

Universidad de Alcalá  
Escuela Politécnica Superior

GRADO EN INGENIERIA DE COMPUTADORES

**Trabajo Fin de Grado**

Desarrollo de aplicación para comunicación y colaboración entre  
alumnos



**Autor:** Sergio Sanz Cacho

**Tutor/es:** Rosa María Estriégana Valdehita

ESCUELA POLITECNICA  
SUPERIOR

2021

**UNIVERSIDAD DE ALCALÁ**

**Escuela Politécnica Superior**

**Grado en ingeniería de computadores.**

**Trabajo Fin de Grado**

Desarrollo de aplicación para comunicación y colaboración entre alumnos

**Autor:** Sergio Sanz Cacho

**Tutor/es:** Rosa María Estriégana Valdehita

**TRIBUNAL:**

**Presidente:** Antonio José de Vicente Rodríguez

**Vocal 1º:** Manuel Prieto Mateo

**Vocal 2º:** Rosa Estriégana Valdehita

**FECHA:** .....

## Índice de contenidos

Índice de figuras	4
<b>1. Resumen</b>	<b>7</b>
<b>2. Abstract</b>	<b>8</b>
<b>3. Palabras Clave</b>	<b>9</b>
<b>4. Introducción</b>	<b>10</b>
<b>5. Herramientas empleadas en la realización del proyecto</b>	<b>12</b>
5.1 Lenguajes utilizados	12
5.2 Base de datos	12
5.3 Servidor local	13
5.4 Programas y herramientas empleadas durante el desarrollo	13
5.4.1 NetBeans	13
5.4.2 FileZilla	14
5.4.3 XAMPP	14
5.4.4 phpMyAdmin	15
<b>6. Desarrollo del proyecto</b>	<b>16</b>
6.1 Fase de adaptación	16
6.1.1 Cambios en el patrón MVC	16
6.1.2 Modificación del formato CSV	18
6.1.2 Modificación en la vista Login	20
6.1.3 Solución error en vista modificar usuario.	23
6.1.4. Cambio en el sistema de selección de grupos en cambio de rol	24
6.1.5. Mas comprobación a la hora de votar	27
6.1.6. Otros cambios	30
6.2 Fase de ampliación	31
6.2.1. Edición de notas	31
6.2.2. Foros	35
6.2.2.1 Foro Privado	35
6.2.2.2 Foro Publico	40
6.2.3. Sistema de penalización	42
<b>7. Estructura de la aplicación</b>	<b>46</b>
<b>8. Planificación</b>	<b>48</b>
<b>9. Futuras mejoras</b>	<b>50</b>
<b>10. Conclusiones</b>	<b>51</b>
<b>11. Presupuesto</b>	<b>52</b>
<b>12. Agradecimientos</b>	<b>53</b>

<b>13. Bibliografía</b>	54
<b>14. Referencias</b>	55
<b>ANEXO I - Tutorial de Uso Profesor</b>	56
1. Creación de un nuevo usuario	56
2. Editar Usuario y cambio de contraseña	57
3. Iniciar cambio de Rol	58
4. Importar usuarios a la base de datos a través de un archivo CSV	59
5. Acceso Foro Publico	60
6. Editar Notas	61
<b>ANEXO II - Tutorial de Uso Alumno</b>	63
1. Experto	63
2. Alumno	63
2.3 Acceso Foro Publico	63
2.3 Foro privado	65
<b>ANEXO III – Extractos de Código fuente</b>	66
Controlador de Foros	67
Modelo Foro Privado	69

## Índice de figuras

Ilustración 1 Logo NetBeans	13
Ilustración 2 Logo FileZilla	14
Ilustración 3 Logo XAMPP	14
Ilustración 4 Logo phpMyAdmin	15
Ilustración 5 index.php	17
Ilustración 6 Ejemplo llamada desde controlador	18
Ilustración 7 Extracto base antigua	19
Ilustración 8 Extracto base actual	19
Ilustración 9 Código SQL para fusionar columnas	19
Ilustración 10 Formato CSV	19
Ilustración 11 Formato CSV antiguo	19
Ilustración 12 Codificación utf-8	20
Ilustración 13 Mensaje email no valido	21
Ilustración 14 Mensaje de usuario/contraseña incorrectos	21
Ilustración 15 Mensaje usuario repetido	21
Ilustración 16 Mensaje usuario inexistente	21
Ilustración 17 Selector de pestaña	22
Ilustración 18 Mensajes de register	22
Ilustración 19 Código contraseña incorrecta	22
Ilustración 21 Editar usuario antiguo	<b>¡Error! Marcador no definido.</b>

Ilustración 22 Editar usuario actual -----	<b>¡Error! Marcador no definido.</b>
Ilustración 23 Código encargado de cargar el rol -----	<b>¡Error! Marcador no definido.</b>
Ilustración 24 Rotación actual -----	<b>¡Error! Marcador no definido.</b>
Ilustración 25 Rotación antigua -----	25
Ilustración 26 Sección encarga de cargar la vista -----	<b>¡Error! Marcador no definido.</b>
Ilustración 27 Desplegable de cursos -----	<b>¡Error! Marcador no definido.</b>
Ilustración 28 Función que carga los grupos -----	<b>¡Error! Marcador no definido.</b>
Ilustración 29 Votación antigua -----	<b>¡Error! Marcador no definido.</b>
Ilustración 30 Puntuar actual -----	28
Ilustración 31 Valor seleccionado no valido -----	29
Ilustración 32 Mensaje de aviso -----	29
Ilustración 33 Comprobación de muestra -----	<b>¡Error! Marcador no definido.</b>
Ilustración 34 Script de alerta -----	<b>¡Error! Marcador no definido.</b>
Ilustración 35 Carga de tutorial alumno -----	30
Ilustración 36 Función del controlador encargada de mostrar la edición de notas -----	<b>¡Error!</b>
<b>Marcador no definido.</b>	
Ilustración 37 Función de carga de notas -----	<b>¡Error! Marcador no definido.</b>
Ilustración 38 Captura del menú de edición de notas -----	<b>¡Error! Marcador no definido.</b>
Ilustración 39 Función del controlado r encargada de actualizar notas -----	<b>¡Error! Marcador no definido.</b>
<b>definido.</b>	
Ilustración 40 Función update notas -----	<b>¡Error! Marcador no definido.</b>
Ilustración 41 Código vista notasAdmin -----	<b>¡Error! Marcador no definido.</b>
Ilustración 42 Ventana de acceso a los subforos -----	<b>¡Error! Marcador no definido.</b>
Ilustración 43 Interior de un subforo -----	<b>¡Error! Marcador no definido.</b>
Ilustración 44 Respuesta de Hilo -----	<b>¡Error! Marcador no definido.</b>
Ilustración 45 Creación de hilo -----	<b>¡Error! Marcador no definido.</b>
Ilustración 46 Estructura foroPrivado -----	39
Ilustración 47 Foro controller-Registrar Publicación Privada -----	<b>¡Error! Marcador no definido.</b>
Ilustración 48 Foro privado1 -----	<b>¡Error! Marcador no definido.</b>
Ilustración 49 Foro Privado 2 -----	<b>¡Error! Marcador no definido.</b>
Ilustración 50 Vista del foro publico -----	<b>¡Error! Marcador no definido.</b>
Ilustración 51 Acceso foro público profesor -----	42
Ilustración 52 Vista Admin -----	<b>¡Error! Marcador no definido.</b>
Ilustración 53 Código penalización -----	<b>¡Error! Marcador no definido.</b>
Ilustración 54 Controler penalizar -----	<b>¡Error! Marcador no definido.</b>
Ilustración 55 Estructura de la aplicación -----	46
Ilustración 56 Planificación inicial -----	<b>¡Error! Marcador no definido.</b>
Ilustración 57 Crear usuario anexo -----	56
Ilustración 58 Primer login anexo -----	57
Ilustración 59 Editar usuario anexo -----	58
Ilustración 60 Iniciar cambio de Rol anexo -----	59
Ilustración 61 Importar CSV anexo -----	59
Ilustración 62 Formato CSV anexo -----	59
Ilustración 63 Acceso foro público anexo -----	60
Ilustración 64 Foro Publico anexo -----	60
Ilustración 65 Nuevo hilo anexo -----	61
Ilustración 66 Responder anexo -----	61
Ilustración 67 Editar notas anexo -----	61

## Desarrollo de aplicación para comunicación y colaboración entre alumnos

Ilustración 68 Foro público alumno anexo -----	64
Ilustración 69 Nuevo hilo alumno anexo -----	65
Ilustración 70 Responder alumno anexo -----	65
Ilustración 71 estructura de archivos 2 -----	66
Ilustración 72 Estructura de archivos 1 -----	66

## 1. Resumen

En la asignatura de fundamentos de ingeniería de computadores se dispone de varias aplicaciones web enfocadas al aprendizaje mediante juegos, y al aprendizaje colaborativo. Una de estas aplicaciones enfocada al aprendizaje colaborativo es una llamada Puzle donde los alumnos trabajan en grupos de 4 (o 5) alumnos y en cada actividad hay un alumno que tiene el papel de “experto”. El experto será el encargado de resolver con la profesora las dudas que tenga en la realización de la actividad y de que todos los miembros del equipo entiendan todas las tareas. Una vez finalizada la actividad, los alumnos votan la implicación y el trabajo realizado por el experto y el experto vota el trabajo realizado por cada uno de los alumnos.

El objetivo principal de este Trabajo fin de grado es mejorar las funcionalidades de dicha aplicación y añadir elementos que faciliten la comunicación entre los alumnos. El trabajo consta de dos partes, la primera está más enfocada a solucionar errores existentes en la aplicación, haciendo así más fácil su uso y gestión, como por ejemplo, algunos problemas a la hora de registrar alumnos en el sistema o problemas que tienen los alumnos a la hora de hacer las votaciones.

La segunda parte del proyecto consiste en ampliar las capacidades de la aplicación mediante la implementación de nuevas funcionalidades, como podrían ser la capacidad de editar las votaciones por parte de la profesora e incorporar a la aplicación un sistema de comunicación entre alumnos similar a un foro.

## 2. Abstract

In the subject of computer engineering fundamentals, there are several web applications focused on learning through games, and collaborative learning. One of these applications focused on collaborative learning is a so-called Puzzle where students work in groups of 4 (or 5) students and in each activity, there is a student who has the role of "expert". The expert will be in charge of solving with the teacher the doubts that he has in carrying out the activity and that all the members of the team understand all the tasks. Once the activity is finished, the students vote on the involvement and the work done by the expert and the expert votes on the work done by each of the students.

The main objective of this Final Degree Project is to improve the functionalities of said application and add elements that facilitate communication between students. The work consists of two parts, the first is more focused on solving existing errors in the application, thus making its use and management easier, such as some problems when registering students in the system or problems that students have at the time of voting.

The second part of the project consists of expanding the application's capabilities by implementing new functionalities, such as the ability to edit the teacher's votes and incorporate a forum-like communication system between students into the application.



## 3.Palabras Clave

PHP, HTML, MYSQL, JAVASCRIPT, SQL, JIGSAW, FORO

## 4.Introducción

Las habilidades para interactuar, cooperar con otros y saber trabajar en equipo, son valores muy demandados actualmente por las empresas. El desarrollo del trabajo en equipo ofrece importantes beneficios profesionales, sociales y también de aprendizaje ya que el desarrollo cognitivo depende en gran medida de la interacción social y la colaboración con otras personas (Estriégana et al., 2021).

Por otra parte, los alumnos aprenden mejor cuando se sienten parte de un grupo. El “idioma” común que comparten los alumnos ayuda al aprendizaje, fomenta la socialización, el compañerismo, la responsabilidad y el trabajo en equipo.

Es por estas razones que se desarrolló una aplicación web llamada Puzle para motivar e implicar a los alumnos a colaborar y a aprender entre ellos y que además permitiese valorar el trabajo realizado por los compañeros.

Dicha aplicación web, que se realizó hace unos años como Trabajo Fin de Grado, está basada en la metodología Jigsaw (o rompecabezas) descrita por Aronson, (1978). La metodología Jigsaw consiste en dividir a los alumnos en grupos, asignar parte de la lección a cada alumno dentro del grupo, dar tiempo de estudio y reflexión, después se agrupa a los alumnos de los distintos grupos por áreas para que debatan y resuelvan dudas y finalmente cada alumno regresa a su grupo donde presenta o explica su parte de contenido.

Pero existen múltiples variaciones posibles de esta metodología, tantas como deseemos. En este caso, la metodología Jigsaw ha sido adaptada para las prácticas de laboratorio de la asignatura Fundamentos de Tecnología de Computadores, aunque puede emplearse en cualquier asignatura y para diferente tipo de actividades. Esta versión de Jigsaw, consiste en dividir a los alumnos en pequeños grupos, en los cuales se asigna a un alumno con el rol de experto, dicho experto es el encargado de dirigir la actividad y de asegurarse de que el resto de los alumnos colaboren, comprendan y aprendan en el desarrollo de dicha actividad. Además el experto es el que debe solucionar las dudas que puedan surgir a sus compañeros de grupo.

Para hacer la metodología más atractiva para los alumnos y más fácil de gestionar por el profesor, se ha creado una aplicación web llamada Puzle. Donde se automatiza la gestión de grupos, la asignación de expertos y el proceso de votaciones. Así, una vez que una actividad termina, los alumnos deben puntuarse entre ellos. El experto evalúa a cada alumno de su grupo y el resto de los miembros evalúan al experto, tras esto se rotan los papeles y se comienza la siguiente actividad.

El principal objetivo de este trabajo consiste en mejorar la aplicación Puzle y en dotarla de más capacidades para hacer su uso más cómodo, ya que la versión original de Puzle presentaba ciertos fallos y carecía de algunas características de comunicación entre los alumnos.

El trabajo se ha planteado en dos fases principales: La primera, a la que llamaremos fase de adaptación, en la fase en la que se han solucionado fallos de la versión anterior de Puzle, como problemas en la gestión de la aplicación. Por ejemplo, los alumnos debían votar una vez terminada la actividad, pero antes de que se extinguiese el tiempo de activación y se activase una nueva actividad ya que cambian los roles de los alumnos y había algunos alumnos que no recordaban hacerlo a tiempo. Esto suponía errores en las votaciones y más trabajo en la gestión de la aplicación. Además, se ha modificado el funcionamiento interno de la aplicación para que se encuentre más estructurado y las distintas partes estén bien diferenciadas.

También en la fase de adaptación se ha modificado el formato mediante el cual se introducen los datos de los alumnos facilitando así el pre-registro de los alumnos a partir de las listas que la Universidad proporciona a los profesores. Se han añadido elementos como mensajes y comprobaciones para evitar que los alumnos voten de forma incorrecta, y otros cambios que detallaremos más adelante que hacen que el uso y la gestión de la aplicación sea más sencillo.

La segunda fase es la que llamaremos la fase de ampliación. En esta fase se han añadido nuevas funcionalidades como serían los dos foros para la comunicación, uno privado, interno de cada grupo de trabajo y otro público abierto al resto de compañeros y que sirve también para poder comunicarse con la profesora. También se ha añadido un módulo para que la profesora, en caso de que algún alumno se equivoque al realizar la votación, pueda corregir las notas, y un sistema de penalizaciones para así animar a los alumnos a que realicen la votación ya que en muchos casos los alumnos se olvidaban de votar. Así, cuando finaliza una actividad y se ejecuta la rotación de alumnos, todos aquellos que no hayan realizado la votación tendrán una penalización. Habrá un contador que se irá incrementado en un punto por cada alumno al que no hayan votado.

## 5.Herramientas empleadas en la realización del proyecto

### 5.1 Lenguajes utilizados

La aplicación puzle sobre la que está realizado este proyecto se encuentra desarrollada principalmente sobre PHP, por este motivo ha sido este lenguaje el elegido para el desarrollo de este Trabajo fin de grado. Además, PHP es un lenguaje muy útil a la hora de desarrollar aplicaciones web, debido a su fácil integración con HTML. Sumado a esto PHP es un lenguaje que nos permite la creación de sitios dinámicos, posee una conexión sencilla con múltiples bases de datos, además de poder recopilar con facilidad datos de formularios y cookies.

Para el correcto funcionamiento de toda la aplicación se emplean pequeños scripts a lo largo de todo su código, estos scripts se encuentran en su mayoría programados empleando JavaScript, este es un lenguaje sencillo que suele ser interpretado y a pesar de poder ser usado fuera del navegador, principalmente se conoce como un lenguaje para la realización de scripts en páginas web.

Otro lenguaje que se ha usado en menor medida, pero que ha sido muy útil a la hora de modificar elementos de las propias vistas a sido JQuery. JQuery es otro lenguaje para realización de scripts que simplifica el desarrollo de las aplicaciones web al manipular los elementos empleando código sencillo.

En cuanto a la estructuración de las vistas se ha empleado HTML para darles forma, ya que este es el lenguaje por excelencia para estructurar aplicaciones web. Se ha empleado CSS para dar estilo y formato a todos elementos que forman las distintas vistas.

Por último, pero no menos importante, al tratarse de un aplicación que interactúa con un base de datos, se ha empleado SQL para programar las diferentes consultas que es necesario realizar a la base de datos, tanto para insertar, consultar o actualizar datos referentes a los usuarios, foros, o actividades.

### 5.2 Base de datos

En cuanto a la base de datos mencionada anteriormente, se ha empleado MySQL. MySQL es un gestor de bases de datos relacionales de código abierto.

Como se mencionará más adelante, para gestionar esta base de datos se emplea phpMyAdmin, una interfaz que nos permite emplear al máximo todas las posibilidades

que nos ofrece MySQL sin la necesidad de emplear SQL para modificar nuestra base de datos.

### 5.3 Servidor local

Durante el proceso de desarrollo se copió tanto la base de datos como el código fuente de la página a un servidor local desplegado en XAMPP en el cual se han ido haciendo todos los cambios hasta obtener la versión final la cual ha sido publicada en el servidor de la universidad: <http://www3.uah.es/forole/puzle/index.php>

Al realizar todo sobre un servidor local se ha asegurado el poder volver siempre a la versión original en caso de encontrar algún problema o sufrir una pérdida de código o corrupción de datos.

De igual manera, antes de modificar algún aspecto crítico de aplicación se han ido realizando copias del progreso que se tenía hasta ese instante para poder deshacer los cambios fácilmente.

### 5.4 Programas y herramientas empleadas durante el desarrollo

#### 5.4.1 NetBeans



*Ilustración 1 Logo NetBeans*

NetBeans es un entorno de desarrollo que personalmente me resulta muy cómodo además de ser muy completo y potente. Ha sido elegido ya que había trabajado con el anteriormente y aunque nunca lo había usado para programar en php, ya que esta es la primera vez que he usado dicho lenguaje, sabía que era compatible con el mismo.

NetBeans también ofrece una ayuda a la hora de programar resaltando posibles errores del código y ofreciendo sugerencias a la hora de programar lo cual lo hace una herramienta perfecta para el desarrollo de esta aplicación.

#### 5.4.2 FileZilla



*Ilustración 2 Logo FileZilla*

En este caso se ha optado por este cliente ftp para realizar la conexión con el servidor a la hora de tener que bajar archivos o subir archivos al mismos.

Se ha tomado esta decisión ya que se trata de un cliente ftp con el cual ya había trabajado con anterioridad sintiéndome cómodo con él.

#### 5.4.3 XAMPP



*Ilustración 3 Logo XAMPP*

Esta herramienta ha sido seleccionada para desplegar el servidor local a la hora del desarrollo.

Se trata de un servidor independiente a la plataforma en la que se despliega, principalmente consta de los siguientes servicios: un servidor apache con compatibilidad php para el despliegue de páginas web y un servidor MySQL para bases de datos. Estos servicios lo convierten en el entorno perfecto para el desarrollo de este proyecto, al asemejarse en gran medida al servidor de la universidad en el que se alojará este proyecto. Además, cuenta con el mismo gestor que se emplea en los servidores de la universidad para las bases de datos phpMyAdmin.

#### 5.4.4 phpMyAdmin



*Ilustración 4 Logo phpMyAdmin*

Esta herramienta es el gestor de bases de datos empleado. La decisión de emplear este gestor para la creación y modificación de tablas dentro de la base de datos viene marcada principalmente por ser el gestor que ya dispone preinstalado XAMPP además de ser el gestor empleado en el servidor en el cual se alojara la aplicación.

A pesar de ser un poco forzado el uso de este gestor frente a otros, se trata de un herramienta completa y fácil de usar por lo tanto también se ha considerado la elección más adecuada para el proyecto.

## 6.Desarrollo del proyecto

En cuanto al desarrollo del proyecto este se encuentra dividido en dos fases a la primera fase la denominaremos la fase de adaptación y a la segunda la fase de ampliación.

### 6.1 Fase de adaptación

En esta fase del desarrollo de la aplicación se han realizado los cambios necesarios en el diseño existente. Dicho diseño sigue el patrón MVC, Modelo-Vista-Controlador.

Se han realizado bastantes cambios en cuanto a la estructura del código. A continuación, se van a exponer los cambios más significativos, ya que los demás cambios se pueden considerar derivados de estos.

#### 6.1.1 Cambios en el patrón MVC

Para entender correctamente los cambios y el porqué de estos, primero hemos de entender cómo funciona el patrón MVC.

La aplicación consta de un **index.php** y de un archivo llamado **routes.php**. Cuando se realiza una acción la orden es pasada al índice, y este le pasa al archivo routes el controlador y la orden, siendo este último archivo el encargado de cargar el controlador y de ejecutar la orden. El controlador realiza la lógica de dicha orden y posteriormente carga la vista necesaria.

Anteriormente la aplicación solo empleaba esta línea de ejecución para ordenes que careciesen de parámetros, mientras que toda aquella acción que requiriesen parámetros para su ejecución era mandada directamente a los controladores.

Con la intención de que el hilo de ejecución que siga cualquier acción sea el mismo, tanto al **index.php** como la archivo **routes.php** se les ha añadido la capacidad de trabajar con un array de parámetros mediante el cual el controlador puede recibir todos los datos necesarios para las distintas acciones sin necesidad de ser llamado directamente desde una vista.

Otro pequeño cambio llevado a cabo por cuestiones de seguridad es que todos los datos necesarios para el correcto funcionamiento de la aplicación (controladores, acciones y parámetros) se envían mediante el método post por lo que queda oculto al usuario, de esta manera conseguimos que no se pueda invocar ninguna vista llamando



directamente al **index.php** y empleando un método get, estando de esta manera los distintos controladores, acciones y parámetros ocultos al usuario.

```
<?php
if (!isset($_SESSION)) {
    session_start();
}
//Importamos el servicio de conexión con la base de datos
require_once('connection.php');

//Caso en el cual se recibe tanto controlador como acción como parámetros
if (isset($_POST['controller']) && isset($_POST['action']) && isset($_POST['args'])) {
    $controller = $_POST['controller'];
    $action = $_POST['action'];
    $args = $_POST['args'];
    //Nos aseguramos de que los parámetros estén en forma de array, sino separamos por comas y creamos array
    if (is_array($args)) {
        $parametros = $args;
    } else {
        $parametros = explode(",", $args);
    }
}

//Caso en el cual no se reciben parámetros
} elseif (isset($_POST['controller']) && isset($_POST['action'])) {
    $controller = $_POST['controller'];
    $action = $_POST['action'];
    //generamos una variable parámetros vacía
    $parametros = '';
}

//Caso que se da al cargar la página pero aún existe la sesión anterior en el navegador
//Se inicia sesión sin necesidad de login
} elseif (isset($_SESSION['id'])) {
    $controller = 'usuario';
    $action = 'index';
    $parametros = '';
}

//Por último caso en el que se inicia una nueva sesión, se carga el login
} else {
    $controller = 'usuario';
    $action = 'show_login';
    $parametros = array('false','', '');
}
require_once('routes.php');
?>
```

Ilustración 5 *index.php*

En la imagen anterior podemos observar la estructura del archivo **index.php**, este archivo es el núcleo principal ya que junto al **routes.php** son los encargados de dirigir la ejecución, ejecutando la acción del controlador adecuado pasándole a este los parámetros necesarios.

En este archivo hay una serie de posibles escenarios, el primero es el caso de que reciba tanto el controlador como la acción como parámetros este es el escenario de ejecución más común ya que es el que resulta de la mayoría de las acciones dentro de la aplicación.

El segundo escenario al igual que el primero es de los más comunes y se da en aquellas acciones que no requieren de parámetros por lo tanto el **index.php** solo va a recibir el controlador y la acción.

El tercer escenario solo se da cuando se cierra la pestaña en la que se encuentra la aplicación, pero la sesión se mantiene iniciada en el navegador por lo tanto al volver a entrar a la web, se detecta la existencia de la sesión por lo tanto se carga la vista principal con los datos de dicha sesión.

Y el último de los escenarios se da cuando se entra a la aplicación por primera vez, es decir, que no existe ninguna sesión creada bien sea porque se ha cerrado sesión o porque se abre el navegado de nuevo, en este caso se envía la orden de cargar la ventana de Login, con los parámetros adecuados, es decir la opción por defecto.

Tras cualquiera de esto escenarios se carga el archivo **routes.php** el cual contiene la lógica que se encarga de cargar el controlador y ejecutar la acción, además de esto también se encarga de comprobar que el controlador existe y en ese caso comprueba que la acción solicitada existe para dicho controlador, si cualquiera de las comprobaciones falla carga una vista de error.

Como último cambio relacionado con el modelo de la aplicación, al cambiar el método de comunicación con el **index.php**, ha sido necesario cambiar la forma de llamada al mismo desde los controladores, antes pasaban los parámetros empleando el método get, ahora se ha creado un pequeño script (HTML + JavaScript) para así poder enviar las ordenes al **index.php** desde los controladores.

```
echo ' <form name="f" id="vueltaAtras" action="" method="post">';
echo ' <input type="hidden" name="controller" value="foro" > ';
echo ' <input type="hidden" name="action" value="show_foro_actv" > ';
echo ' <input type="hidden" name="args[curso]" value="' . $parametros["curso"] . "'>';
echo ' <input type="hidden" name="args[grupolab]" value="' . $parametros["grupolab"] . "'>';
echo ' <input type="hidden" name="args[grupo]" value="' . $parametros["grupo"] . "'>';
echo ' <input type="hidden" name="args[id_actv]" value="' . $parametros["id_actv"] . "'>';
echo ' <input type="hidden" name="args[activo]" value="' . $parametros["activo"] . "'>';
echo ' </form> ';
echo ' <script type="text/javascript"> ';
echo ' document.getElementById("vueltaAtras").submit(); ';
echo ' </script>';
```

Ilustración 6 Ejemplo llamada desde controlador

### 6.1.2 Modificación del formato CSV

Uno de los cambios más importantes a la hora de mejorar el funcionamiento de la aplicación ha sido el del formato del CSV por el cual se introducen los alumnos al sistema.

## Desarrollo de aplicación para comunicación y colaboración entre alumnos

El anterior era complejo, con muchas columnas vacías, por ello, se ha decidido optar por un formato más sencillo que facilite la introducción de datos a la profesora.

Además de para simplificar el formato del CSV, este cambio se ha llevado a cabo porque la aplicación tenía muchos problemas a la hora de extraer del CSV nombres y apellidos compuestos. Para solucionar el problema se ha decidido fusionar ambas columnas de apellidos en una sola y emplear el formato dado por la universidad (APELLIDO-1 APELLIDO-2, NOMBRE) para separar el nombre de los apellidos.

<input type="checkbox"/>	2	nombre	varchar(30)
<input type="checkbox"/>	3	apellido1	varchar(30)
<input type="checkbox"/>	4	apellido2	varchar(40)

Ilustración 7 Extracto base antigua

En la imagen anterior podemos ver un extracto de la base de datos antigua, para adaptarla a los cambios se ha creado un nuevo campo llamado apellidos, mediante una instrucción SQL se han fusionado las dos columnas de apellidos y copiado a esta. Posteriormente se han eliminado estas dos columnas para evitar errores a la hora de cargar nuevos datos en el sistema.

<input type="checkbox"/>	1	grupo	varchar(20)
<input type="checkbox"/>	2	nombre	varchar(30)
<input type="checkbox"/>	3	apellidos	varchar(40)

Ilustración 8 Extracto base actual

```
UPDATE `usuarios` SET `apellidos`=CONCAT(CONCAT(`apellido1`,` ``,`apellido2`) WHERE 1;
```

Ilustración 9 Código SQL para fusionar columnas

A	B	C	D
DNI	APELLIDO-1 APELLIDO-2, NOMBRE	GRUPO	CURSO

Ilustración 10 Formato CSV

En la imagen anterior podemos observar un ejemplo del formato CSV a emplear, el cual además de eliminar las columnas vacías del formato anterior, opta por ocupar una sola columna para los apellidos y nombre siguiendo el formato de la universidad como ya hemos comentado anteriormente.

A	B	C	D	E	F	G	H	I	J
DNI		CURSO	GRUPO				NOMBRE	APELLIDO-1	APELLIDO-2

Ilustración 11 Formato CSV antiguo

Además del cambio del formato a la hora de importar el CSV, se ha añadido la capacidad de sobrescribir un usuario. Como ejemplo, pongamos el caso de que un alumno está repitiendo la asignatura, con el sistema antiguo ocurriría un error a la hora de introducirlo de nuevo ya que se encontraba registrado de otro año, de esta manera cada vez que se importa un nuevo CSV si se encuentran coincidencias el usuario antiguo es sobrescrito.

De igual manera se ha añadido un mensaje que alerta en caso de tener un error en el CSV para así evitar un error en la aplicación que causaría un fallo en su funcionamiento.

En cuanto al sistema de importación CSV el último arreglo que se ha realizado ha sido el codificar correctamente cómo se guardan los datos, ya que, al disponer de caracteres propios del alfabeto latino, había problemas a la hora de guardar los alumnos. Esto se ha solucionado dando el formato adecuado a la hora de importar el CSV.

```
} $file = fopen($fileName, "r");  
while ($row = fgetcsv($file, 0, ";")) {  
    $row = array_map("utf8_encode", $row);  
}
```

*Ilustración 12 Codificación utf-8*

### 6.1.2 Modificación en la vista Login

En la vista de login se han llevado a cabo varias acciones, todas ellas enfocadas a un uso más cómodo e intuitivo.

Primero se han añadido mensajes de alerta en caso de que un usuario inexistente intente registrarse, de igual manera se contempla la posibilidad de que un usuario se intente registrar por segunda vez, en cuyo caso también aparecería un mensaje alertando de la situación.

Junto a los mensajes de alerta anteriores se ha añadió el mensaje de usuario o contraseña incorrectos, junto a la comprobación de que el correo electrónico posea un formato correcto a la hora de registrarse.

Por último, cuando se realiza un registro correcto, la aplicación autóloga al usuario para que este no tenga que volver a introducir sus datos.

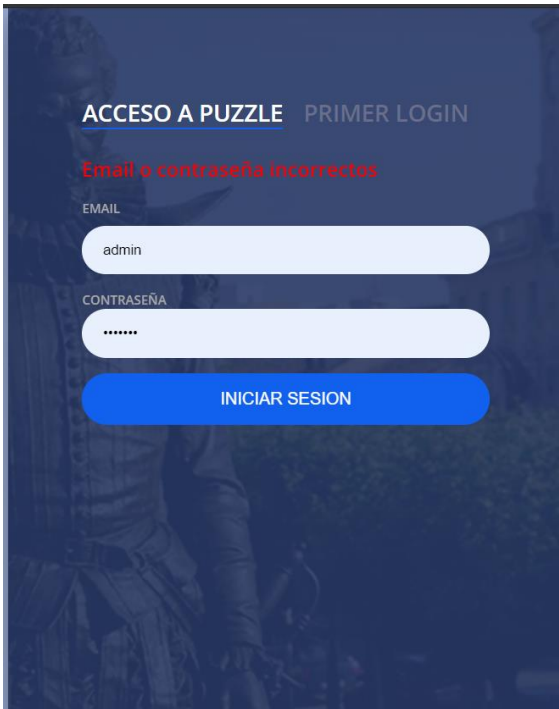


Ilustración 14 Mensaje de usuario/contraseña incorrectos

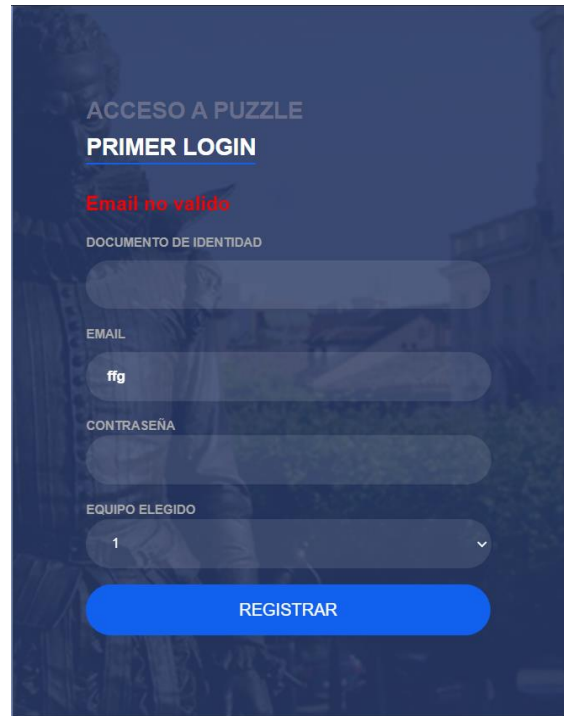


Ilustración 13 Mensaje email no valido

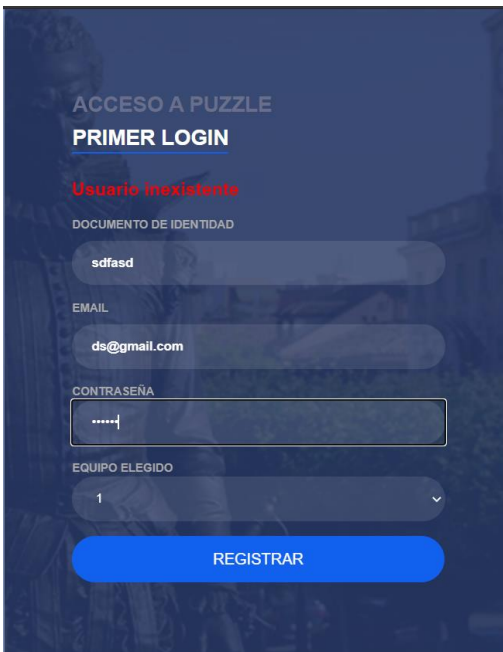


Ilustración 16 Mensaje usuario inexistente

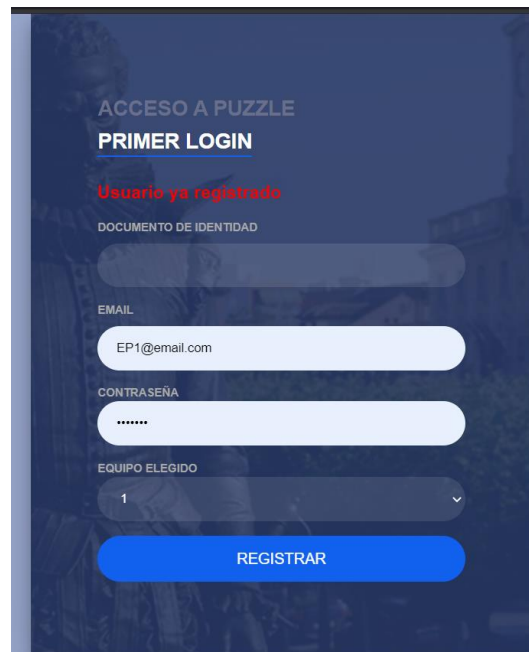


Ilustración 15 Mensaje usuario repetido

```

<div class="login-wrap">
  <div class="login-html">
    <?php if (isset($_POST["opcion"]) && $_POST["opcion"] == "registro") { ?>
      <input id="tab-1" type="radio" name="tab" class="sign-in"><label for="tab-1" class="tab">Acceso a puzzle</label>
      <input id="tab-2" type="radio" name="tab" class="sign-up" checked><label for="tab-2" class="tab">Primer Login</label>
    <?php } else { ?>
      <input id="tab-1" type="radio" name="tab" class="sign-in" checked><label for="tab-1" class="tab">Acceso a puzzle</label>
      <input id="tab-2" type="radio" name="tab" class="sign-up"><label for="tab-2" class="tab">Primer Login</label>
    <?php } ?>
  </div>
</div>

```

Ilustración 17 Selector de pestaña

Para poder mostrar todos estos mensajes de error se ha de recargar la vista login con los parámetros adecuados, debido a esto, se ha añadido una funcionalidad a la vista para que, en función de en qué pestaña se haya producido el error, al realizar la recarga de la vista esta se inicie en la venta correcta. Este mecanismo es el mostrado en la imagen anterior.

Además del parámetro encargado de seleccionar la pestaña inicial, esta vista también recibe desde el controlador otros parámetros encargados de seleccionar el mensaje de error en caso de ser necesario, a continuación, se pueden observar las dos secciones de código encargadas de mostrar estos mensajes.

```

<div class="group">
  <?php if (isset($_POST["err"]) && $_POST["err"] == 'true') { ?>
    <h3 style="color: red"> Email o contraseña incorrectos </h3>
  <?php } ?>
</div>

```

Ilustración 19 Código contraseña incorrecta

```

<div class="group">
  <?php if (isset($_POST["men"]) && $_POST["men"] == '2') { ?>
    <h3 style="color: red"> Usuario ya registrado </h3>
  <?php } else if (isset($_POST["men"]) && $_POST["men"] == '3'){ ?>
    <h3 style="color: red"> Usuario inexistente </h3>
  <?php } else if (isset($_POST["men"]) && $_POST["men"] == '4'){ ?>
    <h3 style="color: red"> Email no valido </h3>
  <?php } ?>
</div>

```

Ilustración 18 Mensajes de register

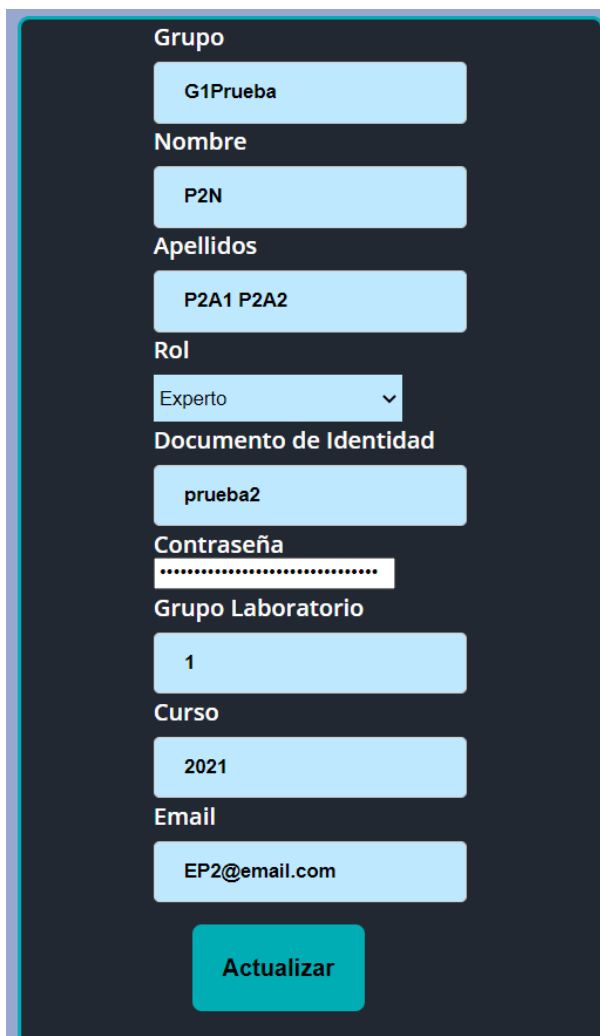
### 6.1.3 Solución error en vista modificar usuario.

En esta vista, como se puede observar en las siguientes imágenes, se han modificado tres cosas principalmente.

Primero se han añadido etiquetas a los campos para saber de qué se trata, ya que anteriormente los campos no disponían de esta etiqueta y podrían llegar a ser confusos.

Segundo, la versión antigua contenía un error por el cual el rol del alumno siempre aparecía como Experto en vez de aparecer como el rol actual del alumno lo que podía llegar a causar que en un grupo hubiera más de un experto si no se tenía cuidado. En la versión actual se ha añadido una comprobación que pone este campo al valor actual del rol del alumno a modificar.

Y tercero, se han remplazado los campos “Primer Apellido” y “Segundo Apellido” por el campo unificado Apellidos.

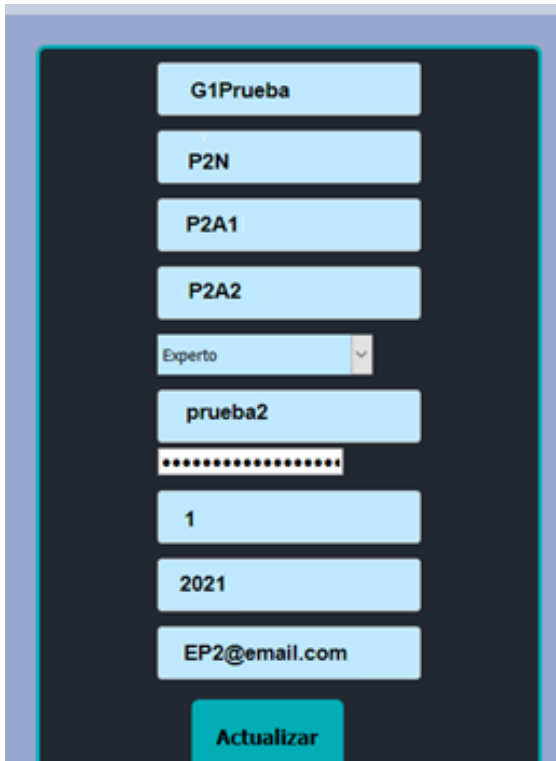


The image shows a user editing form with the following fields and values:

- Grupo:** G1Prueba
- Nombre:** P2N
- Apellidos:** P2A1 P2A2
- Rol:** Experto (dropdown menu)
- Documento de Identidad:** prueba2
- Contraseña:** [Redacted with dots]
- Grupo Laboratorio:** 1
- Curso:** 2021
- Email:** EP2@email.com

At the bottom of the form is a teal button labeled "Actualizar".

Ilustración 20 Editar usuario actual



The image shows a vertical form with several input fields and a button. The fields contain the following text from top to bottom: 'G1Prueba', 'P2N', 'P2A1', 'P2A2', 'Experto' (with a dropdown arrow), 'prueba2', a password field represented by a series of dots, '1', '2021', and 'EP2@email.com'. At the bottom of the form is a teal button labeled 'Actualizar'.

Ilustración 21 Editar usuario antiguo

#### 6.1.4. Cambio en el sistema de selección de grupos en cambio de rol

Anteriormente el cambio de rol se debía hacer introduciendo manualmente tanto el grupo como el curso que se pretendía rotar, lo cual podía llevar a confusiones ya que cabía la posibilidad de introducir de forma errónea el grupo o incluso el año que se pretendía rotar.

Debido a esto, se ha decidido introducir un sistema de rotación que carga automáticamente todos los cursos disponibles ordenándolos de más actual a más antiguo y a su vez carga también todos los grupos correspondientes a cada curso, de esta manera ya no se han de introducir estos datos de forma manual, sino que se encuentran precargados en dos desplegados evitando así los posibles errores y haciendo de este modo mucho más sencilla la rotación.





Ilustración 22 Rotación antigua



Ilustración 23 Rotación actual

Como se puede observar se dispone de dos desplegables, en el primero se cargan todos los cursos que el controlador ha guardado previamente en un array, y una vez seleccionado un curso entra en juego un pequeño script que se encarga de buscar y cargar los grupos correspondientes a dicho curso en el segundo desplegable.

```

<script type='text/javascript'>          function cargarGrupos() {

    //Valor del desplegable curso
    var curso = document.getElementById('curso').value;
    //Objeto desplegable grupo
    var grupo = document.getElementById('grupo');

    //Arrays de grupos y cursos
    var grupos = <?php echo json_encode($grupos) ?>;
    var cursos = <?php echo json_encode($cursos) ?>;

    //Reiniciamos el desplegable grupos
    grupo.innerHTML = '<option value="">Seleccionar grupo</option>';
    if (curso !== '') {
        var i = 0;
        var pos;
        //Buscamos la posición del curso dentro del array
        cursos.forEach(function (c) {
            if (c === curso) {
                pos = i;
            } else {
                i++;
            }
        });

        //Tomamos los grupos correspondientes a esa posición
        gs = grupos[pos];
        //Cargamos cada grupo en el desplegable de grupos
        gs.forEach(function (g) {
            let opcion = document.createElement('option');
            opcion.value = g;
            opcion.text = g;
            grupo.add(opcion);
        });
    }
}
</script></html>

```

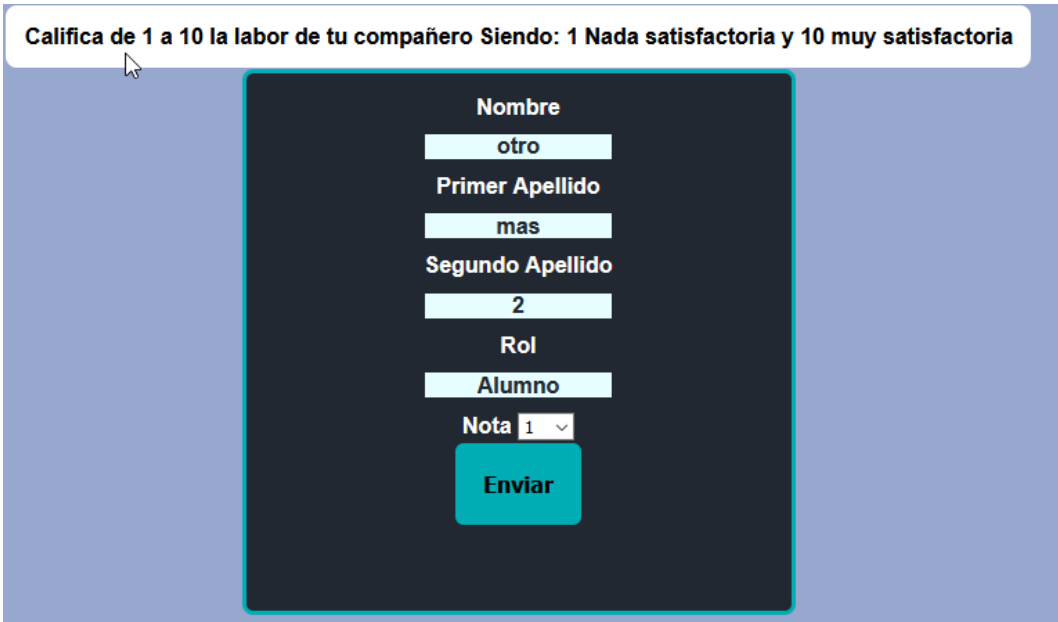
Ilustración 24 Función que carga los grupos

La ilustración anterior muestra el script encargado de cargar el desplegable de grupos.

Tanto el array de cursos, como los arrays de grupos se encuentran en el mismo orden, es decir, el primer array de grupos corresponde al curso situado en la primera posición del array de cursos. Una vez calculada la posición que ocupa un determinado curso dentro del array de cursos, se puede tomar el array de grupo correspondiente a esa posición y así cargar su contenido en el desplegable de grupos.

Esta función se ejecuta cada vez que se modifica la selección en el desplegable de cursos.

### 6.1.5. Mas comprobación a la hora de votar



Califica de 1 a 10 la labor de tu compañero Siendo: 1 Nada satisfactoria y 10 muy satisfactoria

Nombre  
otro

Primer Apellido  
mas

Segundo Apellido  
2

Rol  
Alumno

Nota 1

Enviar

*Ilustración 25* Votación antigua

Como podemos observar, anteriormente la puntuación aparecía por defecto en 1, lo cual hacía que en múltiples ocasiones los alumnos cometiesen errores a la hora de votar, además una vez votado a un compañero este seguía apareciendo disponible para volver a votar, en la nueva versión, como se verá a continuación, se han añadido más niveles de comprobación para asegurar que no se cometen errores a la hora de votar además de ocultar de la pantalla de votación los alumnos que ya han sido puntuados.

ACTIVIDAD: prueba1

Califica de 1 a 10 la labor de tu compañero o compañeros Siendo: 1 Nada satisfactoria y 10 muy satisfactoria

Nombre  
P3N

Primer Apellido  
P3A1

Segundo Apellido  
P3A2

Rol  
Experto

Nota - ▾

Enviar

*Ilustración 26 Votación actual*

En primer lugar, se ha añadido a la parte superior de la vista un título con el nombre de la actividad que se está puntuando, así los alumnos se pueden asegurar de que no cometen el error de puntuar la actividad actual pensando que están valorando la actividad anterior.

En segundo lugar, ya no parece una nota seleccionada por defecto, sino que aparece un -, de tal manera que si se intenta proceder a la votación con este valor seleccionado la aplicación devolverá un mensaje de error no realizando la votación.



Ilustración 27 Valor seleccionado no valido

De la misma manera que en la versión original se dispone de una alerta que obliga a confirmar la votación, en esta nueva versión dicha alerta se hace de una manera más personalizada, como podemos ver a continuación se muestra un mensaje con la puntuación y el nombre de la persona a la que se va a puntuar.

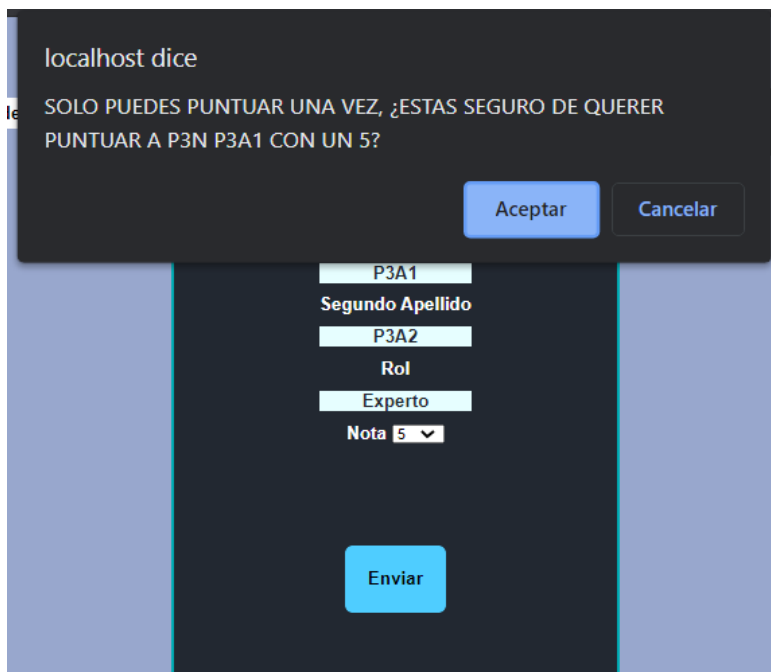


Ilustración 28 Mensaje de aviso

Para asegurarnos de que no se muestran en pantalla aquellos alumnos a los que ya se ha valorado, antes de mostrar el recuadro de votación la aplicación comprueba si el alumno a puntuar existe, ya en un array que recoge los IDs de aquellos alumnos a los que el alumno que realiza la votación ya ha valorado en la actividad.

Los mensajes, tanto el de alerta de que no se ha seleccionado puntuación, como el de confirmación de la votación, se generan empleando un pequeño script que comprueba si se ha seleccionado puntuación en cuyo caso lanza el mensaje de alerta o el de confirmación según corresponda.

### 6.1.6. Otros cambios

Por último, mencionar cambios menores pero que también son de utilidad. En el apartado dedicado a editar el contenido de una actividad se ha añadido la posibilidad de modificar la fecha de inicio ya que anteriormente solo se podía modificar la fecha de vencimiento.

También se ha modificado la forma mediante la cual se muestran los documentos que recogen el tutorial de uso tanto para el alumno como para el profesor, estos manuales anteriormente se debían descargar para su posterior visualización, en la versión actual de la aplicación se han creado dos scripts encargados de lanzar el visor de PDF en una nueva pestaña del navegador y de esta manera ya no es necesario descargar los documentos para su visualización.

```
<?php
$file = "Tutorial_Alumno.pdf";

header('Content-type: application/pdf');
header('Content-Disposition: inline; filename="' . $filename . '"');
header('Content-Transfer-Encoding: binary');
header('Content-Length: ' . filesize($file));
header('Accept-Ranges: bytes');

@readfile($file);
exit();
?>
```

*Ilustración 29 Carga de tutorial alumno*

También se han realizado pequeños cambios en el diseño de las vistas, ya sea para hacer más vistosa la aplicación o para reparar pequeños fallos del diseño de esta.

Además de los cambios mencionados hasta este punto, se han realizado otra serie de cambios en el código existente, pero todos ellos han sido para adaptar las características existentes al nuevo hilo de ejecución o a los cambios ya vistos anteriormente.

## 6.2 Fase de ampliación

En esta fase a la que se ha llamado fase de ampliación, como su propio nombre indica es en la que se han implementado las nuevas funcionalidades de puzle.

A continuación se exponen las acciones llevadas a cabo.

### 6.2.1. Edición de notas

En la versión anterior la profesora podía consultar las notas de un alumno accediendo a la misma vista a la que acceden los alumnos, esta vista le daba la opción de poder consultar las notas, pero no de editarlas individualmente.

Notas en las actividades		
Título de la actividad	Notas	Rol en la actividad
prueba 2	4.00	Experto
prueba1	1.00	Alumno

Nota media en cada rol	
Rol	Nota Media
Alumno	1.00
Experto	4.00

Nota media de los integrantes del grupo		
Nombre	Nota Media Experto	Nota Media Alumno
P1N	4.50	9.00
P2N	4.00	1.00
P3N	-	3.00

Ilustración 30 Ver notas antiguo/ Vista notas alumno

Como se puede ver en la imagen anterior la vista de la que disponía la profesora era la misma de que disponen los alumnos, cosa que no era de mucha utilidad.

Debido a lo comentado anteriormente, se ha implementado una vista con sus respectivas funciones dentro del controlador dedicada a dotar a la profesora de la capacidad de poder modificar de forma individual cada una de las puntuaciones recibidas por un determinado alumno en cada una de las actividades.

Para poder generar esta vista, se han creado funciones encargadas de cargar de forma individual las puntuaciones recibidas por un alumno y el ID del alumno que otorga cada

## Desarrollo de aplicación para comunicación y colaboración entre alumnos

puntuación, de esta manera la profesora puede editar la nota dada por un alumno. Modificando como consecuencia la media final y la media de dicha actividad.

Actividad: prueba1	
Nombre Compañero	Nota
P1N P1A1 P1A2	1

Actualizar

Actividad: prueba 2	
Nombre Compañero	Nota
P1N P1A1 P1A2	3
P3N P3A1 P3A2	5

Actualizar

No data available in table

Nota media en cada rol	
Rol	Nota Media
Alumno	1.00
Experto	4.00

Ilustración 31 Menú de edición de notas

En la imagen anterior se puede ver el aspecto que tiene esta vista para editar las notas, se puede observar que para el alumno que estamos comprobando se disponen de puntuaciones para la primera y la segunda actividad, y además se dispone de la capacidad de modificar cualquiera de estas notas.

Una vez modificadas estas puntuaciones pasamos a actualizar las notas. Dicha funcionalidad además de actualizar la nota que acaba de ser modificada, también debe actualizar las notas medias del alumno.

A continuación, se modifica una de las notas del alumno anterior para ver cómo se modifican las puntuaciones.



## Desarrollo de aplicación para comunicación y colaboración entre alumnos

Show 10 entries		
Actividad: prueba1		
Nombre Compañero	Nota	Rol alumno puntuado
P1N P1A1 P1A2	1	Alumno
	<input type="button" value="Actualizar"/>	

Show 10 entries		
Actividad: prueba 2		
Nombre Compañero	Nota	Rol alumno puntuado
P1N P1A1 P1A2	5	Experto
	<input type="button" value="Actualizar"/>	
P3N P3A1 P3A2	5	Experto
	<input type="button" value="Actualizar"/>	

Show 10 entries		
Actividad: PRUEBA3		
Nombre Compañero	Nota	Rol alumno puntuado
No data available in table		

Show 10 entries		
Nota media en cada rol		
Rol	Nota Media	
Alumno	1.00	
Experto	5.00	

Ilustración 32 Notas actualizadas

Como se puede ver en la ilustración 32, se ha modificado una de puntuaciones de la segunda actividad, y con respecto al instante anterior (ilustración 31) se pueden apreciar los cambios en la nota media de la parte inferior. A continuación podemos observar como, desde la vista del alumno, también se muestran la actualización de las puntuaciones, con respecto al estado anterior (Imagen 30).

Show 10 entries

Notas en las actividades		
Título de la actividad	Notas	Rol en la actividad
prueba 2	5.00	Experto
prueba1	1.00	Alumno

Show 10 entries

Nota media en cada rol	
Rol	Nota Media
Alumno	1.00
Experto	5.00

Show 10 entries

Nota media de los integrantes del grupo		
Nombre	Nota Media Experto	Nota Media Alumno
P1N	4.50	9.00
P2N	5.00	1.00
P3N	-	3.00

Ilustración 33 Vista notas alumno actualizada

En cuanto al código de la propia vista se va a adjuntar la sección más relevante del código encargado de crear y mostrar las tablas de las notas para editar. En este código empleando php dentro del HTML se crea una tabla por actividad y en caso de tener notas para el alumno que se está consultando en una determina actividad rellena las tablas con los datos correspondientes, nombre del alumno que otorgó la calificación, calificación, rol del alumno votado y por último el botón para actualizar dicha calificación.

```
//Se crea una tabla por actividad existente
foreach ($actividades as $actividad) {
    //Cargamos el id de la actividad a a la que corresponde la tabla actual
    $idAct = $actividad->post_id;
    ?>

    <div class="containerNadmin">

        <table id="" >
            <caption style="color: #393e46; margin-bottom: 5px; background: white;">Actividad: <?php echo $actividad->post_tema ?></caption>
            <thead>
                <tr>
                    <th>Nombre Compañero</th>
                    <th>Nota</th>
                    <th>Rol alumno puntuado</th>
                </tr>
            </thead>
            <tbody>
                <?php
                //Si existen notas para la actividad actual
                if ($notas[$i] != "-") {
                    //Rellenamos una fila de la tabla por nota existente
                    foreach ($notas[$i] as $nota) {
                        ?>

                        <tr>
                            <td><?php
                                //Buscamos dentro de los compañeros aquel que tiene el mismo que el id_votante asociado a la nota
                                foreach ($partners as $partner) {
                                    if ($partner->id == $nota->idvotante) {
                                        echo $partner->nombre." ".$partner->apellidos;
                                    }
                                }
                            ?></td>
                        </tr>
                    }
                }
            </tbody>
        </table>
    </div>
}
```

Ilustración 34 Extracto vista ver notas

### 6.2.2. Foros

Una de las partes más importantes de este proyecto es la creación de un sistema para que los alumnos puedan comunicarse entre ellos, en este caso se ha decidido crear una herramienta tipo foro la cual permite a los alumnos comunicarse entre ellos y también con la profesora.

Se ha decidió crear dos foros similares, pero con distinto enfoque.

El primero, denominado foro privado, consiste en un foro dividido en subforos, uno por cada actividad. A este foro solo tiene acceso el grupo de trabajo y su objetivo es que los alumnos se comuniquen con el experto y viceversa. Siendo así la herramienta idónea para que el experto organice el trabajo grupal y para que los miembros del grupo puedan ir exponiendo sus dudas y avances.

El segundo, denominado foro público, engloba a todo el grupo del laboratorio y a la profesora, en este foro los alumnos pueden compartir sus dudas con la clase y de esta manera ayudarse entre todos independientemente de su grupo de trabajo, lo interesante de este foro es la capacidad de interacción con la profesora.

Esta mejora se considera una de las más importante ya que el disponer de una herramienta de comunicación ayuda en gran medida al trabajo grupal. Además viviendo una pandemia como la de los últimos años, el disponer de una herramienta de comunicación integrada en la propia plataforma de trabajo ayuda a los alumnos a relacionarse los unos con los otros, favoreciendo así al trabajo colaborativo y la interacciones sociales.

#### 6.2.2.1 Foro Privado

Como se ha mencionado anteriormente este foro es solo para los miembros del grupo de trabajo y se encuentra dividido en subforos uno por cada actividad disponible.

Dentro de estos subforos, se diferencian los mensajes escritos por el experto de aquellos escritos por el resto de alumno, esta diferenciación se lleva a cabo mediante el uso de colores.

También hay que destacar que los subforos por actividad solo están activos para la actividad que actualmente se encuentra en curso, esto significa que los subforos de actividades pasadas son consultables, pero no se podrán crear hilos nuevos o responder a los existentes.

## Desarrollo de aplicación para comunicación y colaboración entre alumnos

Mensajes de la actividad	Fecha de vencimiento
prueba1	2021-08-31 <span>Actividad vencida</span>
prueba 2	2021-09-02

Ilustración 35 Ventana acceso a subforos

prueba1

Asunto: Hilo de prueba  
Autor: P1N P1A1 P1A2  
Contenido:  
Hola soy el Experto

Asunto: RE: Hilo de prueba  
Autor: P2N P2A1 P2A2  
Contenido:  
Yo estoy respondiendo como alumno

Asunto: Esto es otro hilo  
Autor: P2N P2A1 P2A2  
Contenido:

Ilustración 36 Subforo pasado

prueba 2

Nuevo Hilo

Asunto: Hilo de prueba  
Autor: P2N P2A1 P2A2  
Contenido:  
Soy el experto de esta actividad

Asunto: RE: Hilo de prueba  
Autor: P1N P1A1 P1A2  
Contenido:  
Yo en esta actividad soy un alumno

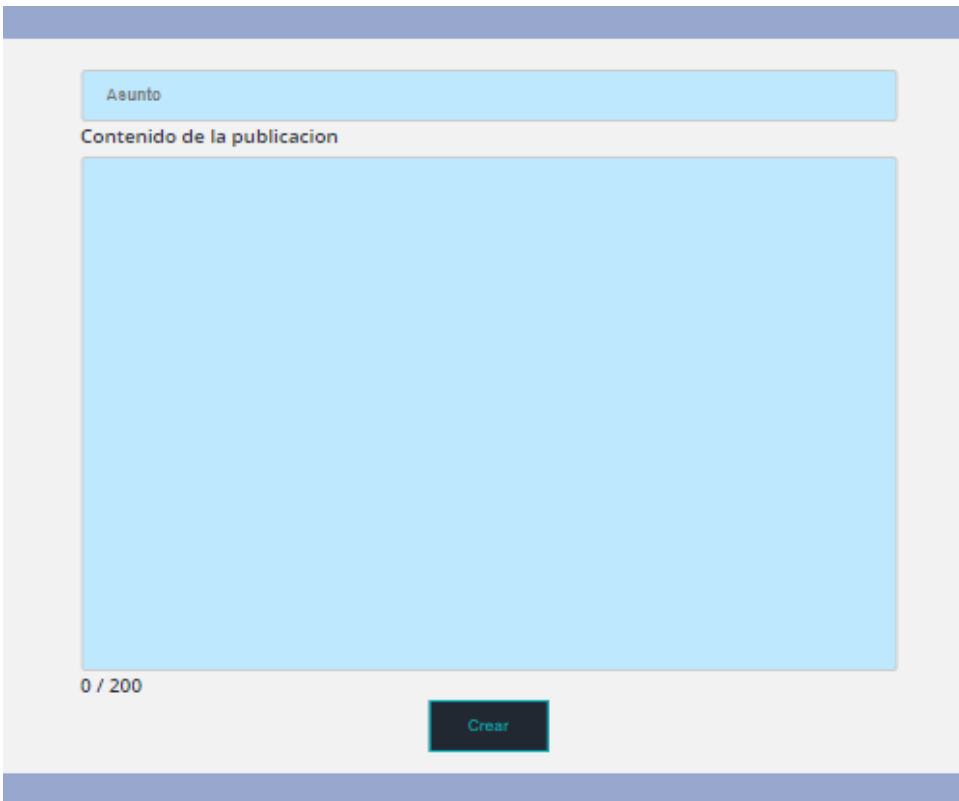
Responder

Ilustración 37 Subforo activo

## Desarrollo de aplicación para comunicación y colaboración entre alumnos

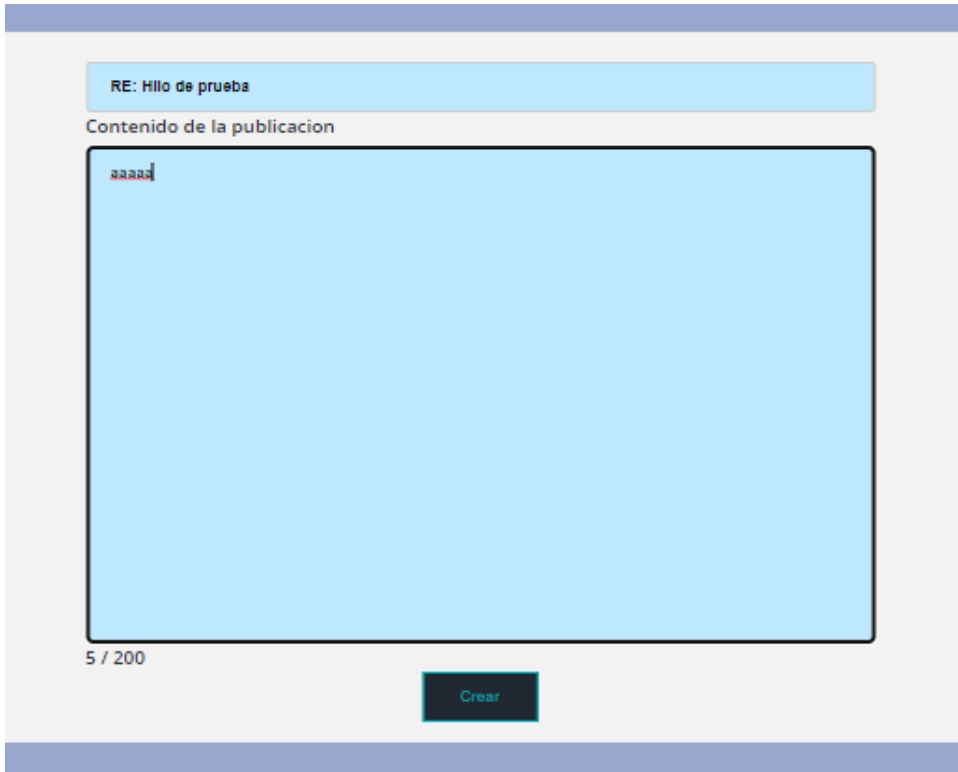
En las imágenes anteriores se puede observar tanto el menú para acceder a los subforos como dos subforos, uno pasado y uno activo. En estas imágenes podemos observar la estructura de uno de estos subforos, se encuentran divididos en distintos hilos y todos los mensajes de un mismo hilo se encuentran englobados en un mismo contenedor

En el subforo activo se tiene la posibilidad de crear un nuevo hilo o responder a uno de los existentes en el caso de responder a uno de los existentes el asunto del mensaje se genera por defecto no dejando modificarlo, de esta manera junto con los contenedores independientes se distinguen los mensajes que pertenecen a cada hilo.



The image shows a web form for creating a new forum thread. It features a light blue header bar at the top. Below it, there is a text input field labeled 'Asunto' (Subject). Underneath the subject field is a larger text area labeled 'Contenido de la publicación' (Content of the publication). At the bottom left of the content area, there is a character count '0 / 200'. At the bottom center, there is a dark blue button with the text 'Crear' (Create).

*Ilustración 38 Creación de un Hilo*



*Ilustración 39 Respuesta a un Hilo*

Como se puede observar en las imágenes anteriores, tanto para crear como para responder a un hilo, la longitud máxima del contenido es de 200 caracteres, esto se ha realizado de esta manera para que los mensajes sean breves y concisos además de que esta forma nos aseguramos de que todo mensaje se almacene en la base de datos y no se pierda información.

Para este foro se ha creado una nueva tabla en la base de datos, para guardar los mensajes. A esta tabla se le ha llamado foro\_priv y sus campos son los siguientes: curso, grupolab y grupo que son para distinguir a que foro pertenece un mensaje y así no tener confusiones entre distintos años, distintos grupos de laboratorio o grupos de trabajo. Id\_User y id\_actv son para distinguir a que actividad pertenece y quien escribe el mensaje. Rol indica de qué color ha de ser el contenedor del mensaje. Asunto y contenido guardan el mensaje en sí. Y por último, hilo, este campo está pensado para identificar a que hilo pertenece un mensaje, este valor se genera automáticamente al crear un nuevo hilo y es pasado por parámetros al responder a un mensaje.

## Desarrollo de aplicación para comunicación y colaboración entre alumnos

Nombre	Tipo
curso	int(10)
grupolab	int(3)
grupo	varchar(20)
id_User	int(5)
id_actv	int(5)
rol	varchar(10)
asunto	varchar(40)
contenido	varchar(250)
hilo	int(3)
id_publi 	int(5)

Ilustración 40 Estructura foroPrivado

```
<?php
//Si el foro esta activo cargamos boton crear hilo
if ($activo) { >>
    <form class="opcionesMenu" id="Publi" action="" method="post" >
        <input type="hidden" name="controller" value="foro">
        <input type="hidden" name="action" value="show_register_publi">
        <input type="hidden" name="args[id_actv]" value="<?php echo $id_actv ?>>
        <input type="hidden" name="args[activo]" value="<?php echo $activo ?>>
        <a style="margin-right: auto; transform: translateX(-50%); margin-left: 50%; width: max-content; background-color:#222831; border: 3px solid #00adb5;color:#00adb5;" href="javascript:;">
    </form>
</php } >>

<?php
//Si existen publicaciones continuamos cargandolos hilos
if (isset($publicaciones)) {

    foreach ($publicaciones as $hilos) {
        //Para cada hilo cargamos el asunto de su primer mensaje y el numero de hilo
        $hilo = $hilos[0]->hilo;
        $asunto = $hilos[0]->asunto;
        >>

        <div class="containerHilo">
            <?php
            foreach ($hilos as $publi) {
                require_once 'Models/usuario.php';
                //para cada publicacion del hilo se cargan los datos del usuario
                $alum = Usuario::getById($publi->id_User);
            }
            >>
            <div class="<?php
            //En funcion del rol se selecciona un tipo de contendor
            if ($publi->rol == 'Experto') {
                echo 'containerPublicacion2';
            } else {
                echo 'containerPublicacion1';
            }
            >>>
        </div>
    }
}
}>>
```

Ilustración 41 Foro privado 1

```
<span> Asunto: <?php echo $publi->asunto; ?> </span>
<br/><br/>
<span> Autor: <?php echo ($alum->nombre) . " " . ($alum->apellidos); ?> </span>
<br/><br/>
<span> Contenido: </span>
<br/><br/>
<span> <?php
    $contenido = str_replace("\n", "<br/>", $publi->contenido);
    echo $contenido;
?> </span><br>
</div>
<?php }
//Si el foro esta activo se muestra elo boton de responder al hilo
if ($activo) { ?>
    <form class="opcionesMenu" id="Publi" action="" method="post" >
        <input type="hidden" name="controller" value="foro" >
        <input type="hidden" name="action" value="show_register_publi">
        <input type="hidden" name="args[hilo]" value="<?php echo $hilo ?>" >
        <input type="hidden" name="args[asunto]" value="<?php echo $asunto ?>" >
        <input type="hidden" name="args[id_activ]" value="<?php echo $id_activ ?>" >
        <input type="hidden" name="args[activo]" value="<?php echo $activo ?>" >
        <a style="margin-right: auto; transform: translateX(-50%); margin-left: 50%; width: max-content; background-color:#222831; border: 3px solid #00adb5;color:#00adb5;" href="#" >
    </form>
    <?php ?>
</div>
<?php
}
?>
```

Ilustración 42 Foro privado 2

Las ilustraciones anteriores muestran el código HTML que da forma a los subforos del foro privado. Su funcionamiento es el siguiente: en primer lugar, comprueba si el subforo esta activo, en cuyo caso cargara el botón para crear un hilo nuevo.

A continuación, comprueba si hay publicaciones, en cuyo caso cargara hilo por hilo.

Para cada hilo guardara el asunto principal y el número de hilo para poder recuperarlos a la hora de responder al hilo.

Después mostrara cada uno de los mensajes del hilo, para cada mensaje cargará los datos del usuario que lo escribió partiendo del ID de usuario asociado a cada mensaje, también hará una comprobación para ver de qué clase es cada mensaje así poder otorgarle el contenedor del tipo adecuado, mensaje de tipo experto o mensaje de tipo alumno.

Por último, a la hora de mostrar el contenido del mensaje sustituirá los saltos de línea “\n” por el carácter de salto de línea de HTML “<br/>”, de esta manera se mostrará el contenido con el formato adecuado.

De igual manera que con el botón de crear hilo, se comprueba si el subforo esta activo para mostrar el botón de responder hilo.

### 6.2.2.2 Foro Publico

Mientras que al foro privado solo tenían acceso los miembros de cada grupo de trabajo, a este foro tienen acceso todos los alumnos de un determinado grupo de laboratorio junto con la profesora.



## Desarrollo de aplicación para comunicación y colaboración entre alumnos

Este foro también hace distinción, mediante contenedores de colores, del rol de la persona que escribió un mensaje, aunque en este caso se dispone del rol de administrador, es decir el profesor, al que se le ha otorgado un color distinto a los dos ya existentes.



Ilustración 43 Foro publico

La principal diferencia entre el foro público y los subforos privados es que en este los hilos, al ser de todo el curso, se ordenan de más reciente as más antiguo, de esta manera los hilos creados más recientemente siempre estarán en la zona superior. En adición al orden cronológico, por una cuestión de comodidad a la hora de buscar un determinado mensaje se les ha añadido un campo fecha.

En cuanto al funcionamiento es idéntico al de los subforos privados disponiendo de un máximo de 200 caracteres para el cuerpo del mensaje, y no pudiendo cambiar el asunto a la hora de responder un mensaje.

Para que el profesor pueda acceder a todos los grupos se ha añadido una ventana intermedia con un funcionamiento similar al mecanismo de cambio de rol explicado anteriormente.

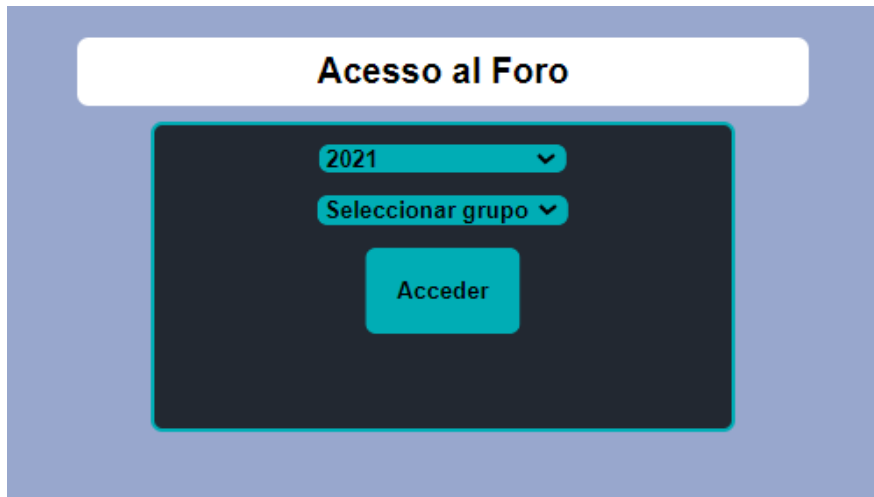


Ilustración 44 Acceso foro público profesor

### 6.2.3. Sistema de penalización

La última funcionalidad añadida a la aplicación es un sistema de penalizaciones, pensado para evitar que los alumnos se abstengan de votar ya que en muchos casos decidían no puntuar a un compañero o directamente ni se preocupaban de acordarse de la votación. De esta manera cada vez que un alumno olvida votar a un compañero, a la hora de hacer la rotación de rol estos se llevan un punto negativo por cada alumno no votado en la última actividad.

Curso	Grupo	Equipo	Nombre	Apellidos	Rol	DNI/NIE	Nota Alumno	Nota Experto	Penalizacion	Editar	Eliminar	Notas
2021	G1Prueba	1	P2N	P2A1 P2A2	Alumno	prueba2	1.00	5.00	0	Editar	Eliminar	Ver/Editar Notas
2021	G1Prueba	1	P3N	P3A1 P3A2	Experto	prueba3	3.00	-	0	Editar	Eliminar	Ver/Editar Notas
2021	G1Prueba	1	P1N	P1A1 P1A2	Alumno	prueba1	9.00	4.50	0	Editar	Eliminar	Ver/Editar Notas

Showing 1 to 3 of 3 entries  
 Previous 1 Next

Ilustración 45 Vista administrador antes de cambio de rol

En la imagen anterior vemos un caso en el cual disponemos de un grupo de trabajo compuesto por dos alumnos y un experto, actualmente todo ellos tienen el contador de

## Desarrollo de aplicación para comunicación y colaboración entre alumnos

penalización a 0, en la siguiente imagen veremos el resultado, tras ejecutar una rotación de rol en la cual solo uno de los alumnos ha realizado la votación.

Curso	Grupo	Equipo	Nombre	Apellidos	Rol	DNI/NIE	Nota Alumno	Nota Experto	Penalizacion	Editar	Eliminar	Notas
2021	G1Prueba	1	P2N	P2A1 P2A2	Alumno	prueba2	1.00	5.00	1	Editar	Eliminar	Ver/Editar Notas
2021	G1Prueba	1	P3N	P3A1 P3A2	Alumno	prueba3	3.00	4.00	2	Editar	Eliminar	Ver/Editar Notas
2021	G1Prueba	1	P1N	P1A1 P1A2	Experto	prueba1	9.00	4.50	0	Editar	Eliminar	Ver/Editar Notas

Showing 1 to 3 of 3 entries  
Previous 1 Next

Ilustración 46 Vista administrador tras cambio de rol

Como podemos observar el alumno que no voto tiene un punto de penalización, y el que era experto, que tampoco voto, tiene dos puntos de penalización ya que no puntuó a ninguno de sus compañeros.

A continuación, se adjunta el código encargado de realizar las rotaciones, seguido de la explicación de este ya que se considera una parte de gran interés en cuanto a las implementaciones llevadas a cabo.

```

public static function penalizar($curso, $grupo, $idActiv) {
    //El id de kla actividad sobre la que se va a comprobar viene dado por parametros

    $db = Db::getConnection();

    //Extraemos todos los usuarios de un grupo y curso determinados
    $select = $db->prepare('SELECT * FROM usuarios WHERE grupo=:grupo AND curso=:curso');
    $select->bindValue('grupo', $grupo);
    $select->bindValue('curso', $curso);
    $select->execute();

    //Para cada usuario comprobamos su rol
    foreach ($select->fetchAll() as $usuario) {
        if ($usuario['rol'] == "Experto") {
            //En caso de experto calculamos a cuantos deberia aver votado restandole 1 al total de su grupo de trabajo
            $compañeros = Usuario::members($grupo, $usuario['grupolab'], $curso);
            $numeroNotas = sizeof($compañeros) - 1;
            $selectN = $db->prepare('SELECT * FROM notas2 WHERE id_actividad=:id_actividad AND idvotante=:idvotante');
            $selectN->bindValue('id_actividad', $idActiv);
            $selectN->bindValue('idvotante', $usuario['id']);
            $selectN->execute();
            $numeroPuntuados = $selectN->rowCount();
            //Se calcula la penalizacion y se actualiza su valor en la base de datos
            $penalizacion = $numeroNotas - $numeroPuntuados;
            $update = $db->prepare('UPDATE usuarios SET penalizacion=penalizacion+:penalizacion WHERE id=:id');
            $update->bindValue('penalizacion', $penalizacion);
            $update->bindValue('id', $usuario['id']);
            $update->execute();
        } else {
            //En caso de ser alumno normal sencillamente comprobamos que haya votado una vez
            $selectN = $db->prepare('SELECT * FROM notas2 WHERE id_actividad=:id_actividad AND idvotante=:idvotante');
            $selectN->bindValue('id_actividad', $idActiv);
            $selectN->bindValue('idvotante', $usuario['id']);
            $selectN->execute();
            $numeroPuntuados = $selectN->rowCount();
            $penalizacion = 1 - $numeroPuntuados;
            $update = $db->prepare('UPDATE usuarios SET penalizacion=penalizacion+:penalizacion WHERE id=:id');
            $update->bindValue('penalizacion', $penalizacion);
            $update->bindValue('id', $usuario['id']);
            $update->execute();
        }
    }
}

```

Ilustración 47 Código penalizar

Al ejecutar la rotación de un determinado grupo también se ejecuta el código encargado de llevar a cabo las penalizaciones.

En un primer lugar extrae todos los usuarios de un determinado grupo y curso, y los separa en expertos y en alumnos.

En el caso de ser expertos para saber cuántos votos ha de haber realizado se toma el total de miembros de su grupo de trabajo y se le resta uno, de esta manera se obtiene el número de compañeros que tiene.

Tras esto se comprueba a cuanta gente ha votado dicho experto, en la actividad pasada por parámetro, una vez comprobado se realiza la diferencia con respecto a los que tendría que haber puntuado y se suma el resultado al valor ya existente de penalización.

En el caso de ser un alumno normal sencillamente se comprueba que haya realizado una votación es decir que haya valorado al experto. Tras esto en caso de no haber votado se suma un punto al contador de penalización, en caso contrario se suma cero puntos.

```
public function penalizar($curso, $grupo) {
    //Lista de actividades
    $listaPost = Post::temas();

    //Para cada actividad se comprueba si ya ha expirado
    foreach ($listaPost as $post) {

        if ($post->post_fecha < date('Y-m-d')) {
            $listaPasados[] = $post;
        }
    }

    if (isset($listaPasados)) {
        //De todas las actividades expiradas se toma el id de la mas reciente
        $id = $listaPasados[0]->post_id;
        $fecha = $listaPasados[0]->post_fecha;
        foreach ($listaPasados as $post) {
            if ($fecha < $post->post_fecha) {
                $id = $post->post_id;
                $fecha = $post->post_fecha;
            }
        }

        UsuarioExtra::penalizar($curso, $grupo, $id);
    }
}
```

Ilustración 48 Controlador penalizar

Dentro del controlador de usuario tenemos la función de penalizar la cual llama a la función mencionada anteriormente (ilustración 47), en esta función se calcula la última actividad expirada para pasarla por parámetro a la función penalizar (ilustración 47).

Para lograr esto primero se toman todas actividades existentes, de estas actividades se extraen todas aquellas que ya han expirado, y de estas últimas se va comprobando su fecha de expiración guardando solo el id de la más reciente.

## 7. Estructura de la aplicación

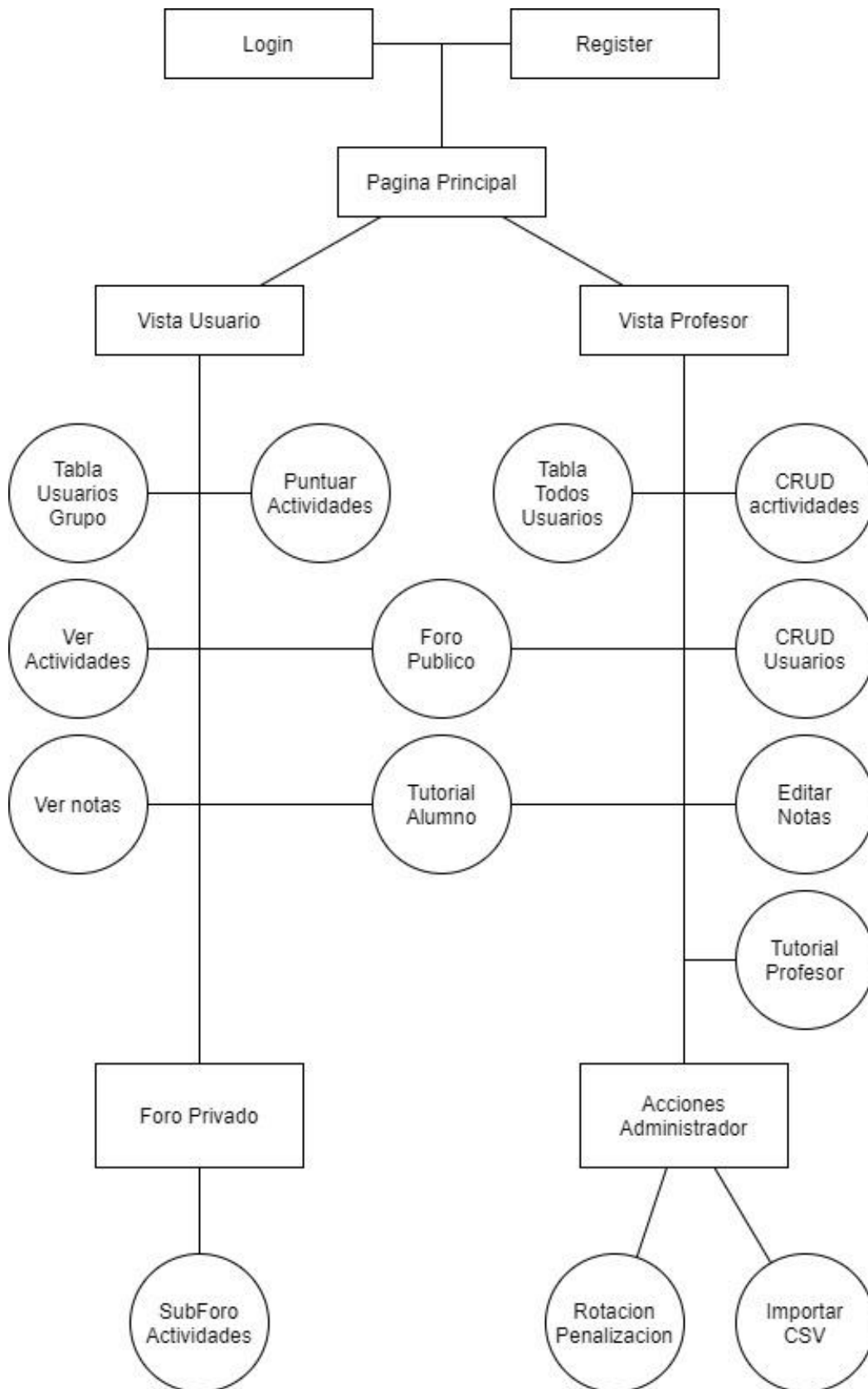


Ilustración 49 Estructura de la aplicación

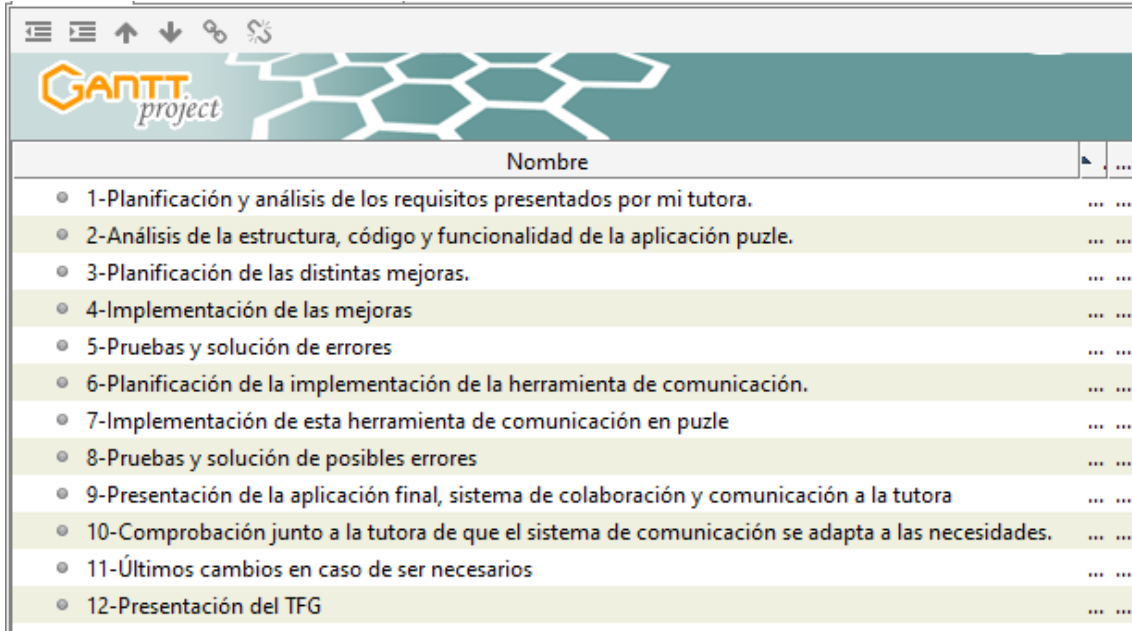
En la imagen anterior, se puede observar la estructura actual de la aplicación, la cual ha cambiado un poco con respecto a la estructura original.

En esta nueva estructura se tienen apartados como serian la edición de notas o los foros, el grafico de la estructura sirve como guía para saber todos los posibles caminos que se pueden llegar a seguir a la hora de usar la aplicación.

Comenzamos en la ventana de login, que posee tanto el login como el register, una vez logeados o registrados nos da acceso a la ventana principal, la cual tiene dos vistas, la vista de usuario desde la que se puede consultar las notas propias, consultar las actividades, consultar el tutorial de alumno, realizar las votaciones a los miembros del grupo y acceder a los foros.

La otra vista disponible es la de administrador esta vista nos deja acceder a todas las acciones sobre los alumnos eliminar, crear, editar, modificar sus notas. Nos deja consultar, modificar eliminar y crear publicaciones. También tendríamos acceso al foro público, a los tutoriales de administrador y alumno y por último desde esta vista se accede a las opciones de administrador desde las cuales se puede rotar el rol e importar un CSV de los alumnos.

## 8. Planificación



Nombre
1-Planificación y análisis de los requisitos presentados por mi tutora.
2-Análisis de la estructura, código y funcionalidad de la aplicación puzle.
3-Planificación de las distintas mejoras.
4-Implementación de las mejoras
5-Pruebas y solución de errores
6-Planificación de la implementación de la herramienta de comunicación.
7-Implementación de esta herramienta de comunicación en puzle
8-Pruebas y solución de posibles errores
9-Presentación de la aplicación final, sistema de colaboración y comunicación a la tutora
10-Comprobación junto a la tutora de que el sistema de comunicación se adapta a las necesidades.
11-Últimos cambios en caso de ser necesarios
12-Presentación del TFG

Ilustración 51 Planificación inicial 1

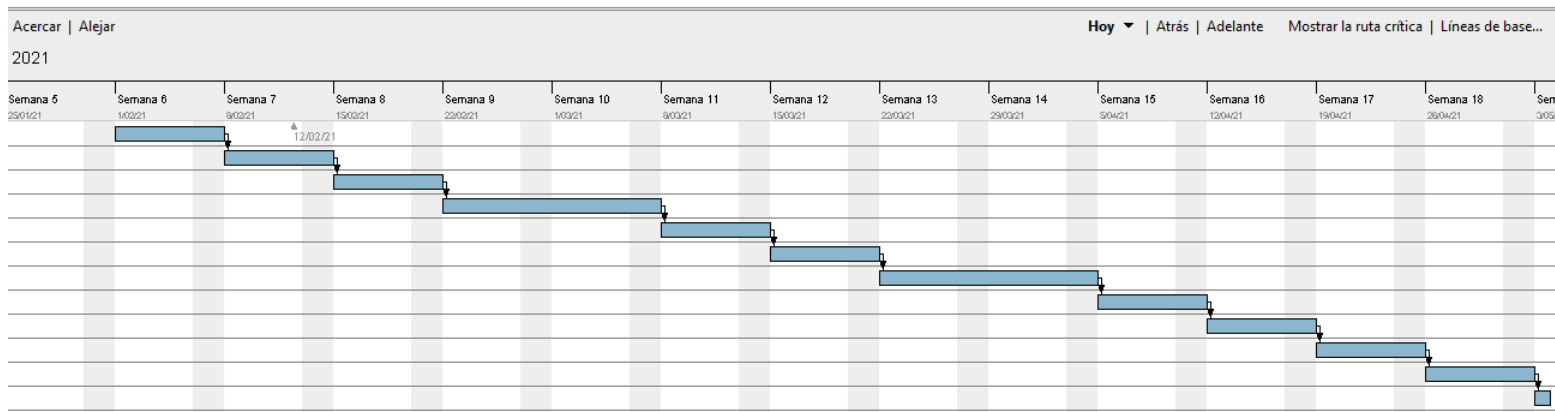


Ilustración 50 Planificación Inicial 2

A pesar de que en la planificación inicial se planteaba la entrega del trabajo en mayo, debido a una serie de contratiempos, tanto personales como académicos, me vi obligado a retrasar el desarrollo del proyecto lo que ha llevado a esta entrega tardía del proyecto en septiembre.

A pesar de los imprevistos durante el desarrollo del proyecto, la planificación inicial fue bastante acertada con respecto al tiempo necesario para desarrollar cada parte del trabajo.



## Desarrollo de aplicación para comunicación y colaboración entre alumnos

He de mencionar también que alguna de las partes ha requerido más tiempo del previsto debido a encontrar fallos más adelante del proceso de implementación que no fueron detectados en fases de pruebas.

## 9. Futuras mejoras

A pesar de haber mejorado en gran medida la aplicación, y haber cumplido los requisitos que se pedían siempre hay hueco para la mejora.

Alguna de las ideas que se me ocurren son:

- 1. Recuperación de contraseña:** Añadir un sistema que, mediante el correo electrónico de la posibilidad de recuperar la contraseña, esto liberaría de trabajo a la profesora ya que los alumnos tienden a olvidar las contraseñas.
- 2. Añadir un porcentaje de penalización:** Además del contador de penalización estaría bien la automatización del porcentaje de nota que se ha de sustraer a un alumno en función de cuantos puntos disponga en este contador. Se podría crear un parámetro modificable que automáticamente vaya restando la nota correspondiente.
- 3. Recordatorio de votar:** Actualmente los alumnos tienden a olvidar en qué fecha se ha de realizar la votación, este problema se podría solucionar implementando un sistema de recordatorios, bien sea por correo o al iniciar sesión en la aplicación.
- 4. Mejor sistema de filtrado de datos:** La tabla que recoge los alumnos en la vista de administrador, dispone de unas casillas de búsqueda, pero al realizar alguna acción estas se reinician al estado inicial, actualmente se ha modificado el orden en el que se muestran los alumnos estando los más actuales primero, pero estaría bien conseguir mantener estos parámetros de búsqueda tras realizar una acción.

## 10.Conclusiones

Tras terminar el proyecto, espero que estas mejoras ayuden a la profesora a seguir empleando la aplicación a la hora de dar clase, ya que estos métodos didácticos ayudan a los alumnos a motivarse a la hora de tomar responsabilidad de dirigir un grupo y trabajar en equipo.

He de decir que espero un futuro en el que cada vez más clases se impartan empleando herramientas como estas y mejor aún si las mismas han sido desarrolladas en parte por los propios alumnos.

El haber trabajado con lenguajes que no había usado en otras ocasiones me ha ayudado a aprender nuevas cosas, además de abrirme los ojos al mundo de las aplicaciones web y su gran potencial.

También he aprendido la importancia de tener una buena documentación, ya que esta ayuda mucho cuando otra persona tiene que trabajar sobre tu trabajo, por eso espero que esta memoria sirva de ayuda si en un futuro alguien ha de trabajar sobre mi trabajo.

Como resumen este proyecto me ha resultado gratificante y he disfrutado con su desarrollo, ya que los conocimientos aprendidos considero que me serán de utilidad en mi futuro tanto profesional como académico.

## 11. Presupuesto

Front-End	Tiempo
Cambios en las vistas	<b>7 días</b>
Creación del foro publico	<b>1 día</b>
Vista de crear publicación en foro	<b>1 día</b>
Creación foro privado	<b>1 día</b>
Creación de accesos a foros	<b>1 día</b>
Edición de Notas	<b>2 días</b>

<b>Total</b>	<b>13 días</b>
--------------	----------------

Back-End	Tiempo
Cambios en la estructura MVC y código original	<b>9 días</b>
Controlador de foros	<b>3 días</b>
Modelos de foros	<b>3 días</b>
Código Editar notas	<b>1 días</b>
Código penalización	<b>2 días</b>

<b>Total</b>	<b>18 días</b>
--------------	----------------

Base de datos	Tiempo
Modificaciones	<b>0,5 días</b>
Tablas foros	<b>1,5 días</b>

<b>Total</b>	<b>2 días</b>
--------------	---------------

PRESUPUESTO FINAL	
DESARROLLO	<b>33 DIAS</b>
PRUEBAS Y AJUSTES	<b>2 DIAS</b>
DOCUMENTACION	<b>6 DIAS</b>
<b>8 HORAS DIARIAS A 10€ LA HORA</b>	
<b>TOTAL</b>	<b>3280€</b>

## 12. Agradecimientos

Gracias a mi tutora por su paciencia, su disponibilidad y su ayuda a la hora de afrontar este TFG, a mi familia por todo el apoyo que me han brindado en estos años de carrera. A mis amigos de toda la vida porque siempre me han apoyado, a los amigos que he hecho durante estos años de universidad ya que sin ellos la carrera no habría sido lo mismo. A Angela, que ha estado ahí estos dos últimos años y me ha apoyado todas las veces que me he saturado.

Gracias a todos ellos y al resto de personas que me han acompañado durante estos años de carrera, ya que sin todos ellos no habría sido tan fácil acabar la carrera.

.

## 13. Bibliografía

- [1] <https://www.w3schools.com/css/>
- [2] <https://www.w3schools.com/php/>
- [3] <https://codepen.io/nagyalex/pen/ZEWLyaw>
- [4] [https://github.com/maranemil/howto/blob/master/mygist\\_files/Open%20a%20pdf%20in%20a%20new%20tab.php](https://github.com/maranemil/howto/blob/master/mygist_files/Open%20a%20pdf%20in%20a%20new%20tab.php)
- [5] <https://www.php.net/manual/es/>
- [6] <https://www.w3schools.com/html/>
- [7] Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 (Learning Php, Mysql, Javascript, Css & Html5) – Robin Nixon
- [8] El gran libro de HTML5, CSS3 y Javascript Primera edición, 2012 © 2012 Juan Diego Gauchat © MARCOMBO, S.A. 2012 Gran Via de les Corts Catalanes, 594 08007 Barcelona [www.marcombo.com](http://www.marcombo.com)

## 14. Referencias

Aronson, E. (1978). *The jigsaw classroom*. Sage.

Estriegana, R., Medina, J., Robina, R., & Barchino, R. (2021, June). Virtual Learning Environment to Encourage Students' Relationships and Cooperative Competence Acquisition. In Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (pp. 53-59).

## ANEXO I - Tutorial de Uso Profesor

### 1. Creación de un nuevo usuario

Para poder incluir un nuevo usuario hay que dirigirse a la página principal, y elegir la opción “Crear usuario” del menú de la barra lateral izquierda.

Una vez aquí la aplicación requerirá que se rellenen los campos indicados como se aprecia en la foto:



El formulario de creación de usuario se muestra en un recuadro con fondo oscuro y una barra lateral azul. Los campos de entrada son:

- Documento de Identidad
- Nombre
- Primer Apellido
- Segundo Apellido
- Grupo
- Curso

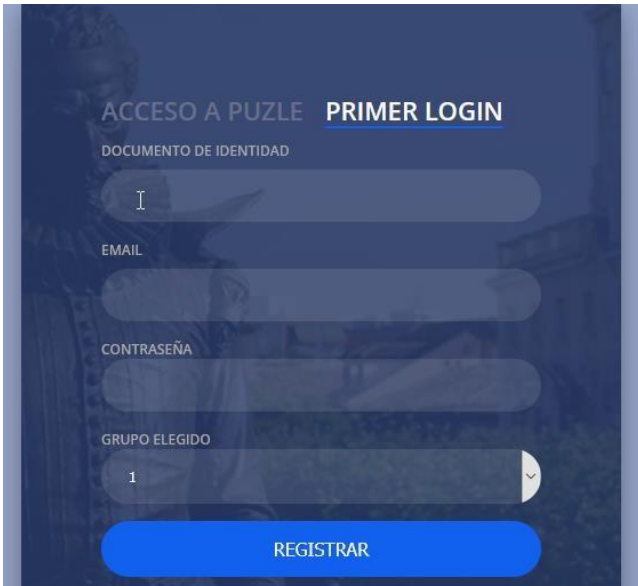
Debajo de los campos hay un botón rojo con el texto "Guardar".

*Ilustración 52 Crear usuario anexo*

Una vez se hayan rellenado todos los campos y pulsemos en registrar, se habrá creado una nueva entrada en la base de datos para ese nuevo usuario.

Lo siguiente para terminar el registro es que el alumno acceda a la opción de “Primer login” para terminar de cumplimentar el registro que se tiene de su usuario en la base de datos, el DNI que indique el usuario debe ser el mismo con el que se le ha registrado previamente por el profesor para que el registro sea completado con éxito.



The image shows a registration form titled 'ACCESO A PUZLE PRIMER LOGIN'. The form is set against a dark blue background with a faint image of a classical statue. It contains four input fields: 'DOCUMENTO DE IDENTIDAD' (with a cursor), 'EMAIL', 'CONTRASEÑA', and 'GRUPO ELEGIDO' (a dropdown menu showing '1'). A blue 'REGISTRAR' button is at the bottom.

*Ilustración 53 Primer login anexo*

Una vez terminado el registro, el usuario podrá iniciar sesión con el correo electrónico y la contraseña establecidas en el “Primer Login”.

## 2. Editar Usuario y cambio de contraseña

Si pulsamos en Editar usuario en la página principal sobre el usuario que queremos cambiar, se nos mostrará la página con el formulario que contiene todos sus datos:

The image shows a dark-themed user editing form. The fields and their values are as follows:

- Grupo:** G1Prueba
- Nombre:** P2N
- Apellidos:** P2A1 P2A2
- Rol:** Experto (dropdown menu)
- Documento de Identidad:** prueba2
- Contraseña:** [masked with dots]
- Grupo Laboratorio:** 1
- Curso:** 2021
- Email:** EP2@email.com

At the bottom of the form is a teal button labeled "Actualizar".

Ilustración 54 Editar usuario anexo

Desde aquí podemos indicar una nueva contraseña para el usuario en caso de la pérdida de esta.

### 3. Iniciar cambio de Rol

Para realizar el cambio de rol, es necesario indicar para que curso y grupo está destinado este cambio. Por lo tanto, en la sección de “funciones de administrador” para iniciar la rotación de experto nos encontraremos un formulario con dos desplegados en este formulario debemos seleccionar el curso y grupo a rotar.

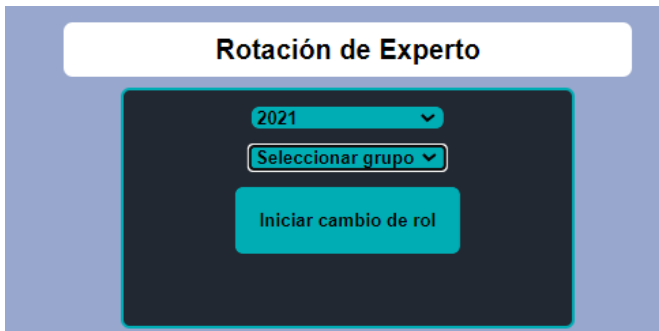


Ilustración 55 Iniciar cambio de Rol anexo

#### 4. Importar usuarios a la base de datos a través de un archivo CSV



Ilustración 56 Importar CSV anexo

Para poder importar con éxito los usuarios a la base de datos de la aplicación, es necesario que el CSV tenga la siguiente estructura:

1ª Columna: DNI

2ª Columna: Apellido-1 Apellido-2, Nombre

3ª Columna: Grupo-Lab

4ª Columna: Curso

La siguiente imagen muestra un ejemplo del formato que debe tener el CSV para todas sus entradas:

A	B	C	D
DNI	APELLIDO-1 APELLIDO-2, NOMBRE	GRUPO	CURSO

Ilustración 57 Formato CSV anexo

## 5. Acceso Foro Publico

En la sección “foro público” seleccionar mediante los desplegables el foro al que se pretende acceder, seleccionando curso y grupo.

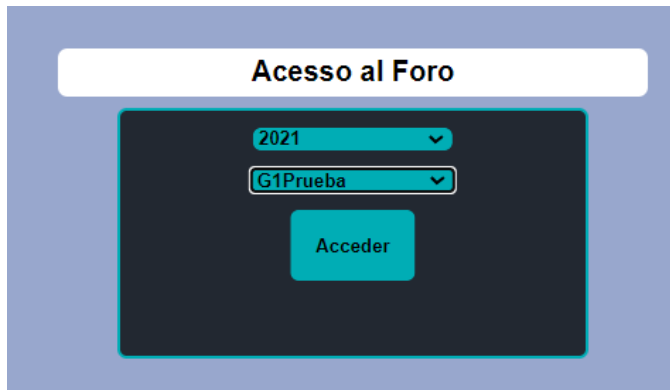


Ilustración 58 Acceso foro público anexo

Una vez dentro del foro se puede crear un nuevo hilo o responder otro. Los colores indican que rol posee la persona que ha escrito el mensaje.

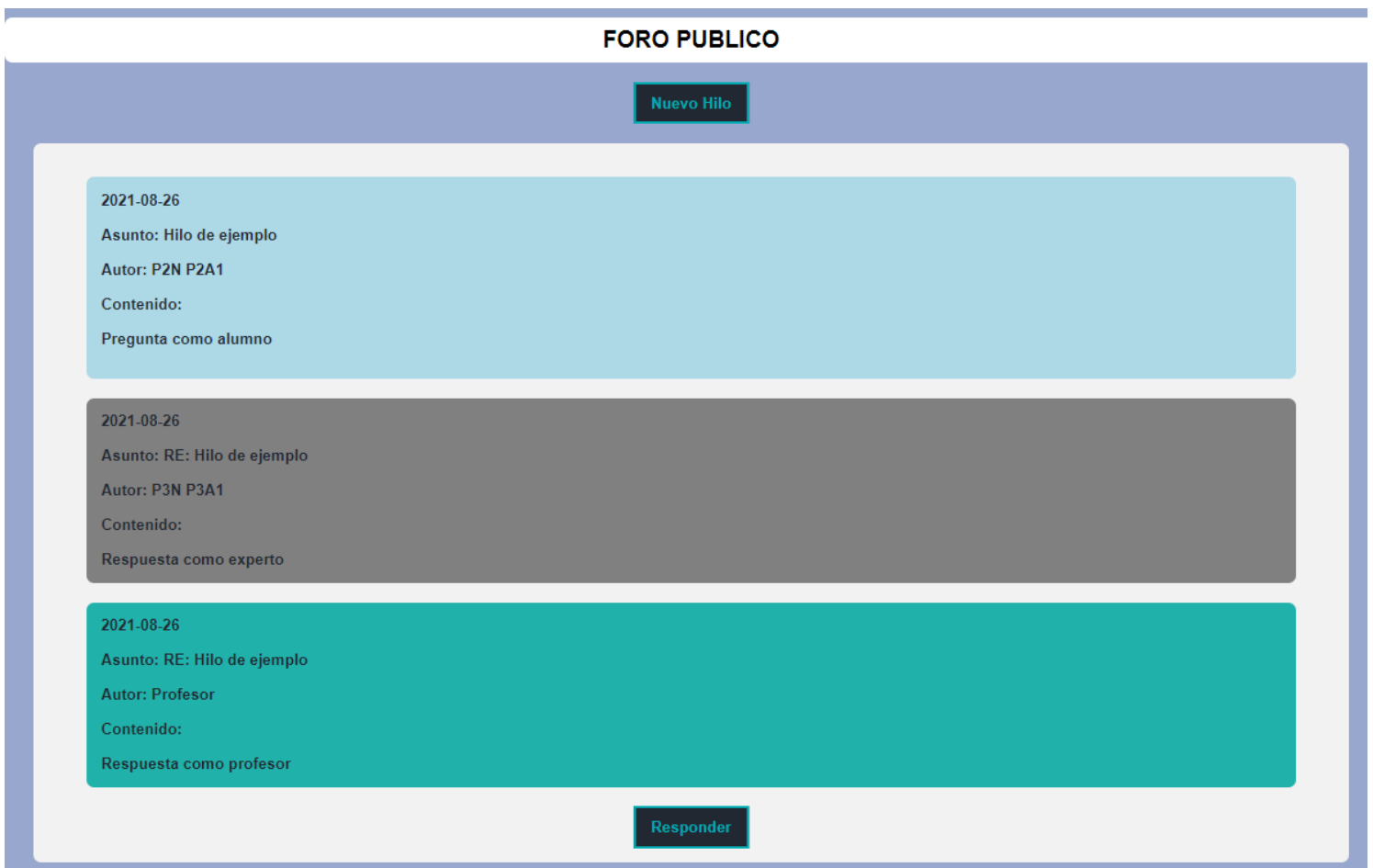


Ilustración 59 Foro Publico anexo

En las opciones de responder o nuevo hilo encontraremos una pantalla con un máximo de 200 caracteres para escribir nuestro mensaje.

Ilustración 61 Responder anexo

Ilustración 60 Nuevo hilo anexo

## 6. Editar Notas

Desde la ventana principal seleccionar la opción Ver/editar notas de un determinado alumno.

Actividad: prueba1		
Nombre Compañero	Nota	Rol alumno puntuado
P3N P3A1 P3A2	7	Alumno
<b>Actualizar</b>		

Actividad: prueba 2		
Nombre Compañero	Nota	Rol alumno puntuado
No data available in table		

Actividad: PRUEBA3		
Nombre Compañero	Nota	Rol alumno puntuado
No data available in table		

Nota media en cada rol		
Rol	Nota Media	
Alumno	7.00	
Experto	-	

Ilustración 62 Editar notas anexo

## Desarrollo de aplicación para comunicación y colaboración entre alumnos

En esta ventana aparecerán las notas que se le han puesto al alumno con la posibilidad de cambiarlas individualmente.

## ANEXO II - Tutorial de Uso Alumno

### 1. Experto

A diferencia de tus compañeros, para esta práctica tendrás que asegurarte de cómo hacer la actividad planteada en la sección de “Publicaciones” donde encontrarás más detalles sobre la ella.

Una vez haya terminado la actividad deberás dirigirte a la sección “Puntuar” de la página principal y elegir la actividad que habéis terminado para poder ponerle nota a tus compañeros de grupo en función de cómo lo hayan hecho durante la sesión.

Si quieres comprobar tus notas actuales, puedes dirigirte a la sección “Ver notas” en la página principal y allí dispondrás de una serie de tablas en las que encontraras un resumen con todas las notas que has recibido durante el curso en cada actividad para el rol que tuviste en ellas.

También dispondrás de tus notas medias para cada rol y una última tabla en la que podrás ver que notas medias poseen tus compañeros para cada rol.

### 2. Alumno

Como alumno tu tarea será realizar la practica en el laboratorio, atendiendo a las explicaciones de tu compañero con el rol de experto para que puedas realizar la actividad con éxito.

Una vez finalizada la sesión de laboratorio puedes dirigirte a la sección de “Puntuar” y allí elegir la actividad que habéis finalizado y puntuar la labor de tu compañero experto en función de cómo lo haya hecho durante la sesión.

Si quieres comprobar tus notas actuales, puedes dirigirte a la sección “Ver notas” en la página principal y allí dispondrás de una serie de tablas en las que encontraras un resumen con todas las notas que has recibido durante el curso en cada actividad para el rol que tuviste en ellas.

También dispondrás de tus notas medias para cada rol y una última tabla en la que podrás ver que notas medias poseen tus compañeros para cada rol.

#### 2.3 Acceso Foro Publico

En la sección foro público tendréis a vuestra disposición un foro para comunicaros con el resto de los alumnos y con la profesora.

## Desarrollo de aplicación para comunicación y colaboración entre alumnos

Una vez dentro del foro se puede crear un nuevo hilo o responder otro. Los colores indican que rol posee la persona que ha escrito el mensaje.



Ilustración 63 Foro público alumno anexo

En las opciones de responder o nuevo hilo encontraremos una pantalla con un máximo de 200 caracteres para escribir nuestro mensaje.





RE: Hilo de ejemplo

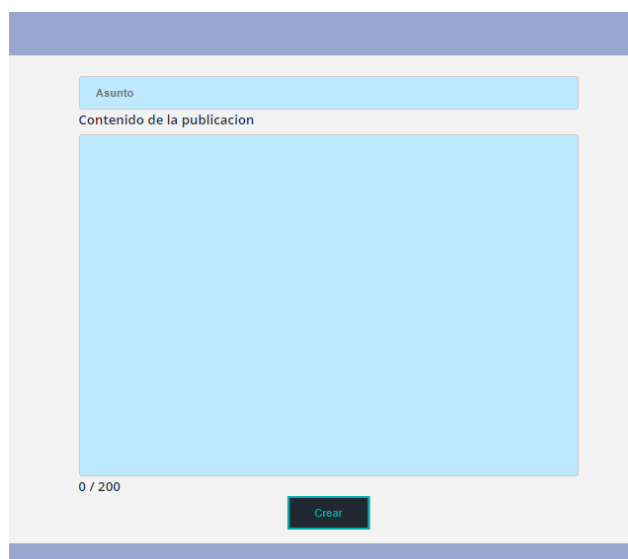
Contenido de la publicación

0 / 200

Crear

This screenshot shows a form for replying to a forum post. At the top, there is a light blue header with the text "RE: Hilo de ejemplo". Below this is a label "Contenido de la publicación" followed by a large, empty light blue text area. At the bottom left of the text area, it says "0 / 200". At the bottom center, there is a dark blue button with the word "Crear" in white.

Ilustración 65 Responder alumno anexo



Asunto

Contenido de la publicación

0 / 200

Crear

This screenshot shows a form for creating a new forum thread. At the top, there is a light blue header with the text "Asunto". Below this is a label "Contenido de la publicación" followed by a large, empty light blue text area. At the bottom left of the text area, it says "0 / 200". At the bottom center, there is a dark blue button with the word "Crear" in white.

Ilustración 64 Nuevo hilo alumno anexo

## 2.3 Foro privado

Del mismo modo que con el foro público, en la sección foro privado tendrás un subforo por cada una de las actividades, podréis entrar en cualquier momento a cualquiera de ellos, pero solo os dejara interactuar con aquellos cuya actividad aun este vigente.

En este foro podréis hablar con el resto de vuestro grupo.

## ANEXO III – Extractos de Código fuente

Debido a la gran cantidad de archivos creados se van a exponer los fragmentos correspondientes al controlador de foros, y el modelo de uno de los foros.

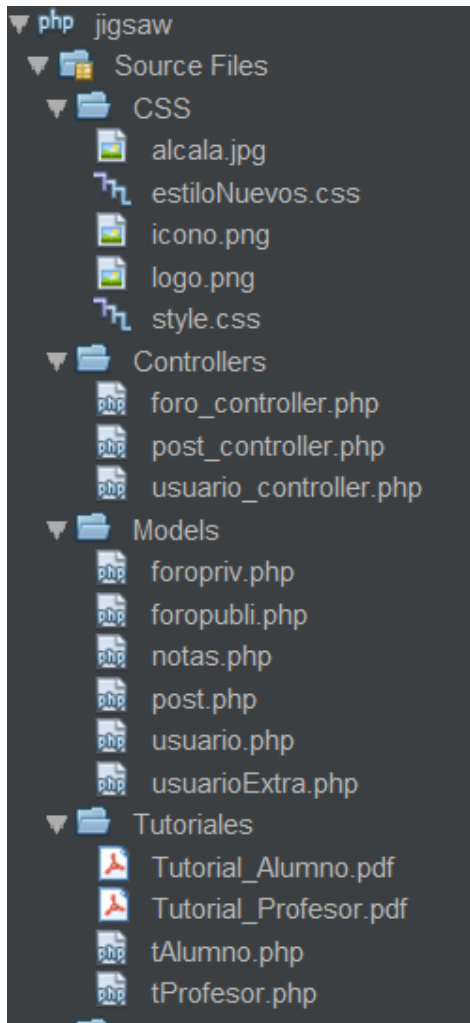


Ilustración 67 Estructura de archivos 1

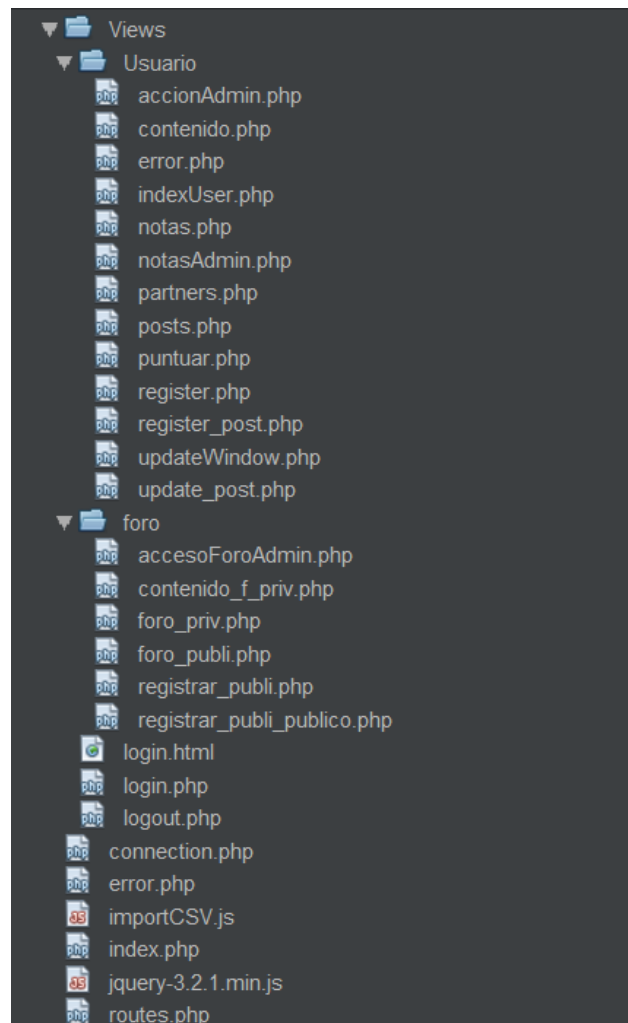


Ilustración 66 estructura de archivos 2

## Controlador de Foros

```

<?php
class ForoController {

    public function __construct() {

    }

    //Funcion encargada de cargar el selector de subforos del privado
    public function index_priv() {
        require_once('Models/post.php');
        //solo necesita cargar las actividades
        $posts = Post::temas();
        require_once('Views/foro/foro_priv.php');
    }

    //Carga el subforo adecuado
    public function show_foro_actv($parametros) {
        require_once('Models/post.php');
        require_once('Models/foropriv.php');
        //Extrae los mensajes correspondientes a dicho foro
        $publicaciones = ForoPriv::getPublicaciones($parametros["curso"], $parametros["grupolab"], $parametros["grupo"], $parametros["id_actv"]);
        $actividad = Post::getById($parametros['id_actv']);
        $titulo = $actividad->post_tema;
        $id_actv = $actividad->post_id;
        $activo = $parametros["activo"];
        require_once('Views/foro/contenido_f_priv.php');
    }

    //Mostramos el registro de un mensaje
    public function show_register_publi($parametros) {

        //Si existe hilo cargamos con hilo y asunto predefinidos

        if (isset($parametros['hilo'])) {
            $hilo = $parametros['hilo'];
            $asunto = "RE: " . $parametros['asunto'];
            $id_actv = $parametros['id_actv'];
        }
        else {
            //sino solo pasamos el id de la actividad
            $id_actv = $parametros['id_actv'];
        }
        //Indicamos si el foro esta activo para posteriores recargas
        $activo = $parametros['activo'];
        require_once('Views/foro/registrar_publi.php');
    }

    //Registramos mensaje en la base de datos
    public function registrar_publi($parametros) {
        require_once 'Models/foropriv.php';

        //Si el hilo no existe cargamos el siguiente hilo disponible mediante la funcion nextHilo
        if (!isset($parametros['hilo'])) {
            $hilo = ForoPriv::nextHilo($parametros['id_actv']);
        }
        else {
            $hilo = $parametros['hilo'];
        }

        //Generamos le objeto ForoPriv que es un mensaje
        $publi = new ForoPriv($parametros["curso"], $parametros["grupolab"], $parametros["grupo"], $parametros["id_user"], $parametros["id_actv"], $parametros["rol"], $parametros["asunto"], $parametros["contenido"], $hilo);

        //Guardamos la publicacion
        ForoPriv::save($publi);

        //Script para recargar el subforo del que veniamos
        echo '<form name="f" id="vueltaAtras" action="" method="post">';
        echo '<input type="hidden" name="controller" value="foro" >';
        echo '<input type="hidden" name="action" value="show_foro_actv" >';
        echo '<input type="hidden" name="args[curso]" value="" . $parametros["curso"] . "">';
        echo '<input type="hidden" name="args[grupolab]" value="" . $parametros["grupolab"] . "">';
        echo '<input type="hidden" name="args[grupo]" value="" . $parametros["grupo"] . "">';
        echo '<input type="hidden" name="args[id_actv]" value="" . $parametros["id_actv"] . "">';
        echo '<input type="hidden" name="args[activo]" value="" . $parametros["activo"] . "">';
        echo '</form>';
        echo '<script type="text/javascript">';
        echo 'document.getElementById("vueltaAtras").submit();';
        echo '</script>';
    }
}

```

```
//Mostramos vista de error
public function error() {
    require_once('Views/Usuario/error.php');
}

//Funcion destinada a cargar el foro publico
public function index_publi($parametros) {
    require_once('Models/post.php');
    require_once('Models/foropubli.php');
    $publicaciones = ForoPubli::getPublicaciones($parametros["curso"], $parametros["grupo"]);

    $grupo = $parametros['grupo'];
    $curso = $parametros['curso'];

    require_once('Views/foro/foro_publi.php');
}

//mismo que en caso privado carga el registrar mensajes
public function show_register_publi_publico($parametros) {

    if (isset($parametros['hilo'])) {
        $hilo = $parametros['hilo'];
        $asunto = "RE: " . $parametros['asunto'];
    }

    $grupo = $parametros['grupo'];
    $curso = $parametros['curso'];

    require_once('Views/foro/registrar_publi_publico.php');
}

//Registra el mensaje del foro publico en l abase de datos
public function registrar_publi_publico($parametros) {
    require_once 'Models/foropubli.php';

    if (isset($parametros['hilo'])) {
        $hilo = ForoPubli::nextHilo();
    } else {
        $hilo = $parametros['hilo'];
    }

    $publi = new ForoPubli($parametros["curso"], $parametros["grupo"], $parametros["id_User"], $parametros["rol"], $parametros["asunto"], $parametros["contenido"], $hilo, $parametros["fecha"]);
    ForoPubli::save($publi);

    echo ' <form name="f" id="vueltaAtras" action="" method="post">';
    echo ' <input type="hidden" name="controller" value="foro" > ';
    echo ' <input type="hidden" name="action" value="index_publi" > ';
    echo ' <input type="hidden" name="args[curso]" value="' . $parametros["curso"] . '" >';
    echo ' <input type="hidden" name="args[grupo]" value="' . $parametros["grupo"] . '" >';
    echo ' </form> ';
    echo ' <script type="text/javascript"> ';
    echo ' document.getElementById("vueltaAtras").submit(); ';
    echo ' </script>';
}

//Carga la ventana de desplegables del foro publico del admion para seccionar foro
public function index_publi_admin() {

    require_once('Models/usuario.php');
    $cursos = usuario::cursos();

    if ($cursos != -1) {
        foreach ($cursos as $curso) {
            $grupos[] = usuario::grupos($curso);
        }
    }

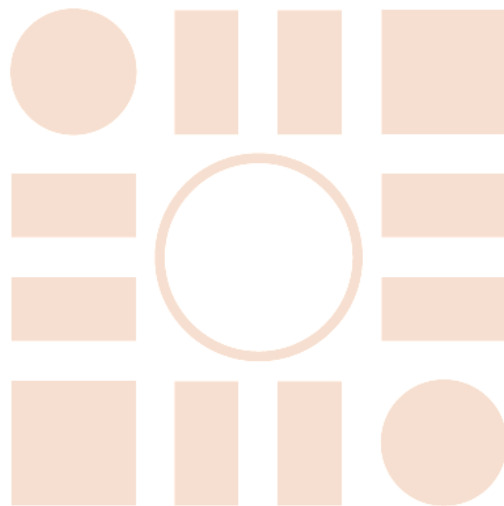
    require_once('Views/foro/accesoForoAdmin.php');
}
}
```

## Modelo Foro Privado

```
ForoPriv {  
  
    blic $curso;  
    blic $grupolab;  
    blic $grupo;  
    blic $id_User;  
    blic $id_actv;  
    blic $rol;  
    blic $asunto;  
    blic $contenido;  
    blic $hilo;  
  
    Constructor de la clase, es un mensaje del foro privado  
    nction __construct($curso, $grupolab, $grupo, $id_User, $id_actv, $rol, $asunto, $contenido, $hilo) {  
        $this->curso = $curso;  
        $this->grupolab = $grupolab;  
        $this->grupo = $grupo;  
        $this->id_User = $id_User;  
        $this->id_actv = $id_actv;  
        $this->rol = $rol;  
        $this->asunto = $asunto;  
        $this->contenido = $contenido;  
        $this->hilo = $hilo;  
  
        Guarda en al base de datos una entrada nueva  
        blic static function save($publicacion) {  
            $db = Db::getConnection();  
            $insert = $db->prepare('INSERT INTO foro_priv VALUES(:curso,:grupolab,:grupo,:id_User,:id_actv,:rol,:asunto,:contenido,:hilo,null)');  
            $insert->bindValue('curso', $publicacion->curso);  
            $insert->bindValue('grupolab', $publicacion->grupolab);  
            $insert->bindValue('grupo', $publicacion->grupo);  
            $insert->bindValue('id_User', $publicacion->id_User);  
            $insert->bindValue('id_actv', $publicacion->id_actv);  
            $insert->bindValue('rol', $publicacion->rol);  
            $insert->bindValue('asunto', $publicacion->asunto);  
            $insert->bindValue('contenido', $publicacion->contenido);  
            $insert->bindValue('hilo', $publicacion->hilo);  
            $insert->execute();  
  
            siguiente hilo disponible  
            blic static function nextHilo($id_actv) {  
                $db = Db::getConnection();  
                //Toma el numero de hilo mayor existente parta una actividad  
                $select = $db->prepare('SELECT DISTINCT hilo FROM foro_priv WHERE id_actv=:id_actv ORDER BY hilo DESC LIMIT 1');  
                $select->bindValue('id_actv', $id_actv);  
                $select->execute();  
                $postDb = $select->fetch();  
                //Y devuelve 0 si no existe y el siguiente si ya existen hilos  
                if ($postDb == null) {  
                    return 0;  
                } else {  
                    return $postDb["hilo"] + 1;  
                }  
            }  
        }  
    }  
}
```

```
//Obtiene las publicaciones de un foro
public static function getPublicaciones($curso, $grupolab, $grupo, $id_actv) {
    $db = Db::getConnection();
    //Tomamos todos los hilos para ese foro
    $select1 = $db->prepare('SELECT DISTINCT hilo FROM foro_priv WHERE curso=:curso AND grupolab=:grupolab AND grupo=:grupo AND id_actv=:id_actv');
    $select1->bindValue('curso', $curso);
    $select1->bindValue('grupolab', $grupolab);
    $select1->bindValue('grupo', $grupo);
    $select1->bindValue('id_actv', $id_actv);
    $select1->execute();
    $hilos = $select1->fetchAll();
    //Si existen estos hilos
    if (is_array($hilos)) {
        foreach ($hilos as $hilo) {
            //seleccionamos y colocamos en un array todos los mensajes de cada hilo
            $select = $db->prepare('SELECT * FROM foro_priv WHERE curso=:curso AND grupolab=:grupolab AND grupo=:grupo AND id_actv=:id_actv AND hilo=:hilo');
            $select->bindValue('curso', $curso);
            $select->bindValue('grupolab', $grupolab);
            $select->bindValue('grupo', $grupo);
            $select->bindValue('id_actv', $id_actv);
            $select->bindValue('hilo', $hilo["hilo"]);
            $select->execute();
            $foroDb = $select->fetchAll();
            $publicaciones = array();
            foreach ($foroDb as $publi) {
                $publicaciones[] = new ForoPriv($publi['curso'], $publi['grupolab'], $publi['grupo'], $publi['id_user'], $publi['id_actv'], $publi['rol'], $publi['asunto'], $publi['contenido'], $publi['hilo']);
            }
            $publicacionesh[] = $publicaciones;
        }
    }
    if (isset($publicacionesh)) {
        return $publicacionesh;
    } else {
        return null;
    }
}
```

Universidad de Alcalá  
Escuela Politécnica Superior



ESCUELA POLITECNICA  
SUPERIOR



Universidad  
de Alcalá