d14a43c8e5a954bbc4f4c70cdc862bd8	13	user	draft	733511235	/	.
1039	fe80fa9496d5906d40d2c00256d6bba7de6d9fae	3144cc18154fb14528e6ee39e122794a673b0e00	37	mod_resource	content	0	/	Viejo.xml
1040	da39a3ee5e6b4b0d3255bfe95601890afd80709	4b45d0de29a43a9680736186a102bc32795456b9	37	mod_resource	content	0	/	.
1041	fe80fa9496d5906d40d2c00256d6bba7de6d9fae	14e237ea9b06a92f0caf3b6e0b3541451184cfb2	13	user	draft	986986037	/	Viejo.xml
1042	da39a3ee5e6b4b0d3255bfe95601890afd80709	5398482b515bcc72e404ef11e242382552f7ea60	13	user	draft	986986037	/	.
1043	fe80fa9496d5906d40d2c00256d6bba7de6d9fae	b08177e4d7e67cdc1e8300c816be0c21b895b794	13	user	draft	319098956	/	Viejo.xml
1044	da39a3ee5e6b4b0d3255bfe95601890afd80709	87546cd3c939ce3d9e4993bbcc94e8a3dced299	13	user	draft	319098956	/	.
1045	68d666d2e3e63c35a931502a6080e98d8e8d75bd	953f7d1aacc87f05cac141ca304dd25013b5daf1	13	user	draft	319098956	/	Nuevo.xml

Tabla 1. BD mdl_files con el nuevo archivo.

Como se logra observar el “itemid” (marcado con amarillo en la Tabla 3) que se utiliza para sacar el nombre del archivo es el mismo, en este caso definido con el número “319098956” (columna itemid de la Tabla 3).

Para poder obtener el valor “itemid” llamamos a la siguiente función anteriormente creada:

```
$itemid = $this->getValue();
```

Donde *getValue()* es una función la cual nos regresa el valor de itemid.

Una vez obtenido el valor “itemid” se utilizó la siguiente secuencia MySQL para poder obtener el “contenthash” y el “id” (ver Tabla 3) del viejo archivo:

```
SELECT contenthash, id FROM mdl_files WHERE itemid=$itemid
```

Así pues, obtenemos la ruta y el “id” del viejo archivo, con este podemos obtener la ruta del segundo archivo con la siguiente secuencia MySQL:

```
SELECT contenthash FROM mdl_files WHERE id=$id_nuevo
```

Por consiguiente ya tenemos ambas rutas, del viejo archivo a actualizar y del nuevo.

Una vez teniendo estos importantes datos, procedemos a hacer la actualización del archivo haciendo click en el botón “Mezcla” que se agregó para realizar dicha función (ver Figura 7).

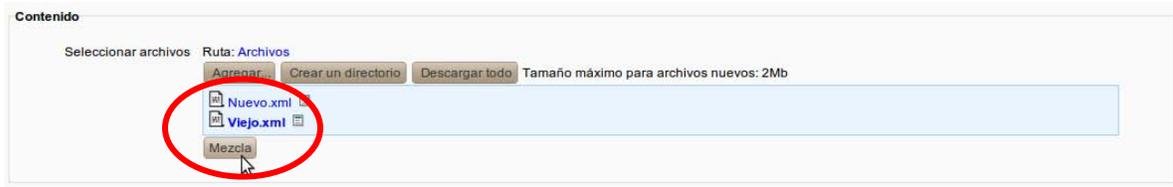


Figura 7. BD mdl_files con el nuevo archivo.

El botón “Mezcla” está creado de la siguiente manera:

```
<form onSubmit="file_manager($itemid)">
<input type="submit" name="mezclar" value = "Mezcla"></form>
```

Donde “file_manager(\$itemid)” es la función que se ejecuta para tener acceso a la base de datos junto con la herramienta de mezcla de archivos.

Se implementa el siguiente algoritmo para hacer la actualización del archivo:

1.- Se crea y se abre un archivo temporal para que se guarden los cambios.

```
$archivo="/var/moodledata/filedir/$carpeta_viejo1/$carpeta_viejo2/temporal.xml";
```

```
$id = fopen($archivo,'w+');
```

2.- Se define la siguiente sentencia con la ruta del viejo, nuevo y archivo temporal:

```
$e = "/var/moodledata/filedir/$carpeta_viejo1/$carpeta_viejo2/temporal.xml
/var/moodledata/filedir/$carpeta_viejo1/$carpeta_viejo2/$viejo
/var/moodledata/filedir/$carpeta_nuevo1/$carpeta_nuevo2/$nuevo";
```

Donde:

\$carpeta_viejo1 y \$carpeta_nuevo1 son las carpetas principales donde se encuentran las carpetas contenedoras del archivo viejo y el archivo nuevo respectivamente.

\$carpeta_viejo2 y \$carpeta_nuevo2 son las subcarpetas donde se encuentran los archivos a mezclar.

\$viejo y \$nuevo son los nombres de los archivos a mezclar.

3.- Ocupamos la función “system()” para hacer llamado a los comandos unix y escribimos el comando de actualización:

```
system('merge -q '.escapshellcmd($e), $retval);
```

Donde:

merge es el comando que hace la mezcla.

escapshellcmd() es un comando unix que se utiliza para omitir cualquier caracter en una cadena que pueda ser usado para engañar a un comando de shell a que ejecute comandos arbitrarios. Esta función puede ser usada para asegurarse de que cualquier dato viniendo de la entrada del usuario sea pasado desapercibido antes de que este dato sea pasado a la función *system()* [15];

\$e es la ruta anteriormente definida.

\$retval es donde se guarda la última línea de ejecución del comando “system()”, la cual para los fines de este proyecto no nos es de utilidad.

4.- Renombramos el archivo actualizado con el nombre del archivo original y borramos el archivo sobrante:

```
$f= "/var/moodledata/filedir/$carpeta_viejo1/$carpeta_viejo2/$viejo";
```

```
system('rm '.escapshellcmd($f),$retval);
```

```
$g="/var/moodledata/filedir/$carpeta_viejo1/$carpeta_viejo2/temporal.xml  
/var/moodledata/filedir/$carpeta_viejo1/$carpeta_viejo2/$viejo";
```

```
system('mv '.escapshellcmd($g),$retval);
```

Donde:

\$f es la ruta del viejo archivo, el cual vamos a eliminar.

rm- es un comando unix para borrar un archivo.

\$g secuencia que se usa para renombrar el archivo actualizado “*temporal.xml*” al nombre del archivo original “*\$viejo*”.

mv- comando unix para renombrar un archivo.

Una vez terminado este procedimiento, cuando el alumno entra al curso, ve la notificación que cierto archivo del curso ha sido actualizado (ver Figura 8).

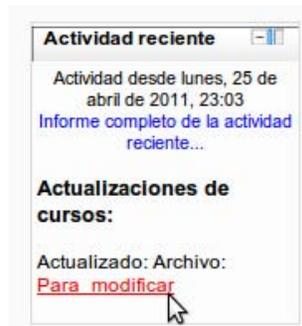


Figura 8. BD Notificación de Actualización de Archivo.

Al acceder al link de la notificación del archivo actualizado se manda al alumno a una página la cual se modificó para agregar estas funciones ("*/moodle/mod/resource/locallib.php*"), la cual contiene la opción de descargar el nuevo archivo o subirlo para hacer la actualización de sus notas como se muestra en la Figura 9.

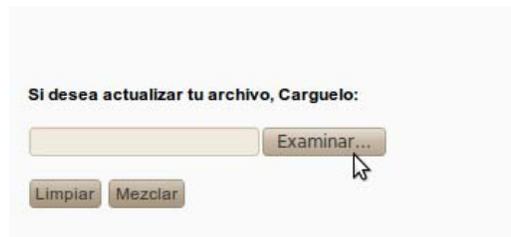


Figura 9. Examinar archivo del alumno.

En esta parte se implementaron las siguientes etiquetas, las cuales le dan la opción al alumno de examinar por su archivo (ver Figura 10) para así poder subirlo.

```
<form method="POST" action="cargando.php" enctype="multipart/form-data">
<input type="file" name="13rchive">
<input type="hidden" name="id_archivo" id="id_archivo" value="<?php echo $filename2;?>">
<br><br><input type="reset" value="Limpiar">
<input type="submit" name="continuar" value="Mezclar">
</form>
```

Donde:

cargando.php es donde se encuentra la función encargada de subir el archivo al servidor.

\$filename2 es una variable con el "id" del archivo modificado del profesor, la cual se manda a la función que está en "*cargando.php*", esta variable la obtenemos con una función llamada "*get_id()*".

Como siguiente paso, cuando se le da click al botón “Mezclar” (Figura 10) se sube el archivo al servidor, ejecutándose así, el algoritmo de comparación explicado en la sección anterior de la actualización del profesor.

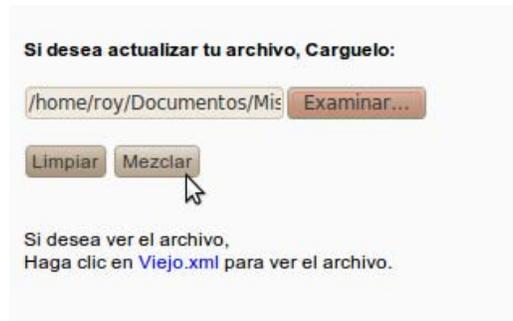


Figura 10. Examinar archivo del alumno.

Al realizarse esto, se manda una notificación (ver Figura 11) de que la actualización de las notas del alumno se realizó satisfactoriamente.

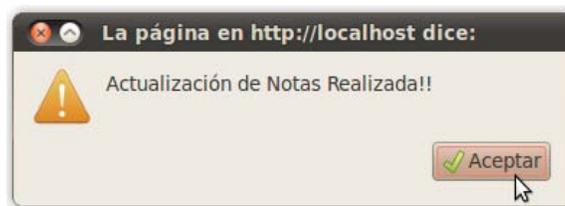


Figura 11. Examinar archivo del alumno.

Como último paso el alumno puede descargar el archivo y guardarlo en la ruta que desee de su PC como se muestra en la Figura 12 y Figura 13.

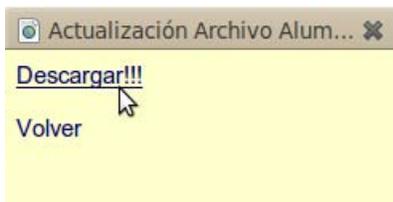


Figura 12. Examinar archivo del alumno.

En la parte de la descarga fue necesario implementar una función para forzar la descarga, ya que si no, simplemente el archivo se abre como una página de internet, la función es la siguiente:

```
<a href="descargar.php">Descargar!!!</a>
```

Donde “descargar.php” es el archivo donde se encuentra la función para hacer la descarga.

Función descarga:

```
$rutafinal = "../temporalalumno/alumno.xml";  
header("Content-type: application/force-download");  
header("Content-Disposition: attachment; filename=\"$rutafinal\"");  
$fp=fopen("$rutafinal", "r");  
fpassthru($fp);
```

Donde “\$rutafinal” es la ruta donde se encuentra el archivo actualizado.



Figura 13. Examinar archivo del alumno.

Glosario de Términos

XML: Por sus siglas en inglés de *eXtensible Markup Language* (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Permite definir la gramática de lenguajes específicos.

GNU: La Licencia Pública General (*General Public License*), está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Apache: Servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.)

MySQL: Sistema relacional, multihilo y multiusuario con más de seis millones de instalaciones.

PHP: Lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

LAMP: Se refiere a un conjunto de subsistemas de software necesarios para alcanzar una solución global, en este caso configurar sitios web o servidores dinámicos con un esfuerzo reducido.

Referencias Electrónicas y Bibliográficas

- [1] Tesina: “MOODLE como plataforma para la enseñanza virtual en el Instituto Politécnico Nacional” realizado en Junio del 2009 por *Barquera García Karen Melanny*.
- [2] <http://docs.moodle.org/es/>
- [3] <http://www.openformats.org/esShowAll>
- [4] http://www.oxygenxml.com/xml_diff_and_merge.html
- [5] <http://kdiff3.sourceforge.net/>
- [6] <http://www.altova.com/diffdog/xml-diff.html>
- [7] <http://www.alphaworks.ibm.com/tech/xmldiffmerge>
- [8] <http://tdm.berlios.de/3dm/doc/index.html>
- [9] <http://www.topologi.com/diffx/>
- [10] http://linux.about.com/library/cmd/blcmdl1_diff.htm
- [11] <http://www.guia-ubuntu.org/index.php?title=PhpMyAdmin>
- [12] http://docs.moodle.org/es/Instalaci%C3%B3n_de_moodle
- [13] http://72.15.209.135/nav.html?_functions/index.html
- [14] “Creación de un portal con PHP y MySQL”, 3ª Edición, Autor: Jacobo Pavón Puertas, Editorial: Alfaomega Ra-Ma.
- [15] <http://www.php.net/manual/es/index.php>
- [16] <http://docs.moodle.org/en/Development>
- [17] <http://moodle.org/mod/forum/>
- [18] <http://www.desarrolloweb.com/faq/89.php>
- [19] <https://csimoodle.ucd.ie/moodle/login/index.php>