UOC

TRABAJO DE FIN DE CARRERA

APLICACIÓN WEB PARA INFORMES DEL TRABAJO EN GRUPOS VIRTUALES

David Gracia Vázquez
Tutor: Ferrán Prados

Contenido

1 Introducción 5

2 Objetivos y Alcance del proyecto 5

2.2 Objetivo 5

2.2 Alcance del proyecto 6

3 La Motivación 6

3.1 La motivación general 6

3.2 Motivación personal 7

4 Método seguido 8

5 Planificación 9

5.1 Calendario 9

5.2 Diagrama de la planificación 9

5.3 Plan de trabajo 10

6 Marco de trabajo y conceptos previos 12

7 Análisis 13

7.1 Esquema general de la aplicación 13

7.2 Análisis de requerimientos no funcionales 14

7.2.1 Requerimientos de rendimiento 14

7.2.2 Requerimientos de distribución 14

7.2.3 requerimientos de seguridad 16

7.2.4 Requerimientos de usabilidad 16

7.3 Tareas y actividades de los actores 17

7.3.1 Tareas del administrador 17

7.3.2 Tareas del coordinador docente 17

7.3.3 Tareas del Profesor 18

7.3.4 tareas del alumno 18

7.4 Análisis de los subsistemas de la aplicación 18

7.4.1 Subsistema de acceso 18

7.4.2 Subsistema de mantenimiento de bases de datos 18

7.4.3 Subsistema de creación de usuarios 18

7.4.4 Subsistema de creación de cursos 19

7.4.5 Subsistema de asignación de cursos 19

7.4.6 Subsistema de asignación de grupos 19

7.4.7 Subsistema de formularios de recogida de datos 19

7.4.8 Subsistema de procesado de datos 19

7.4.9 Subsistema de valoración de trabajos 19

7.5 Análisis de requisitos funcionales 20

7.5.1 Requisitos funcionales de la aplicación 20

7.5.2 Requisitos funcionales del usuario: 20

7.5.3 Compatibilidades e incompatibilidades 20

7.6 Casos de uso 21

8 Diseño 26

8.1. ¿Por qué no usar programación orientada a objetos? 26

8.2 Diseño de la base de datos 26

8.2.1 Tabla usuarios. 27

8.2.2 Tabla grupos 28

8.2.3 Tabla cursos 28

8.2.4 Tabla usuarios curso 28

8.2 .5 Tabla valoraciones 29

8.3. Diseños concretos 30

8.3.1 Diseño del sistema de login 30

8.3.2 Diseño del sistema de valoraciones 32

8.3.3 Otros sistemas diseñados 34

8.4 Aspecto general de la aplicación 35

8.4.1 El menú 35

8.4.2 Evitar los caminos sin salida 35

8.5 Sistemas de JQuery seleccionados 35

8.5.1 Plantilla de admin de medialoot 35

8.5.2 JQuery para la generación de gráficos 36

8.5 Hosting: ovh 36

9 Implementación 37

9.1 Elementos generales 37

9.2 Zona del Administrador/Coordinador 38

9.3 Zona Profesor 42

9.4 Zona alumno 51

9.5 Seguridad 53

10. Puesta en funcionamiento 55

11. Conclusiones finales. 58

12 Trabajo futuro 58

12.1 Mejoras necesarias 58

12.1 .1 Sistema de búsquedas 58

12.1 .2 Mejoras en el sistema de usuarios. 59

12.1 .3 Sistema de borrado. 59

12.1 .4 Sistema de modificación. 59

12.1 .5 Sistema de tareas. 59

12.2 Mejoras extra 60

12.2 .1 Sistema de mensajería. 60

12.2 .2 Sistema de archivos. 60

12.2 .3 Un sistema de foros interno 60

1 Introducción

Actualmente la formación a distancia está a la orden del día sobre todo para personas que están trabajando. La competitividad del sector laboral obliga a los trabajadores a reciclarse y a especializarse en nuevos campos continuamente.

Una de las dificultades a la hora de valorar trabajos en educación a distancia es que al no poder ver a los alumnos realizarlo hay que basarse en el resultado final de este lo cual a veces no permite valorar correctamente si el alumno ha trabajo de manera constante o en el caso de los trabajos en grupo cuanto a colaborado cada alumno.

Por eso es muy interesante tener una aplicación en la que los alumnos puedan presentar informes puntuales de sus progresos para facilitar al profesor la valoración de la actividad y de cada alumno que la realiza. Además es muy interesante dar la posibilidad a los alumnos de valorar a sus compañeros, algunas personas pueden ser reticentes a valorar a su compañeros de equipo pero aun y todo el tutor puede sacar información muy útil de estos. Un alumno que se enfrenta a la tarea de valorar a un compañero lo primero que hace es valorar profundamente su propio trabajo para compararlo con el de sus compañeros y además le obliga a pensar en los puntos fuertes y débiles que ha encontrado en sus compañeros pudiendo adoptar los puntos fuertes y darse cuenta de actitudes o acciones que no debe tomar convirtiendo cualquier actividad en una experiencia más enriquecedora.

En conclusión: un buen sistema de recogida de datos es vital para que el profesor haga un mejor seguimiento de una actividad y valore correctamente a cada alumno.

2 Objetivos y Alcance del proyecto

2.2 Objetivo

El proyecto pretende crear una aplicación que permita la recogida de datos mediante formularios para su posteriormente tratarlos y mostrar gráficos y conclusiones. La recogida de datos se puede dividir en dos grupos. Los datos que envíen los alumnos sobre su propio trabajo y las valoraciones que haga cada alumno de sus compañeros. Será importante valorar que datos podrán ver los alumnos y cuales podrá ver el tutor exclusivamente. Por ello habrá que diseñar también un sistema de usuarios con varios niveles de permisos para tener en cuenta que puede que algunos alumnos (por ejemplo los jefes de proyecto) tengan acceso a determinadas áreas y datos.

La aplicación se desarrollará en PHP con bases de datos en mysql en un hosting. Para darle un mejor aspecto se implementarán también elementos en javascript y jquery. No se plantea usar tecnologías como java o .net dado que se puede realizar un buen

trabajo únicamente con PHP y algunos añadidos. Contra menos lenguajes mezclemos más fácil de entender y documentar será la aplicación.

2.2 Alcance del proyecto

Una vez finalizado el proyecto tendremos una aplicación de recogida de datos y un sistema de usuarios fácilmente implementable en plataformas online de enseñanza como puede ser la UOC aunque también puede resultar útil a nivel empresarial para gestionar grupos de trabajo. En cualquier caso el enfoque que se le va a dar al proyecto va a ser orientado a implementarlo en enseñanza a distancia.

Se procurará que la documentación de la aplicación sea lo suficientemente completa como para que esta pueda ser exportada fácilmente y unida a diferentes plataformas de formación.

Al final esta aplicación pretende eliminar la necesidad de enviar informes en Word (u otro procesador de textos), haciendo que el profesor no tenga que bajárselos todos y valorarlos por separado. Con esta aplicación el profesor tendrá la posibilidad de acceder a los datos de un alumno en concreto o de un grupo en general y podrá además ver gráficas e informes sobre estos datos lo cual facilitará enormemente su tarea de valorarlos.

3 La Motivación

3.1 La motivación general

Actualmente la necesidad de trabajar en grupo en plataformas virtuales ha aumentado muchísimo. Desde hace algunos años es muy común que los trabajadores de una empresa deban trabajar conjuntamente con otros empleados que no se encuentran en la oficina o incluso en el mismo país. Aunque se han dado muchos pasos adelante a la hora de facilitar la comunicación entre grupos de trabajo todavía hay varios impedimentos que hacen que estos grupos en entornos virtuales no trabajen de manera tan buena como grupos presenciales. Uno de esos impedimentos es que es muy difícil para el jefe de proyecto llevar un buen control de los miembros y de los trabajos que se realizan.

Por todo esto me ha resultado muy interesante esta propuesta de proyecto ya que me da la posibilidad de explorar uno de los puntos más importantes del trabajo en equipo en plataformas virtuales: la recogida y posterior procesamiento de los datos y valoraciones de los miembros.

3.2 Motivación personal

Actualmente trabajo en el departamento de desarrollo web de una empresa de cursos a distancia. Además de esto fuera de mi jornada laboral ejerzo como tutor de dos cursos de esta empresa.

Debido a esto debo valorar los trabajos de varios alumnos y me he fijado en un detalle interesante. Valorar determinados trabajos viendo solo el resultado es complicado ya que no sé qué ritmo de trabajo ha llevado el alumno ni que recursos a invertido en él. Por otra parte es interesante ver como los alumnos en los foros internos ponen dudas y trozos de código o ejercicios para que sus compañeros los valoren. Curiosamente aunque pocas veces se discuten mis recomendaciones como tutor en los foros los alumnos discuten, plantean ideas y valoran a sus compañeros desde su propia perspectiva pero llevar un control de esto en los foros por mi parte es bastante complicado ya que hay mucha información y poco orden. Cuando un alumno valora otro se pueden ver muchos datos sobre lo que sabe y ha trabajado la asignatura.

Por ello me resulta muy interesante a nivel personal la posibilidad de establecer un sistema de informes que me permita recoger información sobre las actividades de los alumnos de manera ordenada y con la posibilidad de acceder a gráficos e informes sobre esta.

Además me permitirá poner en práctica diversas asignaturas de la carrera sobre desarrollo de proyectos que me han resultado muy interesantes a nivel teórico y que no he podido utilizar en mi vida laboral ya que en nuestro grupo de trabajo contamos con un jefe de proyecto que se encarga de organizar nuestras tareas.

4 Método seguido

El método seguido para desarrollar la aplicación ha sido ir realizando las PECs propuestas y seguir el planing establecido en la PEC1. Ha habido tres puntos que han sido especialmente interesantes y que han marcado un poco el camino a seguir.

Lo primero que se ha hecho es analizar la aplicación de manera global e identificar que software y hardware se iban a necesitar. No me extenderé ahora en este punto ya que el software utilizado se analizará en profundidad en otro punto pero marcar que ha sido necesario plantearse la necesidad de buscar.

- Un software para hacer gráficas.
- Un diseño inicial para posteriormente adaptarlo a las necesidades de la aplicación.
- Analizar sistemas de usuarios de varias páginas para coger ideas para elaborar el de esta aplicación.

Después se ha procedido a diseñar las pantallas de la aplicación a nivel teórico ya que el aspecto de la aplicación se ha implementado al final. Esta decisión puede ser discutible ya que se podría haber ido dándole estilos a la aplicación de mientras se programaba, pero soy de la opinión de que primero hay que tener una aplicación que funcione y luego pensar en darle un aspecto agradable, como digo es una valoración personal.

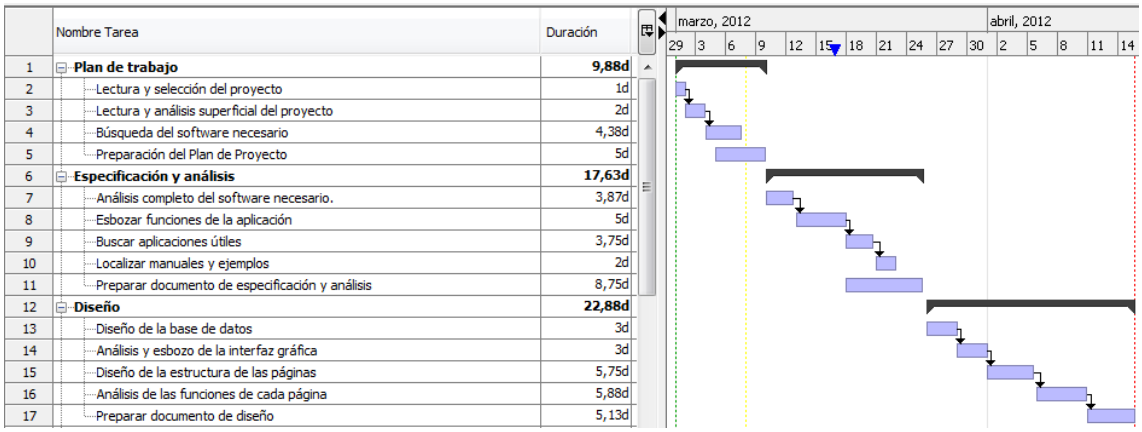
Después de programar y darle aspecto a la aplicación se ha procedido a hacerle una batería de pruebas a la aplicación para comprobar que todo funciona correctamente.

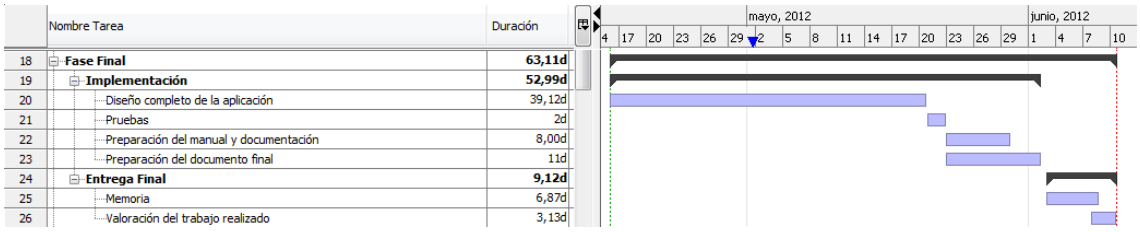
5 Planificación

5.1 Calendario

	Nombre Tarea	Duración	Inicio	Fin
1	**Plan de trabajo**	**9,88d**	**01-mar-2012 8:00**	**09-mar-2012 17:00**
2	Lectura y selección del proyecto	1d	01-mar-2012 8:00	01-mar-2012 17:00
3	Lectura y análisis superficial del proyecto	2d	02-mar-2012 8:00	03-mar-2012 17:00
4	Búsqueda del software necesario	4,38d	04-mar-2012 8:00	07-mar-2012 12:00
5	Preparación del Plan de Proyecto	5d	05-mar-2012 8:00	09-mar-2012 17:00
6	**Especificación y análisis**	**17,63d**	**10-mar-2012 8:00**	**25-mar-2012 16:00**
7	Análisis completo del software necesario.	3,87d	10-mar-2012 8:00	12-mar-2012 16:59
8	Esbozar funciones de la aplicación	5d	13-mar-2012 8:00	17-mar-2012 17:00
9	Buscar aplicaciones útiles	3,75d	18-mar-2012 8:00	20-mar-2012 16:00
10	Localizar manuales y ejemplos	2d	21-mar-2012 8:00	22-mar-2012 17:00
11	Preparar documento de especificación y análisis	8,75d	18-mar-2012 8:00	25-mar-2012 16:00
12	**Diseño**	**22,88d**	**26-mar-2012 8:00**	**15-abr-2012 17:00**
13	Diseño de la base de datos	3d	26-mar-2012 8:00	28-mar-2012 17:00
14	Análisis y esbozo de la interfaz gráfica	3d	29-mar-2012 8:00	31-mar-2012 17:00
15	Diseño de la estructura de las páginas	5,75d	01-abr-2012 8:00	05-abr-2012 16:00
16	Análisis de las funciones de cada página	5,88d	06-abr-2012 8:00	10-abr-2012 17:00
17	Preparar documento de diseño	5,13d	11-abr-2012 8:00	15-abr-2012 17:00
18	**Fase Final**	**63,11d**	**16-abr-2012 8:00**	**10-jun-2012 17:00**
19	**Implementación**	**52,99d**	**16-abr-2012 8:00**	**02-jun-2012 8:01**
20	Diseño completo de la aplicación	39,12d	16-abr-2012 8:00	20-may-2012 17:00
21	Pruebas	2d	21-may-2012 8:00	22-may-2012 17:00
22	Preparación del manual y documentación	8,00d	23-may-2012 8:00	29-may-2012 17:00
23	Preparación del documento final	11d	23-may-2012 8:00	02-jun-2012 8:01
24	**Entrega Final**	**9,12d**	**03-jun-2012 8:00**	**10-jun-2012 17:00**
25	Memoria	6,87d	03-jun-2012 8:00	08-jun-2012 16:00
26	Valoración del trabajo realizado	3,13d	08-jun-2012 8:00	10-jun-2012 17:00

5.2 Diagrama de la planificación





5.3 Plan de trabajo

Plan de trabajo

Periodo del 1 al 9 de marzo

- Lectura del plan de estudios y selección del proyecto a realizar.
- Lectura y profundización en la idea del proyecto.
- Identificación de las tecnologías y lenguajes necesarios para la realización del proyecto. Este es un primer acercamiento muy general que podrá ser ampliado y modificado en futuras bases según se vayan encontrando necesidades concretas.
- Desarrollo del plan de trabajo, apara su entrega el 9 de marzo.

Fase 1: Especificación y análisis

Periodo del 10 al 25 de marzo

- Realización de un documento que analice la aplicación a desarrollar de manera más concreta que en el apartado anterior.
- Análisis ahora más completo del software necesario.
- Esbozar las funciones que queremos que cumpla nuestra aplicación partiendo de las especificaciones del plan docente. Estas irán en tres categorías:
    - Básicas: las funciones que debe cumplir nuestra aplicación para cubrir las necesidades del proyecto.
    - Secundarias(o interesantes): aunque no necesarias intentaré añadir funciones que me parezcan bastante importantes y que añadan funcionalidades adicionales como podría ser la función de que se avise por correo al alumno y al profesor cuando haya un nuevo informe.
    - Terciarias (extras): funciones que aunque no sean necesarias para cubrir las necesidades de este proyecto estaría interesante implementar para darle a la aplicación un extra de funcionalidad. Por ejemplo en esta categoría podría caer la implementación de un muro o foro para que los alumnos y profesores se comuniquen.
- Defender las conclusiones a las que se lleguen en este apartado.
- Buscar aplicaciones que nos vayan a ser útiles en el desarrollo, como jquerys de validación, apertura de ventanas o que embellezcan formularios.
- Ir localizando manuales o ejemplos para realizar más fácilmente la fase de diseño.

Fase 2: Diseño

Periodo del 26 de marzo al 15 de abril.

- Diseño de la base de datos de la aplicación. Este apartado incluirá las tablas y sus relaciones además de una lista clara de claves primarias y secundarias. Se hará un esquema visual completo de tablas y relaciones para tener clara su estructura a la hora de codificar la aplicación.
- Análisis de la interfaz gráfica de nuestra aplicación. En este punto se harán unos bocetos de la zona del alumno y del profesor para tener una idea de dónde vamos a colocar cada elemento.
- Diseño de la estructura de las páginas que compondrán nuestra aplicación. Esto incluirá un organigrama que indique a que página nos lleva cada botón o link que implementemos.
- Análisis de las funciones de cada página de la manera más concreta posible. Por ejemplo si tenemos una página que grabará datos intentaremos centrar que datos va a grabar y a que tablas deberemos acceder.
- Recoger toda esta información en documento estructurado para su entrega.

Fase final

Implementación

Periodo del 16 de abril al 2 de junio.

- Desarrollo completo de la aplicación basándonos en los diseños hechos en el apartado anterior.
- Detección y modificación del diseño inicial si resulta que necesitamos ampliarlo para implementar alguna funcionalidad.
- Introducción de datos de prueba en la base datos.
- Pruebas de la aplicación.
- Preparación de la documentación completa de la aplicación que incluirá las funciones de cada página y que irá de la mano con los comentarios que se hayan ido poniendo en el código. Esto es de cara a que otra persona que no haya participado en la creación de la aplicación pueda modificarla para sus necesidades particulares.
- Preparación del manual de usuario y una pequeña guía para instalar la aplicación desde 0 en un servidor (que puede incluir un archivo sql para la generación automática de las tablas necesarias pro ejemplo).

Entrega final

Periodo del 3 al 10 de junio.

- Redactado de la Memoria.

6 Marco de trabajo y conceptos previos

Para crear la aplicación se ha usado Dreamweaver CS5.5. Existen otras herramientas perfectamente válidas para programar en php pero esta es a la que estoy más acostumbrado. Además esta nueva versión incluye algunas mejoras muy interesantes como puede ser el hecho de que la aplicación te comprueba el código y te avisa de errores comunes como olvidarte un punto y coma o no cerrar algún bucle en tiempo real.

He cogido un hosting en ovh que incluye todo lo que necesito para crear un página web como puede ser un servicio de ftp o un gestor de base datos.

Para manejar la base de datos he usado el programa phpmyadmin, un sistema muy amigable a la hora de gestionar bases de datos.

7 Análisis

7.1 Esquema general de la aplicación

Este punto pretende explicar a grandes rasgos las personas y funciones básicas de la aplicación. Estos puntos serán ampliados en puntos siguientes como descripción de los actores o los casos de uso. Pero me ha parecido interesante condensar un poco el conjunto de ideas menos convencionales que tengo de esta aplicación y justificarlas brevemente. He optado por no hacer un esquema general en gráfico porque en este punto no me decía mucho, dejo este gráfico para la fase de diseño que creo que estará más completo y será más explicativo.

Actores

Empecemos por los actores, un esquema típico sería un administrador que se ocupará de las bases de datos y de la administración de la aplicación, los tutores que tendrán asignados uno o varios cursos y los alumnos que estarán apuntados a uno o varios cursos. Este esquema es simple aunque me parece un poco escaso ya que no creo que un profesor debiera de poder asignarse cursos ni creo que sea tarea del administrador ocuparse de estas tareas por lo que considero que hay que añadir un actor más: El coordinador docente.

El coordinador docente se encargará de crear los cursos, asignar profesores e incluso asignar alumnos a los cursos. La idea es descargar al profesor del curso de estas tareas que caen más dentro de la administración del centro de estudios. A fin de cuentas un profesor debería encontrarse con una interfaz que le permita comunicarse correctamente con los alumnos y manejar su asignatura (subida de material etc…) y poco más. La experiencia que he tenido con tutores de cursos a distancia es que algunos no entienden mucho de informática y hay que darles solo lo que necesitan para realizar sus funciones. Creo que esto ahorrará quebraderos de cabeza al administrador ya que todos los problemas que ocurran se los transmitirá el coordinador docente actuando de filtro de incidencias.

Ausencia de pantalla pública de registro:

En uno de los ejemplos que he visto de una aplicación parecida aparecía la función de que los alumnos y profesores se registraran en la aplicación. Esto no me parece una opción interesante. La tarea de creación de usuarios recaerá sobre el coordinador docente, este se encargará de introducir los datos de los alumnos y profesores. Un buen ejemplo de este sistema es la UOC, solo me llegó el usuario y la contraseña del campus cuando rellené mis datos y realicé las gestiones oportunas.

Es verdad que podría añadir un sistema de registro completo para los alumnos, es decir, que el alumno elija el curso al que se quiera apuntar y este usuario se genere automáticamente, pero

no me parece una función tan importante en el conjunto de esta aplicación. En cualquier caso los usuarios de los profesores los creará el coordinador docente. Si el tutor del proyecto considera interesante añadir un registro público de los usuarios estoy abierto a cambiar esta parte.

Entiendo que esto puede resultar bastante pesado para el coordinador docente si se da el caso de que hay muchos cursos y muchos alumnos, si en la fase de diseño se comprueba que esto es una carga excesiva se diseñará un formulario de registro de alumnos.

Pantallas simples y con lo justo y necesario

Se hará hincapié en crear un interface simple y con funcionalidades claras, no queremos que abrumar al profesor y al alumno con funciones innecesarias. El coordinador docente y el administrador sí que tendrán muchas más pero supondremos que los dos han recibido formación sobre la plataforma y que no tendrán problema en manejarla.

Subsistema de gestión de bases de datos de simple.

Aunque tanto el administrador como el coordinador docente tendrán acceso a funciones de creación de usuarios, asignación de cursos etc… no me parece que sea interesante añadir funciones de creación de tablas, modificación de relaciones etc… Esto será tarea del administrador mediante aplicaciones externas como puede ser phpMyAdmin mucho mejor preparadas que cualquiera que diseñemos nosotros.

7.2 Análisis de requerimientos no funcionales

7.2.1 Requerimientos de rendimiento

Esta aplicación debe ser capaz de ejecutarse de manera rápida en cualquier servidor con PHP y base de datos MYSQL. Plantearemos que para un número “reducido” de alumnos y cursos debería de poder ejecutarse en cualquier servidor compartido que contratemos pero hay que asumir que para un número elevado de cursos y alumnos puede ser necesario contratar un servidor con una alto coste o incluso alquilar un servidor completo para correr la aplicación. A a lo largo del desarrollo se tendrá siempre en cuenta el rendimiento de la aplicación ya que si en la fase de pruebas (en la que hay un número reducido de usuario y cursos) al servidor le “cuesta” ejecutar la aplicación es que algo no está hecho correctamente.

7.2.2 Requerimientos de distribución

Queremos que nuestra aplicación sea fácil de instalar. Esto se conseguirá con un manual de instalación detallado e intentando mantener una estructura fácil e intuitiva (obviamente para un administrador de sistemas).

El objetivo final es que la aplicación pueda guardarse en un archivo comprimido (.zip, .rar etc..) y que el administrador solo tenga que descomprimir los archivos y subirlos al servidor.

Para facilitar la creación de la base datos se incluirá un archivo .sql que generará toda la estructura de tablas necesaria para que la aplicación funcione. Este .sql además creará un usuario administrador para que el administrador pueda crear los usuarios básicos como puede ser el coordinador docente etc..

También habrá unas carpetas de uso habitual de aplicación que sean autoexplicativas para que el administrador tenga claro en un primer vistazo que contiene cada una. Estos pueden ser algunos ejemplos.

- **/img**: en esta carpeta se guardarán todas la imágenes de la aplicación
- /documentos en esta carpeta se guardarán todos los archivos relacionados con los cursos.
- **/includes** Toda aplicación tiene partes del código que se repiten como puede ser la cadena de conexión a la base de datos o el inicio de sesión. Para ahorrar tiempo es recomendable crear un archivo específico para estos datos y luego incluirlo en la página que necesitemos. Otro caso típico suelen ser la cabecera de la página con las opciones de navegación o los menús. Por tanto es recomendable crear una carpeta específica para este tipo de archivos.
- **/CSS** nuestra aplicación usará CSS para definir los estilos, en principio intentaremos crear un solo CSS pero en previsión de que haya más le asignaremos su propia carpeta. Aunque solo sea uno me parece interesante asignarle su propia carpeta ya que es un tipo de archivo especial.

Otro punto interesante pueden los archivos especiales como pueden ser:

- **Index.php** Todos los usuarios tendrán la misma página de inicio, está la página de login que les pedirá usuario y contraseña, según el tipo de usuario que sean se les redirigirá a una página u otra.
- **config.php** el archivo de configuración de nuestra aplicación, en el se guardaran cosas como la cadena de conexión a la base de datos o el servidor en el que se encuentra nuestra aplicación. La función de este archivo es que el administrador pueda acceder de manera fácil a todas las opciones de configuración de la aplicación y modificarlas como considere oportuno.
- **estilos.css** este archivo contendrá todos los estilos gráficos de nuestra aplicación, intentaremos en la medida de lo posible no codificar los estilos en las propias páginas y hacerlo en el CSS para poder reutilizarlos en otras páginas.
- **cabecera.php o menú.php** Estos archivos contendrán elementos de navegación que estén en varias páginas.
- **.htaccess** este archivo nos permite redirigir las páginas y cambiar la forma de la urls. Se nos puede dar el caso de que tengamos que mandar datos por GET de forma que estos se vean en la barra de direcciones del navegador lo cual no queda muy bién de cara al usuario. Con el htaccess podemos modificar estas urls y volverlas “amigables”. Un ejemplo simple es quitar el .php de las páginas para que la url sea más cercana al usuario.

7.2.3 requerimientos de seguridad

El acceso a las páginas estará restringido según el tipo de usuario con el que nos logueemos. Esto se conseguirá con el uso de sesiones, cada vez que entremos en una página esta comprobará que tipo de usuario ha iniciado sesión y le mostrará los datos que necesite.

Esto también funciona como sistema de seguridad por si alguien decide meter la url de una página (digamos por ejemplo panel_profesor.php) si ningún usuario ha iniciado sesión el sistema lo detectará y le enviará a la página de login.

Suponiendo que nadie puede entrar en la aplicación si no está logueado no pondremos sistemas de detección de robots como captchas. Si al final ponemos un sistema público de registro de alumnos este formulario si llevará captchas para evitar que el servidor se sature si un robot comienza a rellenar formularios a toda velocidad.

Los formularios sí que llevarán comprobaciones como por ejemplo comprobar si un usuario ha metido una fecha correcta o si ha metido un email válido. Todo esto de cara a que no se guarden en la base de datos datos absurdos (por ejemplo una fecha de entrega para el día anterior) o erróneos (un mail sin arroba).

Las contraseñas son un dato sensible y se plantea codificarlas (por ejemplo con MD5).

7.2.4 Requerimientos de usabilidad

La usabilidad se refiere a la facilidad que tiene un usuario para utilizar las herramientas de la aplicación.

Como indicamos en el punto 1.3 Pantallas simples y con lo justo y necesario intentaremos que el interfaz sea lo más simple posible para que los usuarios no se sientan abrumados. No sabemos el nivel de conocimientos de navegación van a tener nuestros alumnos y profesores por lo que intentaremos poner solo las opciones necesarias marcadas en un menú claro y conciso. Obviamente también se pensará en estructurar bien las funcionalidades del administrador y del coordinador docente pero supondremos que estos han recibido una formación (mínima al menos) sobre la aplicación y que su nivel de navegación es alto.

Se evitará que las páginas tengan scroll en la medida de lo posible, puede que en algún caso (como un formulario especialmente largo) toleremos algo de scroll pero en general procuraremos que esto no ocurra. A veces esto puede resultar difícil ya que no podemos saber qué tamaño de pantalla y resolución usa el alumno.

En el ejemplo que he leído sobre una aplicación parecida se hablaba de la posibilidad de que la página pudiera verse correctamente en móviles, aunque me parece una idea interesante no la voy a tener en cuenta para esta aplicación. La razón es que por mi experiencia hacer que una página se vea correctamente en móviles implica usar unos estilos muy simples (poco agradables a la vista en una pantalla grande) y una navegación algo engorrosa si no se usa desde móvil. Lo mejor en estos casos es diseñar una sistema de páginas aparte específico para

móviles y redirigir a los usuarios cuando detectemos que se conecta por móvil. No me parece muy útil no aprovechar la potencia que nos da un navegador de un ordenador (PC o portátil). Llegado el caso se podría plantear esto como un desarrollo paralelo pero inicialmente no lo tendremos en cuenta. Se englobaría en el conjunto de funciones terciarias.

A nivel interno todas imágenes llevarán un alt con la explicación de la imagen.

Resumen: Pantallas limpias y simples, poco scroll y mantener una coherencia a la hora de navegar por las páginas.

7.3 Tareas y actividades de los actores

7.3.1 Tareas del administrador

Responsable de mantener la aplicación y las bases de datos. Tareas y actividades posibles:

- Instalación de la aplicación.
- Creación de la base de datos.
- Mantener las bases de datos saneadas, es decir, asegurarse que los datos introducidos son correctos.
- Solución de incidencias que le transmita el coordinador docente.
- Realizar copias de seguridad de los datos
- Creación del usuario coordinador docente (en este caso supondremos que hay uno pero podría haber más).
- Control de logs de acceso.
- Estar preparado para asumir temporalmente las tareas del coordinador docente en caso de que este esté ausente.

7.3.2 Tareas del coordinador docente

Encargado de la parte administrativa de la aplicación.

- Crear usuarios, ya sean profesores o alumnos.
- Crear los cursos.
- Asignar profesores a los cursos.
- Asignar alumnos a los cursos (si no se incluye el registro público).
- Resolver las dudas de los profesores sobre la aplicación.
- Intentar solucionar las incidencias técnicas (dentro de sus posibilidades) y en caso de poder solucionarlas enviar un informe al administrador.
- Alta y baja de usuarios y profesores (Función secundaria)

7.3.3 Tareas del Profesor

Encargado de valorar el trabajo de los alumnos.

- Recoger los formularios de los alumnos.
- Consultar las estadísticas de los alumnos y grupos.
- Calificar a los alumnos y a los grupos.
- Contestar a las dudas de los alumnos.
- Formar grupos de alumnos.

7.3.4 tareas del alumno

Son los que realizarán la tarea en grupo con otros alumnos. Sus tareas son:

- Rellenar los formularios de valoración sobre el grupo, el trabajo y sus compañeros.

7.4 Análisis de los subsistemas de la aplicación

7.4.1 Subsistema de acceso

Actores implicados: Todos (administrador, coordinador docente, profesor, alumno)

Funcionalidades: Se solicitará un usuario y una contraseña y dependiendo del tipo de usuario que sea se le dirigirá a su panel de control. Este sistema establecerá las variables de sesión adecuadas para que el usuario navegue por la aplicación.

También ofrecerá la posibilidad de recuperar la contraseña si el usuario no la recuerda.

7.4.2 Subsistema de mantenimiento de bases de datos

Actores implicados: administrador

Funcionalidades: se hará mediante una aplicación externa como phpmyadmin. Dado que el administrador será el único que posea las claves de acceso a la base de datos será el único que pueda modificarlas.

Con esto el administrador tendrá control absoluto sobre la base de datos, podrá crear, modificar y eliminar cualquier dato, tabla o relación de la base de datos. También podrá comprobar que los datos introducidos son correctos.

7.4.3 Subsistema de creación de usuarios

Actores implicados: Administrador y coordinador docente.

Funcionalidades: Este sistema permitirá crear todo tipo de usuarios, mientras el coordinador podrá crear usuarios de tipo profesor y alumno el administrador podrá crear usuarios de tipo coordinador docente (además de los otros dos tipos).

7.4.4 Subsistema de creación de cursos

Actores implicados: coordinador docente.

Funcionalidades: Mediante este subsistema el coordinador docente podrá crear nuevos cursos a los que posteriormente podrá añadir profesores y alumnos.

7.4.5 Subsistema de asignación de cursos

Actores implicados: coordinador docente.

Funcionalidades: este sistema permitirá asignar profesores y alumnos a los cursos existentes.

7.4.6 Subsistema de asignación de grupos

Actores implicados: Profesor.

Funcionalidades: Este sistema permitirá formar grupos de alumnos al profesor ya que suponemos que vamos a tener suficientes alumnos para formar varios grupos en cada curso.

7.4.7 Subsistema de formularios de recogida de datos

Actores implicados: Alumno.

Funcionalidades: este subsistema presentará los formularios adecuados a cada situación al alumno. Por ejemplo le mostrará el formulario de valoración general del ejercicio cuando este termine. Se encargará de recoger la información facilitada por alumno y grabarla en la base de datos para su posterior tratado

7.4.8 Subsistema de procesado de datos

Actores implicados: Todos (administrador, coordinador docente, profesor, alumno)

Funcionalidades: Este subsistema se encargará de mostrar los datos los datos de manera ordenada a todos los actores. Por ejemplo en el caso del profesor sería un sistema que le permitiera ver las estadísticas de cada grupo o alumno. A los alumnos ver sus notas ordenadas y estadísticas sobre ello.

7.4.9 Subsistema de valoración de trabajos

Actores implicados: profesor

Funcionalidades: Aunque podría encuadrarse en el subsistema de recogida de datos he decidido asignarle su propio subsistema para tenerlo mejor estructurado. Este subsistema permitirá al profesor valorar los trabajos de los alumnos así como valorar el rendimiento de los alumnos de manera individual y como grupo. Además se encargará de guardar los datos de manera ordenada para su posterior procesamiento.

7.5 Análisis de requisitos funcionales

7.5.1 Requisitos funcionales de la aplicación

Este punto indica los requisitos de software y hardware que necesitamos para que la aplicación funcione:

- Un servidor web, puede ser un servidor compartido como puede ser nixiweb o un servidor particular que montemos nosotros. Si lo montamos nosotros será más fácil modificarlo y en principio será más seguro pero requerirá invertir más recursos en el mantenimiento.
- El servidor debe soportar PHP y permitir el uso de sesiones y variables globales.
- Una base de datos MYSQL para guardar toda la información de la aplicación. Debemos tener privilegios de administrador en esta base de datos.

7.5.2 Requisitos funcionales del usuario:

- Una conexión a internet suficientemente potente como para poder navegar de manera fluida por la aplicación. Se intentará que la aplicación funcione con el mínimo de conexión posible pero una conexión de 512kb o 1 mega podría ser un mínimo aceptable.
- Navegador Firefox, Internet Explorer o Chrome. En principio se intentará que la aplicación sea compatible con todos los navegadores existentes pero en principio aspiraremos a que funcione correctamente en los tres indicados. Por volumen de usuarios (aunque esto cambia con el tiempo y los usuarios que tengamos) el orden de prioridad será IE (versión 9), Firefox y chrome. Como he dicho esto dependerá mucho del tipo de usuarios que tengamos, por poner un ejemplo en un curso que realicé todo el mundo usaba Firefox, esto contradice las estadísticas habituales por eso a falta de esos datos tomaremos datos generales obtenidos por internet: http://www.abueloinformatico.es/vertutoriales.php?id=939&titulo=estadisticas_de_uso_de_sistemas_operativos_y_navegadores_en_febrero_de_2012&cat=Varios

7.5.3 Compatibilidades e incompatibilidades

- Un alumno podrá estar matriculado en varios cursos a la vez, cuando se conecte entrará en un panel en el que escogerá el aula al que quiere conectarse.
- Un profesor no podrá ser asignado como alumno a un curso. Si un profesor quiere realizar un curso se le hará un nuevo usuario.
- No hay límite de cursos a los que un alumno puede estar apuntado. No es muy lógico que un alumno esté apuntado a 6 cursos por ejemplo pero se puede dar el caso y mejor no limitarlo.
- Solo se creará y entregará el usuario y la contraseña al alumno cuando este haya hecho todos los trámites.

7.6 Casos de uso

CASO DE USO 1	Login en la aplicación
Subsistema	Subsistema de acceso
Objetivo en el contexto	Loguearse en la aplicación
Entradas	Index.php
Precondición	La aplicación está funcionando corectamente
Salida	Página web solicitando usuario y contraseña
Postcondición si éxito	El usuario entra en la aplicación y ve la un página diferente dependiendo del tipo de usuario que sea.
Postcondición si fallo	El usuario no entra en la aplicación y vuelve a la pantalla de loguin
Actores	Todos (administrador, coordinador docente, profesor, alumno)
Acción paso 1	Insertar usuario y contraseña y pulsar el botón entrar
Acción paso 2	Solicitar una nueva contraseña por email

CASO DE USO 2	Registro de nuevo usuario profesor o alumno
Subsistema	Subsistema de creación de usuarios
Objetivo en el contexto	Crear un nuevo usuario profesor o alumno
Entradas	crear_usuario.php
Precondición	El usuario está logueado como coordinador docente
Salida	El usuario se ha registrado correctamente y se vuelve al panel general del coordinador docente.
Postcondición si éxito	El nuevo usuario queda registrado correctamente.
Postcondición si fallo	Los datos introducidos no son correctos, se indican los errores y se espera a que se corrijan.
Actores	Coordinador docente
Acción paso 1	Pulsar el botón grabar usuario, se vuelve al panel general del coordinador docente.

CASO DE USO 3	Crear un nuevo curso
Subsistema	6 Subsistema de creación de cursos
Objetivo en el contexto	Crear un nuevo curso.
Entradas	crear_curso.php
Precondición	El usuario está logueado en la aplicación como coordinador docente
Salida	Se crea un nuevo curso y este queda grabado en la base de datos. Se le redirige al panel principal del coordinador docente.
Postcondición si éxito	Se graba el nuevo curso en la base de datos.
Postcondición si fallo	Indicar que datos no se han introducido correctamente.
Actores	Coordinador docente
Acción paso 1	Rellenar el formulario de creación con todos lo datos
Acción paso 2	Pulsar el botón crear curso.

CASO DE USO 4	Asignar profesores a un curso
Subsistema	7 Subsistema de asignación de cursos
Objetivo en el contexto	Asignar uno o más profesores a un curso
Entradas	asignar_profesor.php
Precondición	El usuario está logueado en la aplicación como coordinador docente
Salida	El profesor o profesores quedan asignados al curso.
Postcondición si éxito	El profesor o profesores quedan asignados al curso. Se le redirige al panel principal del coordinador docente.
Postcondición si fallo	Se le indica al usuario que restricción no ha cumplido o que error ha sucedido. No se asigna ningún profesor.
Actores	Coordinador docente
Acción paso 1	Seleccionar el curso al que se le van a asignar profesores.
Acción paso 2	Seleccionar el profesor o profesores que se van asignar
Acción paso 3	Pulsar el botón asignar a curso.

CASO DE USO 5	Asignar alumnos a un curso
Subsistema	7 Subsistema de asignación de cursos
Objetivo en el contexto	Asignar uno o más alumnos a un curso
Entradas	asignar_alumno.php
Precondición	El usuario está logueado en la aplicación como coordinador docente
Salida	El profesor o profesores quedan asignados al curso.
Postcondición si éxito	El profesor o profesores quedan asignados al curso. Se le redirige al panel principal del coordinador docente.
Postcondición si fallo	Se le indica al usuario que restricción no ha cumplido o que error ha sucedido. No se asigna ningún profesor.
Actores	Coordinador docente
Acción paso 1	Seleccionar el curso al que se le van a asignar profesores.
Acción paso 2	Seleccionar el profesor o profesores que se van asignar
Acción paso 3	Pulsar el botón asignar a curso.

CASO DE USO 6	Crear grupos de alumnos
Subsistema	8.Subsistema de asignación de grupos
Objetivo en el contexto	Crear grupos de alumnos en un curso
Entradas	crear_grupo.php
Precondición	El usuario está logueado en la aplicación como profesor.
Salida	Un grupo de alumnos quedan grabados como grupo
Postcondición si éxito	El grupo de alumnos se ha formado. Se le redirige al panel principal de profesor.
Postcondición si fallo	Se le indica al usuario que restricción no ha cumplido o que error ha sucedido. No se crea el grupo.
Actores	Profesor
Acción paso 1	Seleccionar los alumnos que van formar el grupo.
Acción paso 2	Rellenar el formulario con los datos del grupo.
Acción paso 3	Pulsar el botón formar grupo.

CASO DE USO 7	Alumno rellena un formulario
Subsistema	Subsistema de formularios de recogida de datos
Objetivo en el contexto	Recoger las valoraciones de los alumnos.
Entradas	formulario_alumno.php
Precondición	El usuario está logueado en la aplicación como alumno.
Salida	El formulario queda grabado en la base de datos.
Postcondición si éxito	El formulario se ha grabado correctamente en la base de datos. Se le redirige al panel principal del alumno
Postcondición si fallo	Se le indica al usuario que restricción no ha cumplido o que error ha sucedido. No se envía el formulario.
Actores	Profesor
Acción paso 1	Seleccionar el formulario que se quiere rellenar.
Acción paso 2	Rellenar el formulario.
Acción paso 3	Pulsar el botón enviar formulario.

CASO DE USO 8	Consultar estadísticas de un alumno
Subsistema	10 Subsistema de procesado de datos
Objetivo en el contexto	Mostrar estadísticas sobre notas del alumno.
Entradas	estadisticas_alumno.php
Precondición	El usuario está logueado en la aplicación como alumno, profesor
Salida	Se muestra un conjunto de estadísticas sobre las notas del alumno.
Postcondición si éxito	Se muestra un conjunto de estadísticas sobre las notas del alumno.
Postcondición si fallo	Se muestra que error ha sucedido. Se le devuelve al panel principal.
Actores	Coordinador docente, profesor, alumno.
Acción paso 1	Seleccionar al alumno (si se es el profesor o el coordinador docente).
Acción paso 2	Pulsar el botón mostrar estadísticas.

CASO DE USO 9	Valorar a un alumno
Subsistema	11.Subsistema de valoración de trabajos
Objetivo en el contexto	Asignarle una nota a un trabajo de un alumno concreto
Entradas	valorar_trabajo.php
Precondición	El usuario está logueado en la aplicación como profesor.
Salida	Un trabajo de un alumno queda valorado.
Postcondición si éxito	Un trabajo de un alumno queda valorado. Se le reenvía al panel principal.
Postcondición si fallo	Se muestra que error ha sucedido o que datos ha introducido incorrectamente. Se le devuelve al panel principal.
Actores	Profesor
Acción paso 1	Seleccionar alumno
Acción paso 2	Seleccionar trabajo del alumno
Acción paso 3	Rellenar el formulario de valoración.
Acción paso 4	Pulsar el botón valorar alumno.

CASO DE USO 10	Dar de alta a un usuario.
Subsistema	14. Subsistema de altas y bajas de usuarios
Objetivo en el contexto	Activar un usuario que esté desactivado
Entradas	activar_usuario.php
Precondición	El usuario está logueado en la aplicación como administrador o coordinador docente. El usuario seleccionado está desactivado
Salida	El usuario ha quedado activado.
Postcondición si éxito	El usuario ha quedado activado.
Postcondición si fallo	Se muestra que error ha sucedido o que restricciones no ha cumplido Se le devuelve al panel general de activación de usuarios..
Actores	administrador, coordinador docente
Acción paso 1	Seleccionar usuario a activar.
Acción paso 2	Pulsar en el botón activar usuario.

CASO DE USO 11	Dar de baja a un usuario.
Subsistema	14. Subsistema de altas y bajas de usuarios
Objetivo en el contexto	Desactivar un usuario que esté activado
Entradas	desactivar_usuario.php
Precondición	El usuario está logueado en la aplicación como administrador o coordinador docente. El usuario seleccionado está activado
Salida	El usuario ha quedado desactivado.
Postcondición si éxito	El usuario ha quedado desactivado.
Postcondición si fallo	Se muestra que error ha sucedido o que restricciones no ha cumplido Se le devuelve al panel general de desactivación de usuarios..
Actores	administrador, coordinador docente
Acción paso 1	Seleccionar usuario a desactivar.
Acción paso 2	Pulsar en el botón desactivar usuario.

8 Diseño

8.1. ¿Por qué no usar programación orientada a objetos?

Aunque PHP soporta desde hace tiempo la programación orientada a objetos la verdad es que la mayoría de las aplicaciones que actualmente están funcionando en la red no usan este sistema. Mi experiencia con este tipo de aplicaciones me dice que una programación en PHP bien estructurada funciona perfectamente y es suficientemente rápida.

A la hora de mantenerla con una buena documentación y una estructura lo más lógica posible es fácil de mantener y sobre todo de ampliar.

8.2 Diseño de la base de datos

En algunos casos veremos que algunas tablas se quedan con simplemente dos campos (sobre todo las de relaciones). En algunos casos como la tabla curso tienen pocos campos ya que he intentado simplificar al máximo las partes de la aplicación que no son prioritarias como puede ser las lecciones de un curso, el número de lecciones etc…

Pero prefiero simplificar esta parte y aspirar a tener una aplicación completamente funcional, es decir, que los usuarios se puedan loguear y realizar todas las acciones posibles (crear cursos, añadir usuarios etc…) que a priori puede parecer poco pero dada la envergadura de esta aplicación llevará un tiempo.

8.2.1 Tabla usuarios.

Esta tabla es de las más importantes que tiene la aplicación, se compondrá inicialmente de los siguientes datos:

- **usuario_id** : un id que defina a cada usuario de forma unívoca, probablemente sea un autonumérico.
- **usuario_freg**: la fecha en que se ha creado el usuario.
- **usuario_email:** el email del usuario.
- **tipo:** de que tipo es el usuario (profesor, alumno….)
- **usuario_nombre**: Este es el nombre de usuario con el que se conectará a ala aplicación, como tiene que ser único se generará a partir del nombre, el apellido y el dni. Por supuesto antes de grabarlo se comprobará si ya existe un usuario con este nombre de usuario por si se da el caso.
- **usuario_clave**: La contraseña del usuario, en principio será el dni del usuario, después el usuario podrá cambiarla por una a su gusto. La única restricción en principio es que debe tener al menos 6 caracteres.
- **activo**: Si el usuario está activo o no, por defecto si lo estará.

8.2.2 Tabla grupos

Se compone de dos tablas:

La primera tabla es grupos que tienen los siguientes campos

Id_grupo

nombre_grupo

La segunda tabla se llamará usuarios_grupo.

Esta tabla relaciona a los profesores y alumnos con los grupos y se compone de los siguientes campos.

- **Id_grupo**
- **Id_usuario.**
- **Id_curso**

8.2.3 Tabla cursos

- **Id_Curso**
- **Nombre_curso**

Esta tabla guarda los cursos que existen en el campus.

8.2.4 Tabla usuarios curso

Una de las tablas más importantes de la aplicación

- id_curso el id_del curso al que pertence
- id_usuario: este campo guarda el código del alumno o el profesor asignado a este curso

Esta tabla me permitirá buscar y asignar de manera fácil los alumnos y profesores a un curso. Como veremos en el apartado anterior tal como he diseñado la aplicación esto es muy importante.

Por último la tabla que guarda el objetivo de esta aplicación, la tabla valoraciones. En esta tabla voy a guardar todas las valoraciones, está preparada de forma que se pueden guardar todos los tipos de valoraciones posibles, por ahora he probado solo las alumno y profesor pero creo que puede guardar también las de grupo.

8.2 .5 Tabla valoraciones

- id_valoracion: no tiene un uso concreto pero es interesante siempre tener un valor único, en principio todos los demás campos no se podrán repetir, por ejemplo, un alumno no podrá valorar dos veces a la misma persona, esto lo evitaremos mediante programación.
- **Id_usuario**: el código de usuario del alumno o profesor
- **Objetivo**: este un campo importante, indica que estamos valorando, si valoramos a un compañero tendrá el id del alumno, si es un grupo en general contendrá la id del grupo.
- **Id_grupo**
- id_curso
- Pregunta1: guarda la nota puesta a la pregunta 1.
- Pregunta2: guarda la nota puesta a la pregunta 2.
- Pregunta3: guarda la nota puesta a la pregunta 3.
- Pregunta4: guarda la nota puesta a la pregunta 4.
- Pregunta5: guarda la nota puesta a la pregunta 5.
- Texto: guarda la valoración textual del alumno.
- tipo: si es una valoración a alumnos y profesores, o es una valoración a un grupo.
- **Texto**: además de la valoración en nota se solicitará al usuario que escriba un pequeño texto sobre la valoración que ha dado.

8.3. Diseños concretos

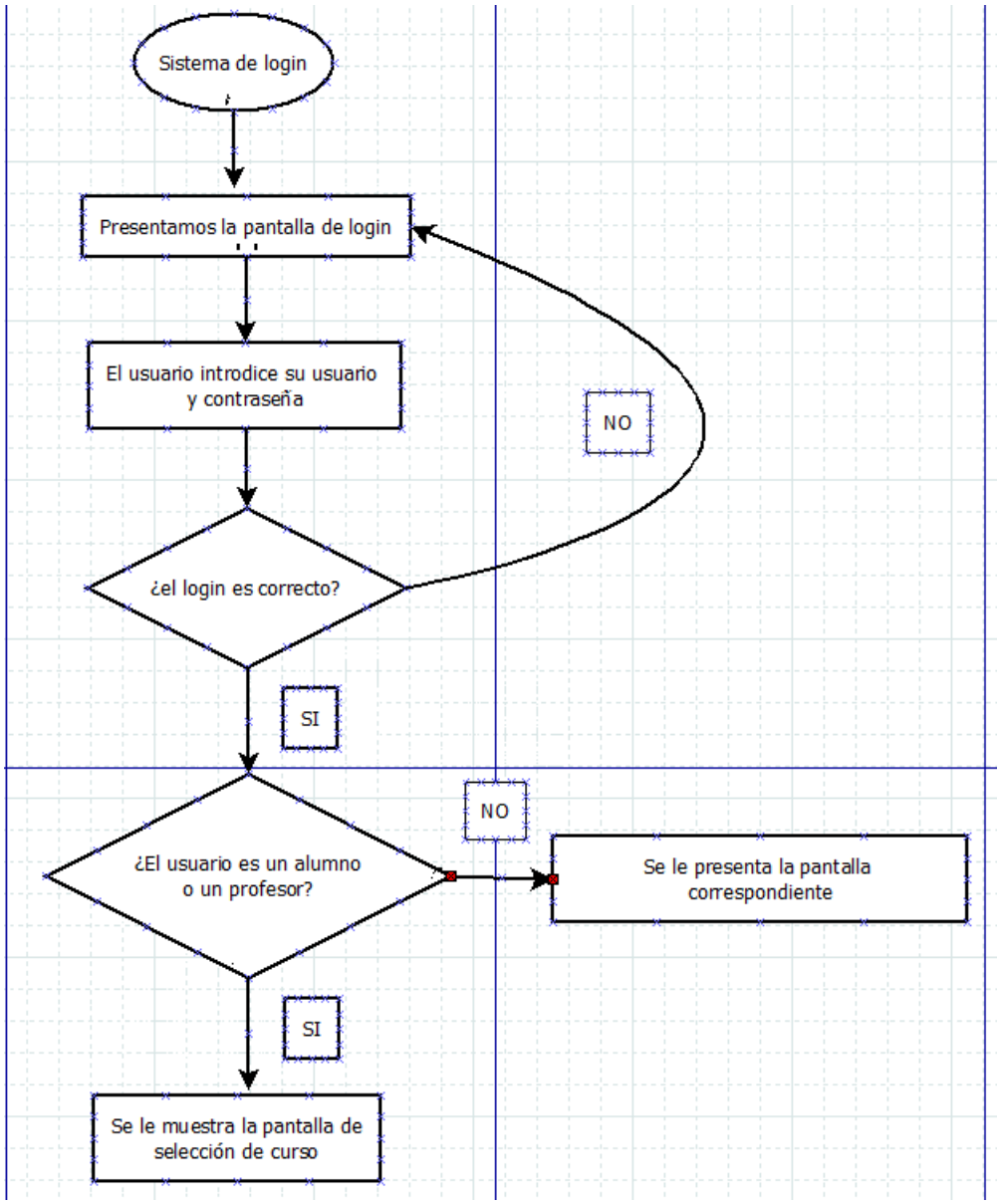
En este punto analizaré un poco más a fondo dos de los sistemas más importantes de la aplicación que considero que requieren su propio apartado. El sistema de login y el sistema de valoraciones.

8.3.1 Diseño del sistema de login

Esta aplicación se basa mucho en el sistema de usuarios, es decir, dependiendo de qué usuario se conecte a la aplicación los menús y las opciones cambian radicalmente. Además los alumnos y profesores pueden tener varios cursos. Para manejar esto después de que un profesor o alumno se le logueen se les presentará una pantalla con la lista de cursos a los que asisten/ son tutores.

Esta parte nos permite mostrar después una pantalla con menos datos. Si no diéramos esta opción y los usuarios tuvieran una pantalla general para todos con todos los cursos sería bastante más lioso.

Podemos ver el proceso en siguiente gráfico:



8.3.2 Diseño del sistema de valoraciones

Siendo la parte más importante de la aplicación le he dado varias vueltas al diseño de las valoraciones así que voy a extenderme un poco en explicar cómo funciona tomando al usuario alumno como referencia:

Alumno:

Cuando un alumno entre en su panel de valoraciones se encontrará con una lista de los compañeros. Cuando elija a un compañero le saldrá un formulario estándar con 5 preguntas de valoración de uno a 5 y un campo para que escriba un comentario.

Este proceso me permite ir recogiendo los valores a guardar en la base de datos de manera sencilla, cuando haga click al seleccionar el trabajo arrastraré una variable con el código del trabajo y cuando seleccione a un compañero arrastraré el código de alumno. Cuando rellene el formulario ya tendré todos los valores necesarios:

Su id de usuario está siempre en una variable de sesión.

El id del curso lo guardamos en una variable de sesión cuando elige el curso.

Por lo que solo queda guardarlo en la base de datos.

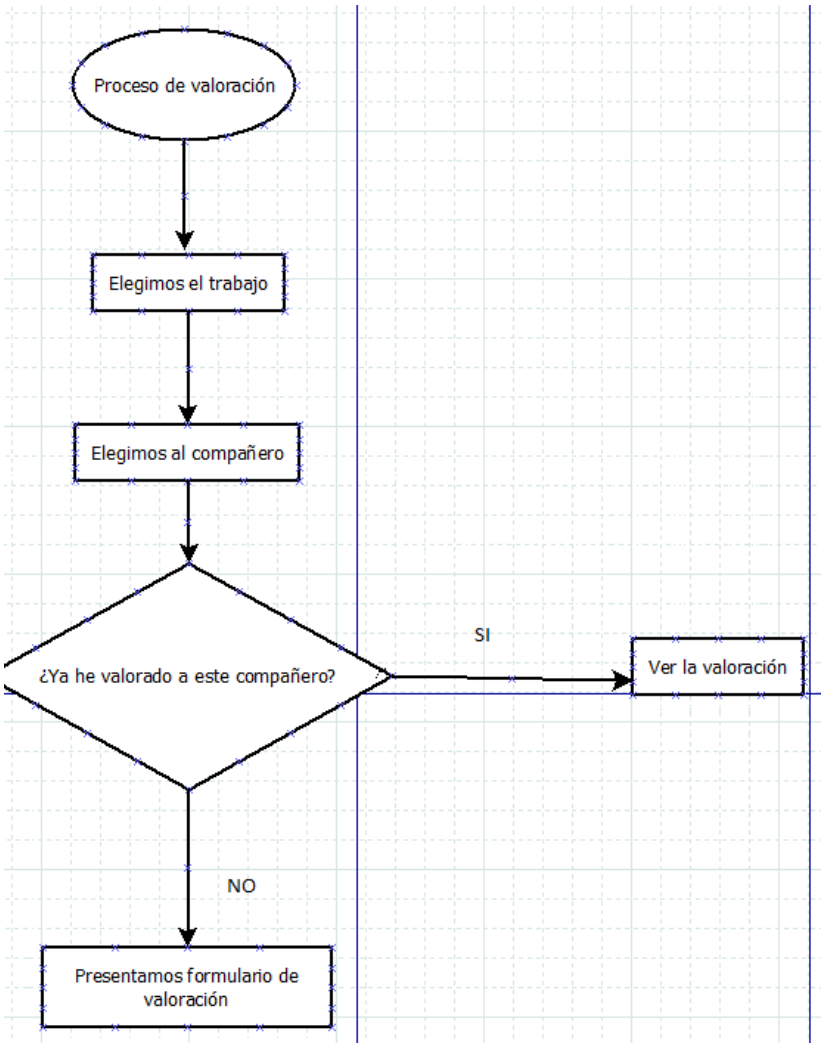
A la hora de mostrar la lista de alumnos se comprobará si el usuarios ya ha valorado a este alumno y si este es el caso en vez de el botón de valorar se le permitirá ver la valoración que ha realizado.

De esta manera como indicábamos antes evitaremos que un alumno valore dos veces a un compañero o un profesor dos veces a un grupo.

Con el profesor es el mismo solo que este tendrá la opción de valorar también los grupos. Al final la filosofía es la misma solo que el profesor tendrá también que

escoger un curso. Después tendrá que elegir si valora a un grupo o a un alumno, después se le presentará una lista de alumnos o grupos y al pulsar en uno se le presentará el formulario de valoración.

Aquí lo podemos ver en un gráfico simple De la parte del alumno



8.3.3 Otros sistemas diseñados

Gestión de usuarios.

Este sistema no requiere mucha explicación, el coordinador docente tendrá un formulario en el que podrá crear los usuarios. Cuando quiera desactivar un usuario seguirá el mismo proceso, se le mostrará una lista y se desactivará o activará el usuario que seleccione.

Sistema de estadísticas

Aunque inicialmente iba hacerlo con el sistema de Google al final me decidí por el Jquery creado por filament group.

http://www.filamentgroup.com/lab/update_to_jquery_visualize_accessible_charts_with_html5_from_designing_with/

El sistema me gustó mucho, la idea que sigue es que el usuario cree una tabla con todos los datos que quiere y partir de esta tabla genera muchos tipos de estadísticas, la que he escogido para este proyecto es la de barras ya que me ha parecido la mejor para mostrar los datos. Pero teniendo otros tipos de estadísticas disponibles de manera fácil permite ampliar la aplicación en un futuro de manera fácil.

El sistema es de pago con modalidad gratuita. La aplicación usa la gratuita que no me permite modificar mucho el aspecto básico pero es más que suficiente para esta.

8.4 Aspecto general de la aplicación

En este apartado explicaré la apariencia básica que quiero que tenga aplicación.

8.4.1 El menú

El menú con todas las opciones disponibles se encontrará en la parte izquierda y será desplegable.

La razón de escoger esta ubicación es que así se deja casi toda la pantalla libre para la información que necesita el usuario y a partir de ahí navegue.

8.4.2 Evitar los caminos sin salida

No hay nada que más moleste a un usuario que después de pasar por varias pantallas se encuentre en una en la que no pueda volver claramente al inicio o moverse a otras pantallas. Por ello se cuidará especialmente el mostrar claramente al usuario al menos un botón para volver a la home en todo momento.

8.5 Sistemas de JQuery seleccionados

Aunque como comenté JQuery es una tecnología muy cómoda y que da unos resultados muy vistosos también tiende a ralentizar la página así que he optado por introducir solo estos dos.

8.5.1 Plantilla de admin de medialoot

Home: http://medialoot.com/item/html5-admin-template/

Una completa suite con todo tipo de funcionalidades entre las que se incluyen para darle a la aplicación una apariencia agradable.

8.5.2 JQuery para la generación de gráficos

Home:
http://www.filamentgroup.com/lab/update_to_jquery_visualize_accessible_charts_with_html5_from_designing_with/

Este Jquery nos permite general varios tipos de gráficos a partir de tablas, fácil de utilizar y muy vistoso.

8.5 Hosting: ovh

Home: http://www.ovh.es/

Después de varios problemas y problemas en un hosting gratuito decidí ahorrarme disgustos y contraté un servicio de pago. Pregunté a varias personas y leí valoraciones por internet y al final me decidí por la compañía ovh, tenían un plan muy barato y las valoraciones en general eran muy buenas.

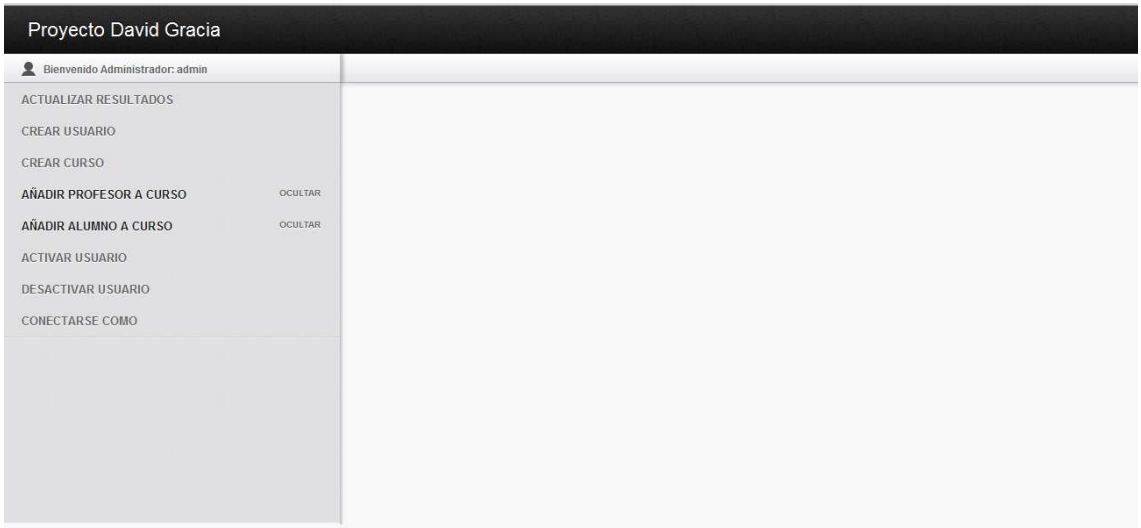
Resultó ser una buena decisión, desde que lo contraté no he tenido ni un solo problema con la aplicación.

9 Implementación

En esta parte iremos paso por paso viendo las opciones de la aplicación con pantallazos de cada zona.

9.1 Elementos generales

Pantalla general



La pantalla de todos los usuarios está dividida en dos zonas, la zona del menú:

La opciones cambiarán según el usuario que usemos pero siempre saldrá un mensaje de bienvenida en la parte superior de la pantalla y la opción de actualizar resultados,

que servirá para ver los cambios que hagamos ya que el menú se modifica según vayamos haciendo acciones.

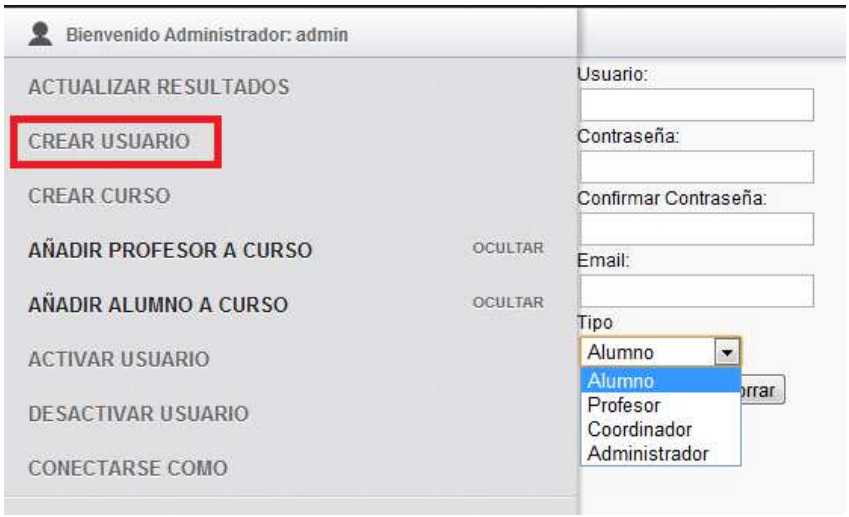
La parte derecha de la pantalla irá cambiando según elijamos opciones, por ejemplo si elegimos crear usuario nos saldrá un formulario para crear usuario como podemos ver en el siguiente pantallazo.

9.2 Zona del Administrador/Coordinador

La zona del administrador y coordinador es igual solo que administrador tiene una opción especial más que es la de conectarse como así que lo mostrado aquí vale para estos dos usuarios.

Crear usuario

Rellenamos el formulario adjunto con los datos que queramos y elegimos el tipo de usuario en el combobox de tipo:

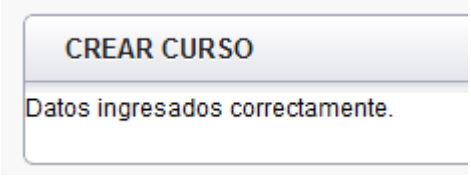


El administrador puede crear los 4 tipos de usuarios, el coordinador por otra parte solo puede crear usuarios de tipo alumno y profesor.

Si creamos correctamente el usuario nos saldrá un mensaje indicándonoslo, si hemos cometido algún error como puede ser que el nombre de usuario ya exista o el email no sea correcto el sistema nos lo indicará con un mensaje.

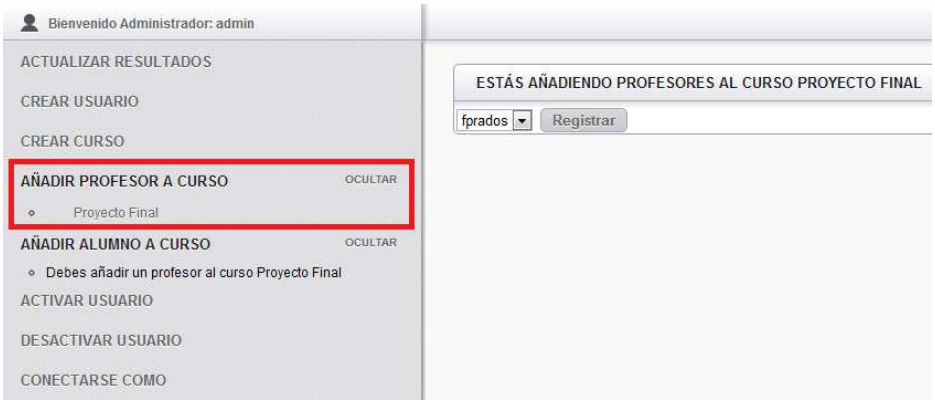
Crear curso

Rellenamos el formulario con el nombre del curso y le damos a registrar, si todo ha ido correctamente saldrá la siguiente pantalla.

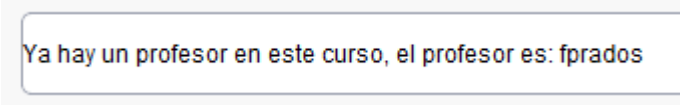


Añadir profesor a curso

Cuando tengamos creado un curso debajo de esta opción nos saldrán los cursos creados y si clickamos en algunos nos dará la opción de agregar un profesor a un curso. Para ello por supuesto debemos haber creado antes un usuario de tipo profesor, en este caso hemos creado previamente el usuario profesor fprados.

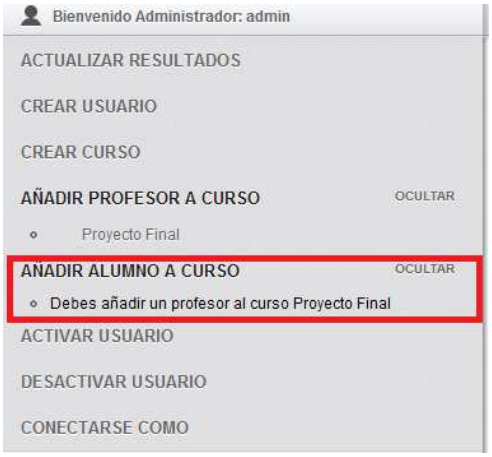


Cada curso solo puede tener un profesor (en esta versión está establecido así) por lo que si intentamos añadir un profesor a un curso que ya lo tiene el sistema nos avisará de esto.

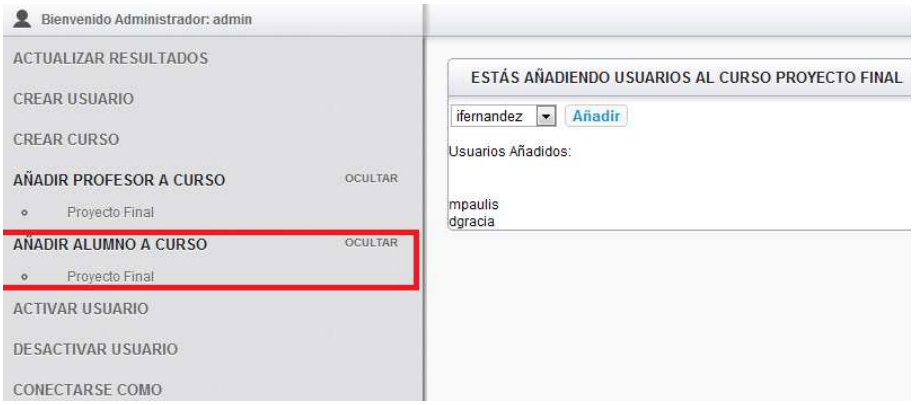


Añadir alumno a curso.

Si el curso al que queremos añadir alumnos no tiene un profesor el sistema nos avisará de este hecho y nonos dejará meter usuarios hasta que hayamos añadido un profesor al curso(recuerda darle al botón actualizar después de agregar usaurios para que el menú se modifique):



Si el curso ya tiene un profesor podremos seleccionar el curso y añadir usuarios:



El combobox se actualiza de manera dinámica, cuando añadimos un usuario este se borra del combobox y se agrega a lista de abajo para tener claro quiénes son parte de ese curso.

Este combobox saca los usuarios que hay disponibles que no son parte de este curso en concreto. Ya que un alumno puede ser parte de varios cursos.

Activar/desactivar usuario.

El admin puede activar y desactivar todos los tipos de usuarios, el administrador solo puede hacerlo con los usuarios alumno o profesor.

Se selecciona el usuario del combo y se le da al botón. En el combo de activar solo aparecen los usuarios inactivos y en el de desactivar solo aparecen los activos.

Conectarse como

Esta opción permite al administrador conectarse como cualquier alumno o profesor para ello debe poner el nombre de usuario de la persona con la que se quiere conectar y el código de curso al que pertenece.

Esto es posible porque la zona de administrador funciona con variables de sesión diferentes que los demás usuarios pudiendo establecer manualmente las que quiera.

Si insertamos datos erróneos como puede ser un usuario que no existe, un código de curso que no existe o que el usuario que pongamos no sea parte del curso (aunque el código corresponda a un curso que exista) el sistema nos avisará de este hecho.

Si todo es correcto se abrirá una nueva pestaña con la sesión del usuario seleccionado.

(Por si quieres probarlo un ejemplo podría ser usuario: fprados, curso:1)

9.3 Zona Profesor

Elegir curso

Cuando un profesor se loguea para a una pantalla donde elige el curso al que quiere acceder de entre los que es parte.

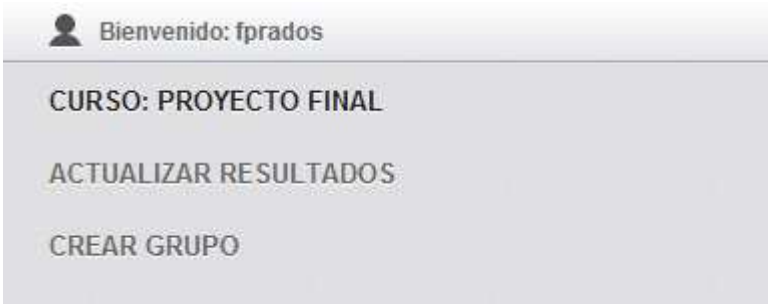
Cuando clicke en uno de los cursos pasará a la pantalla general de ese curso:

Cursos de los que eres profesor

Proyecto Final

Pantalla general.

En la pantalla general veremos varias opciones, la opciones de valoración no aparecen hasta que el profesor crea un grupo y añade alumnos. La pantalla básica es la siguiente:



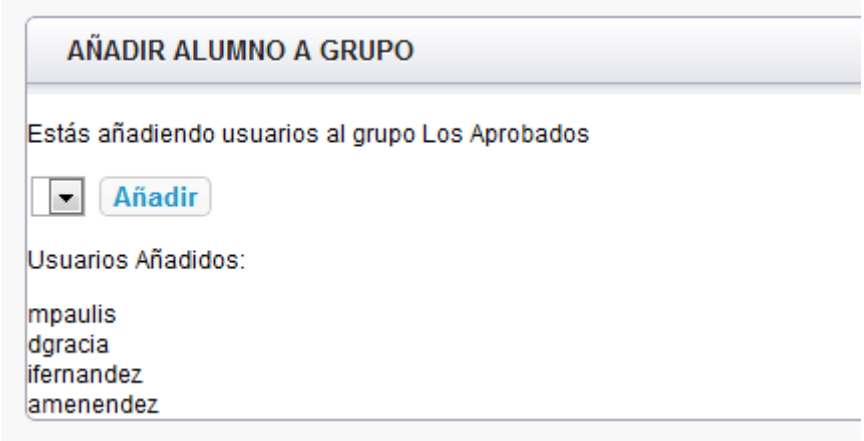
Crear grupo

Rellenamos el nombre del grupo y después le damos a actualizar ya que el menú nos da nuevas opciones cuando creamos el grupo.

Añadir alumnos a grupo

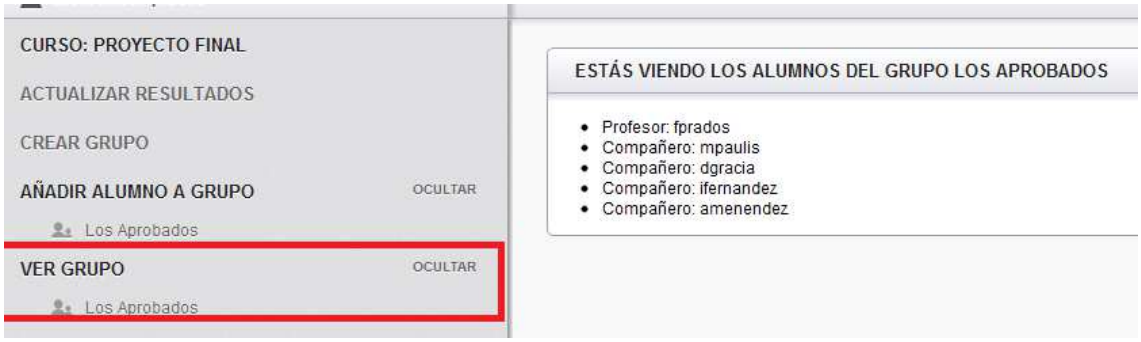
El combo de selección funciona igual que el de añadir usuarios a curso. El combo nos muestra los usuarios que forman parte del curso pero no forman parte de ningún

grupo. Según los vayamos añadiendo se irán eliminando del combo y la lista de usuarios se actualizará dinámicamente.



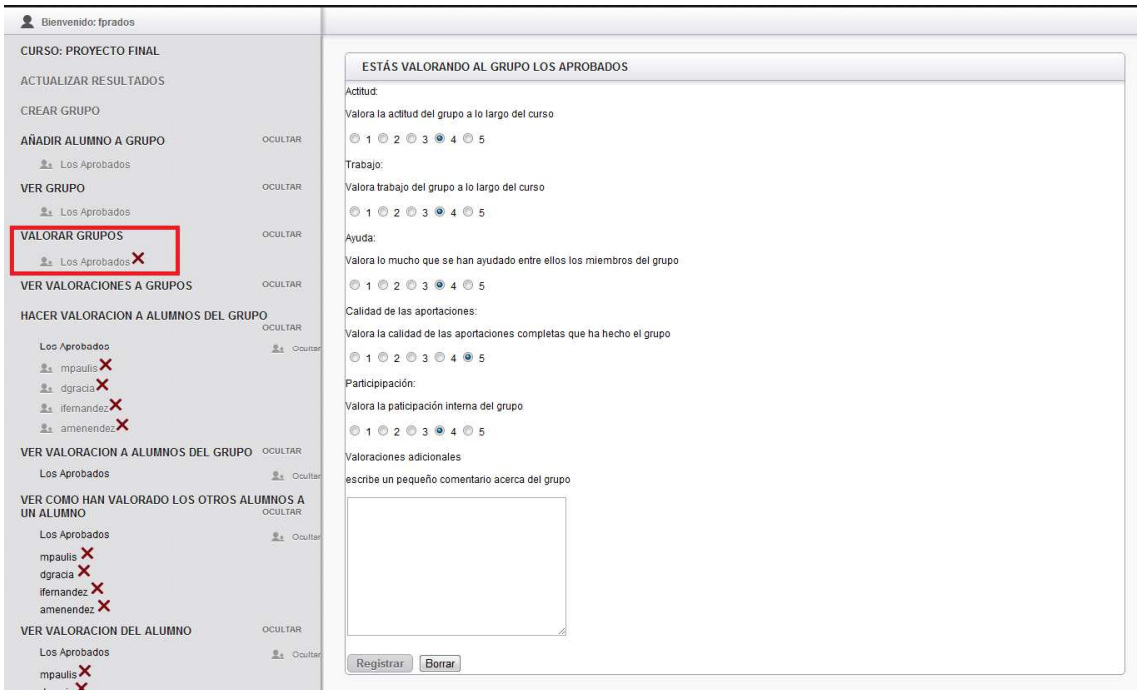
Ver grupo.

Una vez añadamos al menos un alumno al grupo se nos activarán todas las opciones restantes, ver grupo es la primera de ellas. Como su nombre indica nos permite ver todos miembros que tiene actualmente el grupo.

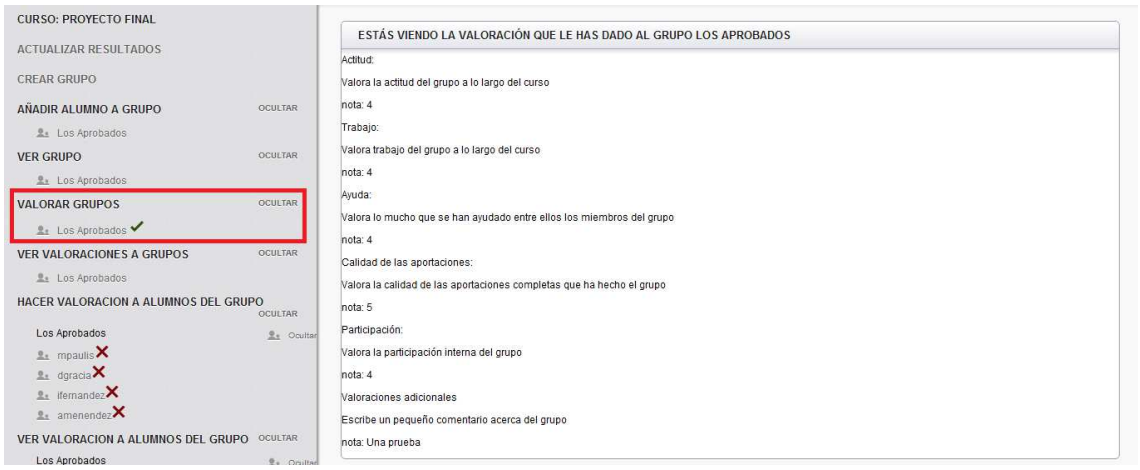


Valorar grupo.

Esta opción nos saca la lista de grupos que hay en este curso, tiene dos estados, con aspa roja al lado nos indica que todavía no hemos valorado a este grupo, si clicamos en el grupo nos saldrá un formulario para que valoremos al grupo.



Cuando hayamos valorado al grupo y hayamos actualizado los datos (con el botón superior) el aspa roja pasará a ser un aspa verde indicándonos que ya hemos valorado a ese grupo. Si clicamos de nuevo en un grupo con un aspa verde podremos ver la valoración que le hemos dado.

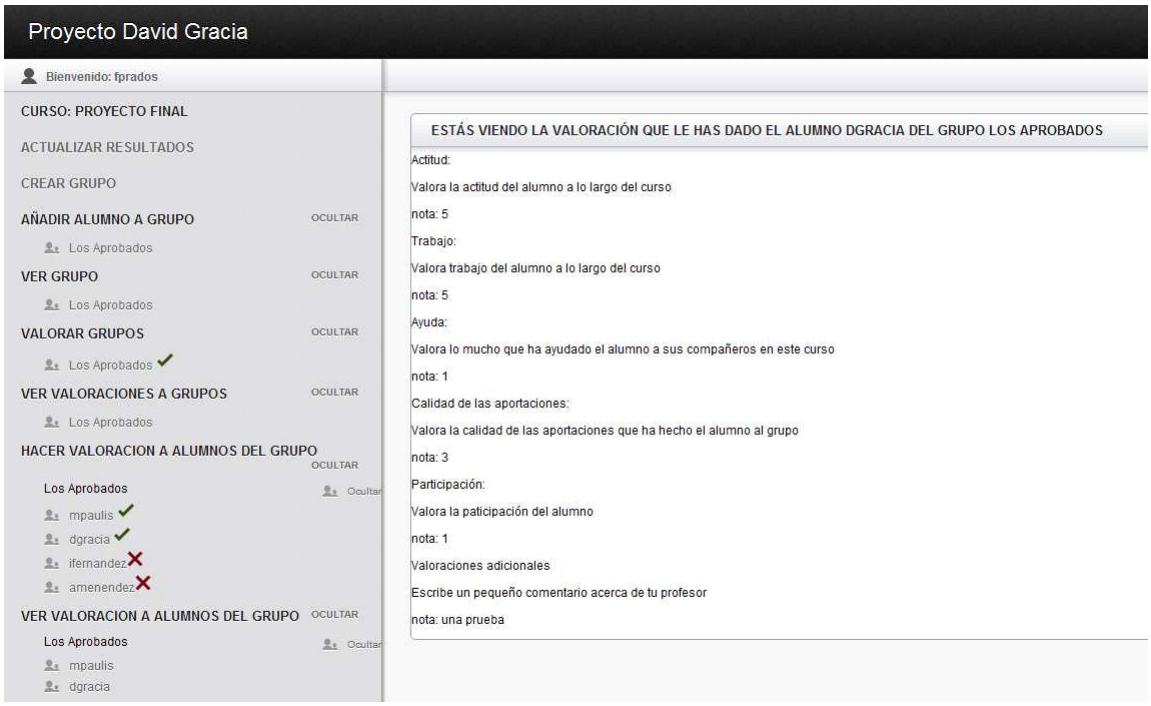


Ver valoraciones a grupos

En esta lista van apareciendo los grupos que el profesor a valorado, si tiene 7 grupos y ha valorado a 2 en esta lista solo saldrán los dos valorados, la información que da es la misma que si clicara en un grupo con aspa verde, pero me ha parecido interesante listarle solo los grupos que ha valorado para que lo tenga en una sola lista.

Hacer valoración a alumnos de grupo.

Esta opción lista todos los grupos y los alumnos que lo componen. Como la valoración de grupos tiene dos estados, si tiene un aspa roja indica que ese alumno no ha sido valorado y si tiene un aspa verde indica que el alumno ya ha sido valorado. Si después de introducir los datos actualizamos la página el estado cambia. Al igual que en el caso anterior si clicamos en un alumno con aspa roja nos sale el formulario de valoración, si clicamos en un alumno con un aspa verde nos muestra la valoración que le hemos dado.



Ver valoración a alumnos del grupo

En esta lista van apareciendo los alumnos que el profesor ha valorado ordenador por grupos, si tiene 7 alumnos y ha valorado a 2 en esta lista solo saldrán los dos valorados, la información que da es la misma que si clicara en un alumno con aspa verde.

Ver cómo han valorado los otros alumnos a un alumno

En esta opción se listan todos los alumnos ordenados por grupos. Si clicamos en un alumno nos permite ver las valoraciones que le han dado sus compañeros de grupo.

Si ningún alumno le ha valorado saldrá un aspa roja y el nombre no será clicable.

Ver valoración del alumno

En esta opción se listan todos los alumnos ordenados por grupos. Si clicamos en un alumno nos permite verla valoración que nos ha dado este alumno.

Si el alumno no ha valorado al profesor saldrá un aspa roja y el nombre no será clicable.

ESTÁS VIENDO LA VALORACIÓN QUE TE HA DADO EL ALUMNO DGRACIA DEL GRUPO LOS APROBADOS

Actitud:

Valora la actitud de tu profesor a lo largo del curso

nota: 5

Trabajo:

Valora trabajo de tu profesor a lo largo del curso

nota: 5

Ayuda:

Valora lo mucho que te ha ayudado tu profesor en este curso

nota: 5

Calidad de las aportaciones:

Valora las aportaciones del profesor

nota: 5

Participipación:

Valora la paticipación de tu profesor

nota: 5

Valoraciones adicionales

escribe un pequeño comentario acerca de tu profesor

nota: Magnifico

Estadísticas de los grupos a mi cargo

Si clicamos en esta opción podremos ver en forma de tabla y en forma de diagrama de barras las valoraciones que le hemos dado a todos los grupos de este curso.

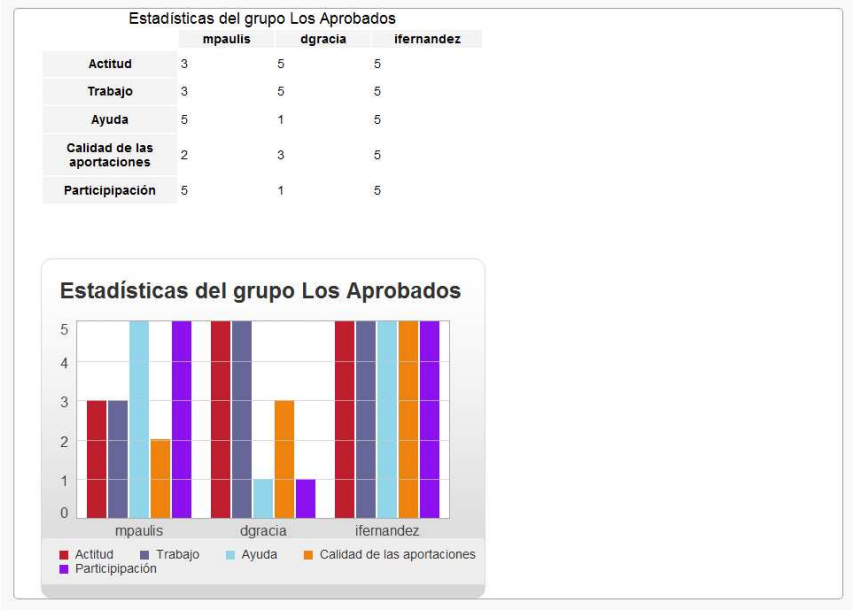
Notas de los grupos

	Los Aprobados
Actitud	4
Trabajo	4
Ayuda	4
Calidad de las aportaciones	5
Participipación	4

Como en este caso solo hay uno solo sale uno.

Estadísticas de los alumnos a mi cargo.

Esta opción nos lista los grupos del curso. Si clicamos en el grupo nos permite ver en forma de tabla y en forma de diagrama de barras las valoraciones que les hemos dado a todos los alumnos de ese grupo.



Estadísticas del grupo Los Aprobados

	mpaulis	dgracia	ifernandez
Actitud	3	5	5
Trabajo	3	5	5
Ayuda	5	1	5
Calidad de las aportaciones	2	3	5
Participipación	5	1	5

Estadísticas de las notas que me han puesto

Esta opción nos lista los grupos del curso. Si clicamos en el grupo nos permite ver en forma de tabla y en forma de diagrama de barras las valoraciones que le han dado al profesor todos los alumnos de ese grupo.

Notas que te ha puesto el grupo Los Aprobados

	dgracia	ifernandez
Actitud	5	4
Trabajo	5	3
Ayuda	5	5
Calidad de las aportaciones	5	4
Participipación	5	4

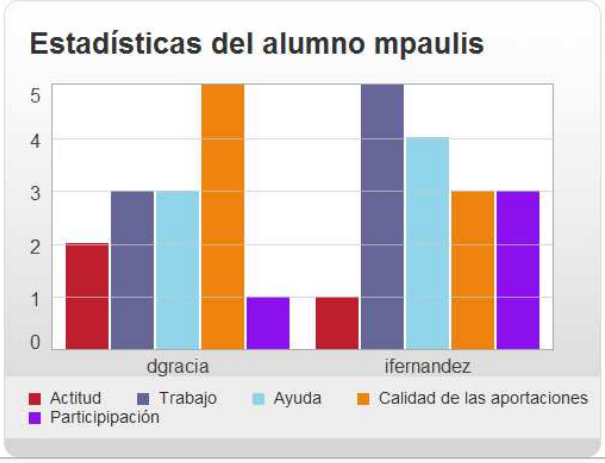
Estadísticas de alumnos entre ellos.

Esta opción nos lista los alumnos de un curso por grupos. Si clicamos en un alumno nos permite ver en forma de tabla y en forma de diagrama de barras las valoraciones que ha recibido de los demás compañeros de grupo.

Si el alumno no ha sido valorado por ningún compañero el nombre no será clicable y tendrá al lado un aspa roja

Estadísticas del alumno mpaulis

	dgracia	ifernandez
Actitud	2	1
Trabajo	3	5
Ayuda	3	4
Calidad de las aportaciones	5	3
Participipación	1	3



9.4 Zona alumno

Elegir curso

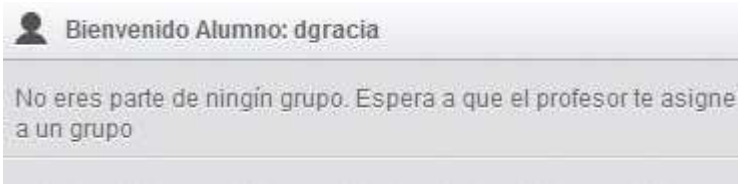
Cuando un alumno se loguea para a una pantalla donde elige el curso al que quiere acceder de entre los que es parte.

Cuando clicke en uno de los cursos pasará a la pantalla general de ese curso:

Cursos de los que eres Alumno

Proyecto Final

Si se da el caso de que el profesor todavía no le ha asignado ningún grupo la aplicación se lo indica y no muestra ninguna opción.

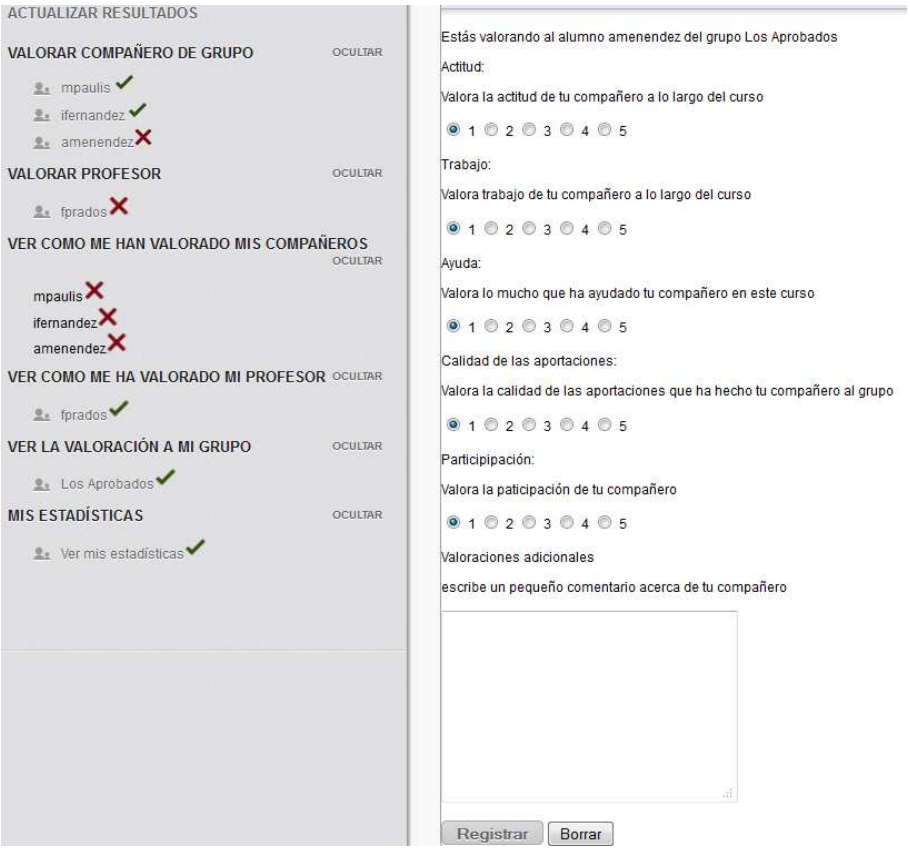


Una vez nos hayan agregado a un grupo se activan todas las opciones.

Valorar compañero de grupo

Esta opción lista todos los compañeros que tenemos en el grupo y nos permite valorarlos.

Si al lado del nombre del compañero hay un aspa roja si clicamos en él nos mostrará el formulario de valoración. Una vez lo valoremos y actualicemos al lado del alumno saldrá un tic verde que nos indicará que el alumno ha sido valorado y si clicamos en su nombre nos mostrará la valoración que le hemos dado.

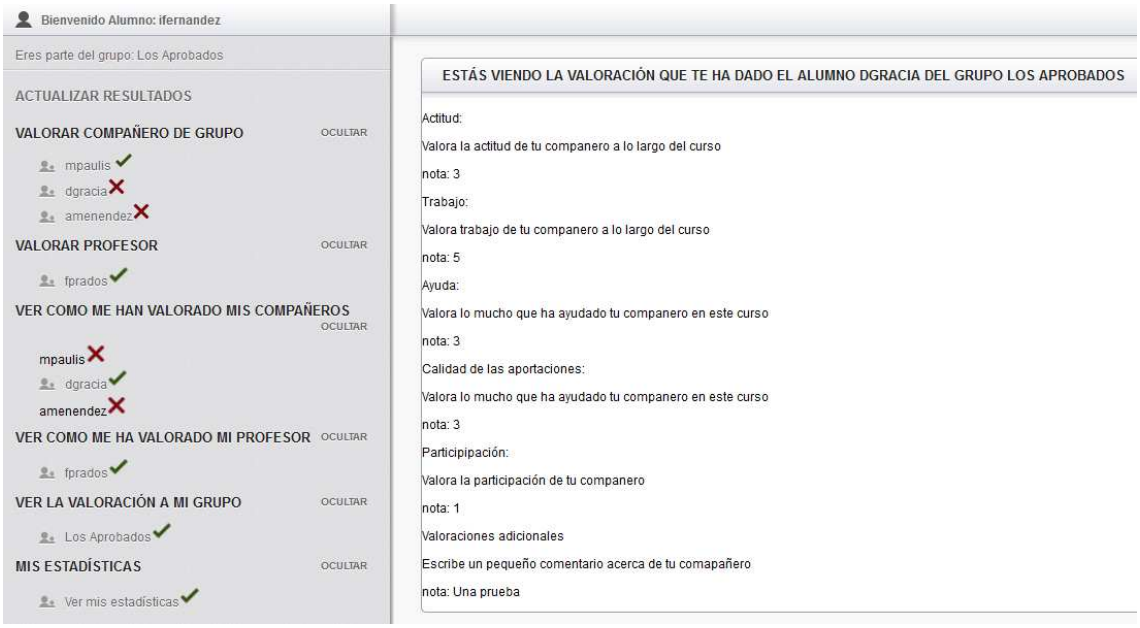


Valorar profesor

El funcionamiento es exactamente el mismo que el alumno solo que en este caso valoramos al profesor.

Ver cómo me han valorado mis compañeros

En esta opción sale una lista de todos los compañeros de grupo. Si el compañero todavía no me ha valorado saldrá un aspa roja y su nombre no será clicable, si sale un tic verde indicará que ese compañero me ha valorado y podré hacer click en su nombre para ver qué valoración me ha dado.



Ver cómo me ha valorado mi profesor

El funcionamiento es exactamente el mismo que el alumno solo que en este caso nos valora el profesor.

Ver la valoración a mi grupo.

El funcionamiento es exactamente el mismo que el alumno solo que en este caso vemos la valoración que se le ha dado al grupo en general.

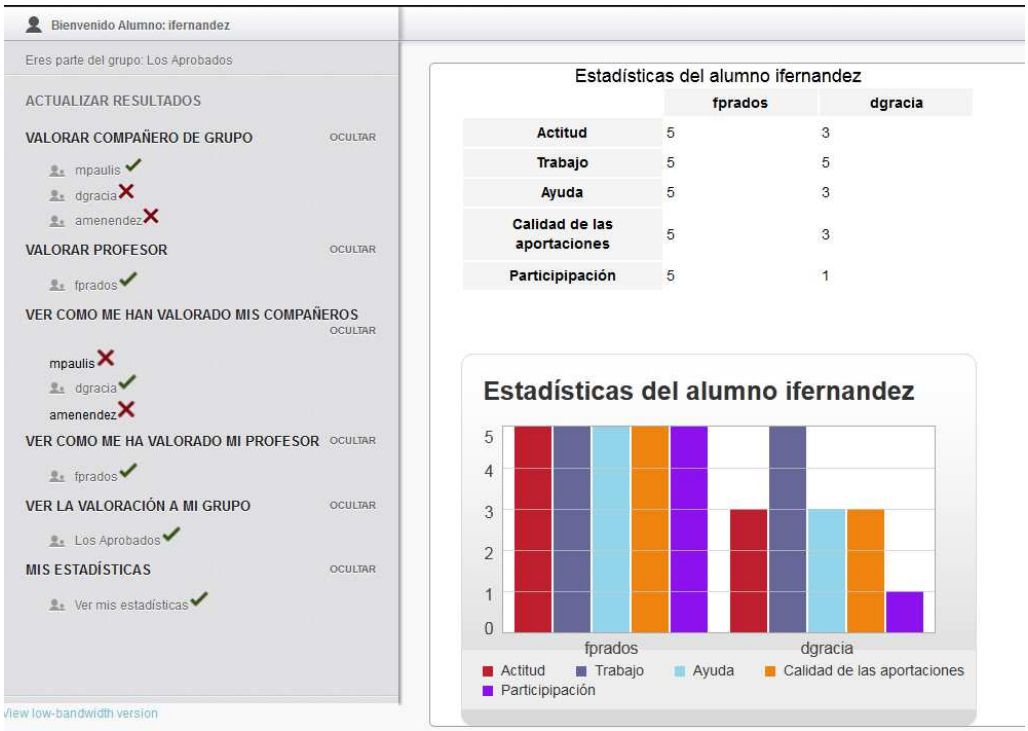
Mis estadísticas.

Si nadie me ha valorado (ni un profesor ni un alumno), esta opción no será clicable. Si alguien me ha valorado podré consultar las valoraciones de mis compañeros y profesor en una tabla y en forma de diagrama de barras.

9.5 Seguridad

El sistema de sesiones incorporado en todas las cabeceras impide que se pueda entrar en una página simplemente poniendo el link en el navegador. Cada página al cargarse

comprueba que están establecidas las variables de sesión adecuadas y si no lo están redirige directamente la página al login.

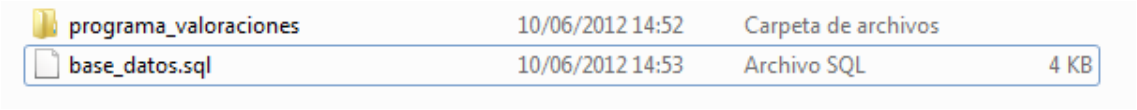


10. Puesta en funcionamiento

La aplicación se instala de manera muy sencilla una vez que tenemos un hosting con bases de datos.

Paso 1: Descomprimir los archivos

Descomprimimos los archivos de la aplicación y veremos una carpeta y un archivo.

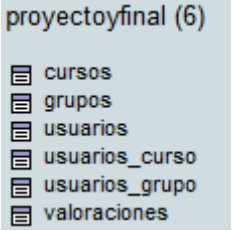


Programa_valoraciones contiene todos los archivos de la aplicación y base de datos.sql nos permitirá crear la base de datos inicial.

Paso 2: Importar la base de datos.

Entramos en el phpmyadmin de nuestro hosting e importamos el archivo base_datos.sql

Si todo ha ido correctamente veremos que se han creado 6 tablas.



Este archivo incluye un usuario administrador necesario para iniciar la aplicación.

Usuario: admin
Contraseña: a

Paso 3 Rellenar el archivo de configuración.

Para que la aplicación funcione debemos rellenar el archivo de configuración con los datos de acceso de nuestra base de datos:

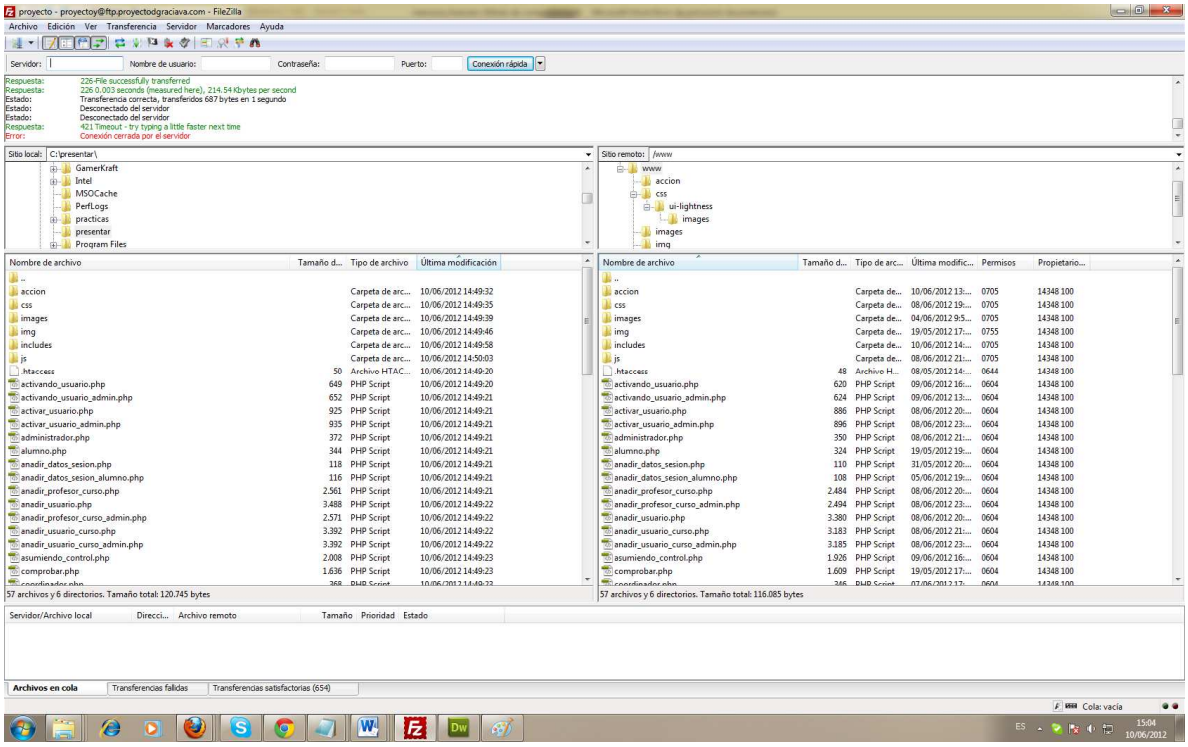
El archivo en cuestión está dentro de la carpeta programa_valoraciones/includes/config.php

```
<?php
    $host_db = ""; // Host de la BD
    $usuario_db = ""; // Usuario de la BD
    $clave_db = ""; // Contraseña de la BD
    $nombre_db = ""; // Nombre de la BD

    //conectamos y seleccionamos db
    mysql_connect($host_db, $usuario_db, $clave_db);
    mysql_select_db($nombre_db);
?>
```

Paso 4 Subir los archivos al servidor

Subimos todo el contenido de la carpeta programa_valoraciones a nuestro servidor



Paso 5 Conectarnos con administrador

Nos conectamos con el administrador:

Y ya accedemos al panel de administrador desde el que podemos realizar todas las acciones para poner la aplicación en marcha como puede ser crear usuarios, cursos etc..

11. Conclusiones finales.

La aplicación cumple sus cometidos básicos y el aspecto es agradable a la vista por lo que considero que los objetivos del proyecto han sido cumplidos. Quedan algunas cosas por desarrollar y algunas cosas que sería interesante implementar para hacer la aplicación fuera completamente autosuficiente como campus virtual que desarrollaré en el siguiente punto.

Ha sido muy interesante desarrollar una aplicación desde cero ya que era un trabajo al que no me había enfrentado nunca, normalmente he tenido que modificar aplicaciones y lo que he tenido que desarrollar han sido módulos extra que cumplieran determinadas funciones. Aunque he desarrollado páginas webs completas el diseño ya me venía dado y una página web corporativa no tiene las funcionalidades de una aplicación.

En general me ha ayudado a ver todas las fases de un desarrollo en especial el diseño (entendiéndolo como el aspecto), un punto que no había trabajado y que me resultado muy enriquecedor ya que me ha ayudado a valorar la dificultad de realizar un buen diseño.

12 Trabajo futuro

12.1 Mejoras necesarias

Estás mejoras son cosas que se me han quedado en el tintero por falta de tiempo y las considero muy importantes de cara a tener una aplicación completa.

12.1 .1 Sistema de búsquedas

Sería interesante montar un sistema de búsqueda de usuarios para los administradores y coordinador para añadir los usuarios a un curso y para el profesor para añadir usuarios a un grupo.

El sistema actual con un combobox es muy cómodo mientras no haya muchísimos usuarios, si hubiera muchos este sistema sería bastante pesado de utilizar.

12.1 .2 Mejoras en el sistema de usuarios.

Actualmente no se guarda el nombre real del usuario, sería interesante implementarlo para que los mensajes de bienvenida le llamen por su nombre.

12.1 .3 Sistema de borrado.

Actualmente no se permite borrar usuarios, valoraciones etc… Es una opción que me plantee al inicio del proyecto y que descarté. Actualmente esto lo puede hacer a mano el administrador desde phpmyadmin. Esta opción la descarté ya que al borrar un alumno se borran todas sus valoraciones y esto no me parecía una buena opción. En su lugar puse la posibilidad de desactivar los usuarios que simplemente evita que los usuarios accedan a la aplicación pero mantienen todo lo que hayan hecho previamente. Actualmente para realizar un borrado es preciso avisar al administrador.

12.1 .4 Sistema de modificación.

Las implicaciones de cambiar a un usuario de grupo resultaban demasiado grandes para manejarlas inicialmente, por eso no se han implementado inicialmente. Por supuesto el administrador puede realizar estos cambios desde la base de datos ya que se supone que tiene suficiente conocimiento para no provocar errores. En cualquier caso esta opción debería ser implementada en un futuro ya que ayudaría bastante a los usuarios. Actualmente para realizar una modificación es preciso avisar al administrador.

12.1 .5 Sistema de tareas.

Actualmente no se pueden crear varias tareas en un curso, todas valoraciones son del propio curso. Estaría bien poder dividir un curso en varias tareas valorables por separado.

12.2 Mejoras extra

Estás mejoras añaden mucha funcionalidad a la aplicación y le permiten funcionar como un campus virtual.

12.2 .1 Sistema de mensajería.

Un sistema que permita mandar mensajes a los diferentes usuarios de la aplicación.

12.2 .2 Sistema de archivos.

Un repositorio de archivos para que los profesores puedan subir documentación y que esta sea descargable por los alumnos de los diferentes grupo.

12.2 .3 Un sistema de foros interno

Para que los alumnos de un grupo o curso puedan comunicarse de manera rápida y eficaz. Podría plantearse como un foro al uso o como un muro al estilo del antiguo Facebook (el nuevo timeline no me parece muy claro).