

UNIVERSITAT OBERTA DE CATALUNYA

Estudios de Ingeniería en Informática

Trabajo Final de Carrera
Aplicaciones web para trabajo colaborativo

**Aplicación web para la corrección
automática de pruebas**

Memoria

Última actualización: 27/05/2012

Alumno:
David Cabrera Rodríguez

Dirigido por Ferran Prados Carrasco

Curso 2011/2012-2

Índice de contenido

1.Introducción.....	3
1.1 Justificación.....	3
1.2 Objetivos.....	3
1.3 Motivación.....	4
2 Organización del proyecto.....	4
2.1 Relación de tareas.....	4
2.2 Planificación temporal.....	5
2.3 Metodología de trabajo y herramientas utilizadas.....	6
2.4 Presupuesto.....	6
3 Marco de trabajo y conceptos previos.....	7
3.1 Definición de las necesidades docentes.....	7
3.2 Análisis de herramientas ya existentes.....	10
4 Especificación y análisis.....	15
4.1 Análisis funcional de la aplicación.....	15
4.2 Requisitos del sistema.....	16
4.3 Arquitectura.....	17
4.3.1 Diagrama de actividad.....	20
4.3.2 Diagrama de secuencia.....	21
4.3.3 Estructura de datos y carpetas.....	22
4.4 Interfaz de usuario.....	24
5 Implementación y pruebas.....	31
5.1 Pruebas de integración.....	32
5.2 Pruebas de aceptación.....	33
6 Implantación y resultados.....	34
7 Conclusiones.....	35
8 Trabajo futuro.....	35
9 Bibliografía.....	36
10 Anexo.....	36

1.Introducción

1.1 Justificación

El Trabajo final de carrera (TFC) es una asignatura que está pensada para realizar un trabajo de síntesis de los conocimientos adquiridos en otras asignaturas de la carrera y que requiera ponerlos en práctica. Normalmente el TFC es un trabajo eminentemente práctico y vinculado al ejercicio profesional de la informática aunque en algunos casos puede ser, o incluir, un trabajo de investigación.

Este TFC está enmarcado dentro del departamento de Aplicaciones Web para el trabajo colaborativo que fija sus estudios en las nuevas tecnologías aplicadas al trabajo en grupo dentro de cualquier tipo de organización, con el objetivo claro de facilitarnos el trabajo.

Este trabajo pretende desarrollar una herramienta web que permita corregir pruebas de forma automática. Surge ante una necesidad presente en mi actual trabajo de profesor de secundaria, donde en determinados tipos de exámenes se dedica mucho tiempo a corregir de una forma un tanto sistemática y en la que se emplea mucho tiempo.

1.2 Objetivos

Se propone crear una herramienta que haga posible al profesor la creación de pruebas y la corrección automática de estas, de forma que el proceso de evaluación sea mucho más rápido y sin ningún tipo de error humano.

Los principales objetivos a lograr serían:

- Estudiar las herramientas que existen actualmente y enumerar sus propiedades más características.
- Desarrollar una aplicación web dinámica que facilite la corrección automática de los ejercicios de los alumnos.
- Profundizar en los conocimientos generales de desarrollo de aplicaciones web haciendo servir tecnología PHP, a parte del reciclaje de los conceptos de ingeniería del software y gestión de proyectos.
- Poner en práctica todos los conocimientos y técnicas aprendidas a lo largo de la carrera sobre el desarrollo de un proyecto.

1.3 Motivación

En mi trabajo diario como docente tengo que corregir muchos ejercicios, trabajos y exámenes, algunas de estas correcciones se escapan del alcance de este proyecto ya que además de la respuesta se valora el procedimiento seguido y el razonamiento. Pero existen otras pruebas por ejemplo las conocidas como tipo test, en las que el proceso de corrección se vuelve repetitivo y además es fácil equivocarse. En estos casos, una máquina realizaría el proceso mucho más rápido y sin errores. El objetivo es realizar un aplicativo web que facilite este trabajo y que se pueda utilizar en el día a día, para reducir el trabajo del docente.

2 Organización del proyecto

2.1 Relación de tareas

Las tareas que se tienen que realizar pueden agruparse en 4 fases que casi coinciden con cada una de las PACs.

Fase 1 – Recogida y análisis de requisitos

- Definición de las necesidades docentes
 - Realizar un estudio del tipo de herramientas que pueden facilitar el trabajo de los docentes en la corrección de pruebas centrado en la educación secundaria. Se realizarán entrevistas personales o comunicación vía email. Entre el 6 y el 9 de Marzo de 2012.
- Análisis de herramientas ya existentes
 - Buscar soluciones ya implementadas que realicen la misma función y analizar sus ventajas e inconvenientes. Se realizará una búsqueda en Internet. Entre el 9 y el 14 de Marzo de 2012.
- Análisis funcional de la aplicación
 - Sacar unas conclusiones del análisis previo realizado y describir el análisis funcional de la aplicación. Entre el 14 y el 25 de Marzo de 2012.

Fase 2 – Diseño

- Diseño del interfaz
 - Realizar un prototipo del interfaz de usuario tanto con perfil profesor como con perfil alumno. Entre el 14 y el 29 de Marzo de 2012.

Fase 3 – Codificación

- Implementación del perfil profesor
 - Realizar la implementación siguiendo el análisis funcional y el interfaz propuesto. Entre el 29 de Marzo de 2012 y el 1 de Abril de 2012.
- Implementación del perfil alumno
 - Realizar la implementación siguiendo el análisis funcional y el interfaz propuesto. Entre el 9 y el 15 de Abril de 2012.

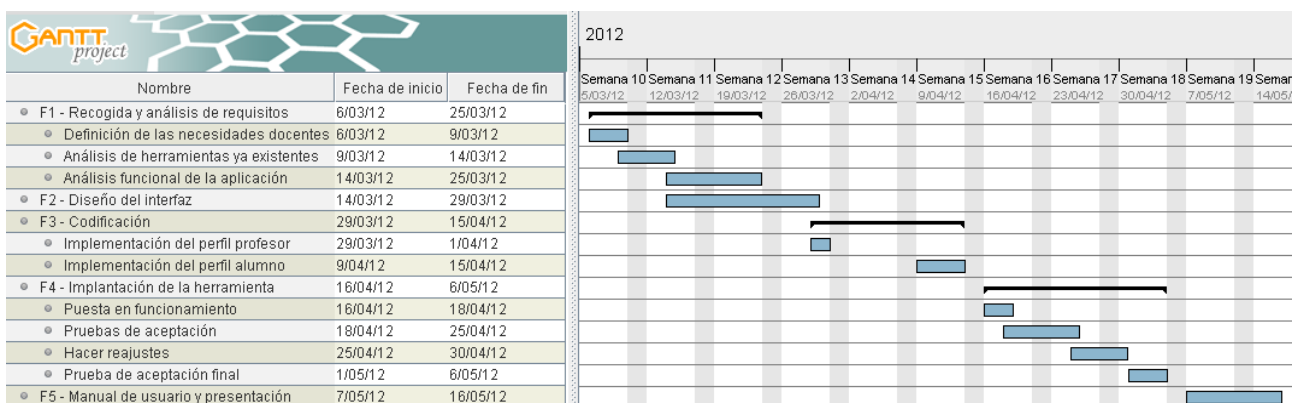
Fase 4 – Implantación de la herramienta

- Puesta en funcionamiento: Entre el 16 y el 18 de Abril de 2012.
- Pruebas de aceptación: Entre el 18 y el 25 de Abril de 2012.
- Hacer reajustes: Entre el 25 y el 30 de Abril de 2012
- Prueba de aceptación final: Entre el 1 y el 6 de Mayo de 2012

Fase 5 – Manual de Usuario y presentación

- Realizar el manual de usuario y la documentación necesaria para presentar el TFC. Entre el 7 y el 16 de Mayo de 2012.

2.2 Planificación temporal



2.3 Metodología de trabajo y herramientas utilizadas

En las primeras fases del proyecto predominará los periodos de estudio y búsqueda de información,

alternándose con periodos para ir perfilando el diseño de la aplicación.

En las fases finales, se implementará todo el diseño previo planificado y se realizaran las pruebas oportunas para poder superar las pruebas de validación.

La relación de herramientas que se utilizarán serán las siguientes:

Hardware

- Ordenador Intel 2.8 Mhz con 1 Gb de memoria RAM y 80 Gb de disco duro.

Software

- Para desarrollar la documentación del proyecto
 - Distribución Ubuntu 12.04
 - LibreOffice Writer para desarrollar la documentación escrita
 - LibreOffice Impress para desarrollar la presentación virtual
 - Gimp para hacer capturas y manipulación de imágenes
 - GanttProject para realizar los diagramas de Gantt del plan de trabajo
 - Visual Paradigm for UML (Community Edition) para realizar los diagramas UML
- Para desarrollar el aplicativo web
 - Distribución Ubuntu 12.04
 - Servidor Web Apache
 - PHP 5.3
 - Sublime Text como editor de código fuente

2.4 Presupuesto

El presupuesto de materiales necesarios para la realización de este proyecto se considera de coste cero.

Respecto a las horas de trabajo utilizadas para su realización se estima una dedicación diaria de una hora en cada una de las fases, a excepción de la fase de codificación que se emplearan dos horas diarias. Con esta distribución y un precio-hora aproximado de 40 euros/hora. Para cada una de las fases se detalla en la siguiente tabla:

	Nº de horas	Precio
Recogida y análisis de requisitos	19	760 euros
Diseño	15	600 euros
Codificación	34	1360 euros
Implantación de la herramienta	20	800 euros
Manual de usuario y presentación	9	360 euros
Totales	97	3880 euros

El coste total para desarrollar el proyecto asciende a 3880 euros + IVA.

Respecto al coste de los medios utilizados es de coste cero, ya que todo el software que se utiliza para su implementación es libre y gratuito.

3 Marco de trabajo y conceptos previos

3.1 Definición de las necesidades docentes

Según las actuales teorías del aprendizaje, las tres funciones de la evaluación son:

- **Diagnóstica** para conocer el nivel actual en el que se encuentra el alumno.
- **Formativa** se realiza a lo largo del curso y tiene que servir para apoyar al alumno en su proceso de aprendizaje, ayudándolo en sus deficiencias y errores.
- **Sumatoria** se realiza al finalizar con el fin de obtener una calificación al aprendizaje alcanzado.

Existen multitud de instrumentos para evaluar a un alumno donde el principal objetivo que se persigue es conocer el nivel de asimilación de los conocimientos adquiridos, pero donde además es importante no descuidar otros factores igual de importantes como son:

- La **Confiabledad**: El resultado de la prueba tiene que ser veraz, estar fuera de cualquier duda y reflejar fielmente el nivel del alumno. En este caso, por un lado juega un papel importante el enfoque y alcance de cada una de las preguntas que se realizan que depende

exclusivamente del docente y donde no podemos hacer nada. Pero por otro lado, está el problema de que los alumnos pueden copiarse de un compañero, sobretodo en las aulas de informática, donde suelen estar muy juntos. Una posible solución sería realizar las preguntas en un orden diferente al del compañero.

- La **Objetividad**: Al corregir una prueba realmente estamos ejerciendo de jueces y es muy importante realizar lo con el mismo patrón para todos los alumnos. En las típicas pruebas escritas, muchas veces perdemos esa objetividad en el momento de leer el nombre del alumno, y es muy fácil ser más estricto o permisivo en función de su actitud en clase. Una forma de solucionar este problema es utilizando etiquetas donde el alumno ya no se identifica por un nombre sino por un código como realizan ya algunas universidades como la UOC. En las pruebas realizadas con ordenador, se puede ocultar la identificación del alumno y además contamos con la ventaja de ser mucho más difícil, por no decir imposible, la identificación por su caligrafía o forma de expresarse.

De este modo, nos encontramos con:

- Técnicas objetivas donde realizamos una valoración cuantitativa.
- Técnicas subjetivas donde realizamos una valoración cualitativa
- Técnicas mixtas

A continuación, vamos a analizar cada una de ellas:

Las técnicas objetivas están dirigidos a evaluar el nivel de conocimientos del alumno

- Cuestionarios de elección múltiple es uno de los más utilizados por la facilidad de corrección y el alto grado de objetividad que permite.
- Cuestionarios de doble alternativa donde cada pregunta tiene dos respuestas habitualmente verdadero-falso, correcto-incorrecto, si-no. No se suelen utilizar porque no refleja realmente el nivel de conocimiento en aquellos alumnos que realizan la prueba al azar.
- Asociar parejas consiste en relacionar por parejas los elementos de dos conjuntos.
- Rellenar huecos consiste en completar los espacios en blanco en una frase o párrafo con un número ilimitado de posibles respuestas para cada uno de los huecos. Por ejemplo, un crucigrama.

- Identificar consiste en ejercicios con objetos que el alumno debe colocar en el sitio adecuada. Por ejemplo, un rompecabezas o puzle.
- Clasificar consiste en clasificar los elementos propuestos según su naturaleza. Por ejemplo relacionar una serie de conceptos con dos ideas principales.

Las técnicas subjetivas incluyen todas las pruebas que implican el juicio del docente y son especialmente útiles para valorar múltiples aspectos del alumno: forma de expresarse, iniciativa, capacidad de afrontar problemas, habilidades sociales, etc.

- Exposición oral y Redacción escrita permite al alumno defender sus ideas a través de un foro, chat o similar, frente al profesor o a sus propios compañeros. El profesor debe establecer una serie de pautas antes de su realización. La ventaja de esta prueba frente a otras es, que permite evaluar la capacidad de riqueza lingüística que tiene el alumno y la implicación con el tema asignado.
- Resolución de problemas consiste en la presentación de una serie de problemas que el alumno deberá solucionar, en un tiempo establecido por el profesor. De esta manera se pide al alumno que aplique los conocimientos que ha adquirido. Se pueden utilizar diferentes herramientas para aplicar esta técnica como, por ejemplo, chat, correo, foro, audioconferencia, videoconferencia, etc.
- Caso Práctico se utiliza para analizar una situación o problemática ya dada o que puede darse, partiendo de los diferentes factores involucrados, el análisis de los antecedentes, condiciones, de la situación, etc. Esta técnica pretende que el alumno reflexione sobre las distintas situaciones que pueden plantearse en la vida real acerca de la materia formativa que está recibiendo. El tutor deberá evaluar aspectos como son la participación de los alumnos, el nivel de profundización de las ideas y la aplicabilidad de las soluciones ofrecidas, la creatividad, la capacidad de resolución de problemas, el número y frecuencia de las respuestas, etc.

Las técnicas objetivas tienen un alcance fijo y conocido por lo que su implementación es más sencilla. En las técnicas subjetivas su alcance es más abierto y hay que delimitar el problema para permitir la autocorrección, por ejemplo para la resolución de problemas se podría aplicar un algoritmo para resolver cada uno de los problemas, pero el algoritmo deberíamos definirlo nosotros.

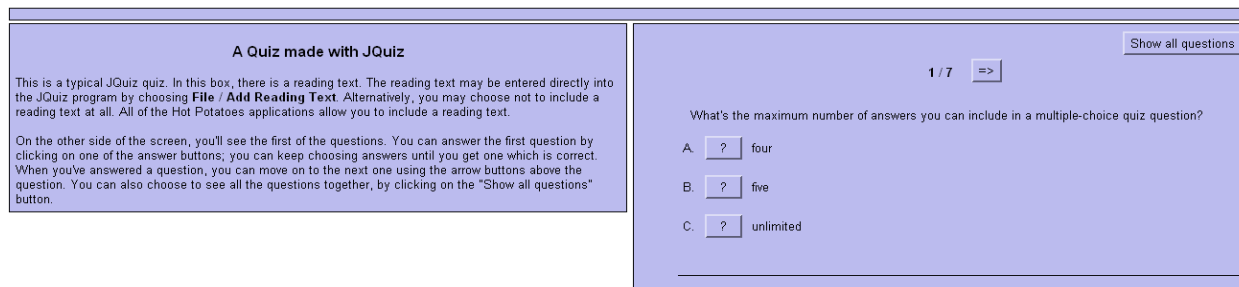
3.2 Análisis de herramientas ya existentes

Entre las aplicaciones que existen actualmente que corrigen pruebas de forma automática hay que mencionar Hot Potatoes, Jclie, QuizFaber, Constructor y Quia.

Hot Potatoes (<http://hotpot.uvic.ca>) es un conjunto de herramientas de autor, desarrollado por el equipo del University of Victoria CALL Laboratory Research and Development, que te permiten elaborar ejercicios interactivos basados en páginas Web, y que se describen a continuación:

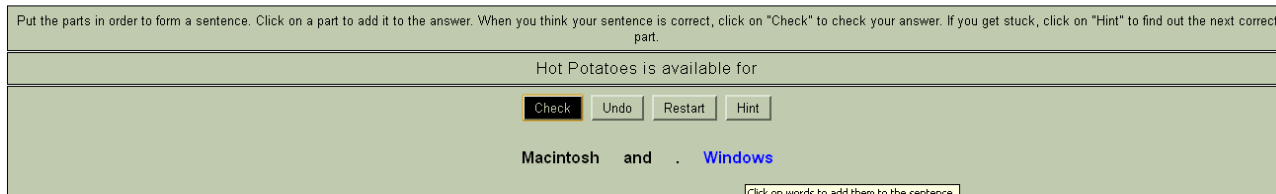
- **JQuiz**: consiste en ejercicios de elección múltiple, donde cada pregunta puede tener tantas respuestas como se quiera y cualquier número de ellas puede ser correcta.

A Quiz made with JQuiz
Choose the correct answer for each question.



- **JMix**: consiste en ejercicios de reconstrucción de frases o párrafos a partir de palabras desordenadas.

A Jumbled-Sentence Exercise made with JMix
Mixed-up sentence exercise



- **JMatch**: consiste en ejercicios de emparejamiento u ordenación

An Ordering Exercise made with JMatch

Matching exercise

Select the correct sequence of actions in creating a Hot Potatoes exercise by choosing from the drop-down lists.

Check

First: ???

Second: ???

Third: ???

Fourth: ???

Fifth: ???

Finally: Set the correct configuration information.
Save your data.
Start a Hot Potato
Click on "File / Create Web page".
Enter your data, questions, answers, etc.
Relax and have a coffee.







A Matching Exercise made with JMatch

Drag-and-drop matching exercise

Drag the words on the right to match the items on the left.

When you have finished this exercise, close this browser window to return to the tutorial.

Check

 beef	
 carrots	pie
	
 beans	lettuce
	
 tomato	

- JCloze: Consiste en ejercicios de rellenar huecos. Se puede poner un número ilimitado de posibles respuestas correctas para cada hueco y el estudiante puede pedir ayuda si tiene dudas y se le mostrará una letra de la respuesta correcta cada vez que pulse el botón ayuda.

A Gap-Fill Exercise made with JCloze

Gap-fill exercise

Fill in all the gaps, then press "Check" to check your answers. Use the "Hint" button to get a free letter if an answer is giving you trouble. You can also click on the "[?]" button to get a clue. Note that you will lose points if you ask for hints or clues!

This is a simple gap-fill exercise made with the [?] program. The user enters his or her answers into the gaps, then presses the "Check" [?] to find out which are correct, and to get a score. For each gap, any number of correct can be accepted. For example, this allows the answers "gap", "space", "blank" and "slot". Try them and you'll . If the user needs help, he or she can on the "Hint" button to get a free letter. To get a free letter in a particular gap, put the cursor in that gap before pressing the "Hint" button. The "Hint" button is optional -- if you want to make the exercise difficult for your [?], you don't need to include it. You can also include a special [?] for each gap if you wish. Finally, you can make answer-checking case-sensitive or not as you wish. This exercise is not case-sensitive -- you should be able to enter answers in upper or case.

Check Hint

Next example

A Gap-Fill with Drop-Downs and Individual Questions

Gapfill exercise

Enter your answers in the gaps. When you have entered all the answers, click on the "Check" button.

1. This exercise is created with
2. It uses drop-down lists instead of text boxes. This is an option that you can choose in the screen.
3. Like all the other Hot Potatoes, JCloze allows you to control the on the Web page, so you can change the appearance of your pages quite easily.
4. You can also insert code directly into Hot Potatoes to create effects like this numbered list of questions.
5. If you want to create a set of numbered like this, to allow students to be able to check each question individually, you should use JQuiz instead of JCloze.

- JCross: Permite generar crucigramas.

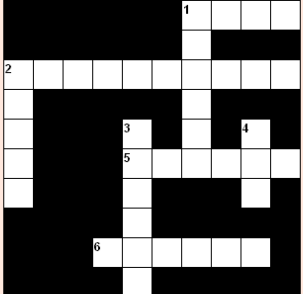
A Crossword made with JCross

Crossword

4:56

Complete the crossword, then click on "Check" to check your answer. If you are stuck, you can click on "Hint" to get a free letter. Click on a number in the grid to see the clue or clues for that number.

This crossword also demonstrates the exercise timer, which you can add to any exercise.

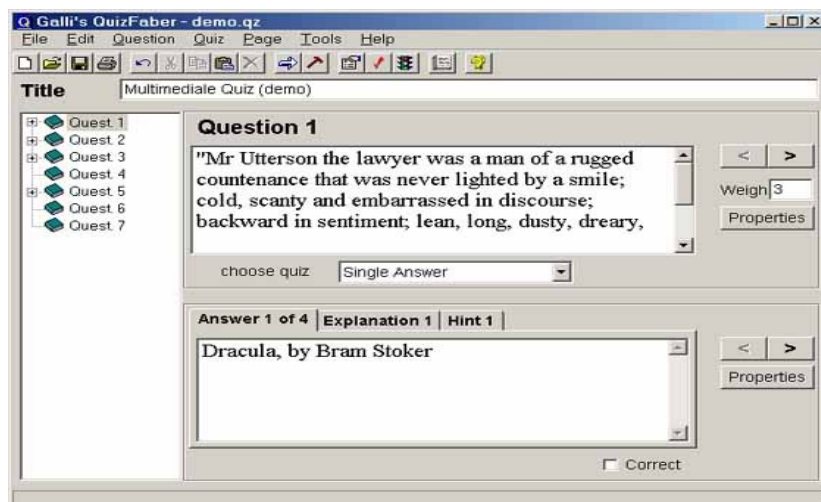


Hay que mencionar que más que corregir pruebas, lo que hacen es indicar en cada pregunta si la opción o respuesta elegida es correcta o incorrecta.

JClick (<http://clic.xtec.cat/es/index.htm>) está formado por un conjunto de aplicaciones informáticas que sirven para realizar diversos tipos de actividades educativas: rompecabezas, asociaciones, ejercicios de texto, palabras cruzadas, etc. Su funcionamiento es casi idéntico a Hot Potatoes, pero esta desarrollado en la plataforma Java y tiene un aspecto diferente.



QuizFaber (<http://www.lucagalli.net/en>) es una aplicación que genera fácilmente páginas html con javascript de ejercicios con preguntas y respuestas múltiples, que posteriormente se pueden incluir en cualquier otra página.



Constructor (<http://constructor.educarex.es/>) es una herramienta de creación de contenidos educativos digitales que incluye 53 modelos diferentes de actividades (crucigrama, sopa de letras, emparejamientos diversos, dictados de palabras o frases, completar u ordenar frases o textos, pirámide, ahorcado, rompecabezas, puzzles, tangram, reconstrucción de figuras, compositor musical,...) además de un conjunto de aplicaciones sumamente avanzadas (calculadoras básica y

científica, fórmulas y gráficas matemáticas, y un completo laboratorio virtual de física). Es necesario descargar una aplicación para generar el contenido posteriormente en html.



Quia (<http://www.quia.com/>) es una herramienta que al igual que las anteriores permite crear 16 actividades online diferentes.



En resumen, todas las aplicaciones tienen un gran parecido. En el aspecto funcional se puede decir que son casi idénticas con la única diferencia que unas implementan más actividades que otras. Y la mayor diferencia está en el aspecto, las más actuales tienen una apariencia más amigable, que las que llevan más tiempo funcionando.

4 Especificación y análisis

4.1 Análisis funcional de la aplicación

Después de analizar las aplicaciones existentes actualmente en el mercado nos damos cuenta que se centran en las técnicas objetivas de corrección, así que para no repetir lo que otras aplicaciones ya realizan sólo se implementará la que se utiliza habitualmente en un centro de educación secundaria que son las preguntas de respuesta múltiple.

Respecto a las técnicas subjetivas de corrección nos centraremos en la resolución de problemas sencillos. La aplicación permitirá que el profesor defina un problema junto a su algoritmo de autocorrección.

Por último, se realizará una pequeña aplicación que aprenderá a corregir, aunque no se trata realmente de una aplicación de autocorrección auténtica, si que facilitará el proceso en gran medida.

Las tres aplicaciones se implementarán como si fueran tres actividades o pruebas de diferentes tipos, y aunque no sea la finalidad de este proyecto incluso podrían entremezclarse y ser completamente indiferentes para el alumno.

A continuación haremos una lista con los requisitos principales y como se implementaran:

- Uso de software libre

Dado que se trata de una aplicación web utilizaremos un lenguaje script PHP que puede ser utilizado en los sistemas operativos más comunes, aunque por nuestro requerimiento de software libre lo implementaremos bajo Linux. Como servidor web utilizaremos Apache.

- El principal requisito de la aplicación es conseguir un uso lo más sencillo posible e intuitivo para el usuario final que no tiene muchos conocimientos como usuario de ordenadores.

Se diseñará una interfaz que no este muy recargada (pocos elementos por página), intentando eliminar el texto por iconos representativos, y que cada pantalla tenga una funcionalidad muy acotada.

- Almacenar las actividades en formato XML

Dado que todavía no está claro un estandard para almacenar este tipo de información se

definirá un único fichero XML que permita guardar los tres tipos de actividades que vamos a implementar.

- Acceso restringido a la aplicación

Las páginas donde se crean y modifican las actividades tendrán un acceso restringido para que los alumnos no puedan entrar y ver las respuestas. Por otro lado, puede ser interesante que el alumno se pueda validar para posteriormente analizar sus resultados.

4.2 Requisitos del sistema

Después de realizar el análisis vamos a enumerar todos los requisitos que debe cumplir nuestra aplicación empezando por los funcionales:

- Programada en la versión 5.3 de PHP
- Completamente funcional en cualquier distribución Linux: Debemos tener especial cuidado en la forma de acceder a los directorios.
- Interfaz simple para facilitar el uso: Para ello se pedirá la opinión de diferentes tipos de usuarios.
- Almacenamiento de la información en ficheros XML: No será necesario el uso de una base de datos, por lo que el proceso de instalación será más sencillo.
- Disponer de una pantalla de validación de usuario: Así se mostrará diferentes partes de la aplicación en función del tipo de usuario.
- Permitir crear, modificar y eliminar actividades.
- Permitir crear, modificar y eliminar preguntas de las actividades de cualquiera de los tres tipos que se implementarán.
- Gestionar respuestas dadas por los alumnos que todavía no han sido evaluadas por el profesor.
- Mostrar y evaluar estas actividades a los alumnos.
- Calcular la nota obtenida en la prueba.

Respecto a los requisitos no funcionales encontramos:

- Estabilidad: Tiene que ser una actividad estable, disponible en cualquier momento. Los servidores web suelen destacar por esta cualidad y sobretodo los entornos con sistema operativo Linux.
- Portabilidad: Que sea fácilmente utilizable en cualquier lugar y cualquier momento sin grandes complicaciones. Por este motivo nos decantamos por una aplicación web ya que no es necesario realizar una instalación en cada equipo.
- Bajo costo: El coste de desarrollo es relativamente bajo, y una vez terminado y validado no necesita ningún tipo de mantenimiento adicional.
- Seguridad: Un usuario no autorizado no debe acceder a ninguna pantalla de la aplicación.

4.3 Arquitectura

La aplicación estará compuesta por 8 pantallas diferentes, aunque algunas de ellas implementan varias funcionalidades.

Antes de ver la funcionalidad de cada una de ellas hay que explicar dos conceptos:

- **Actividad**: Se refiere a un conjunto de preguntas sobre una temática similar, que se agruparan en un mismo contenedor y que se mostrará de forma secuencial al alumno.
- **Pregunta**: Se refiere a cada cuestión que puede ser planteada al alumno. En este proyecto se definen tres tipos diferentes:
 - **Respuesta múltiple (tipo 1)**: Consiste en un enunciado con un número variable de respuestas entre las que se encuentra la correcta.
 - **Problemas sencillos (tipo 2)**: Consiste en un enunciado en el que se van definiendo variables de contenido numérico con un rango de posibles valores y por otro lado se introduce la respuesta en forma de ecuación.
 - **Asistida (tipo 3)**: Muy parecida a las de respuesta múltiple, pero el alumno da una respuesta escrita. Si se encuentra dentro de las respuestas previamente evaluadas se da un resultado, en caso contrario se queda en estado de pendiente hasta que el profesor la evalúa y se añade a las lista de respuestas.

Veamos la funcionalidad de cada una de ellas:

- login
 - Es la pantalla donde se realizará la validación del usuario.
 - Se han definido dos usuarios que son “profesor” y “alumno” donde cada uno corresponde con un perfil diferente.
 - Es la única pantalla que tiene acceso cualquier tipo de usuario aunque no haya sido validado.
 - En caso de intentar acceder a cualquier otra página, sin haber pasado por esta, el usuario será redirigido aquí. Esto se consigue utilizando la variable de sesión “user” que será validada al inicio de cualquier página.

- actividades
 - Sólo puede acceder el perfil “profesor”
 - Muestra un listado con todas las actividades de ese usuario. Cada usuario tiene una propia carpeta que se ubica en */datos/nombre_del_profesor*.
 - Para crear una nueva actividad se crea un nuevo fichero con el nombre “preguntas_nombre_actividad.xml” dentro de la carpeta “/datos/nombre_del_profesor/actividad”.
 - Desde esta pantalla también podemos eliminar una actividad, borrando el fichero que contiene esa actividad.
 - En caso de que la actividad tenga alguna pregunta pendiente de corregir por el profesor (Preguntas de tipo 3) se mostrará un mensaje con el objetivo de aviso.
 - Tiene un enlace para quitar la validación del usuario

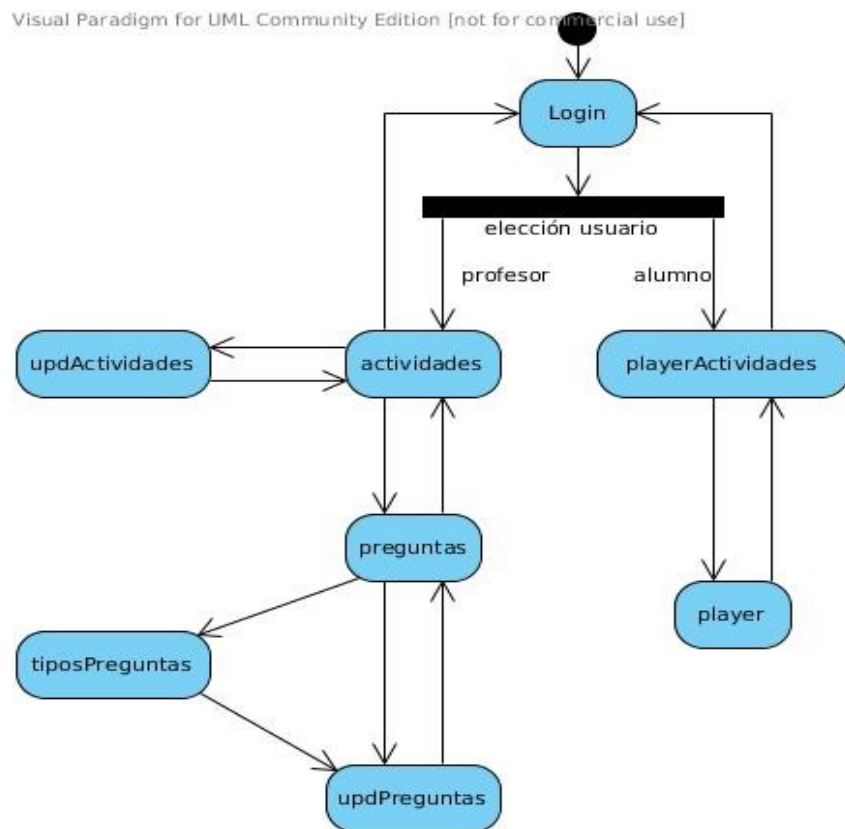
- updActividades
 - Sólo puede acceder el perfil “profesor”
 - Sirve para asignar el nombre de la actividad o para renombrar una actividad previamente creada. Si hay que renombrar la actividad, se cambia el nombre del archivo que contiene la actividad.

- preguntas
 - Sólo puede acceder el perfil “profesor”
 - Muestra todas las preguntas de esa actividad por orden, Las preguntas se encuentran dentro del archivo de la actividad.
 - En función del tipo de pregunta que sea se muestra información diferente:
 - Respuesta múltiple (tipo 1): Se visualiza el enunciado y las posibles respuestas utilizando el color rojo para las incorrectas y el color verde para las correctas.
 - Problema sencillo (tipo 2): Se visualiza el enunciado del problemas con las variables marcadas de color verde. Esas variables se componente de tres parámetros separados por el carácter “|” que se corresponden con el nombre de la variable, el valor mínimo y el valor máximo.
 - Asistida (tipo 3): Igual que el primer caso, pero se añade el color naranja a las respuestas pendientes de evaluar por el profesor.
 - Se permite eliminar una pregunta, eliminándola del fichero xml.
- tiposPreguntas
 - Sólo puede acceder el perfil “profesor”
 - Se accede únicamente cuando queremos crear una nueva pregunta, para elegir de que tipo se trata, después de la elección se redirige a la pantalla “updPreguntas”.
- updPreguntas
 - Sólo puede acceder el perfil “profesor”
 - Edita cada una de las preguntas en función del tipo de que se trate.
 - Las respuestas correctas se marcan seleccionando el checkbox.
- playerActividades
 - Sólo puede acceder el perfil “alumno”
 - Muestra una lista con todas las actividades creadas por el profesor. Además muestra la última nota obtenida en la actividad.
 - Tiene un enlace para quitar la validación del usuario

- player
 - Sólo puede acceder el perfil “alumno”
 - Visualiza cada una de las preguntas para que el alumno pueda responder en función del tipo.
 - Al finalizar todas las preguntas muestra el resultado.
 - En cada iteración se comprueba la respuesta dada, y se almacena en cualquiera de los tres posibles valores: correcta, incorrecta o pendiente del profesor (sólo en tipo 3).

4.3.1 Diagrama de actividad

A continuación se muestra el diagrama de actividad con cada una de las pantallas explicadas en el apartado anterior.



4.3.2 Diagrama de secuencia

Y profundizando un poco más en la funcionalidad estos serían los diagramas de secuencia para cada uno de los perfiles.

Diagrama de secuencia para el profesor

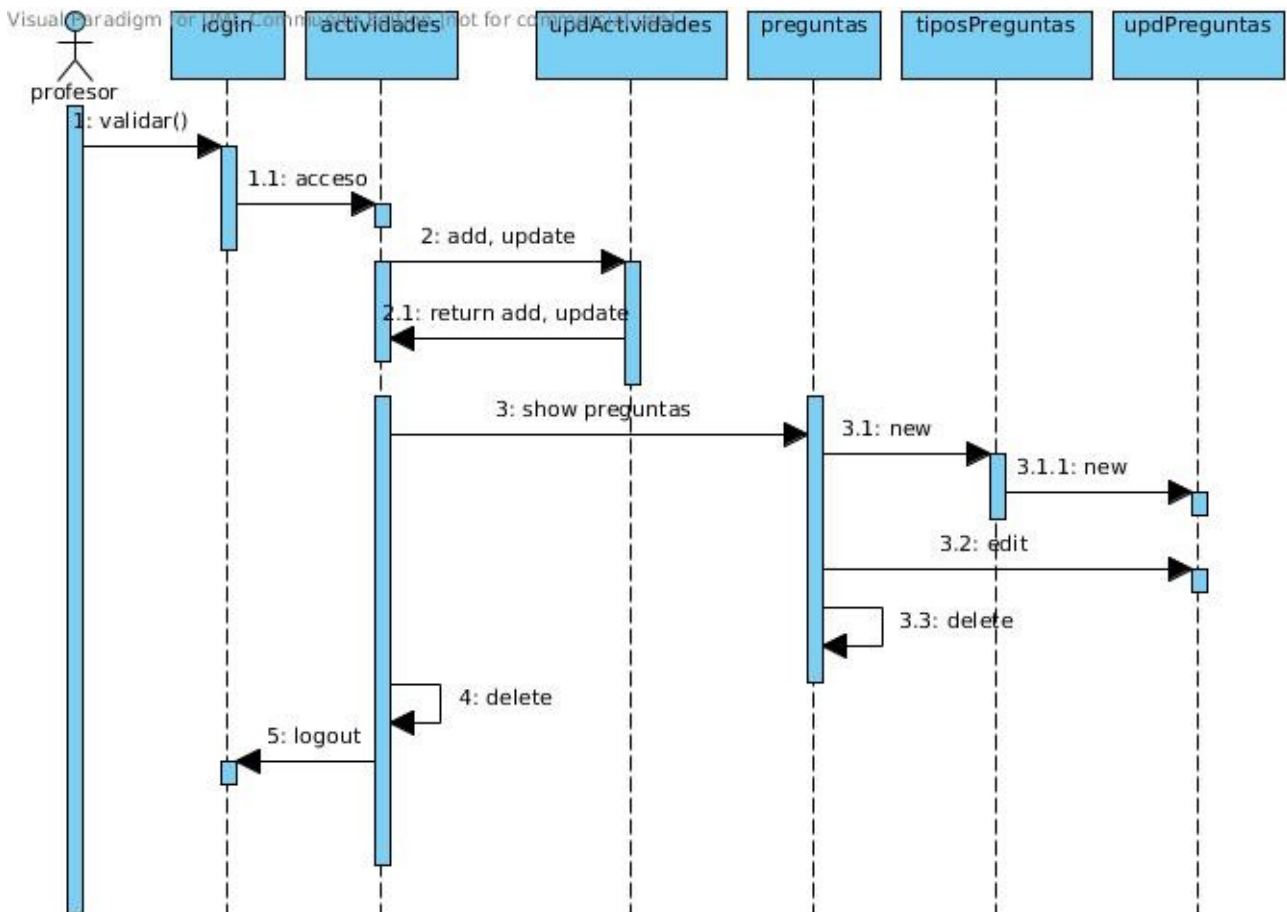
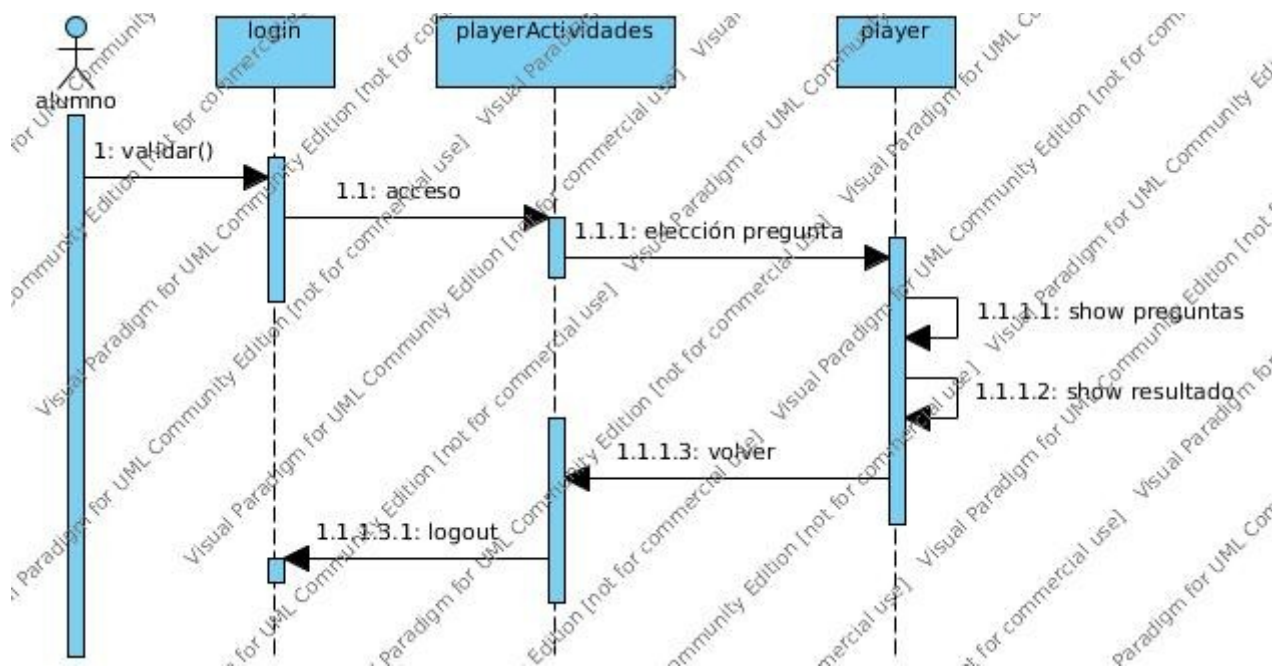


Diagrama de secuencia para el alumno



4.3.3 Estructura de datos y carpetas

Dado que la información que maneja principalmente la aplicación son ficheros y que la información de los usuarios se gestiona desde fuera. Consideramos que no es necesario el uso de una base de datos y almacenaremos todos los datos en una carpeta específica dentro del aplicativo.

La estructura de carpetas de la aplicación será la siguiente:

Carpeta	Funcionalidad
/	Contiene los scripts .php que implementan la funcionalidad de la aplicación.
/templates	Contiene las plantillas html del aplicativo (separando la capa de negocio de la capa de presentación)
/css	Contiene las hojas de estilo
/js	Contiene los scripts necesarios
/img	Contiene las imágenes que son necesarias para el funcionamiento de la aplicación
/datos	Contiene la información creada por los profesores, resultado de las pruebas y el resto de datos que necesita la aplicación.

La carpeta que se utilizará como repositorio de la información es “/datos” y tendrá los siguientes

ficheros:

preguntas_<nombre_actividad>.xml: Se almacenarán las preguntas que posteriormente deberán contestar los alumnos. Un ejemplo sería el que se muestra a continuación con los tres tipos diferentes de preguntas.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<preguntas>
  <pregunta tipo="1">
    <enunciado>Las tijeras que utilizas normalmente para cortar papel es un tipo de palanca ¿De qué tipo?</enunciado>
    <respuesta valor="10">Primer grado</respuesta>
    <respuesta valor="0">Segundo grado</respuesta>
    <respuesta valor="0">Tercer grado</respuesta>
    <respuesta valor="0">Cuarto grado</respuesta>
  </pregunta>
  <pregunta tipo="2">
    <enunciado>Calcula el peso de un niño que se encuentra a [r2,1,10] metros del punto de apoyo, para que se mantenga el balancín en equilibrio. Si enfrente tiene a otro niño que pesa [p1,30,120] kg. y que se encuentra a [r1,1,10] metros del punto de apoyo</enunciado>
    <respuesta>solucion=(p1*r1)/r2</respuesta>
  </pregunta>
  <pregunta tipo="3">
    <enunciado>Dime el nombre de un objeto que se corresponda con una palanca de primer grado</enunciado>
    <respuesta valor="10">Balancín</respuesta>
    <respuesta valor="10">Balanza</respuesta>
    <respuesta valor="0">Fregona</respuesta>
    <respuesta valor="">Destornillador</respuesta>
  </pregunta>
</preguntas>
```

El tipo de pregunta puede tener tres valores:

- **tipo="1"**: Preguntas de respuesta múltiple, donde la opción correcta se marca con la propiedad valor igual a 10, en caso contrario su valor será 0.
- **tipo="2"**: Problemas de resolución sencilla. En la pregunta se definen las variables con el intervalo inicial y final de posibles valores que puede adquirir. Y en la respuesta se escribe el algoritmo que da la respuesta válida.
 - Las variables vienen contenidas dentro de los caracteres “[“ y “]”. Separados los tres valores por el carácter “,”.
 - La respuesta tiene que contener la palabra “solucion” que se sustituirá por la respuesta dada y se validará la condición. Se puede emplear cualquiera de las variables definidas en el enunciado.
- **tipo="3"**: Preguntas de escribir la respuesta, donde se autocorregirá si tiene la respuesta

previamente definida. El parámetro valor define la cantidad de acierto de la respuesta que puede ser 0 (incorrecta), 10 (correcta) o vacío (pendiente de corregir por el profesor).

expediente.xml: Almacenará los resultados obtenidos por el alumno.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<expediente>
  <actividad nombre="Recuperacion" aciertos="7" errores="25" pendientes="0"/>
  <actividad nombre="Mecanismos" aciertos="1" errores="1" pendientes="0" />
</expediente>
```

Contiene un registro para cada actividad realizada con varios parámetros:

- nombre: Corresponde con el nombre de la actividad.
- aciertos: Número de aciertos en la última vez que se realizó la actividad por el alumno.
- errores: Número de fallos en la última vez que se realizó la actividad por el alumno.
- pendientes: Número de respuestas pendientes de evaluar por el profesor en la última vez que se realizó la actividad por el alumno. En caso de ser superior a 0, se crea un nodo hijo llamado “pendiente” con cada una de esas respuestas pendientes de evaluar. En el momento, que el profesor la evalué, automáticamente se autocorregirá.

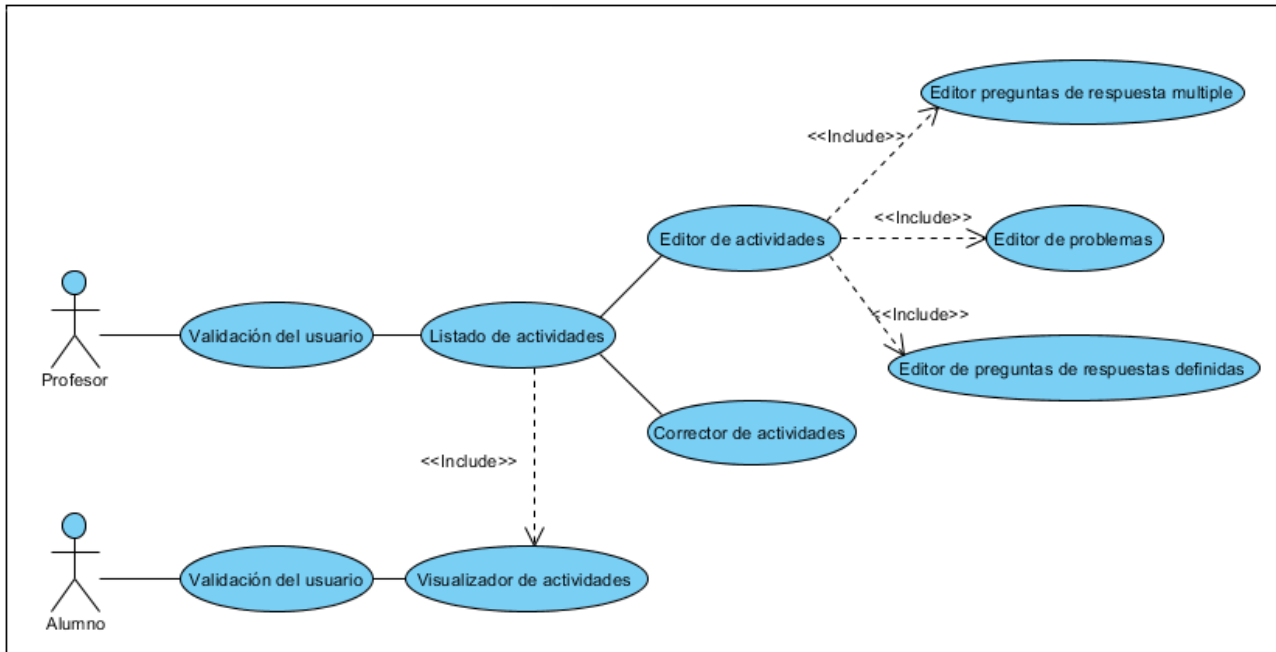
4.4 Interfaz de usuario

Hay que distinguir dos tipos de usuario:

- El perfil alumno podrá resolver las diferentes actividades propuestas por el profesor y obtendrá al finalizar el resultado de la calificación. Como complemento, se puede validar el acceso para almacenar esos resultados.
- El perfil profesor podrá introducir los diferentes tipos de actividades programadas y definir la respuesta correcta o el algoritmo que la resuelve. Dado que este perfil almacena información que no interesa que conozca el alumno, se limitará el acceso con un nombre de usuario.

La aplicación utiliza diferentes pantallas para realizar toda su funcionalidad.

Su diagrama de casos de usos general sería el siguiente:



Validación del usuario

Preguntando un nombre de usuario y una contraseña, identifica al usuario y permite el acceso a la aplicación. Existen dos posibles perfiles:

- Perfil profesor que se redirige al listado de actividades.
- Perfil alumno que se redirige al visualizador de actividades

Formulario de acceso a ejercicios autocorregibles con los siguientes campos:

- Nombre de usuario:
- Contraseña:
- Botón: Entrar

Actualmente hay únicamente dos usuarios de prueba:
Usuario: profesor Contraseña: profesor
Usuario: alumno Contraseña: alumno

Listado de actividades

Sólo tiene acceso el perfil profesor. Muestra una lista con todas las actividades creadas por el profesor autenticado en la pantalla anterior.




Desde esta pantalla, se podrá:




- Crear nuevas actividades: Pulsando sobre el botón “Añadir una nueva actividad”
- Modificar actividades ya creadas: Desde el icono del lápiz

- Modificar el nombre de la actividad: Desde el icono de propiedades
- Eliminar actividades ya creadas: Desde el icono del aspa
- Corregir actividades de tipo 3 (cuando su respuesta todavía no ha sido definida). Desde el enlace “[Corrección pendiente]” que sólo aparecerá en caso de exista alguno.

Logout

Listado de actividades

Estructuras por *profesor* [Corrección pendiente]   

Mecanismos por *profesor* [Corrección pendiente]   

Añadir un nueva actividad ...

Cancelar

Propiedades de la actividad

Nombre de la actividad:

Editor de actividades

Sólo tiene acceso el perfil profesor. Permite visualizar todas la preguntas introducidas previamente. La respuestas en color verde son respuestas correctas y en color rojo incorrectas.

Con el icono del aspa se elimina la pregunta, y con el icono del lápiz se modifica. Para añadir nuevas preguntas pulsamos sobre el botón “Añadir una nueva pregunta”.

[Volver al listado de actividades](#)

Preguntas de la actividad

¿En qué unidad se mide el tiempo en el sistema internacional?

- En horas
- En minutos
- En segundos
- Ninguna de las anteriores

Calcula el peso de un niño que se encuentra a [r2|1|10] metros del punto de apoyo, para que se mantenga el balancín en equilibrio. Si enfrente tiene a otro niño que pesa [p1|30|120] kg, y que se encuentra a [r1|1|10] metros del punto de apoyo

Enumera dos sistemas de transporte terrestre

- Bicicleta
- Moto
- Avión

[Añadir un nueva pregunta ...](#)

Desde aquí se permite crear y modificar los tres tipos de actividades implementadas:

- Editor de preguntas de respuesta múltiple

[Volver al listado de preguntas](#)

Editor de preguntas

Enunciado:

¿En qué unidad se mide el tiempo en el sistema internacional?

Respuestas:

- En horas
- En minutos
- En segundos
- Ninguna de las anteriores

[Añadir nueva solución](#)

Se puede añadir tantas soluciones como sea necesario, pulsando sobre el enlace “Añadir nueva solución”. Las respuestas correcta/s se marcan con el checkbox. Para guardar los cambios se utiliza el botón “Guardar”.

- Editor de problemas

[Volver al listado de preguntas](#)

Editor de preguntas

Enunciado:

Calcula el peso de un niño que se encuentra a $[r2,1,10]$ metros del punto de apoyo, para que se mantenga el balancín en equilibrio. Si enfrente tiene a otro niño que pesa $[p1,30,120]$ kg. y que se encuentra a $[r1,1,12]$ metros del punto de apoyo

Solución:

`solucion=(p1*r1)/r2`

En el enunciado del problema se define este con todas las variables necesarias o deseadas. Para añadir una variable se define entre corchetes [] y contendrá tres datos separados por el carácter “|”.

Estos datos son:

- Nombre de la variable
- Valor mínimo que puede tener
- Valor máximo que puede tener

Es decir que si $r2$ quiero que sea un valor comprendido entre 1 y 10, tendré que escribir $[r2|1|10]$.

Cuando el alumno visualiza el problema se sustituye el variable por un número comprendido entre el 1 y el 10.

En la solución se escribe en formato ecuación con la variable “solucion” como la respuesta. Estas soluciones pueden ser del tipo “ $solucion=4*r2$ ” e incluso “ $2*r2<solucion<4*r2$ ”.

- Editor de preguntas de respuesta definida

Volver al listado de preguntas

Editor de preguntas

Enunciado:
Enumera dos sistemas de transporte terrestre

Respuestas:

- Bicicleta
- Moto
- Avión
- (pendiente) Coche

Guardar

Mismo sistema que el las preguntas del respuesta múltiple, pero las respuestas se añaden a medida que los alumnos dan respuestas.

Aquellas respuestas que todavía no han sido valoradas aparecen con la palabra “pendiente” delante.

Corrector de actividades

Sólo tiene acceso el perfil profesor. En caso de que algún alumno responda una respuesta que todavía no ha sido definida, se indicará por medio de esta pantalla al profesor, para que se califique y se añada a la lista de posibles respuestas.

Se realizará con la ventana anterior.

Visualizador de actividades

Tendrá acceso el alumno y el profesor. Desde aquí se pondrá en práctica el conocimiento de los alumnos respondiendo las preguntas propuestas por el profesor. Al finalizar se indicará la puntuación obtenida y se incorporará a su expediente.

Primero es necesario mostrar una lista de todas las actividades que puede realizar.

Logout

Listado de actividades

Estructuras por *profesor* Última puntuación: pendiente →

Mecanismos por *profesor* Última puntuación: pendiente →

Se muestra las últimas puntuaciones y en caso de querer realizarla se pulsa sobre el icono con una flecha.

Pregunta 1 de 3

¿En qué unidad se mide el tiempo en el sistema internacional?

- En horas
- En minutos
- En segundos
- Ninguna de las anteriores

Se muestran las preguntas con el formato adecuado según del tipo que sean.

Pregunta 2 de 3

Calcula el peso de un niño que se encuentra a 9 metros del punto de apoyo, para que se mantenga el balancín en equilibrio. Si enfrente tiene a otro niño que pesa 105 kg. y que se encuentra a 12 metros del punto de apoyo

23

Responder

Pregunta 3 de 3

Enumera dos sistemas de transporte terrestre

tren y bicicleta

Responder

Al finalizar hay una pantalla con el resultado obtenido.

Resultado de la actividad

El resultado obtenido en esta prueba ha sido:

Número de preguntas: 3

Número de aciertos: 0

Número de errores: 2

Número de respuestas pendientes de corregir: 1

Calificación: 0

(0%)

5 Implementación y pruebas

La implementación se realizó siguiendo la planificación establecida y en ningún momento se produjo desviaciones superiores a 3 días. Lo cual produce una gran satisfacción comprobar que conozco mis habilidades y que soy capaz de ajustar el tiempo con tanta precisión.

Respecto a los algoritmos implementados, quiero detenerme en algunos. Aunque parezca que esto es una aplicación sencilla, realmente es sencillo las preguntas de respuesta múltiple (tipo 1), el resto no es tan obvio.

- En los problemas sencillos, se ha implementado un sistema que es muy sencillo de manipular para el usuario final, permitiéndole crear variables fácilmente y luego nos permite utilizar esas variables para obtener la respuesta. El algoritmo que se utiliza para comprobar que el alumno da una respuesta correcta consiste en reemplazar la palabra “solucion” de la ecuación por la respuesta dada por el alumno. Después adaptamos el formato de la ecuación para que sea correctamente interpretado por la función **eval()** que implementa PHP.
- En las preguntas asistidas, si la respuesta dada por el alumno ya la tenemos, se corrige directamente como si se tratara de una de respuesta múltiple. En caso contrario, tenemos que añadir la respuesta dada al fichero de preguntas para que sea evaluada por el profesor. Y además, tenemos que almacenar esa respuesta dentro del perfil del alumno, para que cuando sea evaluada, se recalculen la nota del alumno o alumnos que dieron esa opción. De esta forma, cuando el profesor corrige la pregunta hay un proceso oculto para el usuario que

actualiza las actividades pendientes.

Por otro lado tenemos:

- Un fichero llamado “security.php” que realiza la comprobación de validación y se que incluye en cada uno de los ficheros que lo implementa.
- Un fichero llamado “common.php” que además de contener el único parámetro de configuración contiene aquellas funciones comunes y que se utilizan en diferentes partes de la aplicación. Un ejemplo es la función que lee nuestras plantillas que se utiliza en el resto de páginas. Una vez se ha leído la plantilla se añade la información dinámica y se envía a la salida.

5.1 Pruebas de integración

Pantalla de Login

Código	Acciones a verificar	Resultado esperado	Verificación
001	Introducir un usuario no valido	Volver a la misma página de acceso	OK
002	Introducir las credenciales del usuario profesor	Acceder a la página con el listado de actividades de ese profesor	OK
003	Introducir las credenciales del usuario alumno	Acceder a la página con el listado de actividades que puede realizar ese alumno	OK
004	Acceder a cualquier otra página sin validación previa.	Redirigir a la página de login	OK

Pantallas del perfil profesor

Código	Acciones a verificar	Resultado esperado	Verificación
005	Muestra todas las actividades del profesor	Comparar con los archivos que hay en la carpeta del profesor	OK
006	Crear actividad	Visualizar la actividad creada en el listado de actividades	OK
007	Modificar nombre de actividad	Visualizar la actividad modificada en el listado de actividades	OK
008	Eliminar actividad	Comprobar que desaparece la actividad del listado de actividades y que se elimina el fichero	OK
009	Editar actividad	Visualiza la lista de preguntas de esa actividad	OK
010	Añadir una nueva pregunta	Primero nos permite elegir el tipo de pregunta y después escribir su contenido. Por último comprobar que	OK

		aparece en la lista de preguntas en la última posición	
011	Eliminar una pregunta	Se elimina de la lista de preguntas	OK
012	Almacenamiento de las respuestas correctas	Comprobar que las respuestas correctas aparecen de color verde y las incorrectas de color rojo	OK
013	Almacenamiento de las respuestas pendientes	Comprobar cuando un alumno da una respuesta que no tenemos almacenada, se da de alta automáticamente y luego puede evaluarla el profesor	OK
014	Evaluar respuestas pendientes.	Comprobar que cambia la respuesta del color naranja a verde o rojo. Y por otro lado se actualiza la puntuación del alumno	OK
015	Permite hacer un logout	Elimina la autenticación y ya no deja acceder a ninguna página	OK

Pantallas del perfil alumno

Código	Acciones a verificar	Resultado esperado	Verificación
016	Muestra todas las actividades disponibles para el alumno	Comparar con los archivos que hay en la carpeta del profesor	OK
017	Visualiza el último resultado obtenido	Tiene que coincidir con la nota visualizada al finalizar la actividad.	OK
018	Utilizar una actividad con varios tipos de pregunta para comprobar la apariencia de las preguntas.	Comprobar que se corresponde con cada tipo. Y que los problemas muestra valores aleatorios.	OK
019	Resultado de la prueba	Calcular y verificar que el resultado obtenido es el correcto.	OK
020	Actualización del resultado de la prueba	Comprobar que cuando el profesor termina de evaluar las respuestas pendientes, se actualiza la nota del alumno	OK

5.2 Pruebas de aceptación

Código	Requisito	Verificación
001	Uso de software libre	OK
002	Uso de la aplicación sencillo e intuitivo	Aceptable

003	Almacenar información en ficheros XML	OK
004	Acceso restringido a la aplicación	Aceptable
005	Crear, modificar y borrar actividades	OK
006	Crear, modificar y borrar preguntas	OK
007	Permitir a los alumnos responder a las actividades creadas	OK
008	Mostrar el resultado de las pruebas	OK
009	Permitir corrección automática de preguntas tipo 3 una vez evaluadas por el profesor.	OK

6 Implantación y resultados

Para instalar la aplicación en cualquier servidor web es el siguiente:

- Descomprimir el código que se adjunta en la carpeta deseada, manteniendo la misma estructura.
- Editar el fichero “common.php” y modificar la variable de sesión “webPath” con el nombre del dominio donde esta alojada la aplicación. Por ejemplo,

```
$_SESSION['webPath'] = "http://www.qtuto.com/";
```
- Dar permisos de escritura a la carpeta */datos* y todo su contenido.
- Apuntar con el navegador a la carpeta donde se han copiado los archivos. Se ha colgado una versión de la aplicación en la siguiente dirección <http://www.qtuto.com>

Los resultados obtenidos son los esperados, además se ha podido poner ya en práctica con los alumnos de 1º de la ESO del instituto en el que trabajo realizando una prueba para recuperar una evaluación pendiente y su impresión ha sido muy satisfactoria destacando los siguientes puntos:

- El interfaz es mucho más agradable que otras aplicaciones similares que habían utilizado hasta ahora.
- Han mostrado asombro cuando la mayoría obtuvo una nota nada más responder la última pregunta.

- Han mostrado sorpresa cuando el problema que tenían que resolver era diferente al de su compañero, a pesar de que sólo se cambiaban los valores para ellos se trataba de un problema completamente diferente.
- Las preguntas se adaptaban a lo que habían estudiado, ya que muchas veces que hacen ese tipo de preguntas en el ordenador aparecen preguntas que no han dado en clase. Esto se consigue cuando es el propio profesor que imparte la clase quien escribe las preguntas, y la forma de conseguir es simplificando al máximo la parte de creación de preguntas. Y en este aspecto creo que lo he conseguido.

7 Conclusiones

Después de desarrollar el proyecto estoy muy contento con el resultado obtenido, creo que he logrado centrar las bases de lo que puede ser una aplicación con mucha más flexibilidad y potencia. He conseguido definir un proyecto con toda su funcionalidad estableciendo una planificación que he cumplido a lo largo de estos meses de trabajo, y después de enseñar la aplicación a compañeros y ponerla en práctica con mis alumnos, la satisfacción es doble y quiero seguir trabajando en ello con los puntos que defino en el siguiente apartado.

Cuando empecé a pensar en el proyecto pensaba en algo muy parecido a lo que he conseguido, es una aplicación sencilla que puede mejorarse en muchos aspectos como puede ser la gestión de usuarios, pero la intención no era crear un completo gestor de actividades ya que para eso no había tiempo material. Mi objetivo era crear algo sencillo que se pudiera usar realmente en el día a día de un profesor, al mismo tiempo que ponía en práctica los conocimientos aprendidos en las diferentes asignaturas de la carrera y practicar con un lenguaje de programación web como PHP.

8 Trabajo futuro

Como ya he señalado en el apartado anterior, creo que he desarrollado una herramienta que me facilita mi trabajo diario y tengo la intención de proseguir el desarrollo con los siguientes puntos:

- Gestión de usuarios para poder dar de alta tanto alumnos como profesores.
- Posibilidad de añadir imágenes y vídeos en los enunciados y respuestas que puede implementarse fácilmente con un editor WYSIWYG como CKEditor.

- Añadir posibilidad de seleccionar varias respuestas como correctas.
- Implementar nuevos tipos de preguntas
- Guardar un historial con todos los resultados obtenidos por un alumno, en lugar de almacenar únicamente el último.
- Enviar un email al alumno cuando su actividad ha sido evaluada completamente por el profesor.

9 Bibliografía

FERNÁNDEZ PRIETO, M. (2004). “Las TIC en las aulas”. *Boletic*, septiembre 2004, pàg. 56-59.

GALO SÁNCHEZ, J. (2007). *Experimentación del proyecto educativo en el aula. Proyecto Descartes*.
[HTTP://DESCARTES.CNICE.MEC.ES/HEDA/DIFUSION/MATERIALES/XIICEAM08/MESA/DESCARTESXIICEAMRED.PPT](http://DESCARTES.CNICE.MEC.ES/HEDA/DIFUSION/MATERIALES/XIICEAM08/MESA/DESCARTESXIICEAMRED.PPT)

MARQUÈS GRAELLS, P. (2000). *Impacto de las TIC en educación: Funciones y limitaciones*.
<http://peremarques.pangea.org/siyedu.htm> [fecha de consulta: 18 de abril, 2012].

<http://hotpot.uvic.ca> [fecha de consulta: 9 de marzo, 2012].

<http://clic.xtec.cat/es/index.htm> [fecha de consulta: 10 de marzo, 2012].

<http://www.lucagalli.net/en> [fecha de consulta: 11 de marzo, 2012].

<http://constructor.educarex.es/> [fecha de consulta: 12 de marzo, 2012].

<http://www.quia.es/> [fecha de consulta: 13 de marzo, 2012].

10 Anexo

Videotutorial

<http://youtu.be/rfVnAxnxMEY>

- Se explica como realizar cada una de las funcionalidades que implementa.