



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Sitio web para la gestión de evaluación entre
pares con pseudónimos

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Félix Pérez Collado

Tutor: Antonio Martí Campoy

2016-2017

Resumen

Este proyecto se centra en el desarrollo de una aplicación web que facilite a los profesores la gestión de la evaluación anónima entre pares. Este tipo de evaluación permite que los alumnos corrijan las tareas que han realizado ellos mismos. La idea principal del proyecto es permitir que un profesor pueda crear un grupo de alumnos y asignarles una tarea. A la vez que se crea esa tarea también se asignaran los alumnos que tendrán que corregir esa tarea. Para el desarrollo de este proyecto se hará uso de los lenguajes y tecnologías HTML, PHP, SQL y CSS principalmente.

Palabras clave: aplicación web, evaluación anónima, evaluación entre pares, HTML, PHP, SQL, CSS.

Resum

Aquest projecte es centra en el desenvolupament d'una aplicació web que faciliti als professors la gestió de l'avaluació anònima entre pars. Aquest tipus d'avaluació permet que els alumnes corregeixin les tasques que han fet ells mateixos. L'idea principal del projecte es permetre que un professor pugui crear un grup d'alumnes y assignar lis una tasca. Al mateix temps que es crea una tasca, també s'assignaran els alumnes que tindran que corregir aquesta tasca. Per el desenvolupament d'aquest projecte es farà us dels llenguatges y tecnologies HTML, PHP, SQL y CSS principalment.

Paraules clau: aplicació web, avaluació anònima, avaluació entre pars, HTML, PHP, SQL, CSS.

Abstract

This project focuses on the development of a web application that provides teachers with the management of the anonymous peer assessment. This type of assessment allows students to correct the tasks they have done between themselves. The main idea of the project is to allow a teacher to create a group of students and assign them a task. When the teacher creates a task the application will assign the students who will correct it. For the development of this project the languages and technologies HTML, PHP, SQL and CSS have been mainly used.

Keywords: web application, anonymous peer evaluation, HTML, PHP, SQL, CSS.

Tabla de contenidos

1.	Introducción.....	7
1.1	Objetivos.....	8
1.2	Estructura.....	8
2.	Requisitos.....	10
2.1	Introducción.....	10
2.2	Descripción general.....	12
2.3	Requisitos específicos.....	15
3.	Casos de uso.....	24
3.1	Diagrama de casos de uso.....	25
3.2	Descripción casos de uso.....	26
4.	Diseño.....	31
4.1	Modelo-vista-controlador (MVC).....	31
4.1.1	Vista.....	32
4.1.2	Controlador.....	33
4.1.3	Modelo.....	34
4.2	Base de datos relacional.....	34
5.	Implementación.....	37
5.1	Tecnologías.....	37
5.2	Herramientas.....	39
5.3	Estructura de los ficheros y directorios.....	40
6.	Pruebas.....	46
6.1	Pruebas de uso.....	46
6.2	Pruebas de visualización.....	50
6.3	Validador W3C.....	56
7.	Manuales de usuarios.....	62
7.1	Manual profesor.....	62
7.2	Manual alumno.....	72
8.	Conclusiones.....	77
9.	Bibliografía.....	78

1. Introducción

Una forma muy útil de mejorar el aprendizaje de los alumnos es hacer que ellos mismos corrijan las tareas que han realizado, esto es lo que llamamos coevaluación entre pares o evaluación entre iguales.

La coevaluación entre pares mejora la comprensión de los estudiantes en su propio aprendizaje y permite que participen de una manera más activa en el proceso de aprendizaje. La idea básica de este tipo de evaluación es hacer que los alumnos evalúen una tarea realizada por otro compañero. Dentro de las diferentes tipologías de evaluación y coevaluación, podemos distinguir dos variantes: la anónima y la no anónima. En el caso de la corrección anónima, ni el alumno que ha realizado la tarea sabe quién se la corregirá, ni el corrector sabe a quién está corrigiendo la tarea. Por el contrario en la corrección no anónima, el corrector si sabrá quién es el autor de la tarea que está corrigiendo, y el autor de la tarea sabrá quién le ha corregido [1].

La idea de este TFG surge de la necesidad de varios profesores de la UPV de querer tener a su disposición una herramienta informática que les permita gestionar este tipo de coevaluación, de una manera rápida y sencilla [2].

Este trabajo tiene el propósito de cubrir la necesidad de los profesores a partir de la implementación de una herramienta que de soporte y simplifique la organización y gestión de la coevaluación entre pares.

Cabe mencionar que Poliformat, desde el curso 2016/2017, incluye una herramienta que permite este tipo de evaluación, pero en la aplicación se incluirán algunas funcionalidades que no incluye la herramienta de Poliformat. Los profesores que demandan esta aplicación ya conocían la herramienta de Poliformat, pero necesitaban una herramienta más completa y personalizada que les diera más opciones a la hora de llevar a cabo este tipo de evaluación.

1.1 Objetivos

El objetivo principal de este trabajo es diseñar e implementar una aplicación web. De este modo, mediante esta aplicación un profesor podrá gestionar una evaluación entre pares, tanto de forma anónima como de forma pública en la que tanto autor y evaluador conocerán sus identidades.

Dentro de la aplicación se distinguirá entre dos tipos de roles de usuario, uno de profesor y otro de alumno. Se desarrollará una interfaz de usuario distinta para cada rol, ya que cada uno podrá llevar a cabo unas acciones distintas.

El profesor podrá crear grupos de alumnos a partir de un archivo con formato CSV. Una vez tenga los grupos creados podrá asignar tareas a los alumnos para que las realicen, y una vez realizadas, la aplicación web también permitirá que los alumnos corrijan la tarea y la puntúen. Una vez las tareas estén finalizadas, el profesor tendrá acceso a ellas para verlas y ver las correcciones.

Por otro lado, los alumnos, desde su interfaz de usuario podrán ver las tareas que tienen pendientes tanto de realizar como de evaluar. Tendrán la opción de realizar estas tareas, y también tendrán que corregir las tareas de otros usuarios.

1.2 Estructura

Esta memoria está estructurada en 9 puntos. El primero de los puntos es en el que estamos actualmente, la introducción. En el siguiente capítulo se definirán los requisitos software siguiendo el estándar IEEE 830/1998.

En el punto 3 se explicarán los distintos casos de uso mediante un diagrama de casos de uso y a continuación se detallará la información relativa a cada caso de uso mediante una tabla.

En la siguiente fase del proyecto se explicará el diseño que se llevará a cabo. El diseño de la aplicación seguirá la arquitectura que siguen un gran número de aplicaciones web en la actualidad, la arquitectura Modelo-Vista-Controlador. Dentro de este punto también se definirá el esquema de la base de datos que utilizará la aplicación.

Tras la etapa de diseño pasaremos a la fase de implementación. En este punto se detallarán las tecnologías y las herramientas que se harán servir para llevar a cabo el desarrollo, así como la estructura de directorios y ficheros resultante.

Una vez implementada la aplicación procederemos a realizar las pruebas. La aplicación se probará en distintos navegadores web a distintas resoluciones. También se probará el correcto funcionamiento de los requisitos definidos.

A continuación se incluirán dos manuales de usuario para explicar al detalle el funcionamiento de la aplicación, uno para cada tipo de usuario que tendrá la aplicación. Se redactará un manual para los profesores y otro para los alumnos, ya que las tareas que pueden hacer cada uno de estos usuarios dentro de la aplicación son diferentes.

Para concluir esta memoria tendremos un punto con las conclusiones obtenidas tras la ejecución del proyecto, y otro, con la bibliografía que se ha consultado para obtener la información necesaria para llevar a cabo la ejecución del proyecto.

2. Requisitos

Dentro de este apartado, voy a definir los requisitos software que deberá cumplir la aplicación web. Para la especificación de requisitos seguiré el estándar IEEE 830/1998[3].

Este estándar define la estructura a seguir para desarrollar un documento de requisitos completo.

2.1 Introducción

Para definir los requisitos software, como ya he comentado, seguiré el estándar IEEE 830/1998. Este estándar define los pasos a seguir para llevar a cabo una buena especificación de requisitos. En los siguientes puntos sigo las directrices que marcan este estándar para definir los requisitos.

2.1.1 Propósito

El objetivo principal de la especificación de requisitos es definir de una manera concreta las funcionalidades que la aplicación podrá llevar a cabo. También señalaré lo que mi aplicación no hará.

2.1.2 Ámbito del sistema

El objetivo de este trabajo es desarrollar una aplicación web que permita la evaluación entre iguales. La aplicación que desarrollaré recibirá el nombre de Coevaluación entre pares.

La aplicación permitirá a un profesor crear varios grupos de alumnos a partir de un fichero CSV de una manera automática. El profesor podrá crear tareas y asignarlas a los grupos que tenga creados.

Una vez estén creadas estas tareas, los usuarios podrán acceder a la aplicación y realizar dichas tareas. Además de realizar su tarea, los alumnos podrán corregir las tareas de otros compañeros y puntuarlas. La aplicación llevará a cabo un control de acceso, de tal manera que solo los usuarios creados por el profesor podrán acceder mediante usuario y contraseña.

El profesor además de crear los diferentes tipos de tareas y los grupos, podrá hacer un seguimiento del estado de cada tarea. También podrá puntuar cada tarea, aunque esta ya haya sido puntuada por un alumno.

El principal beneficio de esta aplicación es que automatizará todo el proceso de creación y corrección de una tarea. Además, el profesor podrá llevar a cabo un seguimiento del estado de la tarea y podrá obtener un archivo en formato CSV con las calificaciones de cada alumno.

2.1.3 Definiciones, Acrónimos y Abreviaturas

CSV (comma-separated values): es un tipo de documento en formato abierto para representar datos en forma de tabla, en las que las columnas se separan por comas o punto y coma y las filas por saltos de línea.

2.1.4 Referencias

IEEE Recommended Practice for Software Requirements Specification. AN-SI/IEEE 830/1998

2.1.5 Visión general del documento

El capítulo de especificación de requisitos consta de las secciones de introducción, descripción general y requisitos específicos. En la introducción, he proporcionado una visión general del documento. En la sección de descripción general describiré todos los factores que afectan a la aplicación y a sus requisitos. Por último, en el apartado de requisitos específicos, definiré los requisitos con el suficiente nivel de detalle para poder llevar a cabo su implementación y poder planificar las pruebas que indiquen si se han cumplido estos requisitos.

2.2 Descripción general

En este apartado se van a describir todos los factores que afectan a la aplicación y a sus requisitos. También, describiré el contexto que rodea a la aplicación, para posteriormente definir los requisitos de una manera más específica en el siguiente punto.

2.2.1 Perspectiva del producto

La aplicación que voy a desarrollar requerirá del uso de un navegador web para acceder a ella. Al ser una aplicación web el navegador web es la herramienta a través de la cual se tendrá acceso a la aplicación.

2.2.2 Funciones del producto

En este punto detallaré las principales funciones que tendrá que ser capaz de realizar la herramienta.

Antes de definir las funciones que realizará mi aplicación, creo que es necesario comentar que distinguiré entre dos tipos de roles de usuarios que tendrán acceso a la aplicación. Por un lado, tendré al tipo de usuario profesor, y por otro, al tipo de usuario alumno. Cada uno de estos dos tipos de usuario tendrá una interfaz propia, desde la que accederán a unas funciones propias y diferentes. El profesor tendrá acceso a todas las tareas relacionadas con la administración, mientras que los alumnos solo tendrán

acceso a las funciones relacionadas con realizar y corregir las tareas que tengan asignadas.

A continuación, se muestran a grandes rasgos las funciones que realizará mi aplicación:

- Administrar grupos de alumnos: El profesor podrá crear un grupo de alumnos importando un fichero CSV. También podrá eliminar un grupo o ver los alumnos que pertenecen a ese grupo.
- Crear tareas: El profesor podrá crear diferentes tipos de tareas y asignarlas a un grupo que hubiera creado anteriormente.
- Ver tareas: El profesor podrá ver las tareas que haya creado, podrá ver el estado de la tarea y corregirla.
- Configurar cuenta: Tanto el profesor como los alumnos podrán cambiar su contraseña de acceso y guardar un correo electrónico. Este correo se utilizaría en caso de que algún usuario necesite recuperar su contraseña, para lo que se enviaría un email con la contraseña.
- Ver tareas a realizar: Los alumnos podrán ver las tareas que tienen pendientes de entregar, y podrán entregarlas.
- Ver tareas a corregir: Una vez una tarea haya sido entregada, los usuarios podrán ver las tareas que han de corregir y corregirlas.

2.2.3 Características de los usuarios

Esta aplicación está pensada para su uso en un ámbito académico, en el que un profesor es el administrador de la aplicación y es el que se encarga de dar de alta a los alumnos, que serán los usuarios de la aplicación. Por tanto a la aplicación solo tendrá acceso el profesor que sea el administrador y los alumnos a los que este profesor haya registrado. En la aplicación pueden estar registrados varios profesores a la vez, no tiene por qué haber un único profesor. Los alumnos no podrán registrarse por sí mismos en la aplicación, y por tanto no tendrán acceso a no ser que el profesor los registre.

En principio la aplicación está pensada para el ámbito universitario, pero no habría ningún problema en utilizar en cualquier otro nivel educativo preuniversitario.

En cuanto a la experiencia, hoy en día cualquier persona que esté estudiando tiene habilidades técnicas suficientes como para usar un navegador web y acceder a este tipo de aplicaciones.

2.2.4 Restricciones

Las principales restricciones con las que mi aplicación deberá convivir son:

- La aplicación deberá funcionar correctamente en los navegadores web Google Chrome, Internet Explorer y Mozilla Firefox
- La aplicación tendrá que visualizarse perfectamente en una pantalla de 1920x1080 pixeles

2.2.5 Suposiciones y Dependencias

La aplicación está diseñada para funcionar correctamente en los navegadores web mencionados, en el caso de utilizar otro navegador es posible que se produzca algún fallo de visualización o se produzca algún funcionamiento que no es el esperado.

El rendimiento de la aplicación puede variar en función del servidor web en el que esté alojada.

2.2.6 Requisitos Futuros

Algunos requisitos que sería interesante implementar en un futuro son:

- Permitir que un profesor tenga los grupos separados en varios cursos.
- Realizar tareas mediante la aplicación, en lugar de depender del intercambio de ficheros.
- Añadir comentarios a las calificaciones.
- Añadir distintos tipos de calificaciones, como podrían ser A, B, C, D.

2.3 Requisitos específicos

En este apartado del documento de especificación de requisitos según el estándar IEEE 830/1998 voy a proceder a explicar los requisitos de una manera detallada y completa.

2.3.1 Interfaces Externas

Para el desarrollo de esta aplicación será necesario implementar dos interfaces de usuario, una para los profesores y otra para los alumnos. Cada una de estas interfaces dará acceso a unas funciones distintas, en función de cada usuario.

2.3.2 Funciones

A continuación voy a explicar todas las acciones que deberá llevar a cabo la aplicación, para ello las enumerare en una lista explicando cada función o acción con un nivel de detalle suficiente para su posterior implementación.

Los requisitos que voy a definir los voy a organizar en función de los dos tipos de usuarios que existirán en mi aplicación.

Primero definiré los requisitos asociados al rol profesor:

Número	1
Requisito	Control de acceso profesores
Prioridad	Alta
Descripción	La aplicación tiene que llevar a cabo un control de acceso de los profesores. Para identificar a un profesor se utilizará su DNI como nombre de usuario y una contraseña que el usuario podrá cambiar. En caso de no recordar la contraseña la aplicación le enviara la contraseña por email a un correo que hubiera guardado previamente.

Número	2
Requisito	Crear grupo
Prioridad	Alta
Descripción	El profesor creará un grupo de alumnos a partir de un fichero CSV. La aplicación le permitirá importar este archivo y creará el grupo de manera automática.

Número	3
Requisito	Borrar grupo
Prioridad	Media
Descripción	El profesor podrá borrar un grupo que había creado anteriormente.

Número	4
Requisito	Visualizar grupo
Prioridad	Alta
Descripción	El profesor verá los grupos que ha creado. La aplicación le mostrará una lista de todos los alumnos que pertenecen a ese grupo.

Número	5
Requisito	Crear tarea sin corrector
Prioridad	Alta
Descripción	El profesor podrá crear una tarea y asignarla a un grupo de alumnos. Para crear esta tarea tendrá la posibilidad de subir un archivo, añadir una descripción y poner una fecha de entrega.

Número	6
Requisito	Crear una tarea en la que el alumno elige a quien corrige:
Prioridad	Alta
Descripción	La tarea se crea de la misma forma que en el requisito número 5, pero además se añade la posibilidad de que el profesor seleccione la opción de que un alumno elija a que alumno le corregirá la tarea.

Número	7
Requisito	Crear tarea con correctores aleatorios anónimos
Prioridad	Alta
Descripción	El profesor creará una tarea del mismo modo que en el requisito número 5. Además, podrá elegir el número de alumnos que corregirán esta tarea. Este número será ente 1 y 5, los correctores asignados a cada alumno serán elegidos de manera aleatoria por la aplicación. También tendrá la opción de seleccionar una fecha límite para la entrega de la corrección. Los alumnos que corrijan la terea no deberán saber a qué alumno se la están corrigiendo.

Número	8
Requisito	Crear tarea con correctores aleatorios no anónimos
Prioridad	Alta
Descripción	El proceso de crear la tarea es idéntico al del requisito número 7, con la diferencia de que los alumnos que corrijan esta tarea si sabrán a que alumno se la están corrigiendo.

Número	9
Requisito	Crear tarea eligiendo los correctores, anónima
Prioridad	Media
Descripción	El proceso de crear la tarea es idéntico al del requisito número 7, con la diferencia de que solo se asignara un corrector a cada alumno, además, cada corrector será elegido por el profesor, el profesor también podrá ser seleccionado como corrector. Los alumnos que corrijan la tarea no deberán saber a qué alumno se la están corrigiendo

Número	10
Requisito	Crear tarea eligiendo los correctores, no anónima
Prioridad	Media
Descripción	El proceso de crear la tarea es idéntico al del requisito número 9, con la diferencia de que los alumnos que corrijan la tarea si tendrán que saber a qué alumno se la están corrigiendo.

Número	11
Requisito	Ver tareas creadas
Prioridad	Alta
Descripción	La aplicación deberá permitir que el profesor pueda ver las tareas que ha creado por grupo. Se tendrá que mostrar toda la información relativa a cada tarea.

Número	12
Requisito	Corregir tareas
Prioridad	Alta
Descripción	La aplicación deberá permitir que el profesor corrija cada tarea que haya creado a cada alumno, independientemente del tipo de tarea que sea. Además, como en los requisitos número 9 y 10 el profesor puede ser asignado como corrector de una tarea, también deberá poder corregir esas tareas igual que lo harán los alumnos.

Número	13
Requisito	Ver las notas de tarea
Prioridad	Alta
Descripción	La aplicación deberá mostrar al profesor una tarea completa, mostrándole una lista de todos los alumnos que han de realizar esa tarea. También mostrará todas las notas que hayan puesto todos los correctores, incluido el profesor, y la nota media.

Número	14
Requisito	Exportar notas en un fichero CSV
Prioridad	Alta
Descripción	El profesor deberá tener la posibilidad de exportar las notas de una tarea en un archivo con formato CSV

Número	15
Requisito	Consultar pseudónimo
Prioridad	Media

Descripción	La aplicación utilizará un sistema de pseudónimos para garantizar el anonimato entre los alumnos en los casos en los que sea necesario, pero el profesor deberá poder saber en todo momento quien es cada alumno.
-------------	---

Número	16
Requisito	Actualizar contraseña
Prioridad	Media
Descripción	El profesor tendrá que poder cambiar su contraseña en caso de que lo desee.

Número	17
Requisito	Guardar correo electrónico
Prioridad	Media
Descripción	El profesor podrá guardar un correo electrónico. Este correo se utilizara para recuperar la contraseña.

A continuación se describen los requisitos asociados al rol de usuario alumno:

Número	18
Requisito	Control de acceso alumnos
Prioridad	Alta
Descripción	La aplicación tiene que llevar a cabo un control de acceso de usuarios. Para identificar a los alumnos se utilizará su DNI como nombre de usuario y una contraseña que el alumno podrá cambiar. En caso de no recordar la contraseña la aplicación le enviara la contraseña por email a un correo que hubiera guardado previamente.

Número	19
Requisito	Ver tareas
Prioridad	Alta
Descripción	El alumno podrá ver todas sus tareas, tanto las pendientes de realizar como las realizadas.

Número	20
Requisito	Realizar tareas
Prioridad	Alta
Descripción	El alumno podrá realizar las tareas que tenga asignadas. Podrá ver toda la información de la tarea, descargarse el archivo de la tarea y subir otro archivo con su solución.

Número	21
Requisito	Corregir tareas
Prioridad	Alta
Descripción	El alumno podrá corregir las tareas que tenga asignadas. Tendrá la posibilidad de ver la tarea que ha realizado otro alumno descargándose el archivo y podrá puntuar la tarea.

Número	22
Requisito	Ver notas
Prioridad	Media
Descripción	El alumno podrá ver las notas de las tareas que ha entregado y que han sido corregidas.

Número	23
Requisito	Actualizar contraseña
Prioridad	Media
Descripción	El alumno tendrá que poder cambiar su contraseña en caso de que lo desee.

Número	24
Requisito	Guardar correo electrónico
Prioridad	Media
Descripción	El alumno podrá guardar un correo electrónico. Este correo se utilizara para recuperar la contraseña.

2.3.3 Requisitos de Rendimiento

Los requisitos de rendimiento en este caso estarán sujetos al rendimiento que ofrezca en servidor en el que este alojada la aplicación.

El sistema no tendrá que soportar una carga demasiado grande de trabajo, pero tendrá que ser capaz de dar acceso a varios usuarios de forma simultánea.

La base de nuestra aplicación es la realización de tareas. Para llevar a cabo estas tareas los alumnos se descargarán un archivo y subirán otro con la resolución de la tarea, por tanto, la aplicación gestionará un número importante de archivos, por este motivo, el tamaño de máximo de archivo que se podrá subir al servidor es de 5 megabytes. Por otro lado, el servidor deberá tener una capacidad de almacenamiento mínima para soportar esta carga de ficheros.

Una capacidad de almacenamiento recomendable para el servidor seria 10 gigabytes. De esta manera haciendo un cálculo rápido el servidor permitiría almacenar como mínimo 2000 archivos:

Capacidad del servidos -> 10 GB=10000MB;

Tamaño máximo de archivo -> 5MB

Nº máximo de archivos -> $10000\text{MB}/5\text{MB}=2000$ archivos

En el peor de los casos la aplicación tendría que dar acceso simultáneamente a todos los usuarios que estén registrados en la aplicación, aunque es algo bastante improbable. El caso más sencillo para la aplicación sería que solo un usuario estuviera conectado.

2.3.4 Restricciones de diseño

El diseño de la interfaz de la aplicación está limitada por las posibilidades de diseño que ofrece una aplicación web.

2.3.5 Atributos del sistema

En este punto voy a detallar los atributos de calidad del sistema. Estos son fiabilidad, mantenibilidad, portabilidad y la seguridad.

La aplicación tiene que ser fiable, tendrá que funcionar correctamente y ofrecer todas las funciones que se han especificado en los requisitos.

Tendrá que ser administrable por los profesores que la utilicen, ya que serán los que tengan el papel de administradores de la aplicación.

En cuanto a la seguridad, la aplicación llevará a cabo un control de acceso mediante un nombre de usuario y contraseña.

Por último, la aplicación ha de ser lo suficientemente portable como para, en el caso que sea necesario, poder instalarla en otro servidor web diferente al que está actualmente.

3. Casos de uso

La técnica de los casos de uso se utiliza para capturar la información de cómo se desea que trabaje el sistema, nos permitirá capturar los requisitos funcionales del sistema que deseamos desarrollar.

Para definir los casos de uso utilizaremos el diagrama de casos de uso. En este diagrama tenemos a los actores y los casos de uso.

Un actor es una entidad que intercambia información con el sistema. Los casos de uso contienen una secuencia de transacciones que intercambian los actores y el sistema cuando se desea ejecutar alguna funcionalidad.

A continuación incluiré el diagrama de casos de uso, y posteriormente, haré una descripción detallada de cada uno.

3.1 Diagrama de casos de uso



3.2 Descripción casos de uso

Caso de uso	Iniciar sesión
Actores	Profesor y Alumno
Propósito	Tener un control de acceso a la aplicación
Resumen	El usuario accede a la aplicación con un navegador. Lo primero que deberá hacer es identificarse con su usuario y contraseña.
Precondiciones	El usuario ya había sido dado de alta en el sistema

Caso de uso	Recuperar contraseña
Actores	Profesor y Alumno
Propósito	Permitir que un usuario recupere la contraseña en caso de que la hubiera olvidado.
Resumen	El usuario solicita que se le envíe la contraseña al correo que hubiera guardado en su cuenta.
Precondiciones	El usuario ya había sido dado de alta en el sistema y había guardado un correo electrónico.

Caso de uso	Actualizar contraseña
Actores	Profesor y Alumno
Propósito	El usuario puede cambiar la contraseña si lo desea.

Resumen	El usuario puede cambiar la contraseña de su cuenta.
Precondiciones	El usuario ya había sido dado de alta en el sistema y ha iniciado sesión en la aplicación.

Caso de uso	Actualizar email
Actores	Profesor y Alumno
Propósito	El usuario puede guardar un correo electrónico.
Resumen	El usuario podrá guardar un correo electrónico, que será utilizado para recuperar la contraseña.
Precondiciones	El usuario ya había sido dado de alta en el sistema y ha iniciado sesión en la aplicación.

Caso de uso	Crear grupo
Actores	Profesor
Propósito	El profesor puede crear grupos de alumnos.
Resumen	El profesor podrá crear tantos grupos como desee, esto lo hará subiendo un archivo de extensión csv.
Precondiciones	El profesor ha iniciado sesión en la aplicación.

Caso de uso	Ver grupo
Actores	Profesor
Propósito	El profesor puede ver los grupos que ha creado.
Resumen	El profesor podrá visualizar toda la información de los grupos que haya creado.
Precondiciones	El profesor ha iniciado sesión en la aplicación y hay algún grupo creado.

Caso de uso	Borrar grupo
Actores	Profesor
Propósito	El profesor puede borrar los grupos que ha creado.
Resumen	El profesor podrá borrar los grupos que haya creado de uno en uno.
Precondiciones	El profesor ha iniciado sesión en la aplicación y hay algún grupo creado.

Caso de uso	Crear tarea
Actores	Profesor
Propósito	El profesor puede crear tareas que harán los alumnos.
Resumen	El profesor creará una tarea, del tipo que desee, y se la asignará a un grupo de alumnos.
Precondiciones	El profesor ha iniciado sesión en la aplicación y hay algún grupo creado al que asignarle la tarea.

Caso de uso	Ver tarea
Actores	Profesor
Propósito	El profesor podrá ver las tareas que ha creado
Resumen	El profesor podrá elegir a un grupo y ver todas las tareas que ese grupo tiene asignadas.
Precondiciones	El profesor ha iniciado sesión en la aplicación, hay algún grupo creado y ese grupo tiene tareas asignadas.

Caso de uso	Corregir tarea
Actores	Profesor
Propósito	El profesor podrá corregir una tarea
Resumen	El profesor podrá ver la tarea que ha realizado un alumno y podrá puntuarla
Precondiciones	El profesor ha iniciado sesión en la aplicación, hay algún grupo creado y ese grupo tiene tareas asignadas.

Caso de uso	Consulta pseudónimo
Actores	Profesor
Propósito	El profesor podrá saber que alumno se corresponde con un pseudónimo.
Resumen	El profesor podrá introducir un pseudónimo y obtener la información de que alumno es.
Precondiciones	El profesor ha iniciado sesión en la aplicación y hay algún grupo creado.

Caso de uso	Hacer tarea
Actores	Alumno
Propósito	El alumno podrá realizar una tarea
Resumen	Para realizar una tarea el alumno vera toda la información de la tarea, se descargará el archivo con los datos de la tarea y subirá otro archivo con la resolución.
Precondiciones	El alumno ha iniciado sesión en la aplicación y tiene tareas creadas

Caso de uso	Corregir tarea
Actores	Alumno
Propósito	El alumno podrá corregir una tarea
Resumen	El alumno podrá corregir la tarea que otro alumno haya realizado previamente
Precondiciones	El alumno ha iniciado sesión en la aplicación y tiene tareas a corregir

Caso de uso	Ver notas
Actores	Alumno
Propósito	El alumno podrá ver las notas de sus tareas
Resumen	El alumno vera las notas de las tarea que haya realizado y estén corregidas
Precondiciones	El alumno ha iniciado sesión en la aplicación y tiene tareas asignadas

4. Diseño

La herramienta que desarrollaremos será una aplicación web. A la hora de implementar aplicaciones web, en la actualidad, es muy común utilizar la arquitectura software modelo-vista-controlador (MVC). Esta arquitectura separa los datos y la lógica de negocio de la aplicación de la interfaz de usuario.

Para almacenar toda la información que tendrá que gestionar nuestra aplicación, utilizaremos una base de datos relacional.

4.1 Modelo-vista-controlador (MVC)

La arquitectura MVC propone la construcción de tres componentes distintos, que son el modelo, la vista y el controlador. El uso de este patrón de arquitectura software nos permite reutilizar código y separar conceptos. Estas características nos facilitarán la tarea del desarrollo y el futuro mantenimiento de la aplicación [4].

El MVC consta de los siguientes componentes:

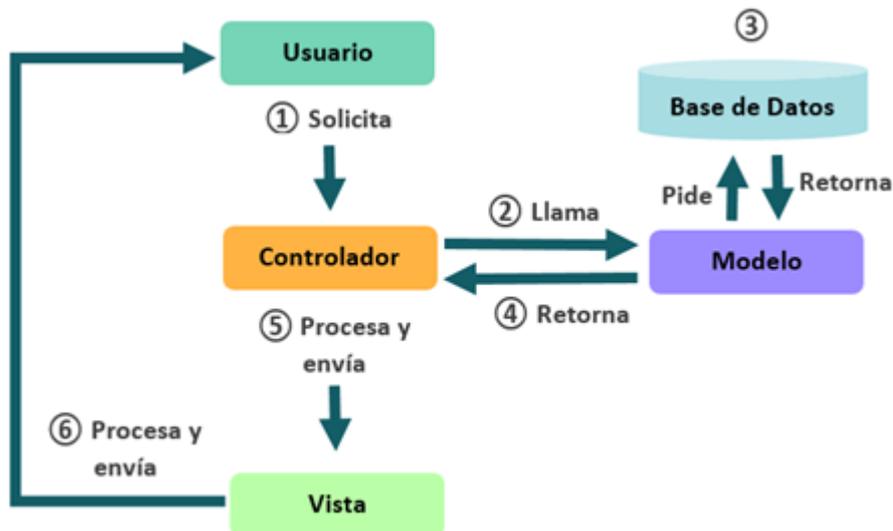
- **Modelo:** es la representación de la información con la que opera el sistema. En otras palabras, gestiona todos los accesos a la información almacenada en la base de datos. Envía a la vista la información que sea demandada para ser mostrada. Las peticiones de acceso o edición de la información llegan a través del modelo.
- **Vista:** es la parte visible de la aplicación. Contiene el código necesario para visualizar las interfaces de usuario necesarias.
- **Controlador:** es la capa que enlace la vista con el modelo. El controlador no manipula los datos, solo se encarga de enlazar la vista con el controlador.

En resumen, todo lo relacionado con las interfaces de usuario estará dentro de las vistas. Todos los accesos a la base de datos se harán desde el modelo, y el controlador será el intermediario que unirá todo.

El funcionamiento del MVC es el siguiente:

- El usuario accede a un controlador y este le genera la vista correspondiente.
- El usuario interactúa con la interfaz y genera algún evento.
- El controlador recibe la notificación de la acción realizada por el usuario.
- El controlador accede al modelo para obtener o actualizar la información necesaria en la base de datos.

- Los datos son enviados a la vista para que los muestre al usuario, conforme a la acción que hubiera realizado.
- La interfaz de usuario quedara a la espera de nuevas interacciones del usuario.



<http://rodrigr.com/blog/modelo-vista-controlador/>

4.1.1 Vista

Uno de los motivos por los que hemos elegido la arquitectura MVC es porque permite reutilizar gran parte del código. La aplicación web estará estructurada en 3 partes. La parte superior, a la que llamaremos *header*, la parte intermedia a la que llamaremos *body* y la parte inferior que llamaremos *footer*. Estructurar la aplicación de esta manera nos permitirá reutilizar el código de *header* y del *footer* para todas las vistas.

Tanto el *header* como el *footer* serán comunes para todas las interfaces de la aplicación. Dentro del *body* estará todo el contenido de cada vista. A continuación se incluyen los diseños del *header*, un ejemplo de un *body* y el *footer*.

Header

Usuario

Coevaluación entre pares

Body

En el caso del *body* se ha diseñado la interfaz correspondiente a la creación de una tarea a modo de ejemplo.

Option 1	Título de la tarea: <input type="text"/>
Option 2	Decripción:
Option 3	<input type="text"/>
	Elegir grupo: <input type="text" value="Grupo"/>
	Fecha de entrega: <input type="text" value="6/28/2017"/>
	<input type="button" value="Importar archivo"/>
	<input type="button" value="Guardar tarea"/>

Footer

4.1.2 Controlador

En cuanto al diseño de controlador, la estructura que seguiremos será la del controlador único o `homeController`. Este tipo de controlador gestiona todas las llamadas al resto de controladores.

Por tanto tendremos un controlador que se encargará de realizar las llamadas necesarias a los controladores correspondientes. Tendremos un controlador por cada funcionalidad o requisito que tenga que llevar a cabo nuestra aplicación. Cada controlador se encargará de montar la vista correspondiente y gestionará la petición de los datos al modelo en función de los eventos que reciba desde la vista.

4.1.3 Modelo

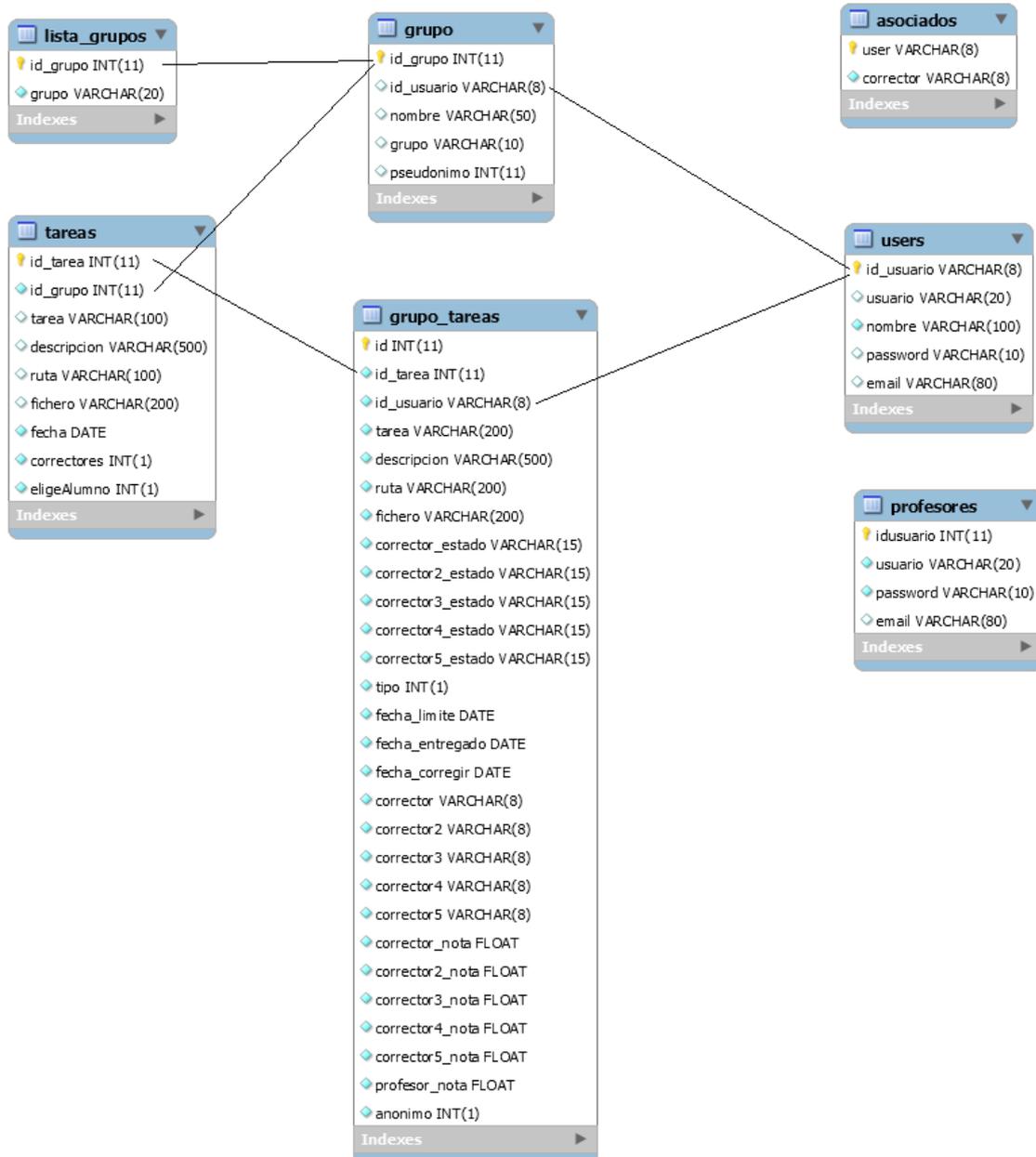
En la arquitectura MVC el modelo es el encargado de gestionar todos los datos. En nuestra aplicación el modelo será el único punto desde el que accedamos a la base de datos, sea para la operación que sea. El modelo por tanto, constará de todas las funciones necesarias para hacer todos los intercambios de información con la base de datos.

4.2 Base de datos relacional

La aplicación web que implementaremos requiere de una base de datos que almacene toda la información necesaria. Para diseñar la base de datos seguiremos el modelo relacional.

El modelo relacional fue inventado por IBM en 1970. Este modelo se basa en el uso de relaciones. Este tipo de base de datos está formado por tablas, y estas tablas esta compuestas de registros. A continuación podemos ver el esquema de la base de datos que hemos diseñado para la aplicación.

Sitio web para la gestión de evaluación entre pares con pseudónimos



Estos son los detalles más importantes del diseño de la base de datos.

Los campos unidos con las flechas son los campos que servirán para relacionar la información entre las distintas tablas:

- id_usuario VARCHAR(8): mediante este campo se identificarán a los alumnos.
- id_grupo INT(11): campo utilizado para identificar a los grupos.
- id_tarea INT(11): campo utilizado para identificar cada tarea.

Tablas:

- Tabla users: esta tabla almacenará la información de todos los usuarios que tengan acceso a la aplicación.

- Tablas profesores: esta tabla almacenará la información de todos los profesores que tengan acceso a la aplicación.
- Tabla lista_grupos: con esta tabla almacenaremos todos los grupos que tengamos creados en nuestra aplicación.
- Tabla tareas: tabla para guardar todas las tareas que han sido creadas. Tendremos una entrada por cada tarea.
- Tabla grupo: esta es la tabla en la que almacenaremos la información de todos los alumnos que pertenecen a un grupo. En este caso la tabla tiene el nombre grupo porque es el diseño, pero en la aplicación real esta tabla tendrá el nombre de cada grupo que cree el profesor. Este tipo de tabla por tanto se creara cuando el profesor cree un nuevo grupo, y habrá una tabla por cada grupo creado.
- Tabla grupo_tareas: en esta tabla almacenaremos toda la información de cada tarea de cada usuario. Al igual que con la tabla grupo, esta tabla se creara con el nombre del grupo que haya elegido el profesor en el momento de crearlo.
- Tabla asociados: esta tabla la utilizaremos para implementar la tarea en la que el profesor elige a los correctores manualmente.

5. Implementación

Una vez tenemos diseñado todo lo necesario, podemos proceder a implementar la aplicación. Para explicar la implementación que llevaremos a cabo primero explicaremos las tecnologías que utilizaremos, después seguiremos con las herramientas que nos permitirán desarrollar la aplicación para acabar explicando la estructura de ficheros que conformará la aplicación.

5.1 Tecnologías

La aplicación que se ha desarrollado es una aplicación web que requiere de una base de datos para almacenar la información. La base para desarrollar este tipo de aplicaciones son los lenguajes HTML, PHP, CSS y SQL.

A continuación detallaremos las versiones de cada lenguaje que utilizaremos y porque hemos elegido estos lenguajes en concreto.

5.1.1 HTML5

HTML es el lenguaje utilizado a día de hoy para el desarrollo de aplicaciones o páginas web. Este lenguaje es interpretado por los navegadores web, que se encargarán de leer este código y montar la página que visualizará el usuario [5].

En nuestro caso utilizaremos la versión HTML5, ya que es la versión más actual de este lenguaje.

Con HTML hemos implementaremos todas las vistas de nuestra aplicación.

5.1.2 CSS3

CSS es un lenguaje de diseño gráfico para crear la presentación de un documento. En nuestro caso, este lenguaje nos permitirá establecer el diseño visual de nuestra aplicación. Este es el lenguaje utilizado por las páginas web para tener una apariencia más atractiva de cara al usuario [6].

Para el desarrollo de nuestra aplicación emplearemos la versión CSS3, ya que es la última versión de este lenguaje y es la versión más completa.

Junto con HTML, CSS nos ayudará a dotar a las vistas de una apariencia más atractiva, clara y estructurada.

5.1.3 PHP

PHP es un lenguaje de código abierto, cuya principal característica es que puede ser incrustado dentro de código HTML. Por este motivo, actualmente es un lenguaje muy empleado en el desarrollo de aplicaciones web. PHP es ejecutado por el servidor, al contrario que HTML [7].

El lenguaje PHP también permite hacer conexiones con una base de datos, haciendo todas las manipulaciones de datos necesarias en las tablas.

Para el desarrollo de nuestra aplicación emplearemos este lenguaje para implementar los controladores. La versión de PHP que utilizaremos es la 5.6.24. Esta no es la última versión que existe de PHP, pero la hemos elegido porque es una versión compatible con muchos servidores. Las últimas versiones de PHP suelen dar problemas de configuración en los servidores, por eso hemos elegido una versión más antigua ya que es más probable que no surjan problemas.

5.1.4 SQL

SQL es un lenguaje que permiten tanto hacer consultas como hacer cambios en una base de datos relacional. Concretamente este lenguaje permite la inserción de datos, consultas, actualizaciones y borrado, la creación y modificación de las tablas y el control de acceso a los datos [8].

La última versión de este lenguaje es la SQL:2012, que es la que utilizaremos. Este lenguaje lo emplearemos en el modelo, junto con PHP, para realizar las acciones necesarias sobre la base de datos.

5.1.5 Bootstrap

Bootstrap es un framework de código abierto que facilita el diseño de aplicaciones web. Actualmente bootstrap es utilizado principalmente para hacer aplicaciones web que se adapten a todas las resoluciones de pantalla que existen en la actualidad [9].

Utilizaremos Bootstrap junto con HTML y CSS para implementar las vistas de nuestra aplicación. La versión que utilizaremos es la 3.3.7.

5.2 Herramientas

A continuación detallaremos las herramientas que emplearemos para llevar a cabo el desarrollo de nuestra aplicación.

5.2.1 Xampp

La aplicación que vamos a desarrollar es una aplicación web, y como tal requiere de un servidor en el que estar alojada. En nuestro caso utilizaremos el software libre Xampp. Xampp es un paquete de instalación que nos permite instalar en el equipo que deseemos varias herramientas relacionadas con el desarrollo de aplicaciones web.

En nuestro caso hemos elegido Xampp porque nos permite instalar de una manera sencilla un servidor web Apache y un sistema de gestión de base de datos MySQL. Con esta herramienta el desarrollo de la aplicación lo llevaremos a cabo de una manera local, desde nuestro equipo.

5.2.1 Notepad++

Notepad++ es un editor de texto y de código fuente libre. Es un editor potente que ofrece una amplia gama de posibilidades ya que soporta un gran número de lenguajes de programación.

Hemos elegido este editor de texto porque soporta todos los lenguajes de programación que vamos a utilizar para desarrollar la aplicación (HTML, PHP, SQL y CSS), además otro motivo por el que hemos elegido este editor de código es que contamos con una amplia experiencia con este editor.

La versión que hemos utilizado es la Notepad++ v7.3.3. No es la última versión de este editor, pero la hemos elegido porque ya la conocíamos y ofrece un gran rendimiento.

5.2.3 HeidiSQL

HeidiSQL es un software libre y de código abierto. Este software es un cliente de bases de datos, que permite conectarse a un servidor MySQL.

Xampp ya viene con phpMyAdmin instalado, que es otra herramienta que permite la gestión de una base de datos, pero por la experiencia que tenemos trabajando con HeidiSQL, consideramos que es un mejor gestor de bases de datos, más rápido, completo y sencillo, y por eso lo hemos elegido.

La versión que utilizaremos es la 9.4.0.5150. No es la última versión, pero es de las más actuales. Este software sufre constante actualizaciones.

5.2.4 Google Chrome, Mozilla Firefox e Internet Explorer

A la hora de desarrollar una aplicación web es necesario disponer de un navegador web para visualizarla. En nuestro caso, para llevar a cabo la implementación de la aplicación web utilizaremos indistintamente tanto Google Chrome como Mozilla Firefox, mientras que el navegador Internet Explorer lo utilizaremos básicamente para testear la aplicación. Los tres navegadores tienen por defecto herramientas que permiten el desarrollo de aplicaciones web.

La versión de Google Chrome es la 58.0.3029.100, mientras que la de Firefox es la 53.0.3. La versión de Internet Explorer es la 11.1358.14393.0

Hemos elegido estos tres navegadores porque son los navegadores web utilizados por la mayoría de usuarios. Además tanto Chrome como Firefox cuentan con unas buenas herramientas para el desarrollo web.

5.2.5 GIMP 2

GIMP es un programa de edición de imágenes. Es un software libre y gratuito. Utilizaremos la versión GIMP 2.8.22.

Hemos elegido este editor de imágenes porque es un editor simple y libre. Utilizaremos este programa para editar las imágenes que vayamos a necesitar tanto a la hora del desarrollo como a la hora de la redacción de la memoria.

5.3 Estructura de los ficheros y directorios

La estructura de los ficheros y directorios que seguiremos estará directamente relacionada con la arquitectura MVC que hemos elegido. Recordemos que esta arquitectura divide la estructura de la aplicación en tres bloques: el modelo, la vista y el controlador. Por tanto tendremos por un lado todos los ficheros relacionados con las vistas, por otro lado los ficheros de los controladores y por último los ficheros pertenecientes al modelo.

En esta primera imagen podemos ver la estructura general de directorios que hemos utilizado.

Nombre	Fecha de modifica...	Tipo	Tamaño
controllers	21/06/2017 18:01	Carpeta de archivos	
css	21/06/2017 18:01	Carpeta de archivos	
fonts	21/06/2017 18:01	Carpeta de archivos	
img	21/06/2017 18:01	Carpeta de archivos	
includes	21/06/2017 18:04	Carpeta de archivos	
tareas	21/06/2017 18:01	Carpeta de archivos	
templates	21/06/2017 18:01	Carpeta de archivos	
index	07/03/2017 17:15	Archivo PHP	1 KB

Vamos a comentar brevemente los directorios y el archivo que vemos en la imagen. Los directorios templates, includes, controllers y tareas los explicaremos de forma más detallada posteriormente, ya que son los más importantes.

- **Index.php:** este es el archivo al que se accede cuando un usuario entra a nuestra aplicación a través del navegador. Este archivo es el que actúa como controlador único de la aplicación y por tanto gestiona todas las llamadas a la aplicación.
- **css:** en este directorio se encuentran todos los archivos .css. Estos son los archivos que hemos utilizado para darle estilo a nuestra aplicación. Dentro de este directorio también se encuentran los archivos pertenecientes al framework bootstrap.
- **fonts:** dentro de este directorio están las fuentes que hemos utilizado en la aplicación.
- **img:** esta carpeta contiene todas las imágenes que necesitamos visualizar desde la aplicación web.

Ahora vamos a explicar de una manera más extensa los principales directorios de nuestra aplicación.

Directorio controllers:

Como hemos definido en el punto el controlador es el encargado de montar la vista correspondiente y gestionar las peticiones de los datos al modelo en función de los eventos que reciba desde la vista.

Lo normal es tener un controlador por cada requisito que hemos definido en la definición de requisitos, aunque para implementar algunos de los requisitos más complejos hemos tenido que utilizar más de un controlador.

En la imagen podemos ver todos los controladores que hemos implementado. Los controladores están implementados con el lenguaje PHP.

Nombre	Fecha de modifica...	Tipo	Tamaño
 alumno	02/05/2017 17:43	Archivo PHP	1 KB
 consultaPseudonimo	04/04/2017 18:21	Archivo PHP	1 KB
 correcciones	05/06/2017 17:42	Archivo PHP	2 KB
 corregir	06/06/2017 17:54	Archivo PHP	2 KB
 corregirEligeAlumno	06/06/2017 17:17	Archivo PHP	1 KB
 crearGrupo	22/05/2017 20:14	Archivo PHP	2 KB
 crearTareaCorrectorAleatorio	25/05/2017 17:02	Archivo PHP	2 KB
 CrearTareaElegiendoCorrector	02/06/2017 17:16	Archivo PHP	4 KB
 crearTareaSinCorrector	02/06/2017 17:33	Archivo PHP	2 KB
 eliminarGrupo	23/05/2017 17:38	Archivo PHP	1 KB
 estadoTarea	24/05/2017 17:46	Archivo PHP	2 KB
 homeController	14/03/2017 16:36	Archivo PHP	1 KB
 loginAlumno	02/05/2017 18:16	Archivo PHP	1 KB
 loginComun	02/05/2017 18:13	Archivo PHP	1 KB
 loginProfesor	02/05/2017 18:15	Archivo PHP	1 KB
 logout	23/03/2017 9:25	Archivo PHP	1 KB
 notas	06/06/2017 18:12	Archivo PHP	1 KB
 profesor	02/05/2017 18:13	Archivo PHP	1 KB
 recuperarContrasena	10/05/2017 17:03	Archivo PHP	1 KB
 tarea	02/06/2017 18:09	Archivo PHP	2 KB
 tareasCorregirProfe	06/06/2017 18:07	Archivo PHP	1 KB
 verGrupo	23/05/2017 17:35	Archivo PHP	1 KB
 verTareas	06/06/2017 17:07	Archivo PHP	1 KB
 verTareasProfe	24/05/2017 17:55	Archivo PHP	1 KB

Directorio templates:

El directorio templates es el directorio que contiene todas las vistas de la aplicación. Recordemos que en la arquitectura MVC la vista es la parte correspondiente a la interfaz de usuario, será lo que el usuario verá de nuestra aplicación. Las vistas las hemos implementado con el lenguaje HTML, aunque también se ha incrustado PHP en alguna ocasión.

Nombre	Fecha de modifica...	Tipo	Tamaño
 consultaPseudonimo	04/04/2017 18:12	Archivo PHP	1 KB
 correcciones	02/05/2017 16:33	Archivo PHP	1 KB
 corregir	06/06/2017 16:48	Archivo PHP	1 KB
 corregirEligeAlumno	05/06/2017 18:52	Archivo PHP	1 KB
 crearGrupo	22/05/2017 20:12	Archivo PHP	1 KB
 crearTareaCorrectorAleatorio	25/05/2017 17:11	Archivo PHP	2 KB
 CrearTareaEligiendoCorrector	25/05/2017 17:11	Archivo PHP	2 KB
 crearTareaSinCorrector	02/06/2017 17:22	Archivo PHP	1 KB
 elegirCorrector	06/06/2017 17:41	Archivo PHP	3 KB
 eligeAlumno	05/06/2017 18:20	Archivo PHP	1 KB
 eliminarGrupo	01/06/2017 16:52	Archivo PHP	1 KB
 estadoTarea	06/06/2017 18:01	Archivo PHP	2 KB
 footer	23/03/2017 9:10	Archivo PHP	1 KB
 head	29/05/2017 17:23	Archivo PHP	3 KB
 head_login	31/05/2017 16:20	Archivo PHP	1 KB
 headTareasProfe	24/05/2017 17:48	Archivo PHP	1 KB
 homeController	23/03/2017 10:23	Archivo PHP	1 KB
 inicioAlumno	13/04/2017 10:45	Archivo PHP	1 KB
 inicioProfesor	22/04/2017 10:17	Archivo PHP	1 KB
 loginComun	13/04/2017 11:15	Archivo PHP	1 KB
 menuAlumno	24/05/2017 16:52	Archivo PHP	1 KB
 menuProfesor	31/05/2017 16:29	Archivo PHP	2 KB
 notas	24/05/2017 17:55	Archivo PHP	1 KB
 recuperarContrasena	13/04/2017 12:22	Archivo PHP	1 KB
 tabla_corregir	31/05/2017 18:31	Archivo PHP	1 KB
 tabla_tarea	16/05/2017 16:46	Archivo PHP	1 KB
 tablaEligeAlumno	05/06/2017 17:59	Archivo PHP	1 KB
 tablaTareasProfe	24/05/2017 17:52	Archivo PHP	1 KB
 tarea	01/06/2017 17:11	Archivo PHP	1 KB
 verGrupo	01/06/2017 16:58	Archivo PHP	1 KB
 verTareas	16/05/2017 17:04	Archivo PHP	1 KB
 verTareasACorregir	06/06/2017 17:27	Archivo PHP	2 KB
 verTareasProfe	23/05/2017 17:40	Archivo PHP	1 KB

En la imagen anterior podemos ver todas las vistas que se han implementado. Hay algunas como pueden ser las vistas *head* o *footer* que son reutilizadas por la mayoría de controladores, mientras que otras son utilizadas únicamente por un controlador, como puede ser la vista *notas*.

Lo normal es que un controlador utilice varias vistas conjuntamente para montar una interfaz de usuario. Por ejemplo el controlador alumno hace uso de las vista head, menuAlumno, inicioAlumno y footer para montar una única interfaz.

Directorio includes:

Dentro de este directorio encontraremos el modelo. Según la arquitectura MVC el modelo es el único punto de la aplicación desde el que se accederá a la base de datos para hacer las operaciones necesarias sobre los datos.

Nombre	Fecha de modifica...	Tipo	Tamaño
 Modelo	07/06/2017 15:44	Archivo PHP	31 KB

Como vemos en la imagen dentro del directorio includes tenemos únicamente un archivo de nombre modelo. Este es un archivo con extensión PHP. Este fichero es uno de los más importantes de nuestra aplicación, ya que dentro de él están todas las funciones que trabajan con la base de datos. Desde este archivo se hacen las operaciones de consulta, inserción, actualización, borrado...

Para hacernos una idea de la magnitud y la importancia de este fichero cabe mencionar que dentro de él tenemos implementadas 64 funciones, las cuales suman un total de 944 líneas de código.

Directorio tareas

Este es otro de los directorios más importantes de la aplicación. Dentro de esta carpeta tendremos un directorio por cada grupo que un profesor haya creado.

Nombre	Fecha de modifica...	Tipo	Tamaño
 grupo_tareas	21/06/2017 18:01	Carpeta de archivos	

Como vemos en la imagen, en este caso tenemos el directorio grupo_tareas. Este es un ejemplo de los directorios que se crearan conforme un profesor cree un grupo en la aplicación. Los directorios se crearan con el nombre que el profesor lo cree, por ejemplo 1a_tareas.

La carpeta que se crea para cada grupo es el directorio que utilizaremos para guardar todos los archivos correspondientes a las tareas que un profesor asigne a un grupo. Guardaremos tanto el archivo que suba el profesor con los datos de una tarea como los archivos que suban los alumnos con sus tareas realizadas. Dentro del directorio de cada grupo se almacenarán los archivos correspondientes a todas las tareas de ese grupo.

6. Pruebas

La fase de pruebas se dedicará a probar el correcto funcionamiento de la herramienta. Este punto se dividirá en dos apartados, por un lado tendremos las pruebas de uso y por otro las pruebas de visualización.

6.1 Pruebas de uso

Hemos llevado a cabo un desarrollo de la aplicación siguiendo los requisitos especificados. En el momento de la implementación hemos probado el funcionamiento de cada requisito de una manera exhaustiva y minuciosa, de tal forma que cuando hemos terminado de implementar un requisito teníamos la seguridad de que funcionaba a la perfección y cumplía con todas las especificaciones que habíamos definido.

A continuación incluiremos unas capturas de pantalla mostrando el correcto funcionamiento de los principales requisitos de nuestra aplicación.

6.1.1 Control de acceso

El control de acceso a la aplicación es un requisito importante, solo los usuarios registrados podrán acceder a nuestra aplicación. En la siguiente imagen vemos la interfaz con la que se produce este control de usuarios.

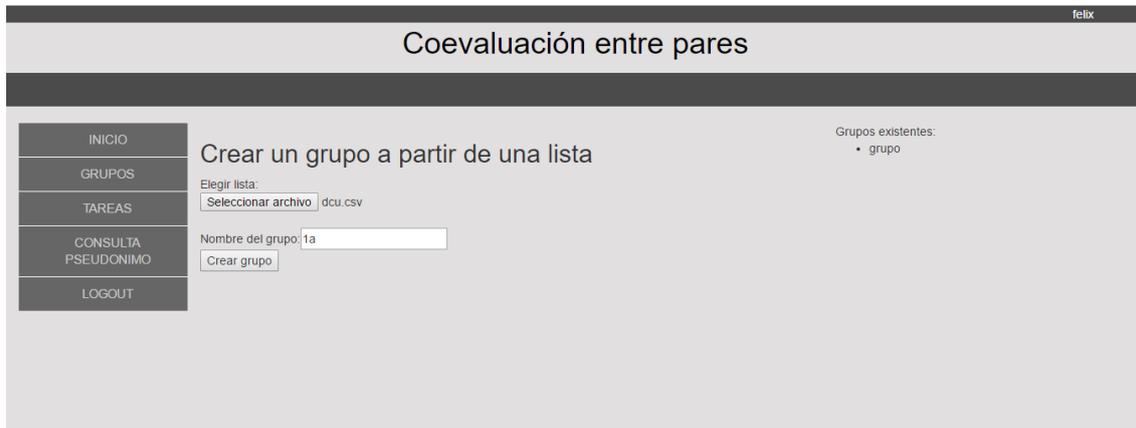


La imagen muestra una interfaz web con un encabezado que dice "Coevaluación entre pares". Debajo de esto, hay un formulario de login titulado "login Profesor". El formulario contiene dos campos de entrada: "Username" y "Password", cada uno con un botón de "login" a su derecha. Debajo de los campos de entrada, hay un enlace azul que dice "Recuperar contraseña".

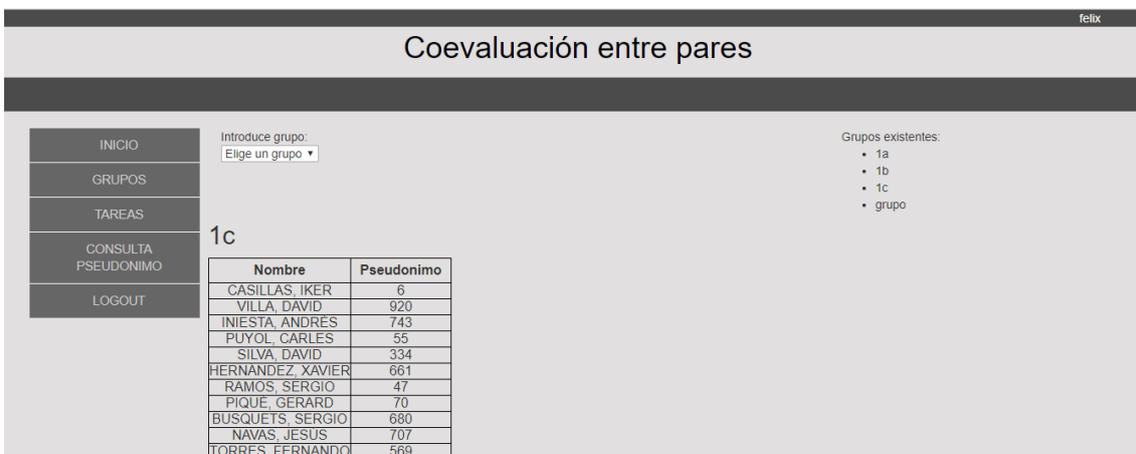
Esta imagen en concreto es la del acceso de los profesores, la de los alumnos es idéntica.

6.1.2 Crear un grupo

Para poder crear tareas y que los alumnos las corrijan será necesario haber creado el grupo de alumnos previamente.



Como vemos en la imagen para crear el grupo subimos un archivo .csv y elegimos el nombre con el que queremos crearlo. La prueba de que este grupo se crea correctamente es que si vamos a la pestaña “Ver grupos” podemos visualizarlo como se ve a continuación.



6.1.3 Crear tareas

Como se ha mencionado anteriormente, el requisito más importante de nuestra aplicación es permitir la creación de diferentes tipos de tareas. Una de las tareas que se pueden crear con nuestra aplicación es la tarea en la que varios alumnos corrigen de forma anónima a otro.

INICIO

GRUPOS

TAREAS

CONSULTA PSEUDONIMO

LOGOUT

Crear una tarea asignando un corrector aleatorio

Tarea:

Corrección no anónima

Descripción:

Toda la información necesaria para realizar esta tarea esta especificada en el archivo.

Fecha entrega:

Fecha correccion:

Subir pdf:

tarea.pdf

Grupo:

Número de correctores:

En la captura de pantalla vemos como se crea este tipo de tarea, el resto siguen un procedimiento similar, aunque puede variar alguna opción. Cuando se cree la tarea el profesor verá todos los datos de la tarea como vemos a continuación.

Una tarea tiene 3 estados:
 - Pendiente: el usuario todavía no la ha hecho
 - Enviado: el usuario ha enviado la tarea, si la tarea tiene correctores seguira en este estado hasta que la corrijan todos
 - Corregido: todos los correctores han corregido la tarea

Usuario	Nombre	Estado	Nota_1	Nota_2	Nota_3	Nota_4	Nota_5	Nota profesor	Nota media	
12345678	CASILLAS, IKER	enviado	NAVAS, JESÚS	TORRES, FERNANDO	BUSQUETS, SERGIO	HERNÁNDEZ, XAVIER	RAMOS, SERGIO	6.5	6.93	Corregir
			5.4	7	8	8.7	6			
87654321	VILLA, DAVID	enviado	PUYOL, CARLES	TORRES, FERNANDO	NAVAS, JESÚS	BUSQUETS, SERGIO	INIESTA, ANDRÉS	7.3	6.78	Corregir
			7.5	5	6.9	6	8			
12348765	INIESTA, ANDRÉS	enviado	VILLA, DAVID	CASILLAS, IKER	PIQUÉ, GERARD	SILVA, DAVID	HERNANDEZ, XAVIER	8.4	7.52	Corregir
			8.2	6.8	7.4	7.3	9			
11223344	PUYOL, CARLES	pendiente	INIESTA, ANDRÉS	RAMOS, SERGIO	HERNÁNDEZ, XAVIER	NAVAS, JESÚS	PIQUÉ, GERARD	0	0	Corregir
			0	0	0	0	0			
22334455	SILVA, DAVID	pendiente	PIQUÉ, GERARD	RAMOS, SERGIO	BUSQUETS, SERGIO	TORRES, FERNANDO	NAVAS, JESÚS	0	0	Corregir
			0	0	0	0	0			

Como vemos, las notas de los correctores están a 0 porque todavía no se ha corregido la tarea, ya que aún no ha sido enviada por el alumno.

La información que el profesor visualiza en todos los tipos de tareas es similar, lo único que puede variar es el número de correctores.

6.1.4 Realizar tarea alumno

Cuando un alumno entre en su cuenta, si tiene tareas asignadas verá lo siguiente

The screenshot shows a web interface titled 'Coevaluación entre pares'. On the left is a navigation menu with buttons: INICIO, VER TAREAS, TAREAS A CORREGIR, NOTAS, and LOGOUT. The main content area displays the group name '1a' and a table of tasks:

Tarea	Archivo	Fecha entrega	Estado	Ver tarea
Ejercicio 1	tarea.pdf	2017-06-23	pendiente	Ver tarea

Below this, the group name 'grupo' is shown, followed by another table:

Tarea	Archivo	Fecha entrega	Estado	Ver tarea
tarea1	ieeee830.pdf	2017-06-22	pendiente	Ver tarea

En la imagen podemos apreciar que este alumno pertenece a dos grupos, y que tiene una tarea pendiente en cada grupo. Estas tareas han sido creadas anteriormente por un profesor. Si el alumno entrara en ver tarea podría realizar dicha tarea.

The screenshot shows the details of a task titled 'Tarea Ejercicio 1'. The navigation menu is the same as in the previous screenshot. The main content area displays the following information:

- Alumno: anonimo
- Descripcion: Toda la información necesaria para realizar esta tarea esta especificada en el archivo.
- Archivo: tarea.pdf
- Estado: pendiente
- Fecha limite: 2017-06-23
- Descargar archivo: [Descargar](#)
- Selección de archivo: Ningún archivo seleccionado
- Subir tarea:

Como vemos esta es una tarea anónima, para hacerla el alumno tiene que subir un fichero con la resolución.

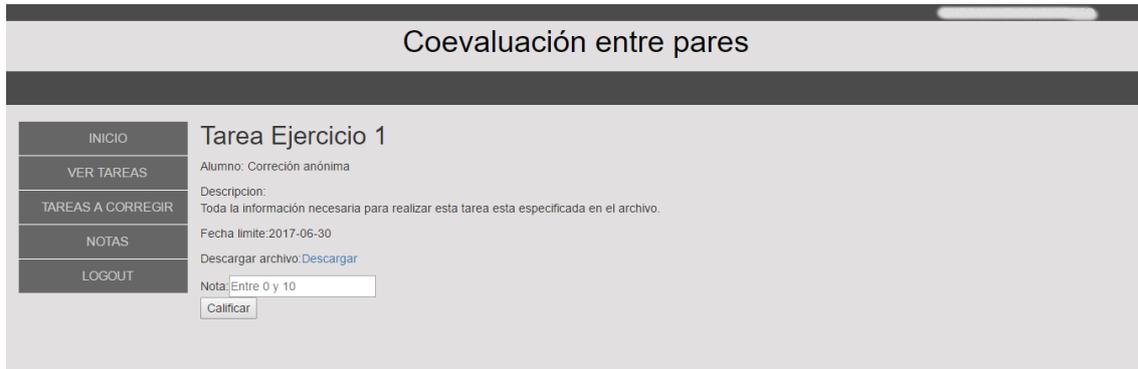
6.1.5 Corregir tarea

Si un alumno ha entregado la tarea, los alumnos encargados de corregirla la verán en "Tareas a corregir"

The screenshot shows the 'Tareas para corregir' section of the website. The navigation menu is the same. The main content area displays the group name '1a' and a table of tasks to be corrected:

Tarea	Fecha entrega	Grupo	Descargar
Ejercicio 1	2017-06-30	1a	Corregir

Si entran en el enlace “Corregir”, podrán ver el archivo con la resolución de la tarea y calificarla.



The screenshot shows a web interface titled "Coevaluación entre pares". On the left, there is a vertical navigation menu with buttons: "INICIO", "VER TAREAS", "TAREAS A CORREGIR", "NOTAS", and "LOGOUT". The main content area is titled "Tarea Ejercicio 1" and contains the following information:

- Alumno: Corrección anónima
- Descripción: Toda la información necesaria para realizar esta tarea esta especificada en el archivo.
- Fecha limite: 2017-06-30
- Descargar archivo: [Descargar](#)
- Nota:
-

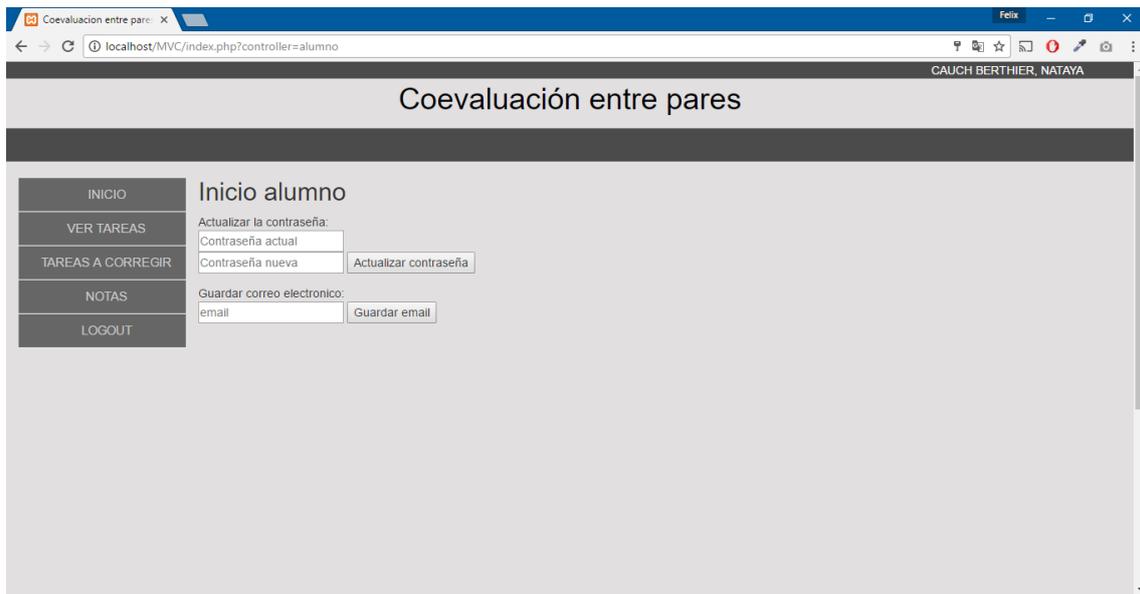
6.2 Pruebas de visualización

Dentro de las pruebas de visualización distinguiremos entre dos tipos de pruebas, por un lado probaremos la aplicación en los navegadores más comunes, y por otro lado, la probaremos a distintas resoluciones.

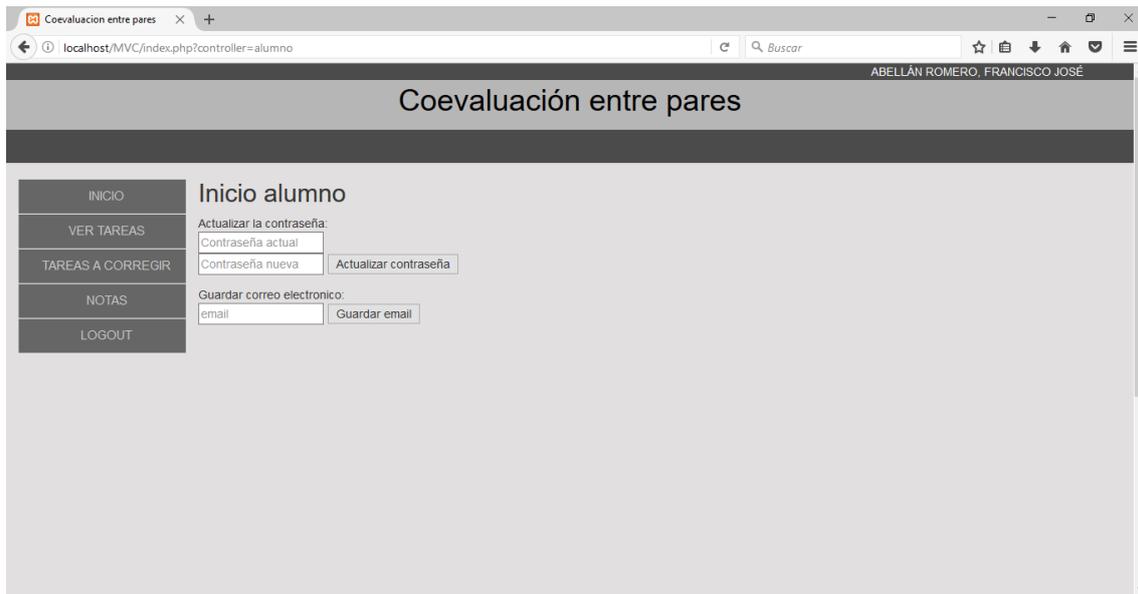
6.2.1 Pruebas en los navegadores más comunes

La aplicación ha sido desarrollada para funcionar correctamente en los navegadores Google Chrome, Mozilla Firefox e Internet Explorer. A continuación probaremos que la visualización es correcta en los navegadores mencionados. Incluiremos dos imágenes por cada navegador, una correspondiente a la interfaz del profesor y otra correspondiente a la interfaz del alumno.

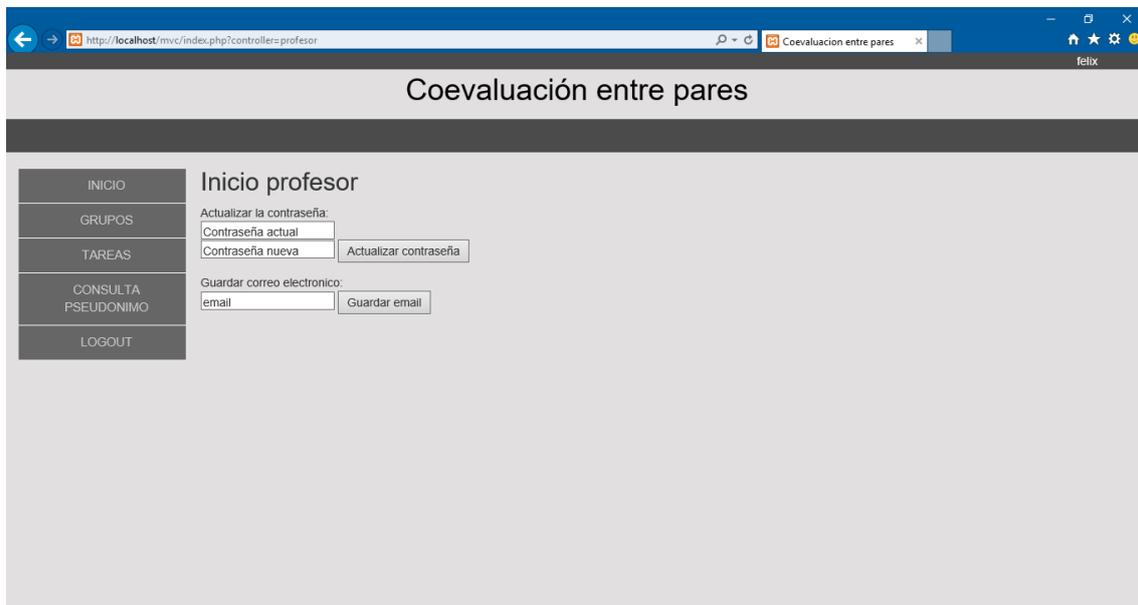
Google Chrome



Mozilla Firefox



Internet Explorer



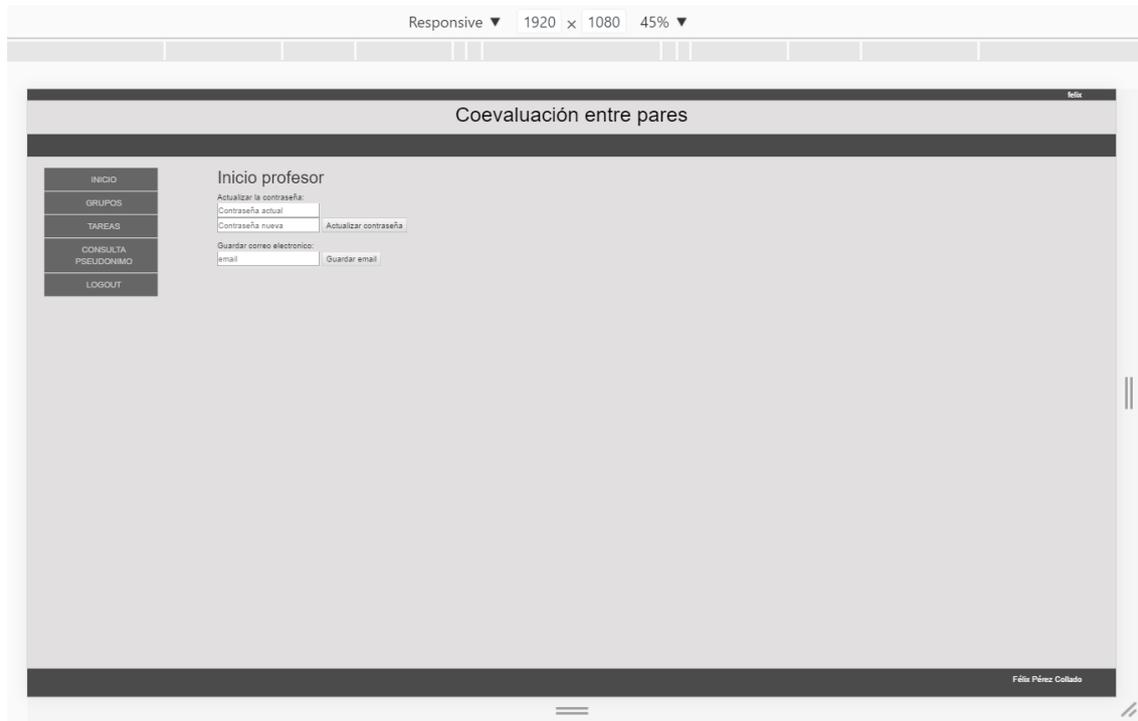
6.2.2 Prueba de distintas resoluciones

Actualmente es fundamental que una aplicación web se visualice correctamente a distintos tipos de resoluciones. Este es el principal motivo por el que hemos utilizado el framework bootstrap. Las pruebas las realizaremos sobre los tamaños de pantalla más utilizados a día de hoy. Para probar las distintas resoluciones utilizaremos la herramienta de desarrollador de Chrome que permite establecer distintas resoluciones de pantalla.

Sitio web para la gestión de evaluación entre pares con pseudónimos

A continuación se muestran capturas de pantalla de las resoluciones más comunes utilizadas a día de hoy.

1920x1080

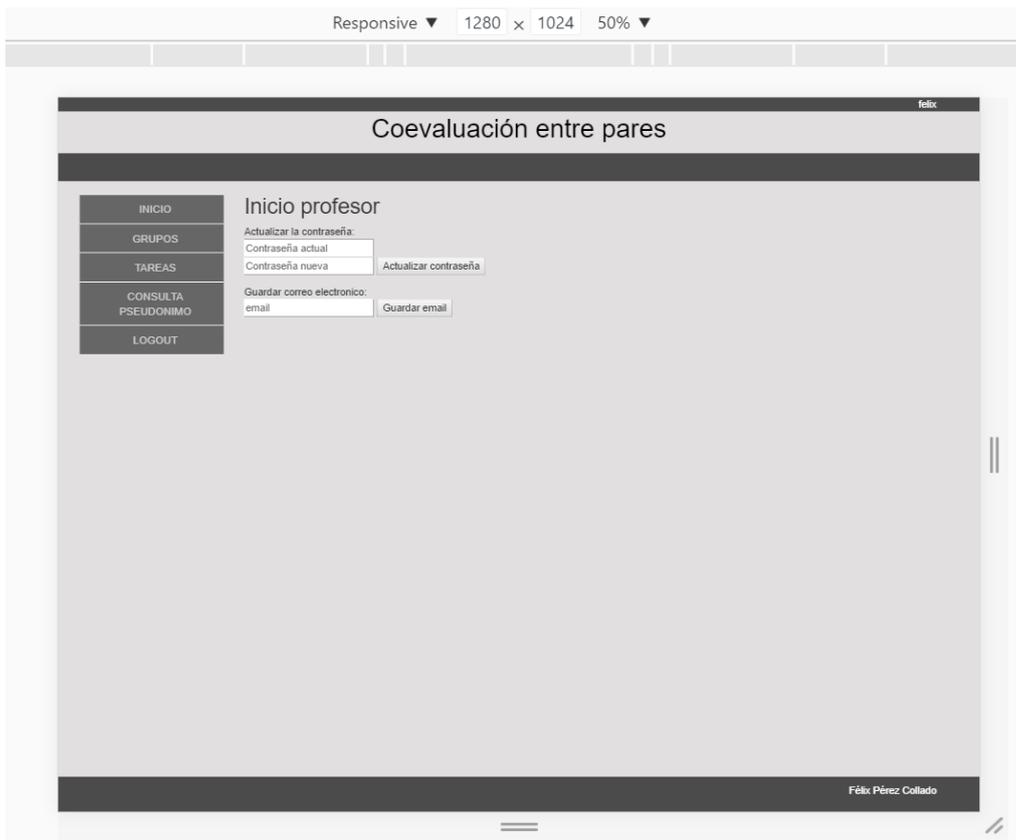


1366x768

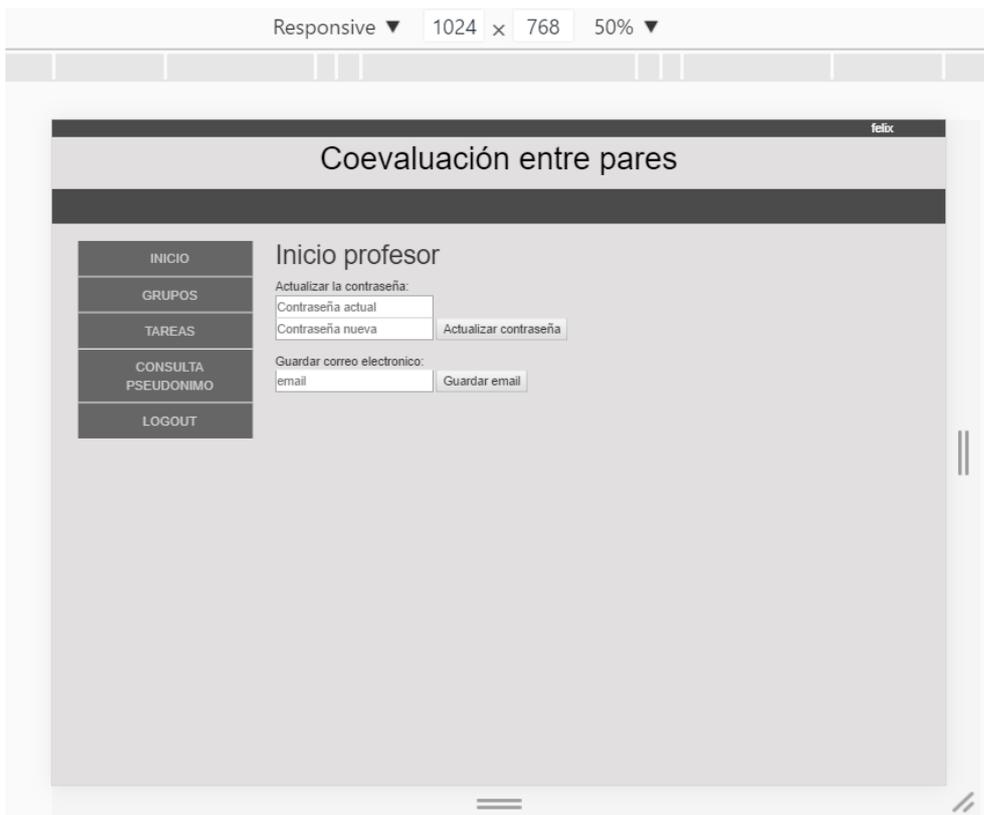


Sitio web para la gestión de evaluación entre pares con pseudónimos

1280x1024



1024x768



800x600



6.3 Validador W3C

Por último, probaremos el código de la aplicación haciendo uso de los validadores de código que proporciona el consorcio W3C. Probaremos tanto el código HTML como el código CSS.

6.3.1 Validación código HTML

La aplicación desarrollada consta de varias ventanas distintas. Todas las ventanas han sido probadas mediante el validador de código que proporciona W3C. A continuación mostraremos las capturas de pantalla de las pruebas se han hecho a las ventanas más importantes de la aplicación.

Para utilizar este validador de código se ha instalado temporalmente la aplicación en un servidor web gratuito, por este motivo en enlace que se ve en las capturas es un poco extraño.

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://blogfelix.000webhostapp.com/evaluacion/index.php?controller=homeController>

Checker Input

Show source outline image report

Check by

<https://blogfelix.000webhostapp.com/evaluacion/index.php?controller=homeController>

Document checking completed. No errors or warnings to show.

Used the HTML parser. Externally specified character encoding was UTF-8.

Total execution time 163 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 17.7.7

En la imagen podemos ver la validación del homeController, esta es la ventana inicial de la aplicación.

A continuación tenemos las capturas de pantalla correspondiente a las interfaces del profesor, como vemos todas son correctas, ya que se han corregido los errores que aparecieron inicialmente.

Login profesor

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://blogfelix.000webhostapp.com/evaluacion/index.php?controller=loginProfesor>

Checker Input

Show source outline image report

Check by

<https://blogfelix.000webhostapp.com/evaluacion/index.php?controller=loginProfesor>

Document checking completed. No errors or warnings to show.

Used the HTML parser. Externally specified character encoding was UTF-8.

Total execution time 354 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 17.7.7

Home profesor

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://blogfelix.000webhostapp.com/evaluacion/index.php?controller=profesor>

Checker Input

Show source outline image report

Check by

Document checking completed. No errors or warnings to show.

Used the HTML parser. Externally specified character encoding was UTF-8.

Total execution time 338 milliseconds.

Crear grupo

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://blogfelix.000webhostapp.com/evaluacion/index.php?controller=crearGrupo>

Checker Input

Show source outline image report

Check by

Document checking completed. No errors or warnings to show.

Used the HTML parser. Externally specified character encoding was UTF-8.

Total execution time 424 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 17.7.7

Ver grupo

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://blogfelix.000webhostapp.com/evaluacion/index.php?controller=verGrupo>

Checker Input

Show source outline image report

Check by

<https://blogfelix.000webhostapp.com/evaluacion/index.php?controller=verGrupo>

Document checking completed. No errors or warnings to show.

Used the HTML parser. Externally specified character encoding was UTF-8.
Total execution time 380 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 17.7.7

Crear tarea

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://blogfelix.000webhostapp.com/evaluacion/index.php?controller=CrearTareaCorrectorAleatorio>

Checker Input

Show source outline image report

Check by

<https://blogfelix.000webhostapp.com/evaluacion/index.php?controller=CrearTareaCorrectorAleatorio>

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

- Warning** The `date` input type is not supported in all browsers. Please be sure to test, and consider using a polyfill.
From line 55, column 16; to line 55, column 47
a entrega:<input type="date" name="fecha">
↵
- Warning** The `date` input type is not supported in all browsers. Please be sure to test, and consider using a polyfill.
From line 56, column 19; to line 56, column 61
orreccion:<input type="date" name="fecha-correccion">
↵

En esta última imagen vemos que tenemos dos warning. Estos son debidos al input de tipo “date”. El aviso dice que este tipo de input no es soportado por todos los navegadores. La aplicación ha sido probada en distintos navegadores y funciona correctamente.



Las siguientes imágenes muestran las pruebas realizadas a las interfaces de los alumnos.

Ver tareas

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://blogfelix.000webhostapp.com/evaluacion/index.php?controller=verTareas>

Checker Input

Show source outline image report

Check by

<https://blogfelix.000webhostapp.com/evaluacion/index.php?controller=verTareas>

Document checking completed. No errors or warnings to show.

Used the HTML parser. Externally specified character encoding was UTF-8.

Total execution time 467 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 17.7.7

Ver tareas a corregir

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://blogfelix.000webhostapp.com/evaluacion/index.php?controller=correcciones>

Checker Input

Show source outline image report

Check by

<https://blogfelix.000webhostapp.com/evaluacion/index.php?controller=correcciones>

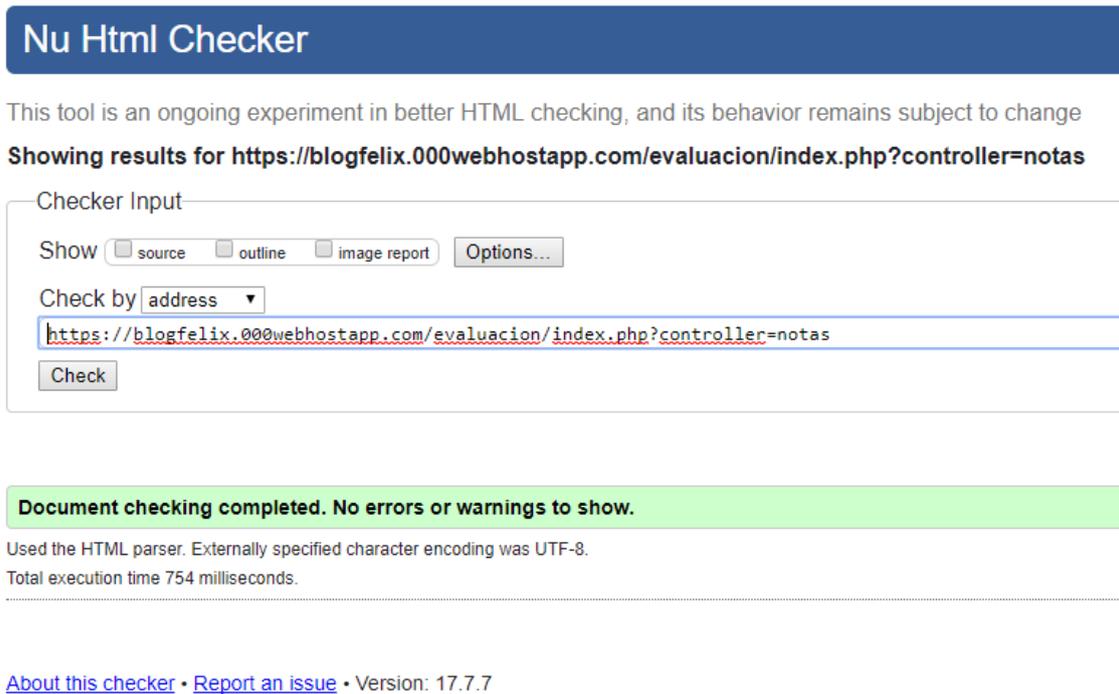
Document checking completed. No errors or warnings to show.

Used the HTML parser. Externally specified character encoding was UTF-8.

Total execution time 389 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 17.7.7

Ver notas



Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://blogfelix.000webhostapp.com/evaluacion/index.php?controller=notas>

Checker Input

Show source outline image report

Check by

<https://blogfelix.000webhostapp.com/evaluacion/index.php?controller=notas>

Document checking completed. No errors or warnings to show.

Used the HTML parser. Externally specified character encoding was UTF-8.
Total execution time 754 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 17.7.7

Como se ve en las imagenes las interfaces de los alumnos son correctas.

6.3.2 Validación código CSS

Por último se han probado los estilos de la aplicación mediante el validador de estilos de W3C.



W3C El Servicio de Validación de CSS del W3C
Resultados del Validador CSS del W3C para <https://blogfelix.000webhostapp.com/evaluacion/index.php?controller=homeController> (CSS versión 3)

Ir a: [Los Errores \(30\)](#) [Las Advertencias \(167\)](#) [Su Hoja de Estilo validada](#)

Resultados del Validador CSS del W3C para <https://blogfelix.000webhostapp.com/evaluacion/index.php?controller=homeController> (CSS versión 3)

Disculpas! Hemos encontrado las siguientes errores (30)

URI : https://blogfelix.000webhostapp.com/evaluacion/css/bootstrap_min.css

Como vemos en la imagen, el validador nos devuelve 30 errores. Estos errores son correspondientes a los estilos del framework bootstrap. Los estilos que se han desarrollado a medida para la aplicación no han generado ningún error.

7. Manuales de usuarios

La herramienta desarrollada durante este trabajo es una aplicación web fácil de usar y administrable por los profesores que vayan a hacer uso de ella, no obstante, a continuación se incluye un manual de usuario para que no quede ninguna duda de su correcto uso. Como la aplicación tiene dos tipos de usuarios claramente diferenciados, se redactarán dos manuales, uno para cada tipo de usuario.

7.1 Manual profesor

7.1.1 Instalación

Para instalar la aplicación, será necesario disponer de un servidor web que pueda alojar la aplicación y que disponga de una base de datos.

Los pasos básicos para instalar la aplicación son:

- Subir la aplicación al servidor.
- Crear una base de datos.
- Editar la función `open_database_connection` del fichero `includes/Modelo.php` con los datos de la base de datos creada.

```
//CONEXION CON SERVIDOR DE BBDD Y SELECCIONAR BBDD
function open_database_connection(){
    $link = mysql_connect('localhost', 'user', 'password');
    mysql_select_db('baseDatos', $link);
    return $link;
}
```

Será necesario cambiar 'user' por el usuario, 'password' por la contraseña de la base de datos y 'baseDatos' por el nombre de la base de datos.

Si todos los datos son correctos y no hay problemas por las versiones de PHP entre el servidor y la aplicación todo debería funcionar correctamente. Nuestra aplicación ha sido desarrollada y probada en la versión 5.6.24 de PHP.

7.1.2 Configurar correo

La aplicación utiliza una cuenta de correo para enviar la contraseña en caso de que algún usuario la pida porque la ha olvidado. Por defecto la aplicación viene configurada con la siguiente cuenta de correo:

correo: coevaluacionentrepares@gmail.com

password:coevaluacion2

Para cambiar la cuenta de correo que utiliza la aplicación editaremos la función enviar_email del archivo includes/Modelo.php. Será necesario cambiar el correo coevaluacionpares@gmail.com por el correo que se vaya a utilizar.

```
function enviar_email($password,$email){  
  
    $titulo = 'Recuperar contraseña';  
  
    $mensaje = '<html>'.  
        '<head><title>Coevaluacion entre pares</title></head>'.  
        '<body><h1>Restaurar contraseña</h1>'.  
        'Esto es un email que se envía en el formato HTML donde enviare la contraseña actual al usuario'.  
        '<hr>'.  
        'Tu contraseña es '.$password.  
        '<hr>'.  
        'Enviado por coevaluacion entre pares'.  
        '</body>'.  
        '</html>';  
  
    $cabeceras = 'MIME-Version: 1.0' . "\r\n";  
  
    $cabeceras .= 'Content-type: text/html; charset=utf-8' . "\r\n";  
  
    $cabeceras .= 'From: Coevaluacion entre pares<coevaluacionpares@gmail.com>';  
  
    $enviado = mail($email, $titulo, $mensaje, $cabeceras);  
  
    if ($enviado){$res='Email enviado correctamente';}  
    else{$res= 'Error en el envío del email';}  
  
    return $res;  
-}
```

El servidor en el que se vaya a instalar la aplicación deberá estar configurado para permitir el envío de correos electrónicos.

7.1.3 Dar de alta a profesores

Para usar la aplicación será necesario tener una cuenta de profesor. Para crear esta cuenta accederemos a la base de datos a tabla profesores y crearemos un nuevo profesor.

7.1.4 Login

Al acceder mediante un navegador a la aplicación, se verá una pantalla en la que elegiremos el tipo de usuario que somos, alumno o profesor. Una vez elegido el tipo de usuario nos identificaremos con nuestro nombre de usuario y contraseña.

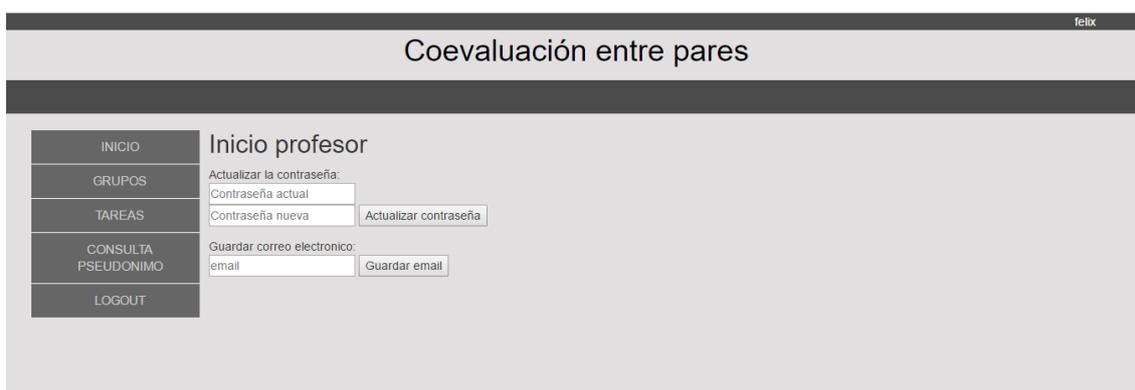




En el caso de los profesores, su usuario y password serán con los que se le haya dado de alta.

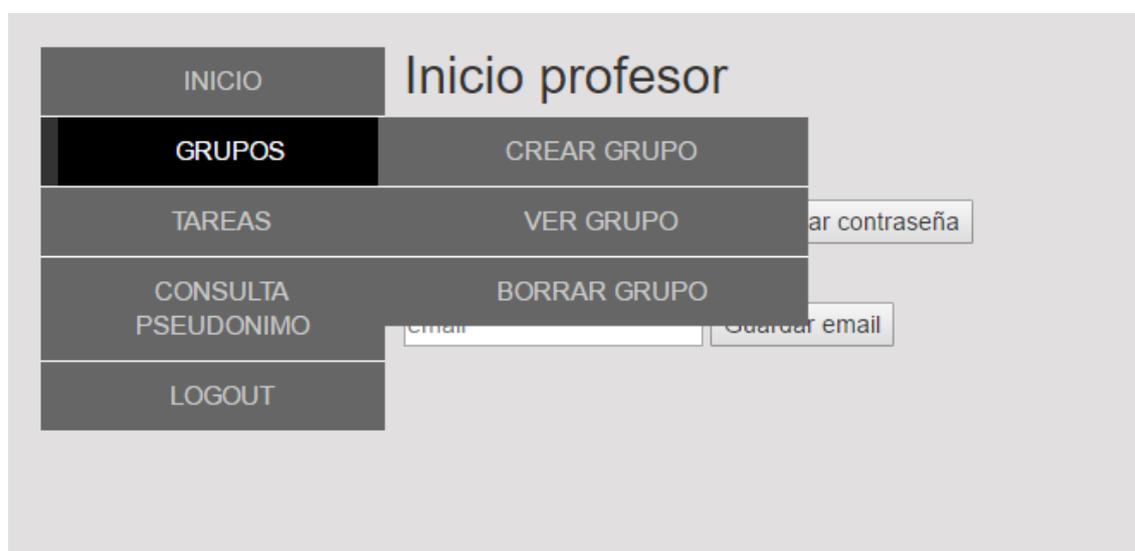


Si los datos introducidos son correctos se accederá a la pantalla principal del profesor. Desde esta interfaz el profesor puede cambiar la contraseña y guardar una cuenta de correo que se utilizara para recuperar la contraseña.



7.1.5 Administrar grupos

Los usuarios profesores son los únicos que pueden llevar a cabo tareas de administración de grupos.



7.1.5.1 Crear grupo

Los profesores podrán crear un grupo importando un fichero csv. El archivo csv tendrá que tener las siguientes columnas separadas por ;

Una primera línea con los nombres de las columnas

DNI; Nombre; Titulación; Tipo de Matrícula; Convocatoria; Comentario Profesor; Condición; Grupo;

Todas las filas que se deseen con los datos de los alumnos.

datos; datos; datos; datos; datos; datos; datos; datos; datos;

Para crear el grupo los profesores entrarán en la ventana crear grupo, una vez allí introducirán el nombre del grupo y seleccionara el archivo. La aplicación no permite crear grupos con nombres repetidos.

Crear un grupo a partir de una lista

Elegir lista:

Ningún archivo seleccionado

Nombre del grupo:

Grupos existentes:

- 1a
- grupo

7.1.5.2 Ver grupo

Los profesores podrán visualizar los alumnos que pertenecen a un grupo desde la pestaña ver grupos, seleccionando el grupo que deseen ver.

Introduce grupo:

▼

1c

Nombre	Pseudonimo
CASILLAS, IKER	6
VILLA, DAVID	920
INIESTA, ANDRÉS	743
PUYOL, CARLES	55
SILVA, DAVID	334
HERNANDEZ, XAVIER	661
RAMOS, SERGIO	47
PIQUÉ, GERARD	70
BUSQUETS, SERGIO	680
NAVAS, JESUS	707
TORRES, FERNANDO	569

Grupos existentes:

- 1a
- 1b
- 1c
- grupo

7.1.5.3 Borrar grupo

Para borrar un grupo, accederemos a la pestaña borrar grupo y elegiremos el grupo a eliminar. Al borrar un grupo se borrarán todos los datos asociados a ese grupo, también se eliminarán las tareas. No se podrá recuperar ningún dato.

Introduce grupo:

▼

Grupos existentes:

- 1a
- grupo

7.1.6 Administrar tareas

Todas las acciones relacionadas con la administración de tareas está la pestaña del menú 'Tareas'.

Para crear una tarea, sea del tipo que sea, el procedimiento es similar:

Accederemos a la pestaña del tipo de tarea que deseemos crear. Una vez allí, las opciones básicas de todas las tareas son las mismas. Pondremos un nombre, una descripción, una fecha de entrega, elegiremos el grupo al que estará asociada la tarea, y subiremos un fichero con los datos de la tarea. El fichero puede ser del formato que se desee.

En función del tipo de tarea que sea deberemos rellenar algún campo más.

A continuación se explican cada una de las opciones que aparecen de una manera más detallada.



7.1.6.1 Tarea sin corrector

Este es el tipo de tarea en la que no se asigna ningún corrector, de tal forma que el único corrector será el profesor. Esta tarea también ofrece la opción de que el alumno elija a quien corrige.

Crear una tarea sin asignar corrector

Tarea:

Descripción:

Fecha entrega:

El alumno elige a quien corrige

Subir pdf:

Ningún archivo seleccionado

Grupo: ▼

7.1.6.2 Tarea corrector aleatorio

En este tipo de tarea la aplicación asigna de manera automática un corrector a la tarea. El número de correctores puede ir desde uno hasta cinco. Esta tarea tiene dos fechas, la fecha de entrega de la tarea y la fecha límite para corregir la tarea.

Crear una tarea asignando un corrector aleatorio

Tarea:

Corrección no anónima

Descripción:

Fecha entrega:

Fecha correccion:

Subir pdf:

Ningún archivo seleccionado

Grupo:

Número de correctores:

7.1.6.3 Tarea eligiendo corrector

Esta tarea permite al profesor elegir que alumno corrige a otro alumno

Crear una tarea eligiendo un corrector

Tarea:

Corrección no anónima

Descripción:

Fecha entrega:

Subir pdf:

Ningún archivo seleccionado

Grupo: ▼

Primero el profesor creará la tarea, una vez la tenga creada tendrá una lista con todos los alumnos. Deberá elegir primero al alumno y después al corrector. Si se desea eliminar una relación bastara con seleccionarla en la columna de la derecha.

PUYOL, CARLES - PROFESOR	<input type="checkbox"/>	Alumnos:	Correctores:
		VILLA, DAVID	<input type="checkbox"/> profesor
		SILVA, DAVID	<input type="checkbox"/> INIESTA, ANDRÉS
INIESTA, ANDRÉS - CASILLAS, IKER	<input type="checkbox"/>	HERNÁNDEZ, XAVIER	<input type="checkbox"/> PUYOL, CARLES
		RAMOS, SERGIO	<input type="checkbox"/> SILVA, DAVID
		PIQUÉ, GERARD	<input type="checkbox"/> HERNÁNDEZ, XAVIER
CASILLAS, IKER - VILLA, DAVID	<input type="checkbox"/>	BUSQUETS, SERGIO	<input type="checkbox"/> RAMOS, SERGIO
		NAVAS, JESÚS	<input type="checkbox"/> PIQUÉ, GERARD
		TORRES, FERNANDO	<input type="checkbox"/> BUSQUETS, SERGIO
			<input type="checkbox"/> NAVAS, JESÚS
			<input type="checkbox"/> TORRES, FERNANDO

Fecha correccion:

7.1.6.4 Ver tareas

En la pestaña ver tareas el profesor puede ver el estado de todas las tareas que existen divididas por grupos

Elige un grupo:

1a

Tarea	Fecha entrega	Correctores		
Ejercicio 1	2017-06-23	5	Estado de la tarea	Generar CSV

Desde esta ventana se puede generar un archivo csv con las notas de los alumnos, también se puede consultar el estado de la tarea.

Una tarea tiene 3 estados:
 - Pendiente: el usuario todavía no la ha hecho
 - Enviado: el usuario ha enviado la tarea, si la tarea tiene correctores seguirá en este estado hasta que la corrijan todos
 - Corregido: todos los correctores han corregido la tarea

Usuario	Nombre	Estado	Nota_1	Nota_2	Nota_3	Nota_4	Nota_5	Nota profesor	Nota media	
12345678	CASILLAS, IKER	enviado	NAVAS, JESÚS	TORRES, FERNANDO	BUSQUETS, SERGIO	HERNANDEZ, XAVIER	RAMOS, SERGIO	6.5	6.93	Corregir
			5.4	7	8	8.7	6			
87654321	VILLA, DAVID	enviado	PUYOL, CARLES	TORRES, FERNANDO	NAVAS, JESUS	BUSQUETS, SERGIO	INIESTA, ANDRÉS	7.3	6.78	Corregir
			7.5	5	6.9	6	8			
12348765	INIESTA, ANDRÉS	enviado	VILLA, DAVID	CASILLAS, IKER	PIQUÉ, GERARD	SILVA, DAVID	HERNÁNDEZ, XAVIER	8.4	7.52	Corregir
			8.2	6.8	7.4	7.3	9			
11223344	PUYOL, CARLES	pendiente	INIESTA, ANDRÉS	RAMOS, SERGIO	HERNÁNDEZ, XAVIER	NAVAS, JESÚS	PIQUÉ, GERARD	0	0	Corregir
			0	0	0	0	0			
22334455	SILVA, DAVID	pendiente	PIQUÉ, GERARD	RAMOS, SERGIO	BUSQUETS, SERGIO	TORRES, FERNANDO	NAVAS, JESÚS	0	0	Corregir
			0	0	0	0	0			

Como se ve en la imagen de cada tarea se muestran todas las notas que le han puesto los correctores, así como la nota media. Se ha implementado un código de colores para ver de una manera gráfica las diferencias entre las notas. Para obtener el color se compara la nota que ha puesto el alumno con la nota que ha puesto el profesor.

Código de colores:

- Verde: la diferencia entre las notas es menor a 1
- Naranja: la diferencia entre las notas está entre 1 y 2
- Rojo: la diferencia es mayor de 2

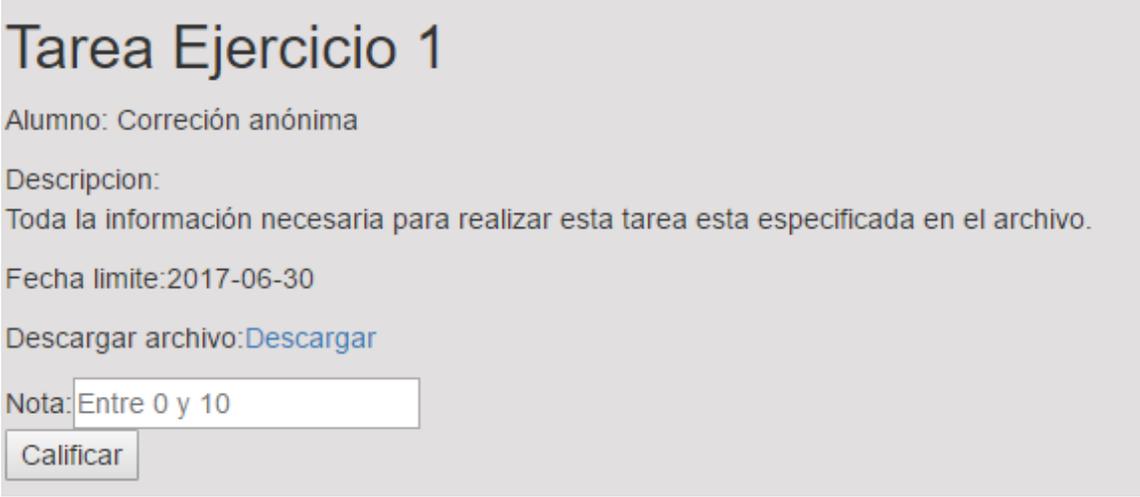
7.1.6.5 Corregir

El profesor puede corregir las tareas de dos formas:

- Desde el estado de la tarea que se muestra en el punto 7.1.6.4, en el enlace corregir. Esta será la manera general de calificar a todos los alumnos de una tarea.

- Desde la opción del menú corregir. El profesor tendrá que acceder a esta sección del menú a corregir una tarea cuando haya creado una tarea del tipo elegir corrector (punto 7.1.6.3) y se haya elegido a sí mismo como corrector de algún alumno.

En ambos casos tendrá la misma interfaz para corregir la tarea.



Tarea Ejercicio 1

Alumno: Corrección anónima

Descripcion:
Toda la información necesaria para realizar esta tarea esta especificada en el archivo.

Fecha limite:2017-06-30

Descargar archivo:[Descargar](#)

Nota:

7.1.7 Consulta pseudónimo

La última opción que aparece en el menú del profesor es la de consultar el pseudónimo. El profesor podrá saber qué alumno está detrás de un pseudónimo



Introduce grupo:

Pseudonimo:

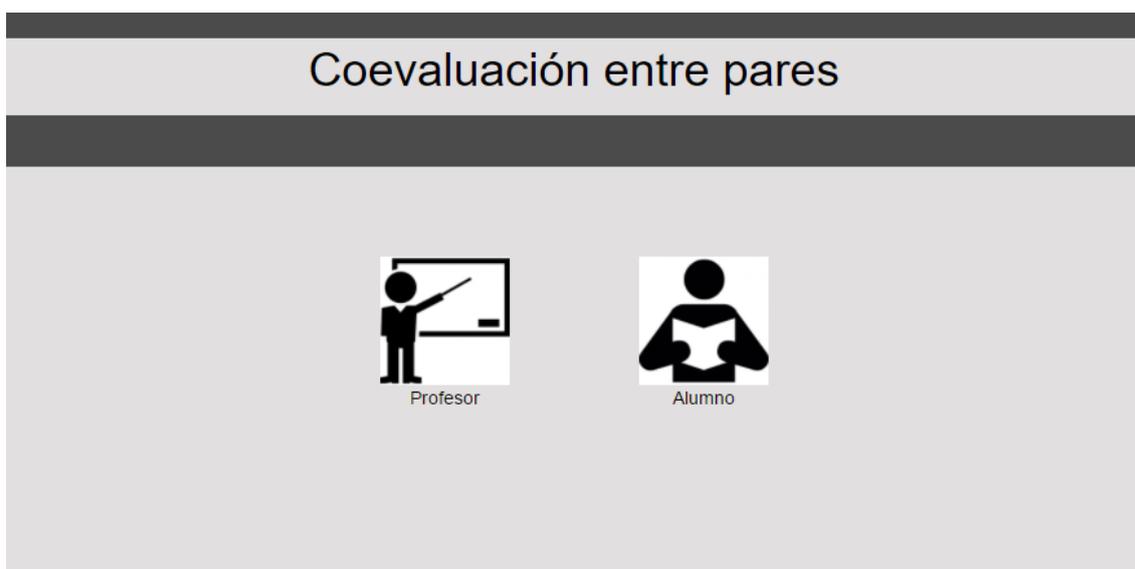
El profesor tendrá que introducir el grupo y el pseudónimo y verá los datos del usuario.

7.2 Manual alumno

Las acciones que tienen disponibles los alumnos son algo más limitadas que las de los profesores. A continuación se explica detalladamente la forma de realizar cada acción.

7.2.1 Login

Al acceder mediante un navegador a la aplicación, nos aparecerá una pantalla en la que elegiremos el tipo de usuario que somos, alumno o profesor. Una vez elegido el tipo de usuario nos identificaremos con nuestro nombre de usuario y contraseña.



El usuario será el DNI del alumno, y el password, si el alumno no lo ha cambiado será el DNI también. Si no recuerda la contraseña puede recuperarla si había guardado un correo.



Una vez que el alumno ha iniciado sesión correctamente tendrá acceso a todas las opciones que le ofrece la aplicación.

Alumno

Coevaluación entre pares

INICIO	Inicio alumno	
VER TAREAS	Actualizar la contraseña:	
TAREAS A CORREGIR	Contraseña actual	<input type="text"/>
NOTAS	Contraseña nueva	<input type="text"/> <input type="button" value="Actualizar contraseña"/>
LOGOUT	Guardar correo electrónico:	
	email	<input type="text"/> <input type="button" value="Guardar email"/>

7.2.2 Ver tareas

Desde la pestaña del menú 'Ver tareas' los alumnos verán todas las tareas que tienen asignadas, separadas por los distintos grupos a los que pueden pertenecer.

1a

Tarea	Archivo	Fecha entrega	Estado	Ver tarea
Ejercicio 1	tarea.pdf	2017-06-23	enviado	Realizar tarea

grupo

Tarea	Archivo	Fecha entrega	Estado	Ver tarea
tarea1	ieeee830.pdf	2017-06-22	pendiente	Realizar tarea
tar	1360176_SIGM@_ImpresoMatricula_MAT_nip_371073_13.07.2016.19.04.09.pdf	2017-06-29	pendiente	Realizar tarea

7.2.3 Realizar tarea

Para realizar cualquier tarea bastará con entrar en el enlace 'Realizar tarea' de la tarea que se desee realizar.

Tarea Ejercicio 1

Alumno: anonimo

Descripcion:
 Toda la información necesaria para realizar esta tarea esta especificada en el archivo.

Archivo: tarea.pdf

Estado: enviado

Fecha limite: 2017-06-30

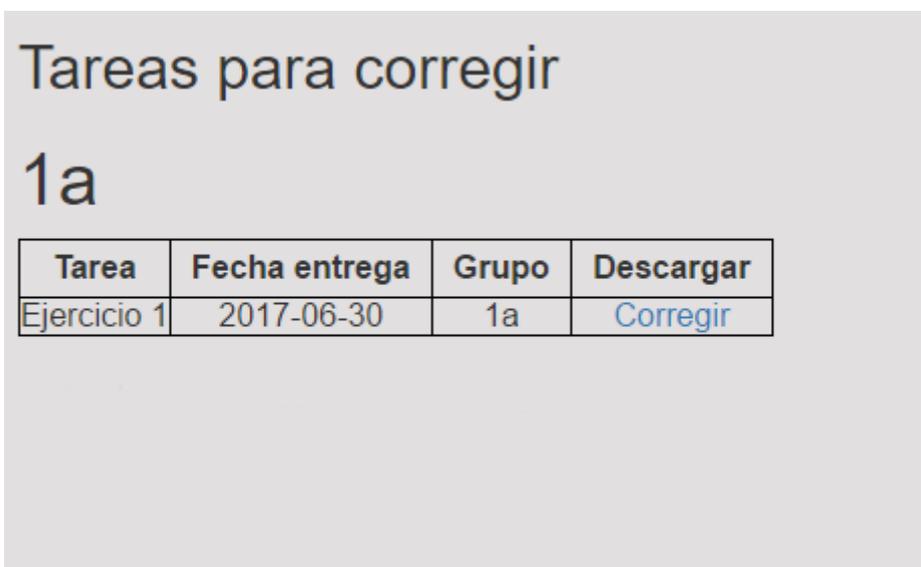
Descargar archivo: [Descargar](#)

Ningún archivo seleccionado

Al entrar en el enlace el alumno verá toda la información de la tarea. Podrá descargarse el archivo con la tarea. Para realizarla tendrá que subir un archivo con la resolución.

7.2.4 Corregir tarea

Para corregir las tareas los alumnos deberán entrar en la pestaña del menú 'Tareas a corregir'. Si el alumno tiene tareas pendientes de corregir le aparecerán en esta ventana.

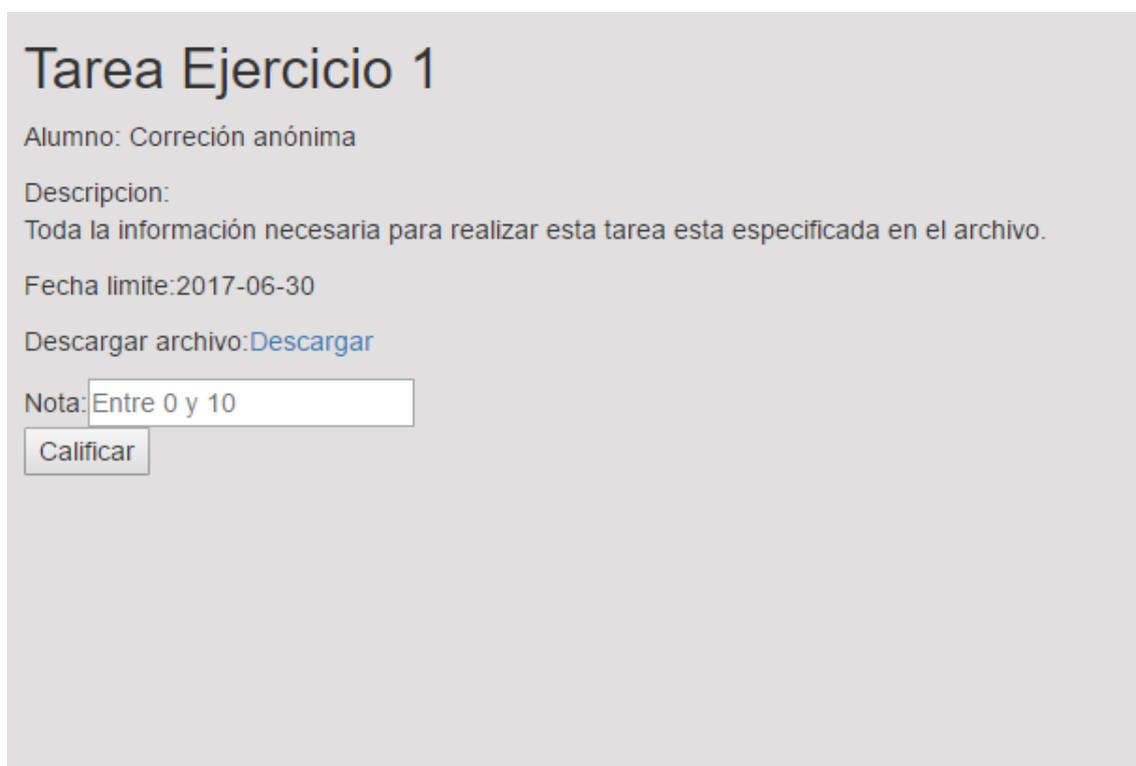


Tareas para corregir

1a

Tarea	Fecha entrega	Grupo	Descargar
Ejercicio 1	2017-06-30	1a	Corregir

Para corregir una tarea el alumno tendrá que acceder al enlace corregir y verá lo siguiente.



Tarea Ejercicio 1

Alumno: Corrección anónima

Descripcion:
Toda la información necesaria para realizar esta tarea esta especificada en el archivo.

Fecha limite:2017-06-30

Descargar archivo:[Descargar](#)

Nota:

El alumno corrector puede ver la tarea realizada y calificarla asignándole una nota.

7.2.5 Notas

Si el alumno accede a la sección 'Notas' podrá ver las notas de todas las tareas que ha realizado.



1a

Tarea	Nota
Ejercicio 1	6.5

grupo

Tarea	Nota
tarea1	7
tar	6

8. Conclusiones

El objetivo principal de este trabajo ha sido desarrollar una aplicación web que permita gestionar de una manera simple y cómoda la coevaluación entre iguales.

El motivo por el que he llevado a cabo la implementación de esta herramienta ha sido dar respuesta a las necesidades de un grupo de profesores de la UPV, que utilizan este tipo de evaluación en sus clases.

He implementado satisfactoriamente todos los requisitos definidos en el punto 2. Los principales requisitos de este proyecto eran todos los relacionados con crear diferentes tipos de tareas, permitir que los alumnos corrigieran las tareas de otros compañeros, y permitir al profesor visualizar las notas y las correcciones.

La aplicación web desarrollada cumple con todos los requisitos establecidos, como se ha demostrado en la fase de pruebas. Por tanto puedo decir, que estoy muy satisfecho con el resultado obtenido.

En cuanto al apartado técnico, las fases más importantes del proyecto han sido definir los requisitos y la fase de diseño. Estas dos fases fueron primordiales, ya que establecieron la base para el desarrollo de la aplicación. Haber definido con claridad estos dos puntos me permitió avanzar con claridad y determinación en el resto de fases del proyecto.

El desarrollo de este proyecto me ha permitido conocer detalladamente todas las fases necesarias para desarrollar un proyecto de este tipo. He ampliado muchos conocimientos adquiridos durante el grado, pero sobretodo, he adquirido una gran cantidad de conocimientos nuevos.

El motivo por el que elegí este trabajo en concreto fue adquirir conocimientos sobre el desarrollo de aplicaciones web, ya que es una de las salidas profesionales a las que me gustaría dedicarme en el futuro. Durante el grado se cursan asignaturas relacionadas con el desarrollo web, pero creo que no te permiten adquirir los conocimientos necesarios para poder desarrollar una aplicación como la que he desarrollado en este trabajo. Gracias a la realización de este proyecto he conseguido ampliar todos los conocimientos que adquirí en el grado con respecto a desarrollo web en general.

9. Bibliografía

1. Sarah Gielen, Elien Peeters, Filip Dochy, Patrick Onghena, Katrien Struyven, Improving the effectiveness of peer feedback for learning, *Learning and Instruction*, Volume 20, Issue 4, 2010, Pages 304-315, ISSN 0959-4752, Consultado el 10 de enero de 2017. Disponible en: <http://dx.doi.org/10.1016/j.learninstruc.2009.08.007>.
2. A. Martí-Campoy, S. Petit, V. Atienza, F. Rodríguez and M. T. Gassó, "Using peer-assessed returnables in multiple stages to improve learning in computer organization courses," 2014 XI Tecnologías Aplicadas a la Enseñanza de la Electrónica (Technologies Applied to Electronics Teaching) (TAAE), Bilbao, 2014, pp. 1-6. Consultado el 10 de enero de 2017. Disponible en: <http://ieeexplore.ieee.org/document/6900158/>
3. Especificación de Requisitos según el estándar de IEEE 830. IEEE Std. 830-1998. (2008). Consultado el 6 de marzo de 2017. Disponible en: <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>
4. Libros web. Capítulo 2. De PHP a Symfony2. Consultado el 11 de enero de 2017. Disponible en: http://librosweb.es/libro/symfony_2_4/capitulo_2.html
5. The world's largest web developer site. HTML tutorial. (2017). Consultado el 12 de enero de 2017. Disponible en: <https://www.w3schools.com/html/>
6. The world's largest web developer site. CSS tutorial. (2017). Consultado el 18 de enero de 2017. Disponible en: <https://www.w3schools.com/css/>
7. Manual de PHP. (2017). Consultado el 12 de enero de 2017. Disponible en: <http://php.net/manual/es/>
8. The world's largest web developer site. SQL tutorial. (2017). Consultado el 12 de enero de 2017. Disponible en: <https://www.w3schools.com/sql/>
9. Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web. (2017). Consultado el 6 de febrero de 2017. Disponible en: <http://getbootstrap.com/>