



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Diseño de un portal web para la gestión de las tareas diarias de un docente de instituto

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Torres Zahinos, Alejandro

Tutor: Sáez Barona, Sergio

2017/2018

Diseño de un portal web para la gestión de las tareas diarias de un docente de instituto

Resumen

Desarrollo de un portal web para uso docente en institutos para la gestión diaria de los grupos en los que imparte clase un profesor. Se trata de una herramienta digital similar al cuaderno del profesor físico que se utiliza habitualmente por los profesores de primaria y secundaria.

La aplicación web está pensada para uso durante la tarea docente de un profesor de manera que pueda gestionar sus grupos de clase y alumnos.

Mediante la aplicación web se podrá llevar el control de los grupos en los que imparte clase el profesor, así como de los alumnos que pertenecen a cada grupo y las actividades calificables que se deben realizar en cada uno de los grupos. Además, la aplicación permitirá gestionar las calificaciones obtenidas por los alumnos en las actividades a realizar y de la actitud de los alumnos del grupo como son los positivos y los negativos.

Para facilitar el registro inicial de los alumnos en el grupo la aplicación permitirá que el profesor active la opción de autorregistro para que los propios alumnos puedan introducir sus datos ellos mismos.

La aplicación web se desarrollará adaptada tanto para su uso desde computadoras de escritorio como para su uso desde teléfonos móviles y tablets de manera que el profesor pueda utilizarla tanto desde el ordenador de escritorio como desde su propio dispositivo móvil.

Palabras clave: web, gestión de tareas, docencia, secundaria, PHP, jQuery, Laravel.

Resum

Desenvolupament d'un portal web per a ús docent en instituts per a la gestió diària dels grups en els quals imparteix classe un professor. Es tracta d'una eina digital similar al quadern del professor físic que s'utilitza habitualment pels professors de primària i secundària.

L'aplicació web està pensada per a l'ús durant la tasca docent d'un professor de manera que pugua gestionar els seus grups de classe i alumnes.

Mitjançant l'aplicació web es podrà portar el control dels grups en què imparteix classe el professor, així com dels alumnes que pertanyen a cada grup i les activitats qualificables que s'han de realitzar en cadascun dels grups. A més, l'aplicació permetrà gestionar les qualificacions obtingudes pels alumnes en les activitats a realitzar i de l'actitud dels alumnes del grup com són els positius i els negatius.

Per facilitar el registre inicial dels alumnes en el grup l'aplicació permetrà que el professor active l'opció d'autoregistre perquè els mateixos alumnes puguin introduir les seues dades ells mateixos.

L'aplicació web es desenvoluparà adaptada tant per al seu ús des d'ordinadors d'escriptori com per al seu ús des de telèfons mòbils i tablets de manera que el professor pugua utilitzar-la tant des de l'ordinador d'escriptori com des del seu propi dispositiu mòbil.

Palabras clave: web, gestió de tasques, docència, secundària, PHP, jQuery, Laravel.

Abstract

Development of a web portal for teaching use in high schools for the daily management of the groups in which a teacher teaches. It is a digital tool similar to the physical teacher's notebook that is commonly used by primary and secondary school teachers.

The web application is intended for use during the teaching task of a teacher so that he can manage his class groups and his students.

Through the web application you can take control of the groups in which the teacher teaches, as well as the students belonging to each group and the qualifying activities that must be done out in each of the groups. In addition, the application will allow managing the marks obtained by the students in the activities to be done out and the attitude of the students of the group, such as positive and negative ones.

To facilitate the initial registration of students in the group, the application will allow the teacher to activate the option of self-registration so that the students themselves could enter their data themselves.

The web application will be developed adapted both for use from desktop computers and for use from mobile phones and tablets so that the teacher can use it both from the desktop computer and from their own mobile device.

Keywords: portal, task management, teaching, secondary education, PHP, jQuery, Laravel.



Tabla de contenidos

1	Introducción	9
1.1	Motivación	9
1.2	Objetivos	9
1.3	Impacto esperado.....	10
1.4	Estructura	10
2	Estado del arte.....	12
3	Especificación de requisitos.....	14
3.1	Introducción.....	14
3.1.1	Propósito	14
3.1.2	Ámbito	14
3.1.3	Definiciones, acrónimos y abreviaturas	14
3.2	Descripción general.....	15
3.2.1	Perspectiva de la aplicación.....	15
3.2.2	Funciones de la aplicación.....	15
3.2.3	Características del usuario.....	16
3.2.4	Restricciones.....	16
3.2.5	Requisitos futuros	16
3.3	Requisitos específicos.....	17
3.3.1	Requisitos funcionales	17
3.3.2	Requisitos de interfaces externas	27
3.3.3	Requisitos de eficiencia	28
3.3.4	Restricciones de diseño	28
3.3.5	Atributos	28
3.3.6	Otros requisitos	29
4	Análisis	30
4.1	Introducción.....	30
4.2	Casos de uso	30
4.2.1	Actores.....	30
4.2.2	Casos de uso	32
4.3	Diagrama de clases.....	57
5	Diseño.....	58
5.1	Introducción.....	58
5.2	Arquitectura física.....	58
5.3	Arquitectura lógica.....	59
5.3.1	Capa de persistencia	59

5.3.2	Capa de presentación	63
5.3.3	Capa lógica	66
6	Desarrollo	67
6.1	Tecnologías usadas	67
6.1.1	HTML 5	67
6.1.2	CSS 3.....	67
6.1.3	Bootstrap 4.....	67
6.1.4	JavaScript	67
6.1.5	jQuery 3.....	68
6.1.6	PHP.....	68
6.1.7	Laravel	68
6.1.8	MariaDB	68
6.2	Entorno de desarrollo	69
6.3	Herramientas utilizadas.....	69
6.4	Estructura de la aplicación	70
7	Pruebas	73
7.1	Introducción.....	73
7.2	Pruebas funcionales.....	73
7.3	Pruebas de visualización.....	73
7.4	Pruebas de tiempo de ejecución	80
7.5	Pruebas de uso	80
7.5.1	Añadir clase	81
7.5.2	Ver clase	82
7.5.3	Añadir alumno	84
7.5.4	Modificar los datos de un alumno	86
7.5.5	Consultar las actividades que tiene que realizar un alumno.....	87
7.5.6	Consultar las calificaciones obtenidas por todos los alumnos de una clase en las actividades a realizar.....	88
7.5.7	Añadir un positivo/negativo a un alumno	89
7.5.8	Modificar la foto de un alumno	91
7.5.9	Eliminar un alumno	93
7.5.10	Mostrar las fotos de un alumno registradas para una actividad.	95
8	Conclusiones	96
9	Glosario	98
10	Referencias.....	99
11	Bibliografía.....	100

Diseño de un portal web para la gestión de las tareas diarias de un docente de instituto

1 Introducción

1.1 Motivación

Generalmente las nuevas tecnologías surgen por la innovación en diferentes campos con la finalidad de facilitar determinadas tareas a las personas. Así, si la aceptación es buena, esas nuevas tecnologías consiguen hacerse un hueco entre las personas, de manera que finalmente su uso se hace indispensable y llega a ser impensable cómo era la vida antes de la aparición de dicha tecnología.

Se pueden encontrar infinidad de ejemplos para corroborar lo anteriormente mencionado. Como muestra, se puede nombrar internet que hoy en día se encuentra al alcance de cualquier persona que viva en un país desarrollado, de hecho, está tan al alcance que la mayoría de las personas llevan en su bolsillo un dispositivo que nos permite la conexión a internet, un Smartphone, otra de esas tecnologías mayoritariamente aceptadas.

Así, mediante la realización de este Trabajo Final de Grado se pretende aplicar y consolidar los conocimientos adquiridos durante la realización del grado mediante la creación de una aplicación web que haga uso de las tecnologías anteriormente mencionadas y con la finalidad de que dicha aplicación pueda llegar a ser beneficiosa para un sector como es el de la docencia.

Como docente de E.S.O, Bachillerato y Ciclos Formativos, con 8 años de experiencia en centros docentes de secundaria de la Comunidad Valenciana, siempre he echado en falta una aplicación multiplataforma y flexible que permita realizar el seguimiento diario del aula en las asignaturas que imparto clase a lo largo de los diferentes cursos académicos.

Aunque existen aplicaciones con la misma finalidad, o bien no son multiplataforma o bien son dependientes de disponer un dispositivo móvil como un Smartphone o un Tablet para poder ser usadas. Además de tener un enfoque hacia la educación infantil o primaria o en su defecto un enfoque demasiado generalista.

Por lo tanto, la aplicación que se va a desarrollar está orientada a poder llevar el control del aula de una manera ágil rápida y fiable. Comprendiendo desde la introducción de alumnos, almacenamiento de las tareas realizadas por parte de los alumnos, hasta las notas obtenidas por estos.

1.2 Objetivos

El objetivo principal de este Trabajo de Final de Grado consiste en el desarrollo de una aplicación web para llevar el control del aula con la finalidad de facilitar de manera considerable las tareas del profesor dentro del aula.

De esta manera la aplicación a desarrollar será accesible tanto desde el ordenador del profesor situado en el aula, como desde cualquier dispositivo móvil con acceso a internet.

El profesor podrá darse de alta en la aplicación y crear todos los grupos necesarios para cada curso académico. El profesor también podrá añadir tanto alumnos como

actividades a dichos grupos de manera que la tarea de evaluar a los alumnos sea más rápida, dando como resultado que el tiempo ahorrado en esos menesteres pueda ser dedicado a atender a los alumnos.

Por parte de los alumnos la única opción que podrán realizar es el autorregistro en el grupo al que pertenecen previa autorización del profesor.

1.3 Impacto esperado

Gracias a la aplicación web que se va a desarrollar se pretende que los profesores de secundaria puedan gestionar de manera más eficiente las tareas diarias que se realizan durante la docencia.

Esto se consigue gracias a que la aplicación será accesible desde cualquier dispositivo de sobremesa o móvil de manera que con mismo teléfono móvil del profesor puede realizar todas las funciones de la aplicación.

Además, como es una aplicación web, los datos son exactamente los mismos desde todos los dispositivos desde los que el usuario se conecte, así, no tendrá que preocuparse en tener que trasladar datos entre dispositivos.

1.4 Estructura

A continuación, se detalla cómo está estructurado el resto de este documento.

Estado del arte

En este punto se realiza un análisis de los productos existentes en el mercado para valorar la necesidad de desarrollar la aplicación web de este proyecto.

Especificación de requisitos

Partiendo de la base que ofrece el estándar IEEE 830:19980 [1] en este punto se realizará la especificación de los requisitos que debe cumplir la aplicación a desarrollar.

Primero se realizará una breve introducción, a continuación, se desarrollará la descripción general y por último se detallarán cada uno de los distintos tipos de requisitos.

Análisis

En este apartado, partiendo del anterior, se definirán de manera concreta los actores que participarán en la aplicación, así como todos los casos de uso para cada uno de esos actores. También mostrará mediante un diagrama de clases el análisis realizado de los elementos que participan en la aplicación.

Diseño

Aquí se detallarán las diferentes partes que hay que desarrollar para la aplicación, empezando por la estructura física de los equipos que intervienen, y terminando con la arquitectura lógica, mostrando cómo será la base de datos mediante el diagrama Entidad-Relación y su transformación a tablas relacionales mediante el modelo

relacional y por último detallando cómo será gráficamente la aplicación mediante diferentes bocetos.

Desarrollo

Este punto sirve como referencia para detallar cuales han sido las tecnologías utilizadas en el desarrollo de la aplicación, así como las herramientas software usadas.

También se detallan algunos aspectos importantes sobre la estructura de los ficheros y directorios de la aplicación.

Pruebas

Se detallan las diferentes pruebas realizadas al software desarrollado. Estas pruebas han sido de los siguientes tipos: de comprobación de que se realizan las funciones requeridas, de visualización en diferentes entornos, de rapidez en ejecución y de que la usabilidad y funcionalidad se corresponde a los casos de uso especificados.

Conclusiones

En este apartado se exponen las conclusiones a las que se ha llegado con la realización del proyecto, relacionándolas con los contenidos y conocimientos adquiridos tanto en la realización del grado como en la realización del propio proyecto.

Bibliografía

Referencias a los documentos que se han utilizado para consulta en la realización del proyecto.

2 Estado del arte

Los cuadernos del profesor [2] siempre han acompañado a los maestros y profesores a lo largo de toda su carrera docente. Es la herramienta analógica mediante la cual queda constancia de todo lo que ocurre en el día a día de las clases que imparte y la manera de tener al alcance de la mano toda la información de los alumnos.

Desde hace unos años han ido apareciendo alternativas digitales al cuaderno digital. La alternativa más básica es el uso de las hojas de cálculo [3] de cualquier suite ofimática. Mediante ellas un docente puede sustituir las hojas con cuadrículas que forman el cuaderno del profesor por las celdas de la hoja de cálculo.

Aunque esta es una alternativa bastante buena a la hora de almacenar calificaciones y generar medias, tiene algunos inconvenientes, entre ellos se encuentra el hecho que para saber utilizar de manera eficiente una hoja de cálculo se necesita un nivel bastante alto en el uso de ellas.

Otro de los inconvenientes es que las hojas de cálculo dependen de un ordenador de escritorio donde ejecutarla, así como dependen de que en dicho ordenador esté instalada la suite ofimática correcta. Esto deriva en una pésima transportabilidad de la información ya que la responsabilidad de transportar la hoja recae en que el docente porte físicamente la hoja de cálculo físicamente para poderla abrir desde los ordenadores en los que la quiera utilizar.

Este último problema se ha solucionado en los últimos tiempos con la implantación de suites ofimáticas en la nube por parte de Google¹ y Microsoft². Esto sumado a que hoy en día casi todo el mundo dispone de un Smartphone de gran pantalla [4], se puede acceder a las hojas de cálculo desde cualquier lugar. Aun así, al seguir siendo hojas de cálculo su uso sigue requiriendo un nivel alto de conocimientos.

Con la proliferación del uso de los Smartphone y de los Tablets, han surgido infinidad de aplicaciones en todos los campos y cómo no también en el campo de la educación. La gran mayoría de aplicaciones relativas a la educación están orientadas a contenidos y materiales de apoyo en clase como por ejemplo enciclopedias interactivas temáticas.

Entre toda esa amalgama de aplicaciones en 2011 surgió una aplicación con el objetivo de sustituir al cuaderno digital, iDoceo³. Al principio esta aplicación disponía de muy pocas funcionalidades, aunque suficientes como para ser utilizada como cuaderno del profesor. Poco a poco los desarrolladores de iDoceo han ido añadiendo funcionalidades convirtiéndola en una gran aplicación. Aunque sigue teniendo el mismo inconveniente que arrastra desde el primer día, solo está disponible para los Apple iPad.

Debido al gran éxito de iDoceo, durante estos años han surgido algunas aplicaciones similares, algunas son aplicaciones web como Learnboost.com⁴ o Engrade.com⁵, otras solo disponibles para dispositivos móviles de Apple como iTeacherBook.com⁶ o incluso

¹ <https://www.google.es/intl/es/sheets/about>

² <https://www.office.com>

³ <https://www.idoceo.es/index.php/es>

⁴ <https://www.learnboost.com>

⁵ <https://www.mheducation.com>

⁶ <http://iteacherbook.com>

multiplataforma como Additioapp.com ⁷ disponible tanto como web como para dispositivos Apple y Android.

Actualmente las dos aplicaciones que copan el mercado como cuaderno del profesor digital son iDoceo y AdditioApp. iDoceo sigue teniendo el inconveniente de ser exclusivo para Apple iPad. Y tanto iDoceo como AdditioApp tienen los siguientes inconvenientes, uno es que la curva de aprendizaje de uso no es algo trivial y el docente que desee usarlas deberá ocupar una buena parte de tiempo en aprender cómo funcionan y el otro inconveniente es que las dos aplicaciones al estar pensadas en la tarea docente sea cual sea el nivel educativo son demasiado generales y su configuración puede ser muy costosa.

Propuesta

Mediante la aplicación web que se plantea desarrollar en este TFG, se ofrece una alternativa digital al cuaderno del profesor a la vez que se pretende solucionar los inconvenientes de las aplicaciones disponibles en el mercado.

Para ello se va a desarrollar una aplicación fácil de usar y muy ágil en lo que lo importante es el uso de la aplicación y no el tiempo de configurar todo el curso para su posterior uso.

Una de las características más importantes de la aplicación a desarrollar es que al ser web se va a diseñar para que tenga un diseño fluido que permita que sea usada desde cualquier dispositivo sea cual sea el tamaño de su pantalla, adaptando dinámicamente los elementos de la aplicación a dicha resolución.

Esta característica permite que los datos estén centralizados en el servidor web de manera que iniciando sesión en cualquier dispositivo siempre se tendrán disponibles todos los datos almacenados en la aplicación.

⁷ <https://www.additioapp.com/es>



3 Especificación de requisitos

3.1 Introducción

3.1.1 Propósito

Mediante la especificación de requisitos (ERS) [1] se pretende tener un punto de partida en la documentación en el que queden definidas y detalladas cada una de las funcionalidades de la aplicación que se va a desarrollar. Estas funcionalidades deben de cumplir las expectativas del cliente, así como las de los posibles futuros usuarios de la aplicación.

3.1.2 Ámbito

La aplicación web que se desarrollará pretende servir como sustituto del cuaderno del profesor tradicional, de esta manera se puede determinar que la aplicación será un cuaderno digital docente.

Gracias al cuaderno digital docente, el profesor tendrá centralizados todos los datos de sus grupos y alumnos, a su vez que podrá mantener una organización lógica similar a la que se tiene con el cuaderno del profesor tradicional, pero aportando las ventajas de la era digital con la incorporación de imágenes relativas a los alumnos y sus tareas.

Además, el profesor podrá realizar las tareas típicas de su labor docente como son anotar las calificaciones obtenidas por los alumnos en las actividades realizadas y realizar anotaciones sobre la actitud de los alumnos en clase.

3.1.3 Definiciones, acrónimos y abreviaturas

En esta sección se pretende realizar una relación de los conceptos y términos que se utilizan en la especificación de requisitos.

ERS: Especificación de Requisitos Software.

Requisitos funcionales: definición de cada una de las funciones que debe realizar un software a desarrollar o uno de sus componentes.

Propósito: necesidad del usuario de realizar una acción en la aplicación.

Proceso: pasos que se realizan para alcanzar el propósito.

Entrada: datos o acciones necesarios para iniciar el proceso.

Salida: datos o resultados que se obtienen al finalizar el proceso.

Interfaces externas: como se realiza la comunicación de la aplicación con elementos o sistemas ajenos a ella.

URL: Uniform Resource Locator. Dirección de los recursos ubicados en internet, mediante la URL se consigue localizar un recurso en internet.

3.2 Descripción general

3.2.1 Perspectiva de la aplicación

Para poner solución a la problemática planteada, se decide realizar una aplicación web para el control del aula para profesores de educación secundaria que permita agilizar las tareas relacionadas con las actividades que se plantean en el aula y las calificaciones obtenidas en estas.

El sistema es totalmente independiente, de manera que no dependerá de ninguna otra aplicación.

La aplicación se desarrollará para que sea accesible desde cualquier sistema operativo disponible en el mercado actual, ya sea de escritorio o de móvil, y esto será posible ya que para poder utilizarla solo será necesario un navegador web.

3.2.2 Funciones de la aplicación

La aplicación dispondrá de las siguientes funcionalidades que se pueden separar según el tipo de usuario que acceda a la aplicación.

Profesor no registrado:

- Registro en el sistema.

Profesor registrado:

- Entrar en el sistema.
- Consultar los datos propios del usuario.
- Registro de grupos.
- Consulta de los grupos del profesor.
- Registro de alumnos.
- Consulta de los alumnos del profesor.
- Registro de actividades.
- Registro de las horas a las que se imparte clase a un grupo.
- Consulta de los datos introducidos de un grupo.
- Consulta de los datos introducidos de un alumno.
- Consulta de los datos introducidos de una actividad.
- Consulta de las calificaciones obtenidas por los alumnos de un grupo.
- Seleccionar la evaluación por defecto.
- Seleccionar el curso académico por defecto.
- Introducción de calificaciones y observaciones sobre esas calificaciones de los alumnos para las actividades realizadas.
- Introducción de imágenes/fotos de las actividades realizadas por los alumnos.
- Consulta de las fotos de una actividad de un alumno.
- Introducción de positivos y negativos para cada alumno.
- Consulta de los emails de los alumnos de un grupo
- Modificación y borrado de los datos introducidos tanto de los grupos, de los alumnos y todos sus datos relacionados, como de las actividades.
- Salir del sistema.

Alumno: la aplicación no está pensada para que la usen los alumnos, pero para facilitar al profesor la tarea inicial de registro de alumnos, el profesor podrá habilitar la siguiente funcionalidad para ellos.

- Autorregistro en el grupo asignado.

3.2.3 Características del usuario

El principal usuario de la aplicación serán profesores de educación secundaria que dispongan en el aula de un ordenador o de un dispositivo móvil como un Smartphone o un Tablet.

Puntualmente también harán uso los alumnos de dicho profesor cuando autorregistren sus datos en la aplicación.

Los dos tipos de usuario disponen de un nivel informático suficiente para el uso de la aplicación ya que la aplicación es una página web.

3.2.4 Restricciones

La aplicación está destinada para el uso tanto en dispositivos fijos como en dispositivos portátiles, para lo cual los únicos requisitos son que estos dispositivos tengan conexión a internet y dispongan de un navegador web.

La aplicación será accesible desde cualquier sistema operativo que disponga de un navegador web y tendrá que funcionar correctamente en los navegadores más importantes del mercado [5] como son Microsoft Edge, Mozilla Firefox, Google Chrome, Safari y Opera.

Los navegadores desde los que se acceda a la aplicación deberán tener el JavaScript activado debido a las funcionalidades que tendrá la aplicación que la dotan de su usabilidad y agilidad.

La aplicación se tendrá que visualizar correctamente independientemente del tamaño de la pantalla del dispositivo con el que se acceda a ella, para ello se usará un diseño adaptativo basado en la filosofía Mobile First [6].

Las contraseñas de los usuarios registrados se deben guardar de manera encriptada en la base de datos de la aplicación.

3.2.5 Requisitos futuros

Dentro de las posibles mejoras futuras de la aplicación se encuentran las siguientes funcionalidades:

- Dotar a las actividades de un porcentaje independiente dentro del tema al que pertenecen.
- Cálculo de las notas medias por evaluación y final de los alumnos.
- Consulta de las notas obtenidas por parte de los alumnos y padres de estos.
- Generación de un horario con las horas en las que el profesor imparte clase.

3.3 Requisitos específicos

3.3.1 Requisitos funcionales

Se definen las acciones que tiene que realizar el software para alcanzar las funcionalidades que se requieren de él. Para ello se reciben unos datos de entrada que se procesan y se genera una salida como respuesta.

RF-01 Registrarse en la aplicación	
Propósito	El usuario puede darse de alta como usuario registrado en la aplicación.
Entrada	Nombre de usuario, email, y contraseña.
Proceso	El usuario rellena el formulario de registro. Se validan los datos introducidos. Se almacenan los datos introducidos. Se inicia la sesión en la aplicación.
Salida	En caso de que el registro sea correcto se mostrará la pantalla de lista de grupos del profesor. Si se produce un error se le mostrará al usuario.

RF-02 Entrar en el sistema	
Propósito	El usuario puede iniciar sesión en la aplicación.
Entrada	Correo electrónico y contraseña.
Proceso	El usuario rellena el formulario de inicio de sesión. Se validan los datos introducidos. Se inicia la sesión en la aplicación.
Salida	En caso de inicio correcto se mostrará la pantalla de lista de grupos del profesor. En caso contrario se le indicará con un mensaje al usuario.

RF-03 Consulta de datos del usuario	
Propósito	El usuario puede consultar sus datos personales y de configuración.
Entrada	Identificador de usuario.

Proceso	El usuario hace clic sobre su nombre de usuario en la barra de menú.
Salida	Se muestra la pantalla con los datos del usuario y con todos los grupos que tiene creados para todos los cursos académicos registrados.

RF-04 Registro de grupos	
Propósito	El usuario puede crear un grupo al que imparte clase.
Entrada	Datos del grupo a crear.
Proceso	<p>El usuario hace clic en el botón “Añadir clase” disponible en la página de lista de clases.</p> <p>Se muestra el formulario de registro de grupo.</p> <p>El usuario rellena los datos del grupo.</p> <p>Se validan los datos introducidos.</p> <p>Se almacenan los datos introducidos.</p>
Salida	<p>Si el registro es correcto en la misma página se añadirá el grupo a la lista de grupos.</p> <p>En caso de error en algún campo del formulario o en el registro se mostrará dicha información al usuario.</p>

RF-05 Consulta de los grupos del profesor	
Propósito	El usuario puede consultar todos los grupos que tiene creados.
Entrada	Palabra “clase”.
Proceso	El usuario hace clic sobre el enlace “Clases” en la barra de menú.
Salida	Se muestra la pantalla con todos los grupos del profesor para el curso académico almacenado por defecto.

RF-06 Registro de alumnos	
Propósito	El usuario puede crear un grupo al que imparte clase.
Entrada	Identificador de la clase a la que pertenece el alumno
Proceso	El usuario hace clic en el botón “Añadir alumno” disponible en la página de lista de alumnos y en la página de información del grupo.

	<p>Se muestra el formulario de registro de alumno.</p> <p>El usuario rellena los datos del alumno.</p> <p>Se validan los datos introducidos.</p> <p>Se almacenan los datos introducidos.</p>
Salida	<p>Si el registro es correcto en la misma página se añadirá el alumno a la lista de alumnos.</p> <p>En caso de error en algún campo del formulario o en el registro se mostrará dicha información al usuario.</p>

RF-07 Consulta de los alumnos del profesor	
Propósito	El usuario puede consultar todos los grupos que tiene creados.
Entrada	Palabra "alumno".
Proceso	El usuario hace clic sobre el enlace "Clases" en la barra de menú.
Salida	Se muestra la pantalla con todos los grupos del profesor para el curso académico almacenado por defecto.

RF-08 Registro de actividades	
Propósito	El usuario puede crear actividades asignadas a un grupo.
Entrada	Identificador del grupo al que se desea asignar la actividad.
Proceso	<p>El usuario hace clic en el botón "Añadir actividad" disponible en el menú de opciones de la clase.</p> <p>Se muestra el formulario de registro de actividad.</p> <p>El usuario rellena los datos de la actividad.</p> <p>Se validan los datos introducidos.</p> <p>Se almacenan los datos introducidos.</p>
Salida	<p>Si el registro es correcto en la misma página se añadirá la actividad a la lista de actividades.</p> <p>En caso de error en algún campo del formulario o en el registro se mostrará dicha información al usuario.</p>

RF-09 Registro de las horas a las que se imparte clase a un grupo	
Propósito	El usuario puede registrar las horas a las que imparte clase a un grupo.
Entrada	Identificador del grupo al que se desea asignar la hora.
Proceso	<p>El usuario hace clic en el botón “Añadir hora” disponible en el menú de opciones de la clase.</p> <p>Se muestra el formulario de registro de la hora.</p> <p>El usuario rellena los datos de la hora.</p> <p>Se validan los datos introducidos.</p> <p>Se almacenan los datos introducidos.</p>
Salida	<p>Si el registro es correcto en la misma página se añadirá la hora a la lista de horas.</p> <p>En caso de error en algún campo del formulario o en el registro se mostrará dicha información al usuario.</p>

RF-10 Consulta de los datos de un grupo	
Propósito	El usuario puede consultar todos los datos relacionados con un grupo: datos propios del grupo, alumnos, actividades y horas.
Entrada	Identificador del grupo
Proceso	El usuario hace clic en cualquier enlace de entrada a un grupo, disponible desde la lista de grupos y desde cualquier elemento que pertenezca al grupo.
Salida	Se muestran los datos del grupo, así como la lista de alumnos del grupo, la lista de horas del grupo y la lista de actividades del grupo para la evaluación seleccionada por defecto.

RF-11 Consulta de los datos de un alumno	
Propósito	El usuario puede consultar todos los datos relacionados con un alumno: datos propios del alumno, actividades realizadas por el alumno y su calificación.
Entrada	Identificador del alumno.
Proceso	El usuario hace clic en cualquier enlace de entrada a un alumnos, disponible desde la lista de alumnos y desde la información de los datos de un grupo.

Salida	Se muestran los datos del alumno, así como las actividades que tiene que realizar y la calificación obtenida.
---------------	---

RF-12 Consulta de los datos de una actividad	
Propósito	El usuario puede consultar todos los datos relacionados con una actividad: datos propios de la actividad y los alumnos que tienen que realizarla.
Entrada	Identificador de la actividad
Proceso	El usuario hace clic en cualquier enlace de entrada a una actividad, disponible desde la lista de actividades de un grupo y desde la lista de actividades a realizar por un alumno.
Salida	Se muestran los datos de la actividad, así como la lista de alumnos del grupo que tienen que realizarla.

RF-13 Consulta de las calificaciones obtenidas por los alumnos de un grupo	
Propósito	El usuario puede consultar todas las calificaciones de los alumnos de un grupo.
Entrada	Identificador del grupo.
Proceso	El usuario hace clic sobre el enlace "Notas" en la barra opciones de un grupo.
Salida	Se muestra la pantalla con todas las actividades asignadas a un grupo con los alumnos del grupo y las calificaciones obtenidas por estos.

RF-14 Seleccionar la evaluación por defecto	
Propósito	El usuario puede cambiar la evaluación actual para guardarla como evaluación por defecto.
Entrada	Identificador del grupo y la evaluación.
Proceso	El usuario hace clic en el campo seleccionable de evaluaciones desde la barra de menú. El usuario selecciona una evaluación. Se almacena la nueva evaluación por defecto en el registro del usuario en la base de datos.

Salida	Se muestra la pantalla de lista de grupos del profesor pero mostrando la nueva evaluación por defecto en el campo seleccionable de la barra de menú.
---------------	--

RF-15 Seleccionar la curso académico por defecto	
Propósito	El usuario puede cambiar el curso académico actual para guardarlo como curso académico por defecto.
Entrada	Identificador del grupo y el curso académico.
Proceso	<p>El usuario hace clic en el campo seleccionable desde la barra de menú.</p> <p>El usuario selecciona un curso académico de entre los existentes.</p> <p>Se almacena el nuevo curso académico por defecto en el registro del usuario en la base de datos.</p>
Salida	Se muestra la pantalla de lista de grupos del profesor con los grupos del nuevo curso académico seleccionado y además se muestra el nuevo curso académico en el campo seleccionable de la barra de menú.

RF-16 Introducción de calificación de un alumno en una actividad	
Propósito	El usuario puede añadir la calificación a un alumno en una actividad realizada.
Entrada	Identificador del alumno, identificador de la actividad y la calificación.
Proceso	<p>El usuario hace clic sobre el espacio reservado para la calificación en cualquiera de las páginas en las que se muestran las calificaciones.</p> <p>Se muestra un campo de formulario en el cual se puede introducir la calificación.</p> <p>El usuario introduce la calificación.</p> <p>Se almacena la calificación en la base de datos.</p>
Salida	<p>Se actualiza en la página el espacio reservado para la calificación con la calificación obtenida.</p> <p>En caso de error se mostrará un mensaje al usuario.</p>

RF-17 Introducción de una observación sobre una calificación de un alumno en una actividad	
Propósito	El usuario puede añadir una observación a una calificación de un alumno.
Entrada	Identificador del alumno, identificador de la actividad y la observación.
Proceso	<p>El usuario hace clic sobre el espacio reservado para la observación en las páginas en las que se muestran las calificaciones.</p> <p>Se muestra un campo de formulario en el cual se puede introducir la observación.</p> <p>El usuario introduce la observación.</p> <p>Se almacena la observación en la base de datos.</p>
Salida	<p>Se actualiza en la página el espacio reservado para la observación con la observación introducida.</p> <p>En caso de error se mostrará un mensaje al usuario.</p>

RF-18 Introducción de una foto sobre una actividad realizada por un alumno	
Propósito	El usuario puede añadir una foto de una actividad realizada por un alumno.
Entrada	Identificador del alumno, identificador de la actividad y la foto.
Proceso	<p>El usuario hace clic sobre el icono de la cámara de fotos que se encuentra en las listas de las actividades y las calificaciones.</p> <p>Se muestra un formulario en el que se permite seleccionar una foto o en caso de acceder mediante un dispositivo móvil activará la cámara para poder capturar una fotografía.</p> <p>El usuario selecciona un archivo o captura una fotografía.</p> <p>Se validan los datos introducidos.</p> <p>Se almacena la foto en el servidor y se crea su registro correspondiente en la base de datos.</p>
Salida	<p>Se actualiza en la página mostrando un icono que permitirá consultar las fotos del alumno para esa actividad.</p> <p>En caso de error se mostrará al usuario un mensaje.</p>

RF-19 Consulta de las fotos de una actividad de un alumno	
Propósito	El usuario puede ver las fotos almacenadas de una actividad realizada por un alumno.
Entrada	Identificador del alumno, identificador de la actividad.
Proceso	El usuario hace clic sobre el icono de una imagen que se encuentra en las listas de las actividades y las calificaciones.
Salida	Se muestra una ventana modal con la todas las imágenes del alumno para esa actividad.

RF-20 Introducción de un positivo a un alumno	
Propósito	El usuario puede poner un positivo a un alumno por una actitud o respuesta positiva.
Entrada	Identificador del alumno.
Proceso	<p>El usuario hace clic sobre el icono de “Positivo” que se encuentra en la página de información de un alumno y en la ficha del alumno en la página de información de un grupo.</p> <p>Se muestra un formulario con los datos a introducir.</p> <p>Se validan los datos introducidos.</p> <p>Se almacenan los datos en la base de datos.</p>
Salida	<p>Si se introduce el positivo correctamente y se está en la página del alumno, se añadirá el positivo a la lista de positivos, si se está en la página de la clase no se mostrará ninguna información.</p> <p>En caso de error se mostrará un mensaje informando al usuario.</p>

RF-21 Introducción de un negativo a un alumno	
Propósito	El usuario puede poner un negativo a un alumno por una actitud o respuesta positiva.
Entrada	Identificador del alumno.
Proceso	<p>El usuario hace clic sobre el icono de “Negativo” que se encuentra en la página de información de un alumno y en la ficha del alumno en la página de información de un grupo.</p> <p>Se muestra un formulario con los datos a introducir.</p> <p>Se validan los datos introducidos.</p> <p>Se almacenan los datos en la base de datos.</p>

Salida	<p>Si se introduce el negativo correctamente y se está en la página del alumno, se añadirá el positivo a la lista de negativos, si se está en la página de la clase no se mostrará ninguna información.</p> <p>En caso de error se mostrará un mensaje informando al usuario.</p>
---------------	---

RF-22 Consulta de los emails de los alumnos de un grupo	
Propósito	El usuario puede necesitar toda la lista de emails de los alumnos de un grupo para hacerles un envío mediante correo electrónico.
Entrada	Identificador del grupo.
Proceso	El usuario hace clic sobre el enlace "Emails" que se encuentra en la barra de opciones del grupo.
Salida	Se muestra la lista de los emails de los alumnos de un grupo junto con el botón "Copiar al portapapeles".

RF-23 Modificación de cualquiera de los datos introducidos en la aplicación	
Propósito	El usuario puede modificar cualquier dato de los grupos, actividades, alumnos y calificaciones.
Entrada	Identificador del elemento propietario del dato a modificar.
Proceso	<p>El usuario hace clic sobre el dato que desea modificar.</p> <p>Aparece un campo de formulario relleno con el dato actual y que permite modificar el dato.</p> <p>El usuario hace clic fuera del campo o pulsa la tecla "Enter".</p> <p>Se valida el dato introducido.</p> <p>Se almacena el dato introducido en la base de datos.</p>
Salida	<p>Si se almacena correctamente el dato modificado se actualiza el dato en la misma pantalla.</p> <p>En caso de error se mostrará un mensaje al usuario</p>

RF-24 Borrado de los datos introducidos en la aplicación	
Propósito	El usuario puede borrar cualquier dato introducido en la aplicación.
Entrada	Identificador del elemento a borrar.

Proceso	<p>El usuario hace clic en el icono “Papelera” situado en las listas de grupos, las listas de alumnos, las listas de actividades, las listas de horas y las listas de calificaciones.</p> <p>Se muestra una ventana modal con los datos del elemento a eliminar y se pide confirmación.</p> <p>El usuario confirma la acción.</p> <p>Se eliminan los datos del sistema.</p>
Salida	<p>Si se elimina el elemento correctamente desaparecerá de la pantalla.</p> <p>En caso de error se mostrará un mensaje al usuario.</p>

RF-25 Salir del sistema	
Propósito	El usuario puede dejar el sistema.
Entrada	
Proceso	El usuario hace clic en “Salir” que se encuentra en la barra de menú.
Salida	Se muestra la pantalla de bienvenida de la aplicación.

RF-26 Activación de autorregistro en un grupo	
Propósito	El usuario puede activar el autorregistro para los alumnos de un grupo.
Entrada	Identificador del grupo y palabra clave.
Proceso	<p>El usuario hace clic sobre la opción “Autorregistro alumnos” que se encuentra en la barra de opciones del grupo.</p> <p>Se muestra un campo de formulario para introducir la palabra clave y un botón para activar el autorregistro.</p> <p>El usuario introduce la palabra clave.</p> <p>El usuario hace clic en el botón de activación</p>
Salida	Se muestra un enlace con la URL para facilitársela a los alumnos. En caso de error se mostrará un mensaje.

RF-27 Autorregistro en el grupo asignado	
Propósito	Un alumno puede introducir sus datos en el grupos asignado previa autorización del profesor.

Entrada	Identificador del grupo y palabra clave
Proceso	<p>El alumno accede a la URL que le facilita el profesor.</p> <p>Se muestra el formulario de registro de alumno.</p> <p>El alumno rellena el formulario con sus datos.</p> <p>Se validan los datos introducidos.</p> <p>Se almacenan los datos introducidos.</p>
Salida	<p>Si el registro es correcto en la misma página se mostrarán los datos y un mensaje de confirmación.</p> <p>En caso de error en algún campo del formulario o en el registro se mostrará dicha información en la misma pantalla.</p>

3.3.2 Requisitos de interfaces externas

Interfaces externas

La aplicación se visualizará en la pantalla de un ordenador o de un dispositivo móvil, en ella el usuario verá toda la información y podrá acceder a todos los servicios de la aplicación. Mediante el ratón y el teclado en el ordenador o mediante pulsaciones con los dedos en los dispositivos móviles el usuario podrá interactuar con el sistema e introducir todos los datos necesarios.

Interfaces Hardware

La aplicación se podrá utilizar en cualquier dispositivo que disponga de navegador web. No se necesitará ningún hardware especial y el dispositivo no requerirá de requisitos mínimos a nivel de hardware para poder disfrutar de los servicios de la aplicación.

Interfaces software

La aplicación podrá ser accesible desde cualquier sistema operativo del mercado siempre y cuando este disponga de un navegador web. Hoy en día cualquier sistema operativo para ordenador o dispositivo móvil dispone de un navegador web instalado o la posibilidad de instalar cualquier otro desde internet.

Interfaces de comunicación

La aplicación, por su característica de ser una aplicación web, se requiere que el dispositivo disponga de conexión a internet para poder hacer uso de ella. Esta conexión puede ser a través de conexión con cable o mediante conexión inalámbrica mediante WiFi o conexiones telefónicas inalámbricas (datos, 3G, 4G).

3.3.3 Requisitos de eficiencia

La aplicación, al ser una aplicación web, está desarrollada para cualquier navegador web capaz de interpretar ficheros con código HTML y capaz de interpretar el lenguaje de programación JavaScript.

La aplicación se instalará en un servidor web accesible desde internet mediante una dirección de dominio (URL). Dicho servidor tendrá instaladas las herramientas necesarias para poder interpretar el lenguaje de programación PHP y un sistema gestor de bases de datos compatible con MariaDB⁸.

3.3.4 Restricciones de diseño

Estándares

La aplicación deberá cumplir las reglas que estipulan los estándares HTML para que los navegadores sean capaces de interpretar correctamente dicho código y mostrar de manera similar y correcta todos los elementos de los que dispone la aplicación.

Limitaciones hardware

Las únicas limitaciones hardware de la aplicación son las limitaciones hardware que tenga el dispositivo desde el que se accede a la aplicación. En cuanto a requisitos mínimos, la aplicación no tiene ninguna restricción mínima.

3.3.5 Atributos

Mantenimiento

Se utilizará una arquitectura que ayude a que el mantenimiento de la aplicación se pueda realizar de manera fácil y rápida. Además, esta arquitectura ayudará a que se cumplan los principios de diseño.

Portabilidad

La aplicación será portable en cuanto se puede acceder a ella desde diferentes sistemas operativos y diferentes navegadores web.

Además, la aplicación será fácilmente portable de un servidor web a otro siempre y cuando el nuevo servidor disponga de un intérprete de PHP y de un sistema gestor de bases de datos compatible con MariaDB.

⁸ <https://mariadb.org/>

Seguridad

Debido a que la aplicación en última instancia depende del servidor web donde se aplica, los fallos en los dispositivos que acceden a ella no afectan en el funcionamiento.

La aplicación tendrá un sistema de autenticación, mediante cuenta de correo electrónico y contraseña, que únicamente permitirá su uso a los usuarios con las credenciales correctas. Además, se usará un mecanismo para que cada usuario solo pueda acceder a los elementos de los que es propietario.

3.3.6 Otros requisitos

Bases de datos

La aplicación hará uso de una base de datos para poder almacenar toda la información que requiera la aplicación.

En la base de datos se almacenará la información de la cuenta de los usuarios (profesores), así como todos los datos que los usuarios introduzcan.

La base de datos que se utilizará será compatible con el sistema gestor de bases de datos MariaDB. Dicho sistema y la base de datos se localizará en el mismo servidor web que aloje la aplicación.

Almacenamiento

Debido a que con la aplicación se pueden almacenarán las fotos de los alumnos y las fotos de las actividades realizadas, el servidor web donde se aloje la aplicación tener que tener capacidad de disco duro suficiente para poder almacenar gran cantidad de este tipo de ficheros.

4 Análisis

4.1 Introducción

Una vez conocidos los requisitos del software que ha de cumplir la aplicación, el siguiente paso es plasmar dichos requisitos mediante un lenguaje de modelado estandarizado que permita crear diagramas fácilmente reconocibles por los diseñadores.

En este proyecto se ha usado el lenguaje estandarizado de modelado UML que es uno de los más conocidos y usados.

4.2 Casos de uso

4.2.1 Actores

Los actores describen las características de los diferentes tipos de usuarios que accederán a la aplicación.

En esta aplicación participan 3 actores, los profesores no registrados, los profesores registrados y los alumnos.

La mayor parte de la funcionalidad de la aplicación requiere estar registrado por lo que el actor principal de la aplicación es el profesor registrado.

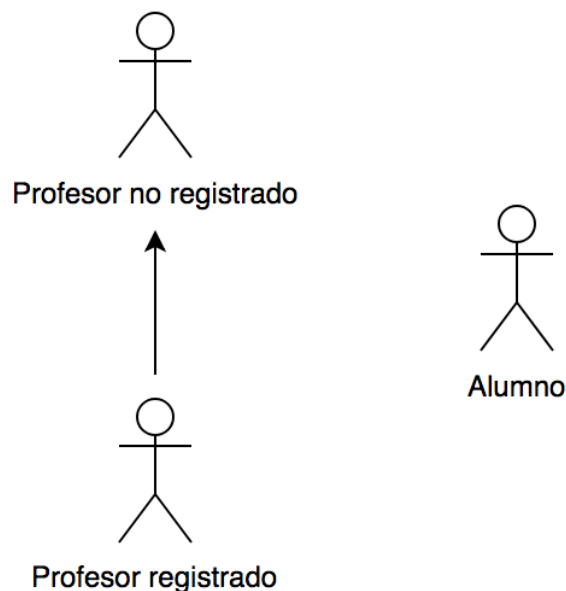


Figura 3.1 – Actores que intervienen en la aplicación

En las siguientes tablas se detallan las características de estos actores.

Profesor no registrado	
Rol de usuario	Invitado
Experiencia tecnológica	Básico, debe tener experiencia en navegar por internet.
Otras características	<ul style="list-style-type: none"> - Debe ser un profesor que imparta clase en un instituto de secundaria. - Debe disponer de una dirección de correo electrónico.

Profesor registrado	
Rol de usuario	Profesor
Experiencia tecnológica	Básico, debe tener experiencia en navegar por internet.
Otras características	<ul style="list-style-type: none"> - Debe estar dado de alta en la aplicación. - Debe disponer de un ordenador o dispositivo móvil con navegador y acceso a internet.

Alumno	
Rol de usuario	alumno
Experiencia tecnológica	Básico, debe tener experiencia en navegar por internet.
Otras características	<ul style="list-style-type: none"> - Debe disponer de un ordenador o dispositivo móvil con navegador y acceso a internet. - Debe disponer del enlace facilitado por el profesor para el autorregistro.

4.2.2 Casos de uso

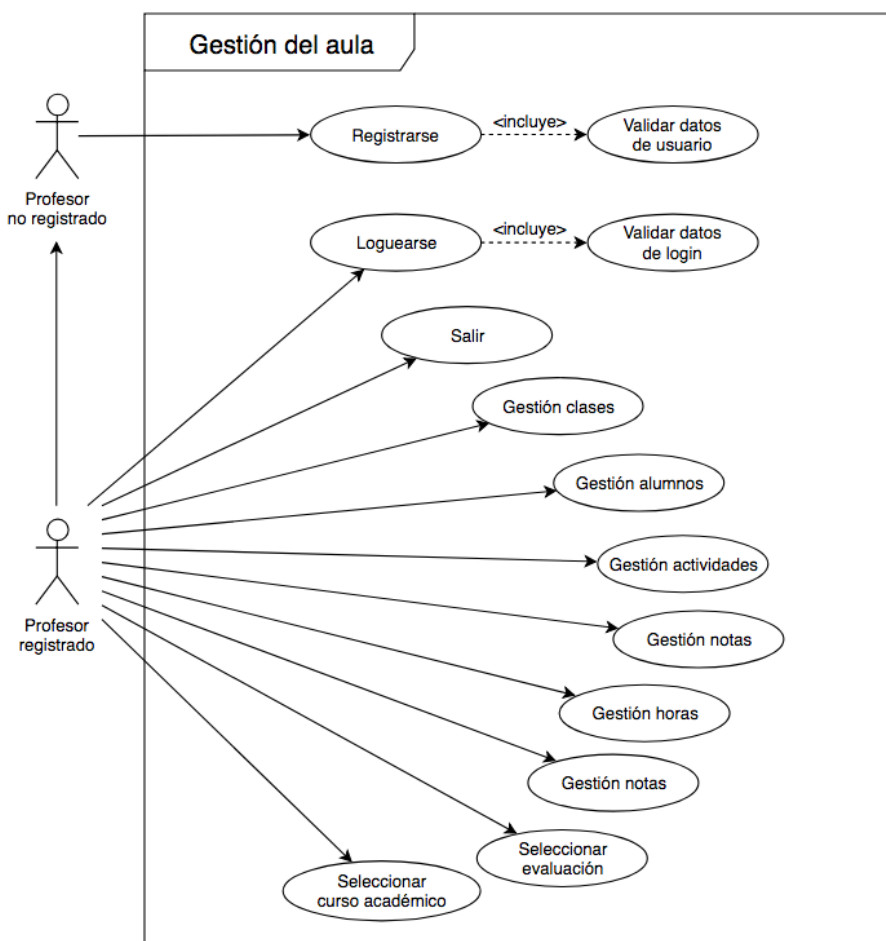


Figura 3.2 – Casos de uso generales de la aplicación

Caso de uso	Registrarse
Descripción	Permite a un usuario crear una cuenta en la aplicación web para así poder disfrutar de los servicios que esta ofrece.
Actores	Profesor no registrado.
Precondiciones	Ninguna.
Requisitos funcionales	RF-01
Requisitos no funcionales	Disponer de una dirección de correo electrónico.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Registrarse” en el menú de la barra superior. 2. El usuario introduce sus datos en el formulario de registro y pulsa “Registrarse”. 3. Se validan los datos introducidos.

Postcondiciones	Se debe crear el nuevo usuario en la base de datos. Se crea una nueva sesión y se redirige al usuario a la pantalla de grupos.
Excepciones	Si el usuario introduce datos no válidos, el sistema informará al usuario y permitirá volver a intentar el registro.

Caso de uso	Loguearse
Descripción	Permite al usuario iniciar sesión en la aplicación.
Actores	Profesor registrado.
Precondiciones	Haber creado una cuenta anteriormente.
Requisitos funcionales	RF-02
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción "Iniciar sesión" en el menú de la barra superior. 2. El usuario introduce el correo electrónico y la contraseña que eligió durante el registro y pulsa "Entrar". 3. Se validan los datos introducidos.
Postcondiciones	Se crea una nueva sesión y se redirige al usuario a la pantalla de creación de grupos.
Excepciones	Si el usuario introduce datos no válidos, el sistema informará al usuario y permitirá volver a intentar el inicio de sesión.

Caso de uso	Salir
Descripción	Permite al usuario cerrar sesión en la aplicación.
Actores	Profesor registrado.
Precondiciones	Haber iniciado sesión anteriormente.
Requisitos funcionales	RF-25
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción "Salir" en el menú de la barra superior. 2. Se muestra la pantalla de bienvenida.
Postcondiciones	Se cierra la sesión y se destruyen los datos asociados a ella.

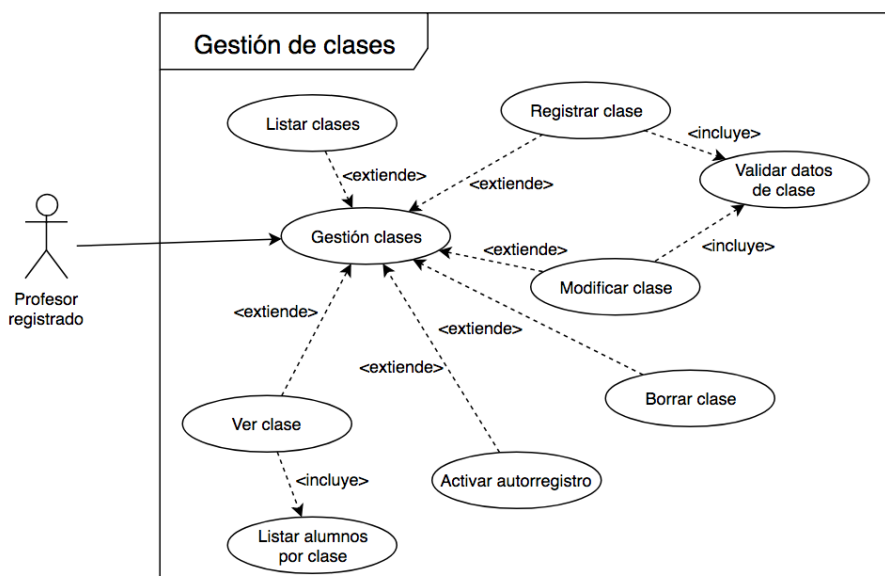


Figura 3.3 – Detalle del caso de uso de gestión de clases

Caso de uso	Listar clases
Descripción	Permite al usuario obtener un listado de todos sus grupos.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión.
Requisitos funcionales	RF-05
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario inicia sesión o pulsa el enlace “Clases” en el menú de la barra superior. 2. Se muestra la lista con los grupos creados por dicho usuario, en caso de no haber grupos se mostrará la lista vacía. 3. El usuario puede navegar por la lista haciendo scroll.
Postcondiciones	Ninguna.
Excepciones	Ninguna

Caso de uso	Registrar clase
Descripción	El usuario crea una clase para su posterior uso en la aplicación.
Actores	Profesor registrado.

Precondiciones	Iniciar sesión.
Requisitos funcionales	RF-04
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la lista de clases. 2. El usuario pulsa el botón "Añadir clase" disponible al final de la lista de clases. 3. El usuario introduce los datos en el formulario para añadir clases y pulsa el botón "Guardar". 4. Se validan los datos introducidos. 5. Se añade la nueva clase a la lista de manera dinámica.
Postcondiciones	<p>Se debe crear un registro en la tabla correspondiente en la base de datos.</p> <p>Si es la primera clase que se registra se recargará la página.</p> <p>Si es la primera clase de un curso académico diferente al curso académico por defecto, se mostrará un mensaje informando al usuario que recargue la página para visualizar correctamente los datos.</p>
Excepciones	Si el usuario introduce datos incorrectos se le informa y permite intentar la creación de la clase de nuevo.

Caso de uso	Modificar clase
Descripción	El usuario modifica los datos de una clase.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión y la clase tiene que estar creada en el sistema por el propio usuario.
Requisitos funcionales	RF-10 RF-23
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la lista de clases. 2. El usuario pulsa sobre cualquier dato de la clase. 3. El dato se convierte en un campo de formulario y el usuario introduce el nuevo valor. 4. Se validan los datos introducidos. 5. Se modifica el dato de manera dinámica.

Postcondiciones	Se modifica el registro correspondiente en la tabla en la base de datos.
Excepciones	Si el usuario introduce datos incorrectos se le informa y se mantiene el dato original.

Caso de uso Borrar clase	
Descripción	El usuario elimina una clase.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión y la clase tiene que estar creada en el sistema por el propio usuario. La clase no puede tener elementos dependientes de ella como alumnos, horas o actividades.
Requisitos funcionales	RF-10 RF-24
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la lista de clases. 2. El usuario pulsa sobre el icono de la papelera correspondiente a la clase a borrar. 3. Aparece una ventana modal informando de la acción a realizar. 4. El usuario pulsa el botón "Eliminar". 5. Se elimina la clase a la lista de manera dinámica.
Postcondiciones	Se elimina el registro correspondiente en la tabla en la base de datos.
Excepciones	Si ocurre un error al eliminar la clase se le informa al usuario.

Caso de uso Ver clase	
Descripción	El usuario quiere visualizar los datos relacionados con una clase.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión y la clase tiene que estar creada en el sistema por el propio usuario.
Requisitos funcionales	RF-10
Requisitos no funcionales	Ninguno.

Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón o enlace de acceder a una clase disponible en diferentes lugares de la aplicación. 2. Se muestra una nueva página con toda la información de la clase y la información dependiente de la clase.
Postcondiciones	Ninguna
Excepciones	Si la clase que se desea ver no está en el sistema se redirigirá a una página de error con la lista de clases del profesor. Si la clase que se desea ver no pertenece al usuario se redirigirá a la página de lista de clases.

Caso de uso	Activación del autorregistro
Descripción	El usuario quiere permitir que los alumnos introduzcan sus datos por ellos mismos.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión y la clase tiene que estar creada en el sistema por el propio usuario.
Requisitos funcionales	RF-10 RF-23 RF-26
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario entra a la página de la clase. 2. El usuario pulsa sobre el botón "Autorregistro alumnos" disponible en la barra de opciones de la clase. 3. Se muestra el formulario de activación de autorregistro. 4. El usuario introduce la palabra clave. 5. El usuario hace clic en el interruptor de activación. 6. Se almacena la palabra clave en el sistema. 7. Se muestra un enlace que se podrá proporcionar a los alumnos
Postcondiciones	Se modifican los campos necesarios (keyword y selfregistration) en el registro de la clase en la base de datos
Excepciones	Si se produce un error guardando la palabra clave se informará al usuario y se le permitirá volver a guardar la palabra clave.

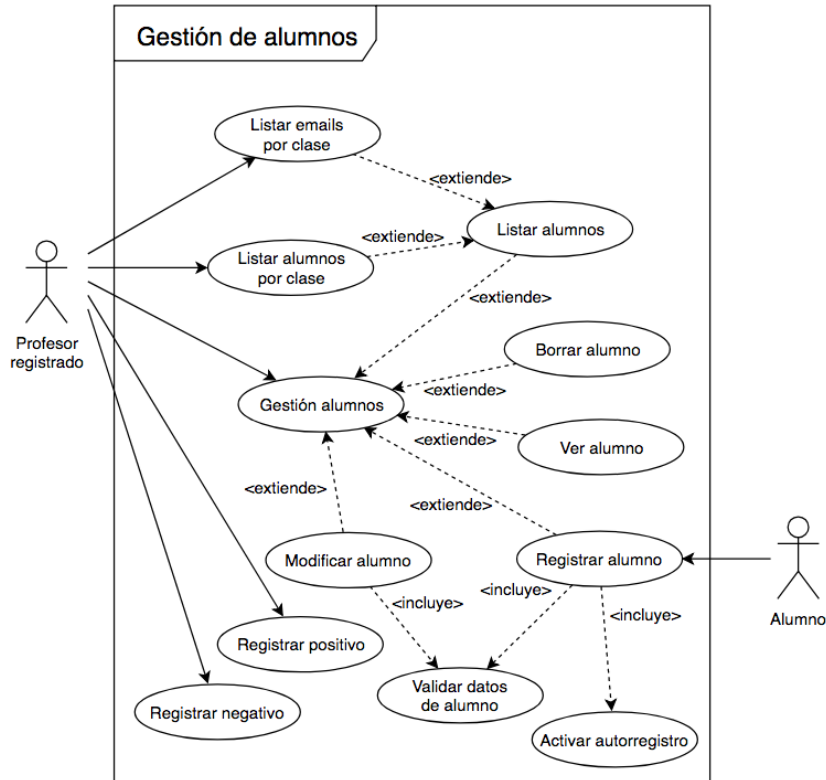


Figura 3.4 – Detalle del caso de uso de gestión de alumnos

Caso de uso	Registrar alumno
Descripción	El usuario crea un alumno en una clase para su posterior uso en la aplicación.
Actores	Profesor registrado
Precondiciones	Iniciar sesión y la clase debe estar creada en el sistema por el propio usuario.
Requisitos funcionales	RF-06
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la lista de alumnos o a una clase. 2. El usuario pulsa el botón "Añadir alumno" disponible al final de la lista de alumnos o al final de la página de la clase. 3. El usuario introduce los datos en el formulario para añadir alumnos y pulsa el botón "Guardar". 4. Se validan los datos introducidos. 5. Se añade el nuevo alumno a la lista de manera dinámica.

Postcondiciones	Se debe crear un registro en la tabla correspondiente en la base de datos.
Excepciones	Si el usuario introduce datos incorrectos se le informa y permite intentar la creación del alumno de nuevo.

Caso de uso Registrar alumno	
Descripción	El usuario crea un alumno en una clase para su posterior uso en la aplicación.
Actores	Alumno
Precondiciones	La clase debe estar creada en el sistema.
Requisitos funcionales	RF-06 RF-27
Requisitos no funcionales	Conocer el enlace de autorregistro correspondiente a la clase.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario introduce la URL del enlace facilitado por el profesor. 2. El usuario introduce los datos en el formulario para añadir alumnos y pulsa el botón "Guardar". 3. Se validan los datos introducidos. 4. Se muestra la información de registro correcto.
Postcondiciones	Se debe crear un registro en la tabla correspondiente en la base de datos.
Excepciones	Si el usuario introduce datos incorrectos se le informa y permite intentar la creación del alumno de nuevo.

Caso de uso Modificar alumno	
Descripción	El usuario modifica los datos de un alumno.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión, el alumno debe pertenecer a una clase que debe estar creada en el sistema por el propio usuario.
Requisitos funcionales	RF-23
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre cualquier dato del alumno.

	<ol style="list-style-type: none"> 2. El dato se convierte en un campo de formulario y el usuario introduce el nuevo valor. 3. Se validan los datos introducidos. 4. Se modifica el dato de manera dinámica.
Postcondiciones	Se modifica el registro correspondiente en la tabla en la base de datos.
Excepciones	Si el usuario introduce datos incorrectos se le informa y se mantiene el dato original.

Caso de uso Borrar alumno	
Descripción	El usuario elimina un alumno.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión, el alumno debe pertenecer a una clase que debe estar creada en el sistema por el propio usuario. El alumno no puede tener elementos dependientes de él como positivos, negativos, fotos de actividades o notas.
Requisitos funcionales	RF-24
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la lista de alumnos general o de una clase. 2. El usuario pulsa sobre el icono de la papelera correspondiente al alumno a borrar. 3. Aparece una ventana modal informando de la acción a realizar. 4. El usuario pulsa el botón "Eliminar". 5. Se elimina el alumno de la lista de manera dinámica.
Postcondiciones	Se elimina el registro correspondiente en la tabla en la base de datos.
Excepciones	Si ocurre un error al eliminar al alumno se le informa al usuario.

Caso de uso Ver alumno	
Descripción	El usuario quiere visualizar los datos relacionados con un alumno.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión, el alumno debe estar creado en el sistema por el propio usuario.

Requisitos funcionales	RF-07 RF-11
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón o enlace de acceder a un alumno disponible en diferentes lugares de la aplicación. 2. Se muestra una nueva página con toda la información del alumno y la información dependiente de él.
Postcondiciones	Ninguna
Excepciones	Si el alumno que se desea ver no está en el sistema se redirigirá a una página de error con la lista de clases del profesor. Si el alumno que se desea ver no pertenece al usuario se redirigirá a la página de lista de clases.

Caso de uso Listar emails por clase	
Descripción	El usuario quiere visualizar los emails de los alumnos de una clase.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión, la clase debe estar creada en el sistema por el propio usuario.
Requisitos funcionales	RF-07 RF-10 RF-11 RF-22
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario entra a la página de una clase. 2. El usuario pulsa el enlace "Emails" que se encuentra en la barra de opciones de la clase. 3. Se muestra la lista de emails de los alumnos registrados en esa clase y un botón para copiar toda la lista de emails.
Postcondiciones	Ninguna
Excepciones	Si no hay alumnos registrados en la clase se mostrará la lista vacía.

Caso de uso		Listar alumnos por clase
Descripción	El usuario quiere visualizar los alumnos de una clase.	
Actores	Profesor registrado.	
Precondiciones	Iniciar sesión, la clase debe estar creada en el sistema por el propio usuario.	
Requisitos funcionales	RF-07 RF-10 RF-11	
Requisitos no funcionales	Ninguno.	
Flujo normal	<ol style="list-style-type: none"> 1. El usuario entra a la página de una clase. 2. Se muestra una relación de fotos y nombres de alumnos de la clase. 	
Postcondiciones	Ninguna	
Excepciones	Si no hay alumnos registrados en la clase se mostrará la lista vacía.	

Caso de uso		Registrar positivo
Descripción	El usuario quiere poner un positivo a un alumno.	
Actores	Profesor registrado.	
Precondiciones	Iniciar sesión, el alumno debe estar creado en el sistema por el propio usuario.	
Requisitos funcionales	RF-11 RF-20	
Requisitos no funcionales	Ninguno.	
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón de positivo para un alumno disponible en la página de la clase o en la página del alumno. 2. Se muestra un formulario para introducir los datos del positivo. 3. El usuario introduce los datos. 4. Se validan los datos introducidos. 5. Si se está en la página del alumno se añade el positivo a la lista de positivos de manera dinámica. 	

Postcondiciones	Se debe crear un registro en la tabla correspondiente en la base de datos.
Excepciones	Si el usuario introduce datos incorrectos se le informa y permite intentar la creación del alumno de nuevo.

Caso de uso Registrar negativo	
Descripción	El usuario quiere poner un negativo a un alumno.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión, el alumno debe estar creado en el sistema por el propio usuario.
Requisitos funcionales	RF-11 RF-21
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón de negativo para un alumno disponible en la página de la clase o en la página del alumno. 2. Se muestra un formulario para introducir los datos. 3. El usuario introduce los datos. 4. Se validan los datos introducidos. 5. Si se está en la página del alumno se añade el negativo a la lista de positivos de manera dinámica.
Postcondiciones	Se debe crear un registro en la tabla correspondiente en la base de datos.
Excepciones	Si el usuario introduce datos incorrectos se le informa y permite intentar la creación del alumno de nuevo.

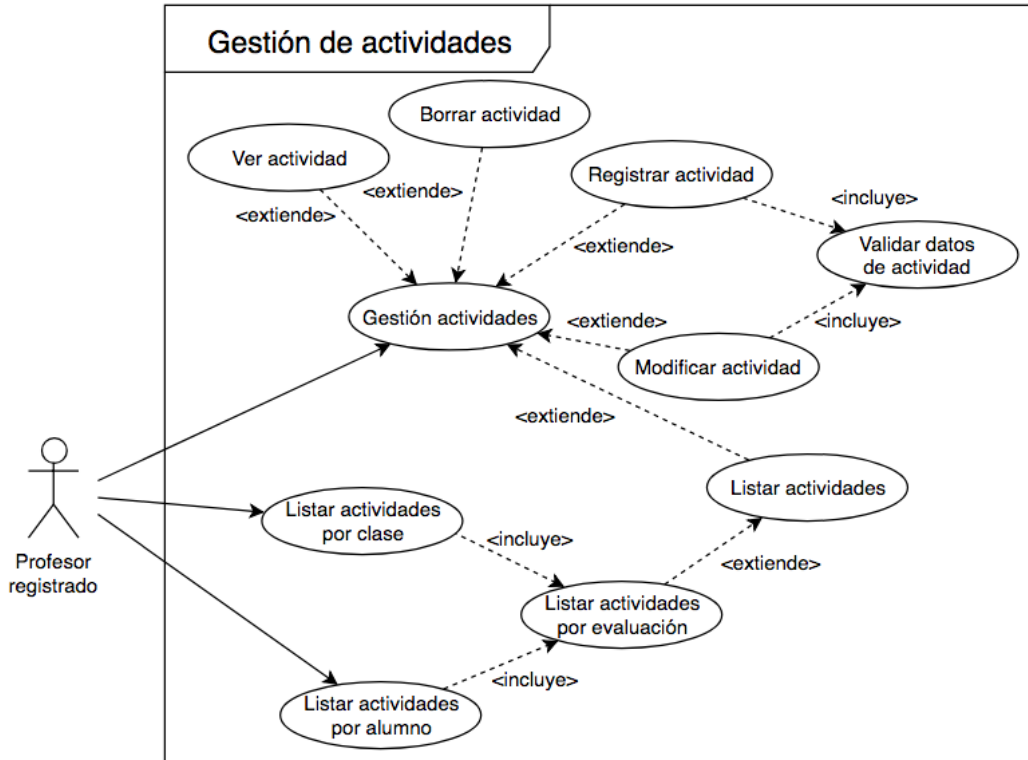


Figura 3.5 – Detalle del caso de uso de gestión de actividades

Caso de uso	Registrar actividad
Descripción	El usuario crea una actividad en una clase para su posterior uso en la aplicación.
Actores	Profesor registrado
Precondiciones	Iniciar sesión y la clase debe estar creada en el sistema por el propio usuario.
Requisitos funcionales	RF-08
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página de una clase. 2. El usuario pulsa el botón "Actividades" disponible en la barra de opciones de la clase. 3. Se muestra la lista de actividades de esa clase para la evaluación por defecto y un botón "Añadir actividad". 4. El usuario pulsa el botón "Añadir actividad" disponible al final de la lista de actividades. 5. El usuario introduce los datos en y pulsa el botón "Guardar". 6. Se validan los datos introducidos. 7. Se añade la nueva actividad a la lista de manera dinámica.

	8. Si la actividad es de una evaluación distinta a la evaluación por defecto se actualiza la lista de actividades con las actividades de esa evaluación.
Postcondiciones	Se debe crear un registro en la tabla correspondiente en la base de datos.
Excepciones	Si el usuario introduce datos incorrectos se le informa y permite intentar la creación de la actividad de nuevo.

Caso de uso	Modificar actividad
Descripción	El usuario modifica los datos de una actividad.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión, la actividad debe pertenecer a una clase que debe estar creada en el sistema por el propio usuario.
Requisitos funcionales	RF-12 RF-23
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la lista de actividades de una clase. 2. El usuario pulsa sobre cualquier dato de la actividad. 3. El dato se convierte en un campo de formulario y el usuario introduce el nuevo valor. 4. Se validan los datos introducidos. 5. Se modifica el dato de manera dinámica.
Postcondiciones	Se modifica el registro correspondiente en la tabla en la base de datos.
Excepciones	Si el usuario introduce datos incorrectos se le informa y se mantiene el dato original.

Caso de uso	Borrar actividad
Descripción	El usuario elimina una actividad.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión, la actividad debe pertenecer a una clase creada en el sistema por el propio usuario. La actividad no puede tener elementos dependientes de ella como notas o fotografías.
Requisitos funcionales	RF-24

Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la lista de actividades de una clase. 2. El usuario pulsa sobre el icono de la papelera correspondiente a la actividad a borrar. 3. Aparece una ventana modal informando de la acción. 4. El usuario pulsa el botón "Eliminar". 5. Se elimina la actividad de la lista de manera dinámica.
Postcondiciones	Se elimina el registro correspondiente en la tabla en la base de datos.
Excepciones	Si ocurre un error al eliminar la actividad se le informa al usuario.

Caso de uso	Ver actividad
Descripción	El usuario quiere visualizar los datos relacionados con una actividad.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión, la actividad debe estar creada en el sistema por el propio usuario.
Requisitos funcionales	RF-12
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón o enlace de acceder a una actividad disponible en diferentes lugares de la aplicación. 2. Se muestra una nueva página con toda la información de la actividad así como la lista de alumnos que han de realizarla y las calificaciones obtenidas.
Postcondiciones	Ninguna
Excepciones	Si la actividad que se desea ver no está en el sistema se redirigirá a una página de error con la lista de clases del profesor. Si la actividad que se desea ver no pertenece al usuario se redirigirá a la página de lista de clases.

Caso de uso	Listar actividades por clase
Descripción	El usuario quiere visualizar las actividades de una clase.
Actores	Profesor registrado.

Precondiciones	Iniciar sesión, la clase debe estar creada en el sistema por el propio usuario.
Requisitos funcionales	RF-10 RF-12
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario entra a la página de una clase. 2. El usuario pulsa el enlace "Actividades" que se encuentra en la barra de opciones de la clase. 3. Se muestra la lista de actividades registradas en esa clase. 4. También se muestran botones para poder ver las actividades de una evaluación concreta o ver las actividades de todo el curso.
Postcondiciones	Ninguna
Excepciones	Si no hay alumnos registrados en la clase se mostrará la lista vacía.

Caso de uso Listar actividades por alumno	
Descripción	El usuario quiere visualizar las actividades que tiene que realizar un alumno.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión, el alumno debe estar creado en el sistema por el propio usuario.
Requisitos funcionales	RF-11 RF-12
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario entra a la página de un alumno. 2. El usuario pulsa el botón "Actividades" en la barra de opciones del alumno. 3. Se muestra la lista de actividades que debe realizar el alumno. 4. También se muestran botones para poder ver las actividades de una evaluación concreta o ver las actividades de todo el curso.
Postcondiciones	Ninguna
Excepciones	Si no hay actividades registradas en la clase se mostrará la lista vacía.

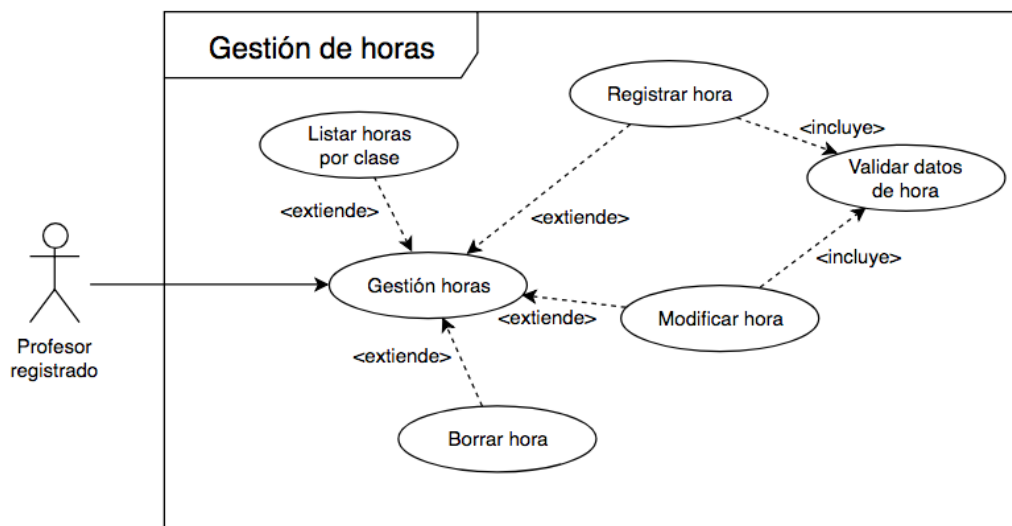


Figura 3.6 – Detalle del caso de uso de gestión de horas

Caso de uso	Registrar hora
Descripción	El usuario crea una hora en una clase.
Actores	Profesor registrado
Precondiciones	Iniciar sesión y la clase debe estar creada en el sistema por el propio usuario.
Requisitos funcionales	RF-09 RF-10
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página de una clase. 2. El usuario pulsa el botón “Horas” disponible en la barra de opciones de la clase. 3. Se muestra la lista de horas de esa clase y un botón “Añadir hora”. 4. El usuario pulsa el botón “Añadir hora” disponible al final de la lista de horas. 5. El usuario introduce los datos en el formulario para añadir horas y pulsa el botón “Guardar”. 6. Se validan los datos introducidos. 7. Se añade la nueva hora a la lista de manera dinámica.
Postcondiciones	Se debe crear un registro en la tabla correspondiente en la base de datos.
Excepciones	Si el usuario introduce datos incorrectos se le informa y permite intentar la creación de la hora de nuevo.

Caso de uso	Modificar hora
Descripción	El usuario modifica los datos de una hora.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión, la hora debe pertenecer a una clase que debe estar creada en el sistema por el propio usuario.
Requisitos funcionales	RF-10 RF-23
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la lista de horas de una clase. 2. El usuario pulsa sobre cualquier dato de la hora. 3. El dato se convierte en un campo de formulario y el usuario introduce el nuevo valor. 4. Se validan los datos introducidos. 5. Se modifica el dato de manera dinámica.
Postcondiciones	Se modifica el registro correspondiente en la tabla en la base de datos.
Excepciones	Si el usuario introduce datos incorrectos se le informa y se mantiene el dato original.

Caso de uso	Borrar hora
Descripción	El usuario elimina una hora.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión, la hora debe pertenecer a una clase que debe estar creada en el sistema por el propio usuario.
Requisitos funcionales	RF-10 RF-24
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la lista de horas de una clase. 2. El usuario pulsa sobre el icono de la papelera correspondiente a la hora a borrar. 3. Aparece una ventana modal informando de la acción a realizar. 4. El usuario pulsa el botón "Eliminar". 5. Se elimina la hora de la lista de manera dinámica.

Postcondiciones	Se elimina el registro correspondiente en la tabla en la base de datos.
Excepciones	Si ocurre un error al eliminar la hora se le informa al usuario.

Caso de uso Listar horas por clase	
Descripción	El usuario quiere visualizar las horas de una clase.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión, la clase debe estar creada en el sistema por el propio usuario.
Requisitos funcionales	RF-10
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario entra a la página de una clase. 2. El usuario pulsa el enlace "Horas" que se encuentra en la barra de opciones de la clase. 3. Se muestra la lista de horas registradas en esa clase.
Postcondiciones	Ninguna
Excepciones	Si no hay horas registradas en la clase se mostrará la lista vacía.

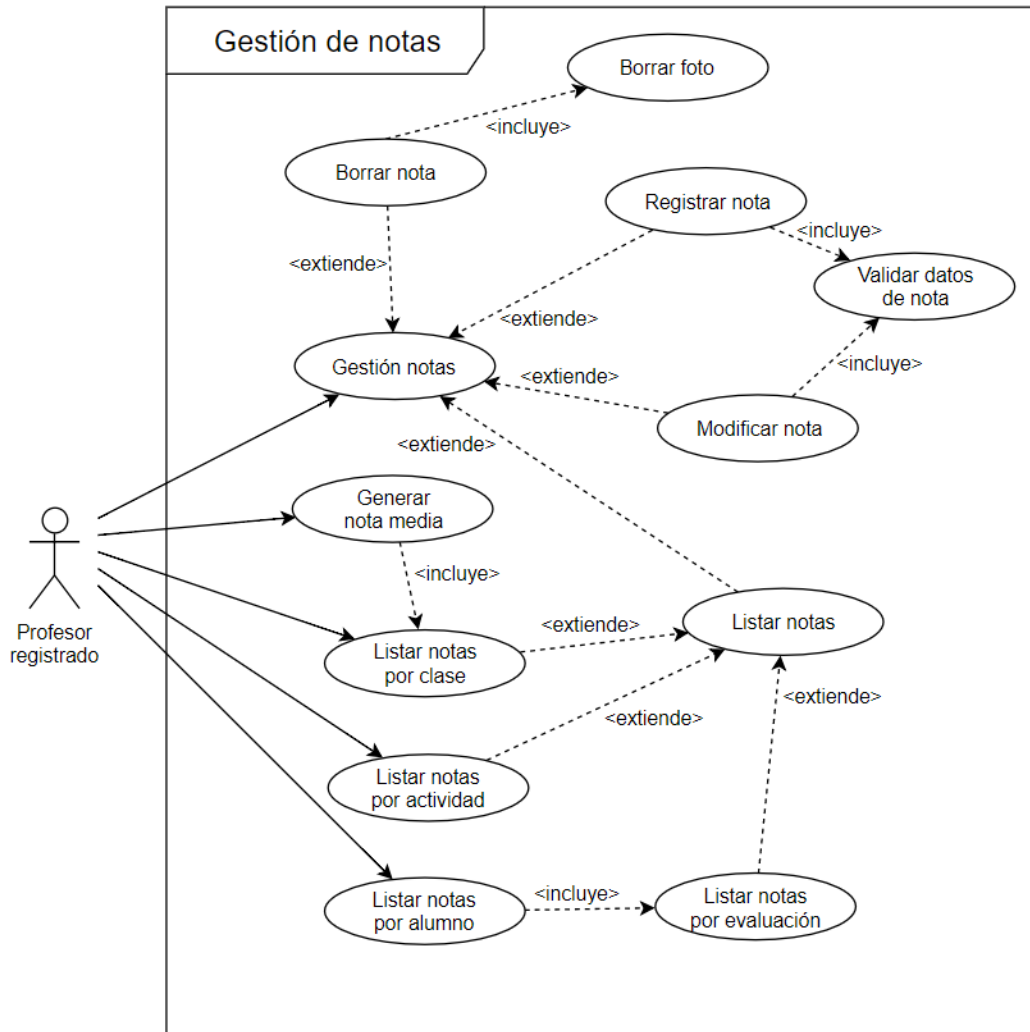


Figura 3.7 – Detalle del caso de uso de gestión de notas

Caso de uso	Registrar nota
Descripción	El usuario asigna una nota a una actividad realizada por un alumno.
Actores	Profesor registrado
Precondiciones	Iniciar sesión y la actividad debe estar creada para una clase que debe estar creada en el sistema por el propio usuario.
Requisitos funcionales	RF-16 RF-17
Requisitos funcionales no	Ninguno.
Flujo normal	1. El usuario accede a la lista de actividades de un alumno, a una actividad o a la lista de notas de una clase.

	<ol style="list-style-type: none"> 2. El usuario pulsa sobre el espacio reservado para la calificación. 3. El espacio se convierte en un campo de formulario y el usuario introduce el nuevo valor. 4. Se validan los datos introducidos. 5. Se modifica el dato de manera dinámica.
Postcondiciones	Se debe crear un registro en la tabla correspondiente en la base de datos.
Excepciones	Si el usuario introduce datos incorrectos se le informa y el espacio reservado se queda como al principio.

Caso de uso		Modificar nota
Descripción	El usuario modifica la nota de una actividad de un alumno.	
Actores	Profesor registrado.	
Precondiciones	Iniciar sesión, la nota debe pertenecer a un alumno que debe estar en una clase creada en el sistema por el propio usuario.	
Requisitos funcionales	RF-23	
Requisitos no funcionales	Ninguno.	
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la lista de actividades de un alumno, a una actividad o a la lista de notas de una clase. 2. El usuario pulsa sobre la nota. 3. El dato se convierte en un campo de formulario y el usuario introduce el nuevo valor. 4. Se validan los datos introducidos. 5. Se modifica el dato de manera dinámica. 	
Postcondiciones	Se modifica el registro correspondiente en la tabla en la base de datos.	
Excepciones	Si el usuario introduce datos incorrectos se le informa y se mantiene el dato original.	

Caso de uso		Borrar nota
Descripción	El usuario elimina una nota de un alumno.	
Actores	Profesor registrado.	
Precondiciones	Iniciar sesión, la nota debe pertenecer a un alumno que debe estar en una clase creada en el sistema por el propio usuario.	

Requisitos funcionales	RF-24
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la lista de actividades de un alumno o a una actividad. 2. El usuario pulsa sobre el icono de la papelera correspondiente a la nota a borrar. 3. Aparece una ventana modal informando de la acción. 4. El usuario pulsa el botón "Eliminar". 5. Se elimina la hora de la lista de manera dinámica. 6. Se eliminan las fotos registradas para esa nota.
Postcondiciones	Se elimina el registro correspondiente en la tabla en la base de datos.
Excepciones	Si ocurre un error al eliminar la hora se le informa al usuario.

Caso de uso	Listar notas por clase
Descripción	El usuario quiere visualizar las notas obtenidas por los alumnos de una clase.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión, la clase debe estar creada en el sistema por el propio usuario, la actividad debe estar creada para esa clase.
Requisitos funcionales	RF-13
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario entra a la página de una clase. 2. El usuario pulsa el botón "Notas" disponible en la barra de opciones de la clase. 3. Se muestra una página todas las actividades de la clase y una relación con todos los alumnos que deben realizar la actividad y las notas obtenidas.
Postcondiciones	Ninguna
Excepciones	Si la clase que se desea ver no está en el sistema se redirigirá a una página de error con la lista de clases del profesor. Si la clase que se desea ver no pertenece al usuario se redirigirá a la página de lista de clases.

Caso de uso Listar notas por actividad	
Descripción	El usuario quiere visualizar las notas obtenidas por los alumnos en una actividad.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión, la clase debe estar creada en el sistema por el propio usuario, la actividad debe estar creada para esa clase.
Requisitos funcionales	RF-07 RF-11 RF-12 RF-13
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón o enlace de acceder a una actividad disponible en diferentes lugares de la aplicación. 2. Se muestra una nueva página con toda la información de la actividad así como la lista de alumnos que han de realizarla y las calificaciones obtenidas.
Postcondiciones	Ninguna
Excepciones	Si la actividad que se desea ver no está en el sistema se redirigirá a una página de error con la lista de clases del profesor. Si la actividad que se desea ver no pertenece al usuario se redirigirá a la página de lista de clases.

Caso de uso Listar notas por alumno	
Descripción	El usuario quiere visualizar las calificaciones obtenidas por un alumno.
Actores	Profesor registrado.
Precondiciones	Iniciar sesión, el alumno debe estar creado en el sistema por el propio usuario.
Requisitos funcionales	RF-11 RF-12 RF-13
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario entra a la página de un alumno.

	<ol style="list-style-type: none"> El usuario pulsa el botón “Actividades” en la barra de opciones del alumno. Se muestra la lista de actividades que debe realizar el alumno junto con las calificaciones obtenidas o el espacio reservado para ellas.
Postcondiciones	Ninguna
Excepciones	Si no hay actividades registradas en la clase se mostrará la lista vacía.

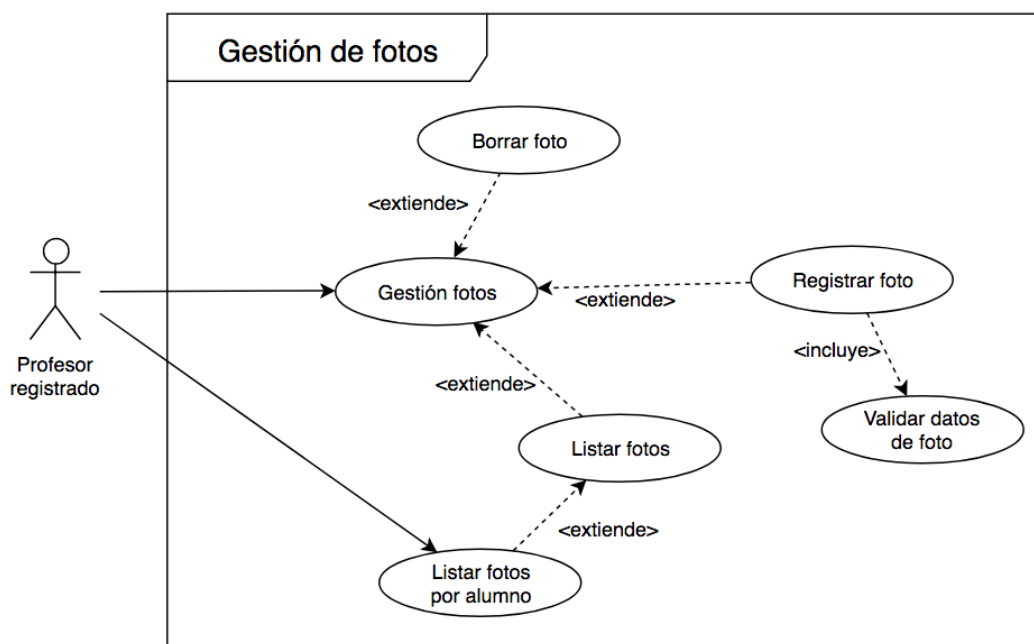


Figura 3.8 – Detalle del caso de uso de gestión de fotos

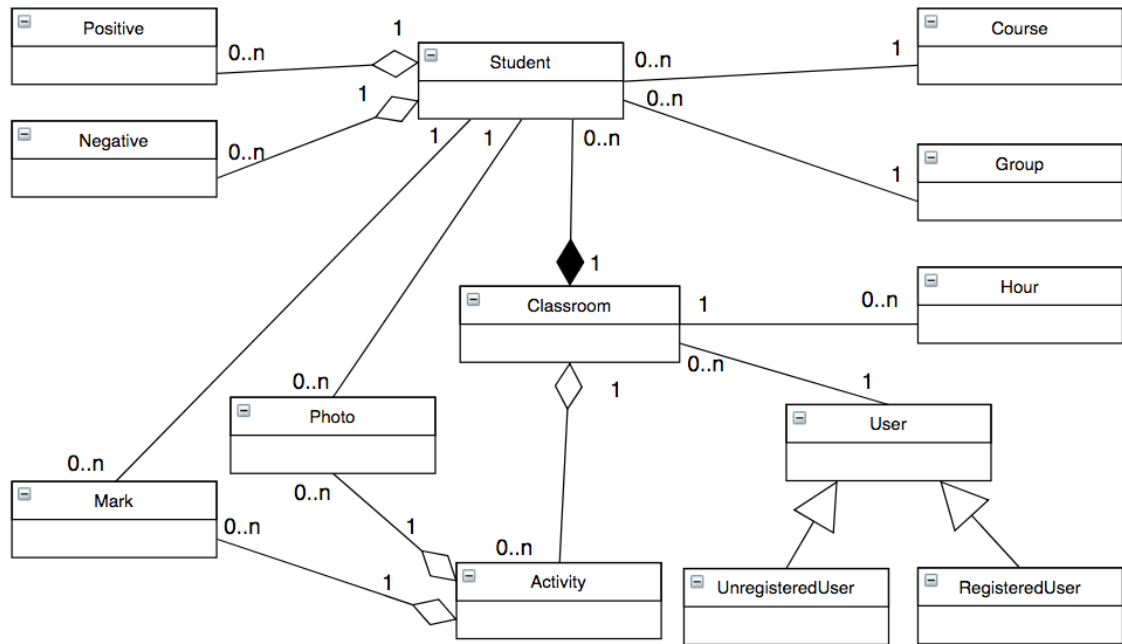
Caso de uso	Registrar foto
Descripción	El usuario asigna una foto a una actividad realizada por un alumno.
Actores	Profesor registrado
Precondiciones	Iniciar sesión y la actividad debe estar creada para una clase que debe estar creada en el sistema por el propio usuario.
Requisitos funcionales	RF-18
Requisitos no funcionales	Ninguno.
Flujo normal	<ol style="list-style-type: none"> El usuario accede a la lista de actividades de un alumno, a una actividad.

	<ol style="list-style-type: none"> 2. El usuario pulsa sobre el icono de la cámara para el alumno correspondiente. 3. Se muestra un formulario en el que se permite seleccionar un archivo para registrarlo en el sistema. 4. Se validan los datos introducidos. 5. Se almacena el archivo en el servidor.
Postcondiciones	Se debe crear un registro en la tabla correspondiente en la base de datos.
Excepciones	Si el usuario introduce datos incorrectos se le informa al usuario permitiendo volver a intentar registrar la foto.

Caso de uso		Borrar foto
Descripción	El usuario elimina una foto de una actividad de un alumno.	
Actores	Profesor registrado.	
Precondiciones	Iniciar sesión, la foto debe pertenecer a un alumno que debe estar en una clase creada en el sistema por el propio usuario.	
Requisitos funcionales	RF-24	
Requisitos no funcionales	Ninguno.	
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la lista de actividades de un alumno o a una actividad. 2. El usuario pulsa sobre el icono de foto correspondiente a las fotos del alumno para esa actividad. 3. Aparece una ventana modal mostrando todas las fotos registradas para ese alumno y actividad. 4. El usuario pulsa sobre el icono "x" de la foto a borrar. 5. Aparece una ventana modal informando de la acción a realizar. 6. El usuario pulsa el botón "Eliminar". 7. Se elimina la foto de la lista de manera dinámica. 	
Postcondiciones	Se elimina el registro correspondiente en la tabla en la base de datos.	
Excepciones	Si ocurre un error al eliminar la foto se le informa al usuario.	

4.3 Diagrama de clases

Mediante el diagrama de clases se muestran las entidades que intervienen en la aplicación, mostrando además cómo se relacionan entre sí. Todas las entidades tienen implementado un sistema CRUD⁹ aunque por simplicidad no se muestra en el diagrama.



⁹ CRUD (Create, Read, Update, Delete).



5 Diseño

5.1 Introducción

Para el diseño de la aplicación a desarrollar se ha tomado el patrón MVC¹⁰ [7] [8], este modelo dispone de tres componentes que son los que indican cada una de sus siglas.

El patrón de software MVC se caracteriza por la separación del código correspondiente a cada uno de los componentes del modelo. De esta forma se permite que estén separados la lógica de negocio, los datos y la interfaz de usuario.

Este patrón además permite que el código se pueda reutilizar y se facilite el desarrollo y mantenimiento de la aplicación.

5.2 Arquitectura física

La arquitectura física muestra cómo se distribuyen los componentes físicos que participan en la aplicación, de esta manera se puede ver cómo se relacionan el cliente con el servidor [9].

En esta arquitectura el servidor se encarga de realizar toda la lógica de negocio, la persistencia de datos y modelar la interfaz al usuario. El cliente se encargará de mostrar la interfaz y servir como intermediario en los eventos creados por las acciones realizadas por el usuario.

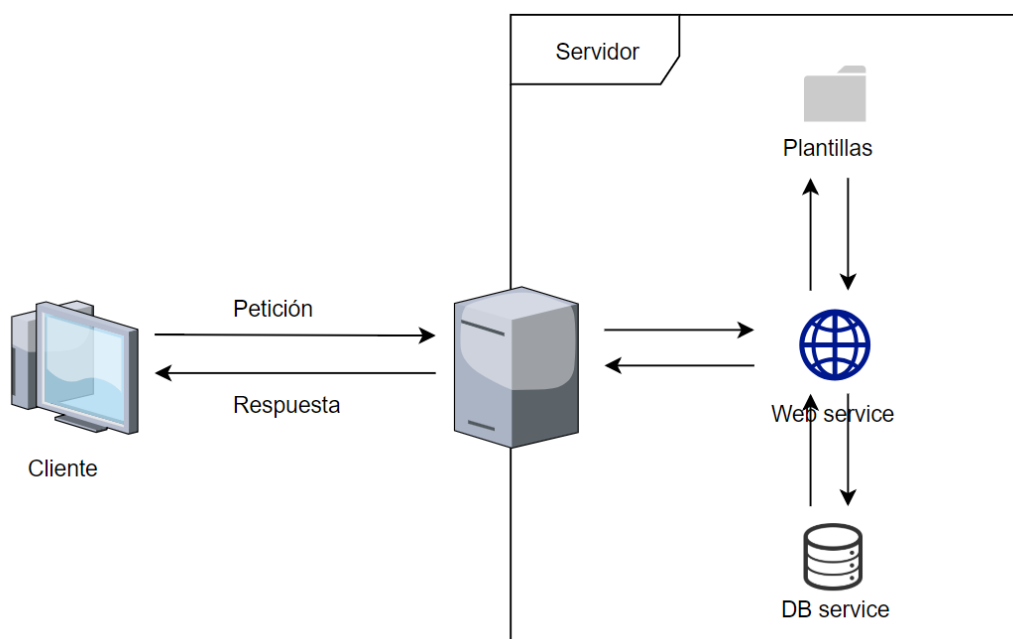


Figura 4.1 – Arquitectura de la aplicación

¹⁰ MVC (Modelo Vista Controlador).

5.3 Arquitectura lógica

La aplicación además de usar el patrón MVC añade un componente adicional que permite procesar las peticiones HTML que realiza el cliente y redirigirlas al controlador correspondiente, este componente es un módulo de enrutamiento. Gracias a este módulo se aporta mayor claridad al código de la aplicación ya que descarga a los controladores de esa responsabilidad.

Esta técnica usada en el desarrollo de aplicaciones web aporta varias ventajas, permite separar responsabilidades, permite reutilizar el código y facilita el trabajo en equipos de desarrollo pudiendo desarrollar de manera independiente el backend y el frontend.

Gracias al patrón MVC se puede separar la aplicación en tres capas: capa de persistencia, capa de presentación y capa lógica.

5.3.1 Capa de persistencia

En esta capa se representan y almacenan los datos que se usan en la aplicación. De esta manera se permite la creación, consulta, mantenimiento y modificación de los datos. Estas operaciones son solicitadas por la capa lógica de la aplicación. Esta capa se encarga de la comunicación entre el servidor web y el servidor de base de datos.

En este proyecto la base de datos es del tipo MariaDB y se gestiona gracias a la aplicación phpmyadmin¹¹.

Modelo

Como se utilizará el framework Laravel para el desarrollo de la aplicación, el modelo permite representar la información de la base de datos mediante un conjunto de clases. Laravel utiliza Eloquent, un ORM¹² que se encarga de realizar una correspondencia entre los objetos y las entidades de la base de datos. Gracias a Eloquent se permite crear de manera sencilla todas las clases del modelo, sus relaciones y las operaciones CRUD, todo ello haciendo uso de objetos del lenguaje PHP, evitando así hacer uso de código SQL.

¹¹ <https://www.phpmyadmin.net>

¹² ORM (Object-Relational Mapping).



Modelo Entidad-Relación

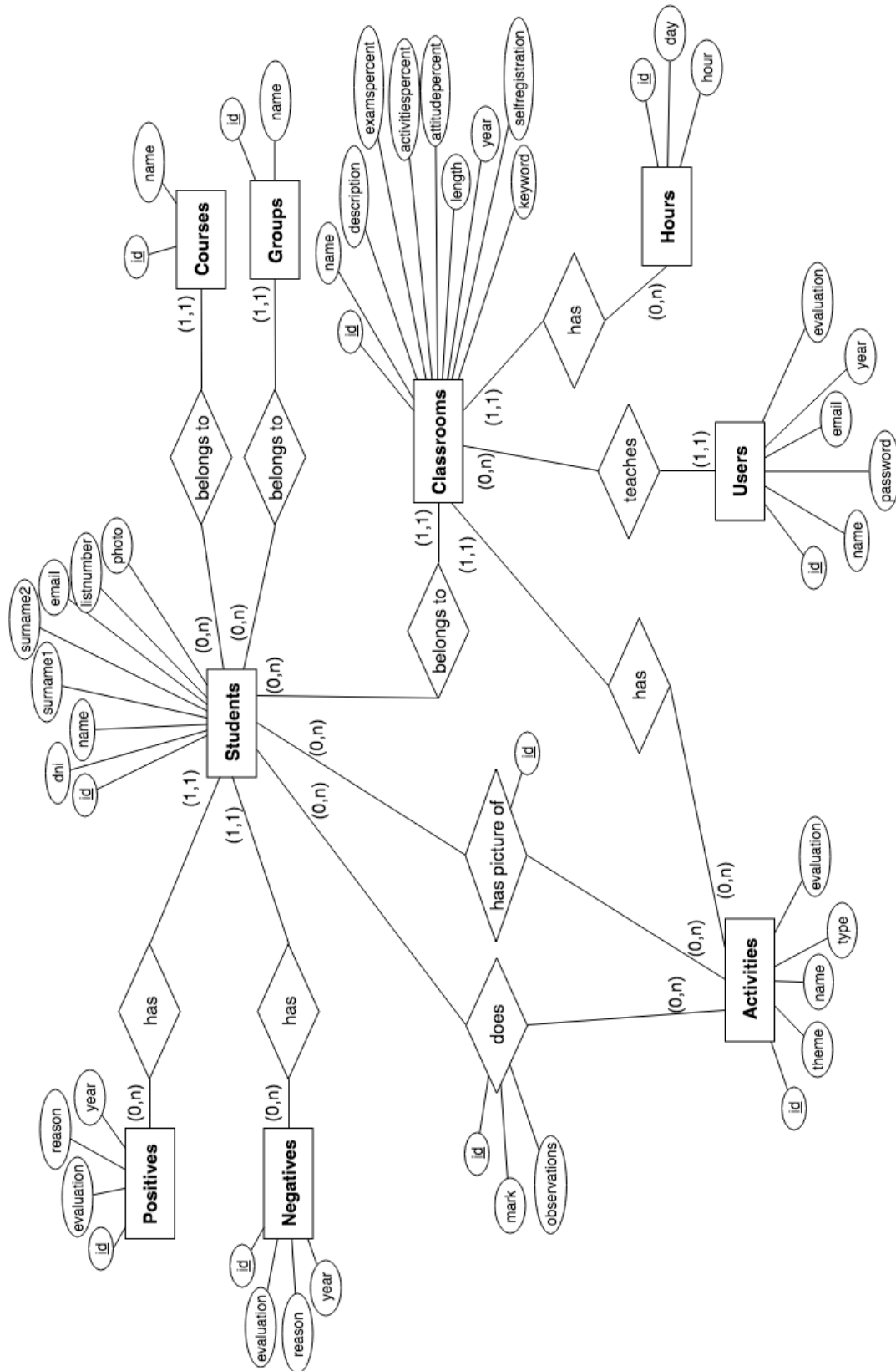


Figura 4.2 – Diagrama Entidad-Relación

Modelo relacional

A continuación, se muestran las tablas obtenidas a partir del modelo Entidad-Relación.

- Tabla classrooms

Columna	Tipo	Descripción
id	int	Identificador de la clase
name	varchar	Nombre de la clase
description	varchar	Descripción de la clase
examspercent	int	Porcentaje que valen los exámenes.
activitiespercent	int	Porcentaje que valen las actividades.
attitudepercent	int	Porcentaje que vale la actitud.
length	int	Duración de la clase en minutos
year	varchar	Descripción del curso académico (p.e. 18-19)
selfregistration	boolean	Indica si está habilitado el autorregistro.
keyword	varchar	Palabra clave para el autorregistro.
user_id	int	Identificador del profesor que crea la clase.
created_at	date	Fecha de creación de la clase.
updated_at	date	Fecha de actualización de la clase.

- Tabla students

Columna	Tipo	Descripción
id	int	Identificador del alumno.
dni	varchar	DNI/NIE del alumno.
name	varchar	Nombre del alumno.
surname1	varchar	Primer apellido del alumno.
surname2	varchar	Segundo apellido del alumno.
email	varchar	Dirección de correo electrónico del alumno.
course_id	int	Identificador del curso al que pertenece el alumno.
group_id	int	Identificador del grupo al que pertenece el alumno.
classroom_id	int	Identificador de la clase a la que pertenece el alumno.
listnumber	int	Número que ocupa en la lista por orden alfabético.
photo	boolean	Indica si hay foto del alumno.
created_at	date	Fecha de creación del alumno.
updated_at	date	Fecha de actualización del alumno.

- Tabla positives

Columna	Tipo	Descripción
id	int	Identificador del positivo.
student_id	int	Identificador del alumno al que pertenece el positivo.
evaluation	int	Evaluación en la que se creó el positivo.
year	varchar	Curso académico en el que se creó el positivo.
reason	varchar	Razón por la que se puso el positivo.
created_at	date	Fecha de creación del positivo.
updated_at	date	Fecha de actualización del positivo.

- Tabla negatives

Columna	Tipo	Descripción
id	int	Identificador del negativo.
student_id	int	Identificador del alumno al que pertenece el negativo.
evaluation	int	Evaluación en la que se creó el negativo.
year	varchar	Curso académico en el que se creó el negativo.
reason	varchar	Razón por la que se puso el negativo.
created_at	date	Fecha de creación del negativo.
updated_at	date	Fecha de actualización del negativo.

- Tabla marks

Columna	Tipo	Descripción
id	int	Identificador de la nota.
student_id	int	Identificador del alumno al que pertenece la nota.
activity_id	int	Identificador de la actividad para la que se califica.
mark	double	La calificación obtenida por el alumno.
observations	varchar	Posible observación a la calificación.
created_at	date	Fecha de creación de la calificación.
updated_at	date	Fecha de actualización de la calificación.

- Tabla photos

Columna	Tipo	Descripción
id	int	Identificador de la foto.
student_id	int	Identificador del alumno al que pertenece la foto.
activity_id	int	Identificador de la actividad a la que pertenece la foto.
created_at	date	Fecha de creación de la foto.
updated_at	date	Fecha de actualización de la foto.

- Tabla courses

Columna	Tipo	Descripción
id	int	Identificador del curso.
name	varchar	Nombre del curso.
created_at	date	Fecha de creación del curso.
updated_at	date	Fecha de actualización del curso.

- Tabla hours

Columna	Tipo	Descripción
id	int	Identificador de la hora.
classroom_id	int	Identificador de la clase a la que pertenece la hora.
day	varchar	Día de la semana a la que se imparte la clase.
hour	time	Hora del día a la que se imparte la clase.
created_at	date	Fecha de creación de la hora.
updated_at	date	Fecha de actualización de la hora.

- Tabla groups

Columna	Tipo	Descripción
id	int	Identificador del grupo.
name	varchar	Nombre del grupo.
created_at	date	Fecha de creación del grupo.
updated_at	date	Fecha de actualización del grupo.

- Tabla activities

Columna	Tipo	Descripción
id	int	Identificador de la actividad.
theme	varchar	Nombre del tema al que pertenece la actividad.
name	varchar	Nombre de la actividad.
type	varchar	Tipo de actividad.
classroom_id	int	Identificador de la clase a la que pertenece la actividad.
evaluation	int	Evaluación en la que se realiza la actividad
created_at	date	Fecha de creación de la actividad.
updated_at	date	Fecha de actualización de la actividad.

- Tabla users

Columna	Tipo	Descripción
id	int	Identificador del usuario.
name	varchar	Nombre del usuario.
email	varchar	Correo electrónico del usuario.
password	varchar	Contraseña encriptada del usuario.
year	varchar	Curso académico por defecto del usuario.
evaluation	int	Evaluación por defecto del usuario.
remembered_token	varchar	Cadena para permitir el inicio de sesión automático.
created_at	date	Fecha de creación del usuario.
updated_at	date	Fecha de actualización del usuario.

5.3.2 Capa de presentación

La capa de presentación es la que permite al usuario interactuar con el sistema. Se encarga de mostrar la información al usuario y de recoger la interacción que tenga el usuario con la aplicación. Además, realiza la comunicación con la capa lógica.

Vistas

Las vistas se encargan de mostrar la información que proporciona el controlador a través del modelo. Las vistas contienen el código HTML que se enviará al cliente para su posterior procesamiento. Al usar el framework Laravel, se hará uso del motor de plantillas Blade que está disponible en el framework. Este motor de plantillas integra de una manera muy sencilla el código HTML con instrucciones PHP de manera que el código queda mucho más limpio.



Bocetos

Mediante los bocetos se realiza un primer acercamiento a cómo será la interfaz de la aplicación web [10]. Se tratan de representaciones simplificadas de las diferentes pantallas de manera que de un vistazo se muestre cómo estará distribuida la información. Muestran la funcionalidad de una manera estática y permiten realizar cambios si es necesario antes de desarrollar el diseño definitivo [11].

Para la realización de los bocetos se ha usado la aplicación web Balsamiq¹³.

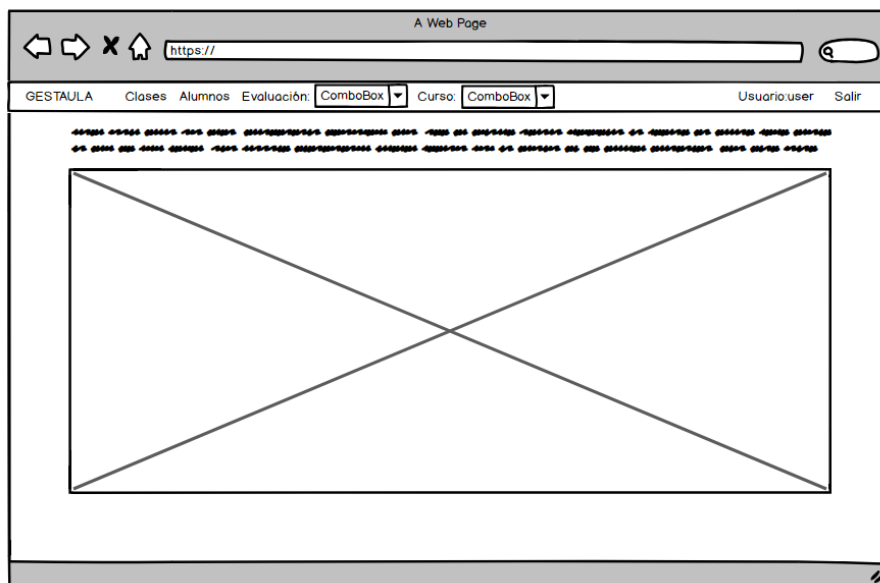


Figura 4.3 – Vista de la pantalla de inicio

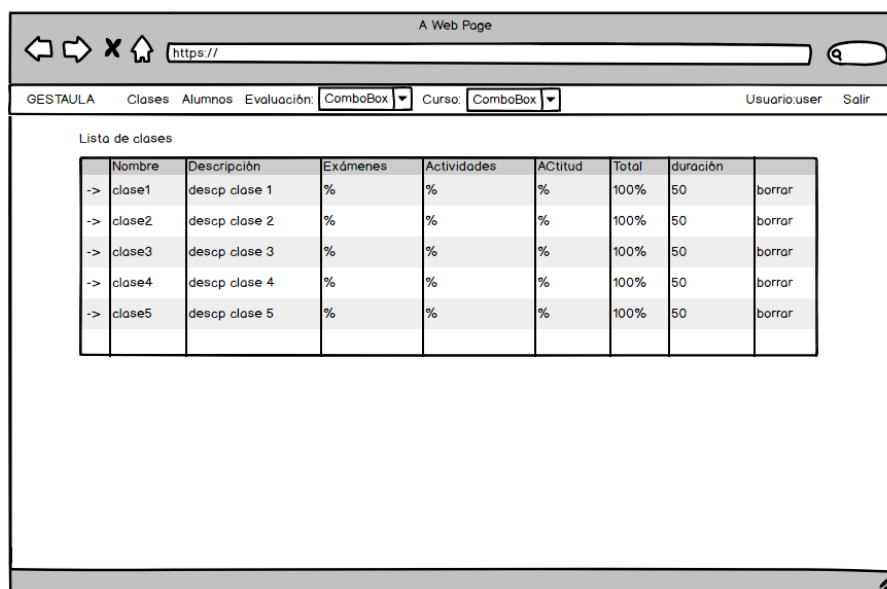


Figura 4.4 – Vista de la lista de clases.

¹³ <https://balsamiq.cloud>

Las listas de alumnos, de notas y la página de una actividad tendrán el mismo aspecto que la anterior con una tabla mostrando los datos

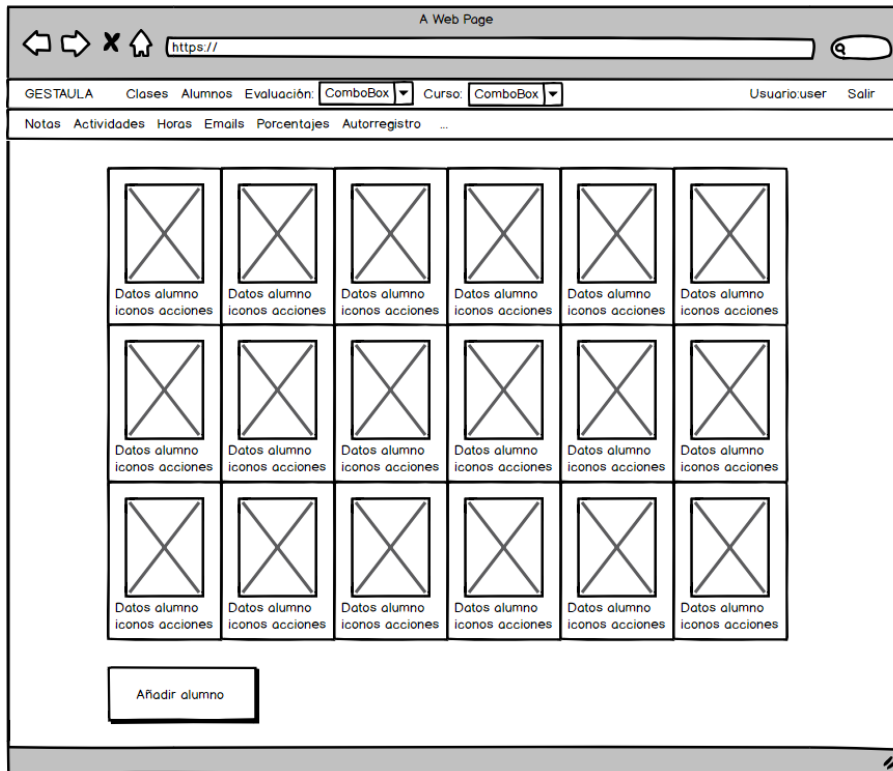


Figura 4.5 – Vista de una clase concreta

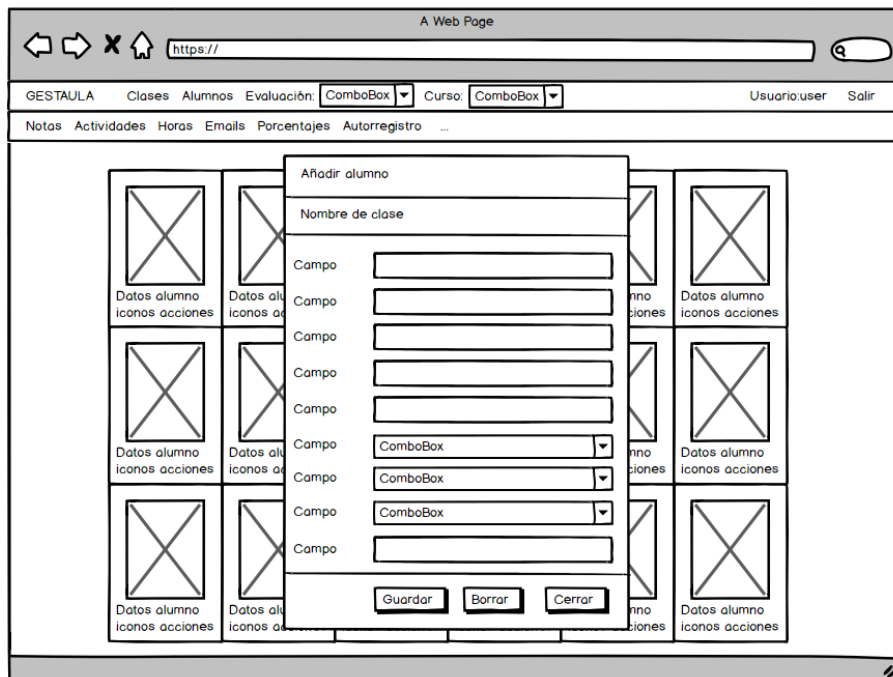


Figura 4.6 – Vista del formulario de introducir alumno.

Todos los formularios de la aplicación aparecerán como este en una ventana modal.

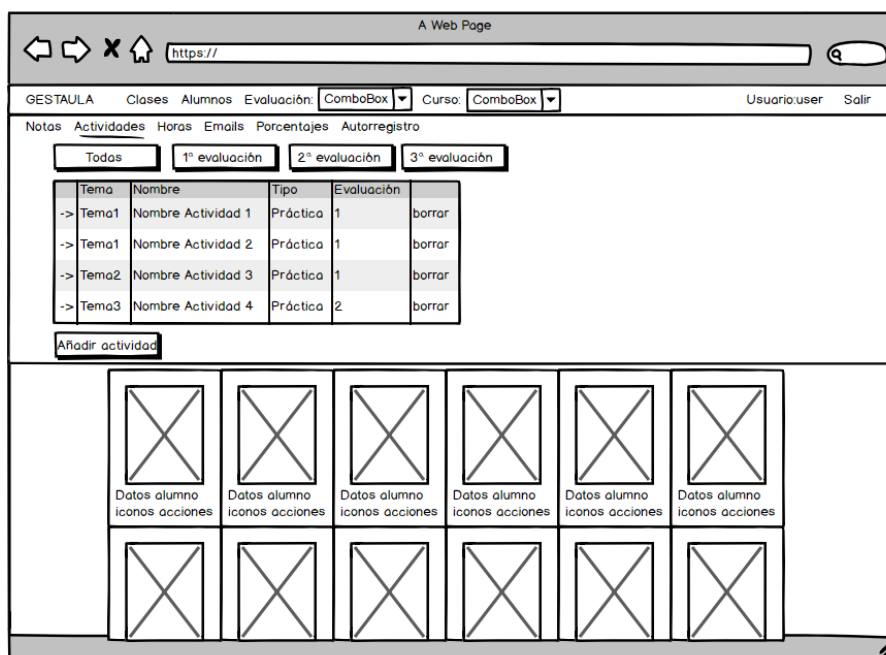


Figura 4.7 – Vista de las actividades de una clase.

Los elementos de la barra de opciones de la clase se mostrarán en la zona desplegable como las actividades, excepto las notas que se mostrarán en una página nueva.

5.3.3 Capa lógica

La capa lógica procesa las interacciones del usuario con la aplicación desde la capa de presentación, también se encarga de realizar las peticiones para consultar o modificar el modelo, y por último se encarga de enviar a las vistas los datos resultantes para que las vistas puedan mostrar la información correcta. Para realizar estas acciones se hará uso del enrutador y de los controladores.

El enrutador recibe las peticiones HTTP que se generan por la interacción del usuario con la aplicación y las redirigen al controlador adecuado. Laravel ofrece un fichero donde centralizar todo el enrutamiento, también ofrece la posibilidad de aplicar determinadas reglas para el control de usuarios registrados o de las peticiones permitidas.

Los controladores se encargan de responder a las peticiones recibidas de manera que se encargan de hacer funcionar toda la lógica de negocio de la aplicación. Los controladores también se encargan de realizar las peticiones CRUD al modelo y enviar los datos obtenidos a las vistas.

En este proyecto la capa lógica estará implementada en un servidor web con intérprete de lenguaje PHP y con una estructura facilitada por el framework Laravel.

6 Desarrollo

6.1 Tecnologías usadas

En la actualidad existen una gran cantidad de tecnologías disponibles a la hora de iniciar un desarrollo de una aplicación web, dentro de ese conjunto de tecnologías hay algunas que no pueden faltar como HTML o CSS y otras donde se puede optar entre varias opciones [9] [12]. A continuación, se muestran en detalle las tecnologías utilizadas.

6.1.1 HTML 5

HyerText Markup Language, HTML, es un lenguaje de marcado para hipertexto. Es el lenguaje utilizado para el desarrollo de aplicaciones web. Mediante este lenguaje se definen los elementos que se mostrarán al usuario en la aplicación web.

En el desarrollo de este proyecto se ha optado por usar la última versión de HTML, HTML 5, que dota al desarrollo web de un significado más semántico.

6.1.2 CSS 3

Gracias a CSS, Cascading Style Sheets, se puede dotar de estilo gráfico a los elementos creados mediante el lenguaje HTML. Para dotar a los elementos de estilo se usan los selectores. Existen infinidad de tipos de selectores, desde los selectores de etiqueta a los selectores de clase, de identificador, hasta selectores más complejos usando la jerarquía de los elementos o los atributos que tienen.

En el proyecto se ha hecho uso de la última versión de CSS, la versión 3, para dotar de estilo a la aplicación. Además, se ha usado como base la biblioteca para CSS Bootstrap.

6.1.3 Bootstrap 4

Bootstrap¹⁴ es una librería para CSS que facilita la creación del diseño de una aplicación web, dotando de diferentes elementos y estilos preestablecidos que además son fácilmente editables.

En el proyecto se ha decidido usar la última versión estable de la librería, Bootstrap 4.1.3.

6.1.4 JavaScript

JavaScript es un lenguaje de programación que se ejecuta en los clientes, los navegadores web. Gracias a JavaScript se consigue descargar al servidor de determinadas tareas, como son, ejecución de animaciones o interacción con el usuario.

¹⁴ <https://getbootstrap.com>



Mediante JavaScript se puede acceder a los elementos de la interfaz de la aplicación web y se pueden capturar y gestionar los eventos que se creen por la interacción del usuario con la interfaz.

6.1.5 jQuery 3

Debido a que trabajar con JavaScript es tedioso y a veces algo complicado surgieron diferentes bibliotecas de funciones que facilitarían el uso de JavaScript. Una de esas librerías es jQuery, que dispone de una gran cantidad de funciones que simplifican el uso de JavaScript gracias a que ofrece la posibilidad de trabajar con los objetos de la aplicación web usando selectores CSS.

En el proyecto se ha usado la última versión estable de jQuery¹⁵, la versión 3.3.1.

6.1.6 PHP

PHP es uno de los lenguajes de programación más usados en el desarrollo de aplicaciones web y desde las últimas versiones ya tienen la implementación de objetos para poder usar el paradigma de programación orientado a objetos.

En el desarrollo del proyecto se ha optado por usar la última versión de PHP por las ventajas que ofrece, PHP 7.

6.1.7 Laravel

Laravel¹⁶ es un framework [13] para el desarrollo mediante el lenguaje de programación PHP. Ofrece una manera elegante y sencilla de desarrollar. Entre sus características se encuentran el uso del patrón MVC, el uso de un enrutador de peticiones, un motor de plantillas y un sistema ORM para convertir en objetos los elementos del modelo.

En el proyecto se ha hecho uso de la última versión de Laravel, versión 5.6.

6.1.8 MariaDB

Como toda aplicación web dinámica, es necesario disponer de un sistema que permita la persistencia de datos. MariaDB es un gestor de bases de datos basado en MySQL. Entre sus características está la posibilidad de usar el motor de tablas InnoDB que permite el uso de transacciones y aporta el control sobre las restricciones de clave ajena de manera automática.

¹⁵ <http://jquery.com/>

¹⁶ <https://laravel.com/>

6.2 Entorno de desarrollo

Para poder desarrollar una aplicación web es indispensable tener un servidor web donde alojar la aplicación y que pueda ejecutarla, para ello se ha hecho uso del entorno XAMPP¹⁷ tanto para sistema operativo Windows como para OS X. XAMPP ofrece tanto el servidor web Apache como el sistema gestor de bases de datos MariaDB como un intérprete para el lenguaje de programación PHP.

Además, la instalación es sencilla y rápida permitiendo que en primera instancia se pueda optar por obviar la configuración del servidor para pasar directamente al desarrollo.

Para el desarrollo de la aplicación se han usado dos entornos diferentes, uno con Windows 10 y otro con OS X El Capitán. En los dos entornos se encontraba instalado XAMPP con una copia de la base de datos en cada uno.

6.3 Herramientas utilizadas

Sublime Text 3

Para el desarrollo en sí de la aplicación se ha optado por el IDE Sublime Text 3¹⁸. Sublime Text es un editor de texto y código fuente multiplataforma con una gran cantidad de características que facilitan el desarrollo de software. Además, dispone de una licencia gratuita.

phpMyAdmin

Para la administración y pruebas sobre la base de datos se ha usado la herramienta phpMyAdmin disponible en el sistema XAMPP. Esta herramienta ofrece una manera visual de crear y gestionar bases de datos.

Cabe destacar que para la creación de la base de datos se ha usado el propio framework Laravel, pero en ocasiones se ha usado phpMyAdmin para realizar pruebas con los registros de la base de datos.

Composer

Composer es un manejador de dependencias, gracias a él se permite gestionar los paquetes que usa un proyecto en PHP. Composer permite declarar las librerías que se van a usar y ayuda a la descarga y mantenimiento de estas.

Artisan

Laravel ofrece la herramienta Artisan, esta es una interfaz para la línea de comandos en la que el desarrollador puede dar instrucciones al sistema mediante comandos de

¹⁷ <https://www.apachefriends.org>

¹⁸ <https://www.sublimetext.com/3>



manera que se pueden realizar diferentes tareas de manera rápida y ahorrando en ocasiones la escritura de ciertas partes del código.

BitBucket

Al haber usado dos entornos diferentes en el desarrollo de la aplicación se ha optado por usar un sistema de control de versiones que permitiera compartir la información entre los dos entornos además de ofrecer las características propias de un sistema de control de versiones.

Una de las características más destacables de un sistema de control de versiones es el mantener todas las versiones de los ficheros con los cambios realizados en estos.

El sistema de control de versiones utilizado ha sido BitBucket¹⁹ y como cliente del sistema se ha usado la aplicación GitKraken²⁰.

6.4 Estructura de la aplicación

La estructura de la aplicación se corresponde a la estructura que genera el framework con el que se ha desarrollado la aplicación. Se ha respetado totalmente la estructura que ofrece Laravel.

Laravel distribuye los ficheros dependiendo de su funcionalidad, de manera que siempre estén agrupados los ficheros que tengan una funcionalidad similar. A continuación, se explica el contenido de los directorios en los que se han creado o modificado ficheros durante el desarrollo de la aplicación.

Archivo .env

En este fichero se guarda la configuración que se personaliza en el servidor. Entre esta información se encuentran el nombre de la aplicación, la url, los datos de conexión de la base de datos o los datos del servidor de correo electrónico para las notificaciones por mail a los usuarios.

Archivo routes/web.php

En este archivo se escribirán todas las rutas necesarias para que funcione la aplicación. Estas rutas serán las usadas por el enrutador para redirigir las peticiones al controlador correspondiente en cada caso.

¹⁹ <https://bitbucket.org/product>

²⁰ <https://www.gitkraken.com>

Directorio app

En el directorio app se almacenan las clases que participan en la aplicación web. Clases que pertenecen al propio framework y clases creadas durante el desarrollo de la aplicación.

Entre las clases creadas en el desarrollo se encuentran las clases para usar el sistema ORM que ofrece Eloquent, estas clases permiten la conversión del modelo relacional a objetos del lenguaje de programación. También se encuentran los Trait, mediante los trait se consigue extender una misma funcionalidad a varias clases, pudiendo simular gracias a esto la herencia múltiple, la cual no está implementada en PHP.

Por último, cabe destacar que entre las clases creadas en el desarrollo también se encuentran los controladores, dentro del directorio app/Http, los controladores como ya se ha explicado anteriormente se encargan de recibir las peticiones y comunicarse con el modelo y pasarle a las vistas los datos obtenidos en la comunicación con el modelo.

Directorio config

En este directorio se mantiene la configuración de la aplicación. Los dos ficheros que interesan son app.php y database.php.

En app.php se puede configurar el lenguaje, la zona horaria, los proveedores de plugins usados y los alias de las clases.

En database.php se configura el motor de la base de datos que se va a utilizar.

Directorio database

En el directorio database se pueden encontrar los ficheros que permiten crear e inicializar las tablas de la base de datos. En el directorio factories se pueden escribir funciones que devuelvan objetos ORM con datos falsos, muy útil en la fase de pruebas. En el directorio migrations se pueden crear los archivos que crean y/o modifican las tablas de la base de datos dando opción a realizar un control de versiones sobre ellas. Y en el directorio seeds aloja las funciones que utilizan las factories para incluir los datos falsos en la base de datos.

Directorio public

En este directorio se almacenan los recursos de la aplicación que deben tener acceso público como son las hojas de estilo css, los ficheros js o las imágenes.



Directorio resources

En este directorio se almacenan los ficheros con las traducciones para la internacionalización de la aplicación y también las plantillas que usarán las vistas, estas últimas se encuentran en el directorio views.

7 Pruebas

7.1 Introducción

Las pruebas han estado presentes durante todas las fases de desarrollo de la aplicación. Mediante estas se ha comprobado tanto que la aplicación cumplía los requisitos y especificaciones requeridas y que no existían fallos.

7.2 Pruebas funcionales

Gracias a las pruebas funcionales se consigue comprobar que cada una de las funciones que debe realizar la aplicación se realiza de manera correcta. Cada vez que finalizaba el desarrollo de una de las funciones se comprobaba que el resultado fuera el esperado.

En todos los formularios de inserción o modificación de datos se ha probado a introducir datos no válidos para comprobar cómo responde la aplicación en esos casos. La validación de datos se ha realizado tanto en el cliente mediante JavaScript antes de enviar los datos como en el servidor una vez recibidos los datos.

En el caso de la prevención de inyección de código no se ha realizado ninguna acción especial debido a que el propio framework Laravel por defecto tiene mecanismos para prevenir este tipo de ataques.

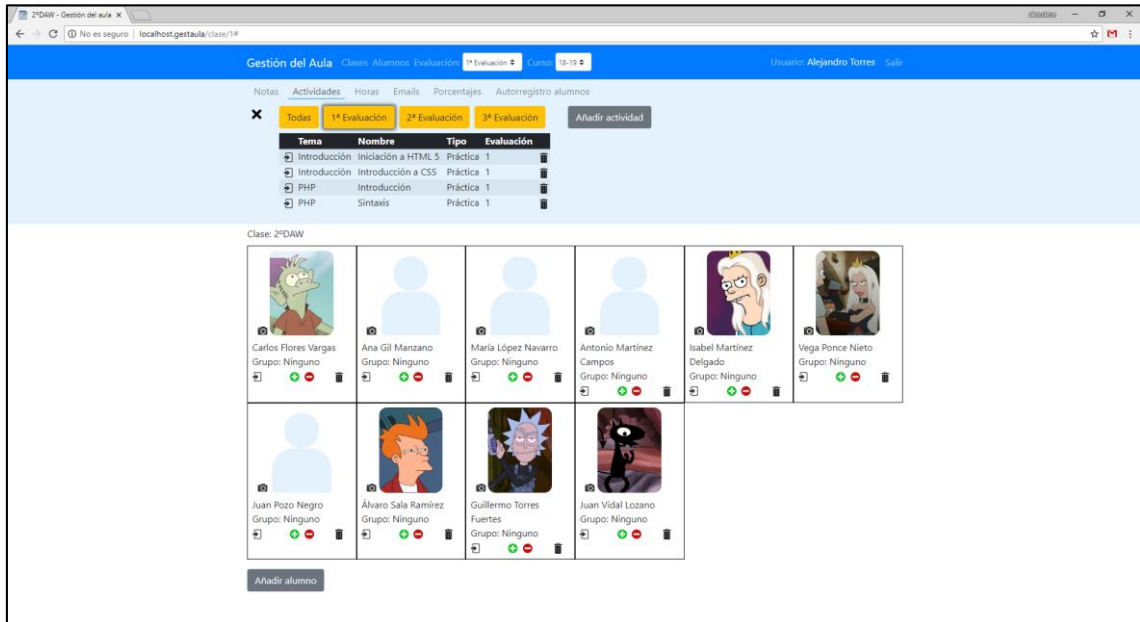
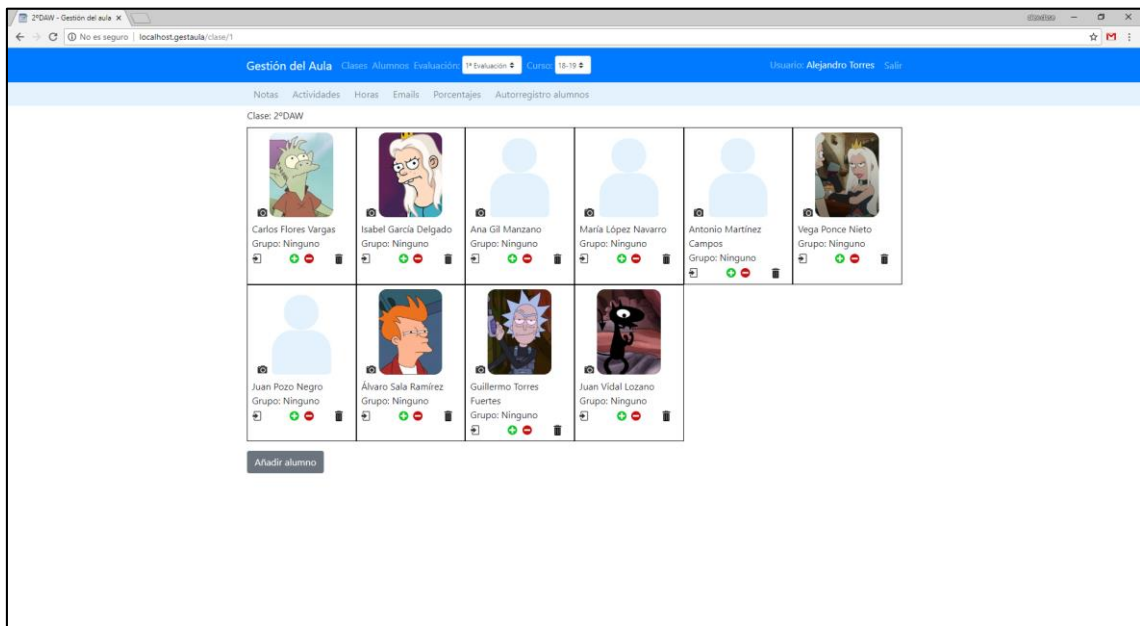
7.3 Pruebas de visualización

Como la aplicación se tiene que poder ejecutar desde ordenadores y dispositivos móviles con diferentes tamaños de pantalla, se han realizado pruebas de visualización con diferentes configuraciones de pantalla para comprobar que la aplicación se adapta correctamente a esos diferentes tamaños.

Las pruebas de visualización se han realizado en 3 tamaños de pantalla y diferentes navegadores web.

Sobremesa

- Google Chrome – resolución: 1920x1080



Alumno: Gestión del Aula | local: localhost:gestaula/alumno

Gestión del Aula | Clase: Alumnos | Evaluación: 11 Evaluación | Curso: 18-19 | Usuario: Alejandro Torres | Salir

Lista de alumnos del profesor: **Alejandro Torres** en el curso 2018-19


DNI	Nombre	1er apellido	2o apellido	Clase	Curso	grupo
11111111H	Carlos	Flores	Vargas	2ºDAW	2 DAW	Ninguno
12345678Z	Ana	Gil	Manzano	2ºDAW	2 DAW	Ninguno
33101849L	María	López	Navarro	2ºDAW	2 DAW	Ninguno
00000000T	Antonio	Martínez	Campos	2ºDAW	2 DAW	Ninguno
66666666Q	Isabel	Martínez	Delgado	2ºDAW	2 DAW	Ninguno
77777777B	Vega	Ponce	Nieto	2ºDAW	2 DAW	Ninguno
87654321X	Juan	Pozo	Negro	2ºDAW	2 DAW	Ninguno
44444444A	Álvaro	Sala	Ramírez	2ºDAW	2 DAW	Ninguno
88888888Y	Guillermo	Torres	Fuertes	2ºDAW	2 DAW	Ninguno
99999999R	Juan	Vidal	Lozano	2ºDAW	2 DAW	Ninguno
98765432M	Marta	Barchino	García	4 ESO	4 ESO	A
33333333P	Saray	Cruz	Esteban	4 ESO	4 ESO	A
55555555K	Carmen	Pons	Montero	4 ESO	4 ESO	A
22222222J	Arturo	Roca	Navarro	4 ESO	4 ESO	B
12344321Z	Carlos	Tarraga	Peña	4 ESO	4 ESO	A

Añadir alumno

Alumno: Isabel Martínez | local: localhost:gestaula/alumno/66

Gestión del Aula | Clase: Alumnos | Evaluación: 11 Evaluación | Curso: 18-19 | Usuario: Alejandro Torres | Salir

Anterior | Siguiendo



Isabel Martínez Delgado

DNI: 66666666Q

Email: igardel@gmail.com

Clase: 2ºDAW

Curso: 2 DAW

Grupo: Ninguno

Actividades | Positivos/Negativos

X Cantidad de positivos del alumno: 3

Razón	Día	Evaluación
Contesta a pregunta	3 septiembre 2018	1
Ejercicio en pizarra	3 septiembre 2018	1
asd asd asdasdasdasd asd asdasd	5 septiembre 2018	1

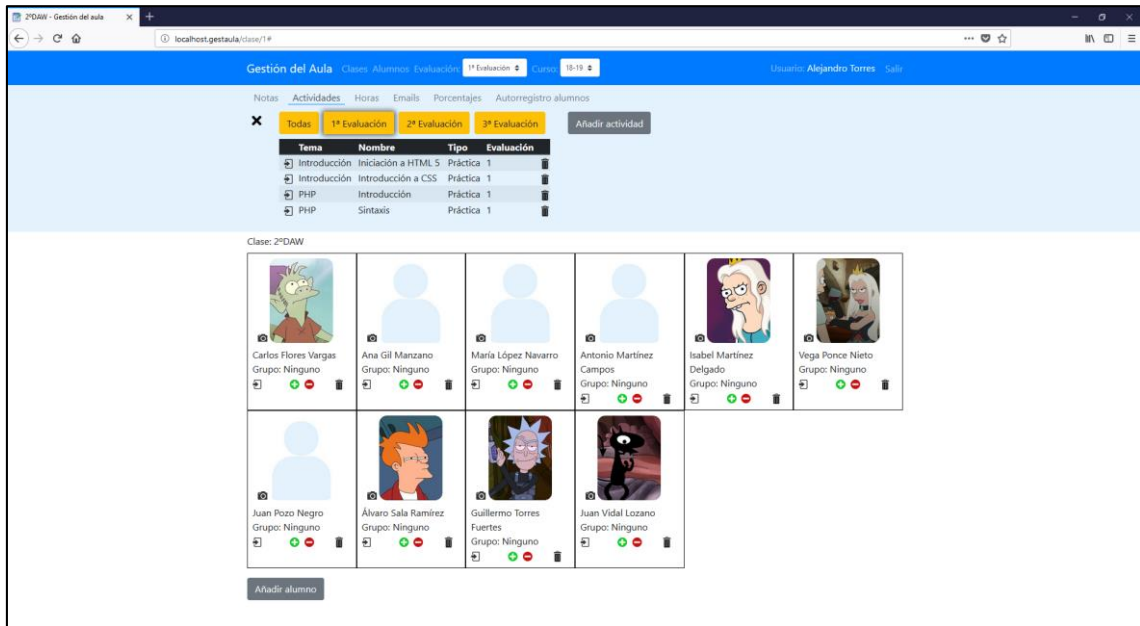
Cantidad de negativos del alumno: 1

Razón	Día	Evaluación
Habla en clase	4 septiembre 2018	1

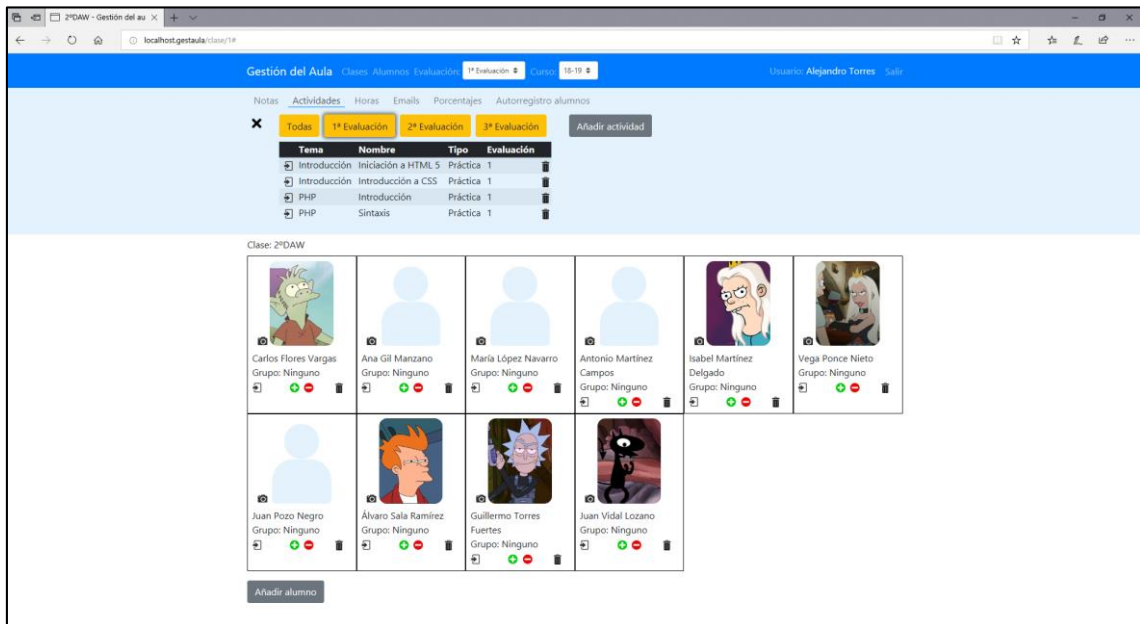


Diseño de un portal web para la gestión de las tareas diarias de un docente de instituto

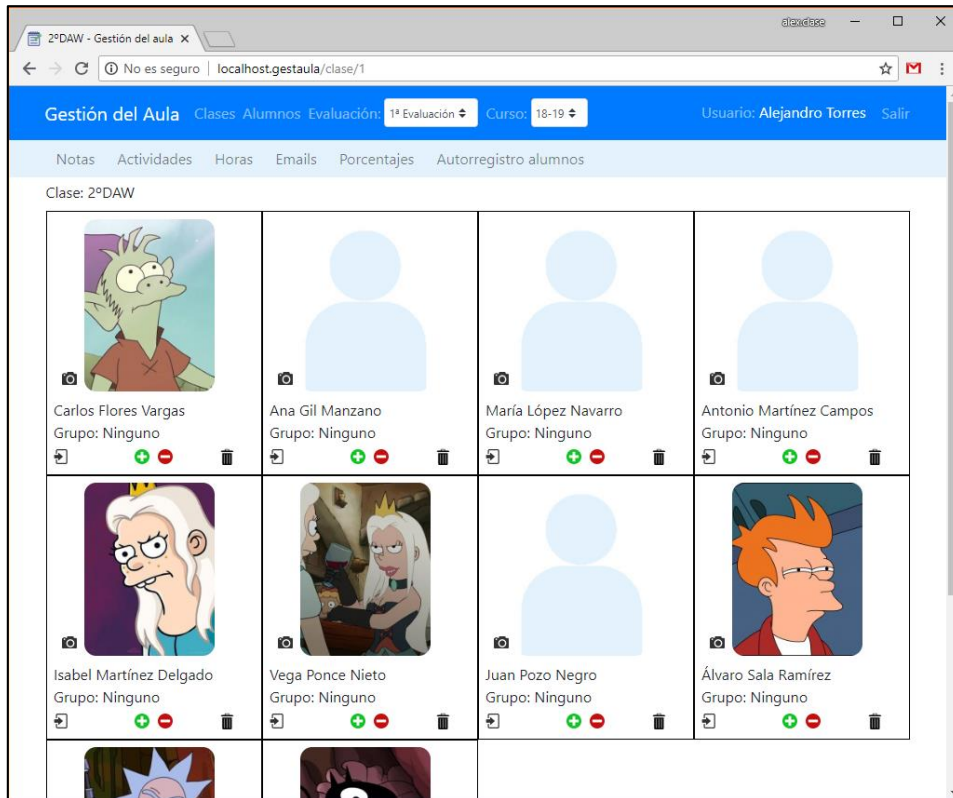
- Mozilla Firefox – resolución: 1920x1080



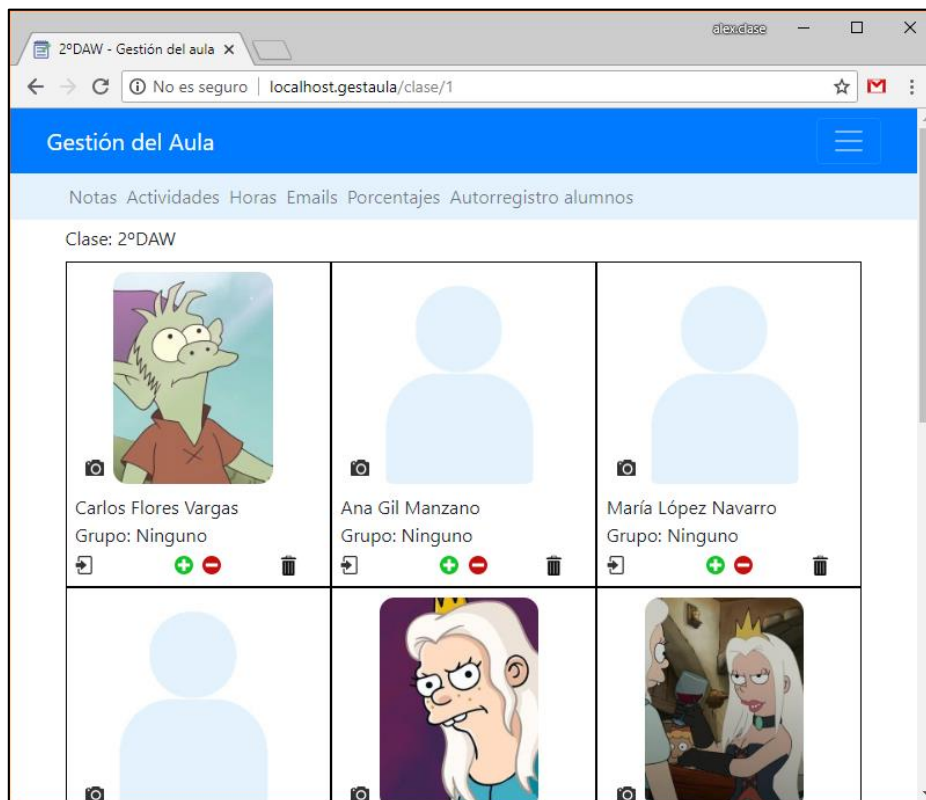
- Microsoft Edge – resolución: 1920x1080



- Google Chrome – resolución: 1024x768



- Google Chrome – resolución: 800x600



Diseño de un portal web para la gestión de las tareas diarias de un docente de instituto

Apple iPad

Gestión del Aula

Notas Actividades Horas Emails Porcentajes Autorregistro alumnos

Clase: 2ºDAW

Grid of student avatars for the 2ºDAW class. Each cell contains an avatar and the student's name and group. The students are: Carlos Flores Vargas, Isabel García Delgado, Ana Gil Manzano, María López Navarro, Antonio Martínez Campos, Vega Ponce Nieto, Juan Pozo Negro, Álvaro Sala Ramírez, and Guillermo Torres Fuertes.

Gestión del Aula

Anterior **Siguiente**

Isabel Martínez Delgado
DNI: 66666666Q
Email: igardel@gmail.com
Clase: 2ºDAW
Curso: 2 DAW
Grupo: Ninguno

Actividades Positivos/Negativos

Todas 1ª Evaluación 2ª Evaluación 3ª Evaluación

Evaluación	Tema	Nombre	Nota	Observaciones	Fotos
1	Práctica	Introducción	Introducción a HTML 5	—	—
1	Práctica	Introducción	Introducción a CSS	—	—
1	Práctica	PHP	Introducción	—	—
1	Práctica	PHP	Sintaxis	—	—

Gestión del Aula

Lista de alumnos del profesor: **Alejandro Torres** en el curso 2018-19

DNI	Nombre	1er apellido	2o apellido	Clase	Curso	grupo
11111111H	Carlos	Flores	Vargas	2ºDAW	2 DAW	Ninguno
66666666Q	Isabel	García	Delgado	2ºDAW	2 DAW	Ninguno
12345678Z	Ana	Gil	Manzano	2ºDAW	2 DAW	Ninguno
33101849L	María	López	Navarro	2ºDAW	2 DAW	Ninguno
00000000T	Antonio	Martínez	Campos	2ºDAW	2 DAW	Ninguno
77777777B	Vega	Ponce	Nieto	2ºDAW	2 DAW	Ninguno
87654321X	Juan	Pozo	Negro	2ºDAW	2 DAW	Ninguno
44444444A	Álvaro	Sala	Ramírez	2ºDAW	2 DAW	Ninguno
88888888Y	Guillermo	Torres	Fuertes	2ºDAW	2 DAW	Ninguno
99999999R	Juan	Vidal	Lozano	2ºDAW	2 DAW	Ninguno
98765432M	Marta	Barchino	García	4 ESO	4 ESO	A
33333333P	Saray	Cruz	Esteban	4 ESO	4 ESO	A
55555555K	Carmen	Pons	Montero	4 ESO	4 ESO	A
22222222J	Arturo	Roca	Navarro	4 ESO	4 ESO	B
12344321Z	Carlos	Tarraga	Peña	4 ESO	4 ESO	A

Añadir alumno

Gestión del Aula

Notas Actividades Horas Emails Porcentajes Autorregistro alumnos

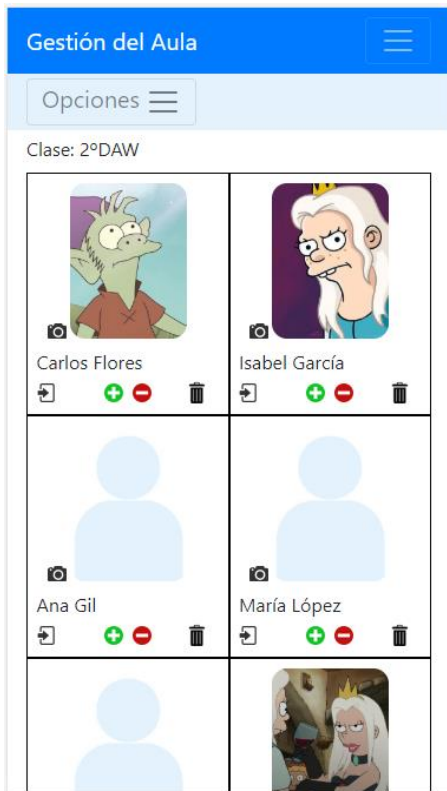
Carlos Flores Vargas <carflor@gmail.com>
Ana Gil Manzano <agilman@gmail.com>
María López Navarro <malona@gmail.com>
Antonio Martínez Campos <anmarcam@gmail.com>
Isabel Martínez Delgado <igardel@gmail.com>
Vega Ponce Nieto <veponni@gmail.com>
Juan Pozo Negro <jupone@gmail.com>
Álvaro Sala Ramírez <alsara@gmail.com>
Guillermo Torres Fuertes <quitofue@gmail.com>
Juan Vidal Lozano <pavilo@gmail.com>

Copia al portapapeles

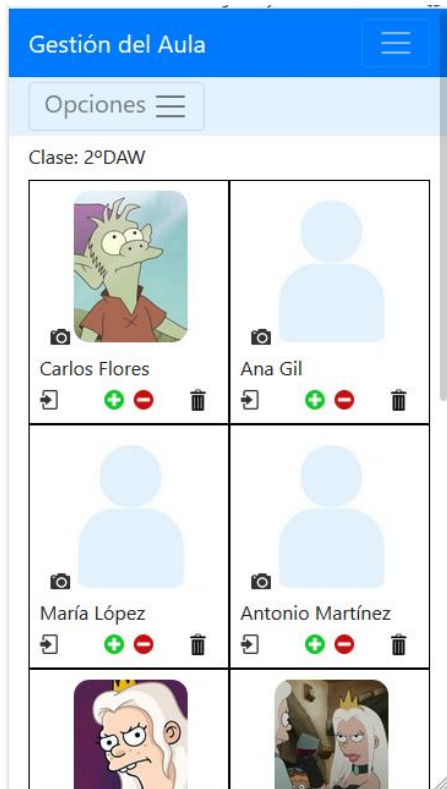
Clase: 2ºDAW

Grid of student avatars for the 2ºDAW class. The students shown are: Carlos Flores Vargas, Ana Gil Manzano, María López Navarro, Antonio Martínez Campos, Isabel Martínez Delgado, and Vega Ponce Nieto.

Apple iPhone 6



Samsung Galaxy S7



7.4 Pruebas de tiempo de ejecución

En las pruebas de tiempo de ejecución se han realizado mediciones tanto en diferentes funciones de código fuente como en la carga de secciones de la aplicación web.

Las mediciones de los tiempos de ejecución de las funciones se han realizado mediante funciones propias del lenguaje PHP. El procedimiento realizado ha sido capturar la marca de tiempo al inicio de la función y al final de la función, de esa manera la resta nos indica el tiempo de ejecución de dicha función. Los tiempos obtenidos han sido despreciables y por ello no se muestran en el presente documento.

Mediante las pruebas de tiempo de ejecución también se ha comprobado si los tiempos de carga de las diferentes secciones de la aplicación web están dentro de unos márgenes aceptables dentro del campo de la usabilidad y no empobrecen la experiencia de usuario cuando se usa la aplicación.

Para realizar estas mediciones se ha hecho uso de las herramientas de depuración que dispone el navegador Google Chrome. Este navegador dispone de una herramienta llamada Network²¹ que calcula el tiempo de carga de una página web. Mediante esta herramienta se han realizado diferentes pruebas sobre las secciones para obtener un tiempo medio de carga aproximado.

A continuación, se indican diferentes tiempos de carga de algunas secciones de la aplicación web:

Sección	Tiempo en ms
Página principal	225 ms
Listado de clases	243 ms
Listado de alumnos	326 ms
Página de clase	389 ms
Página de alumno	346 ms

Aunque estos tiempos se han realizado con la aplicación en pruebas y sin la cantidad de registros de alumnos que podría tener cuando la aplicación esté en uso, se puede observar que los tiempos de carga son aceptables no llegando a 1 segundo.

7.5 Pruebas de uso

Mediante las pruebas de uso se verifica que las funcionalidades y requerimientos que se le piden a la aplicación funcionan correctamente. Para ello se reproducen los casos de uso definidos en este documento.

Estas pruebas se han realizado basándose en los siguientes aspectos:

- Accesibilidad: es fácil acceder a todas las funcionalidades de la aplicación.
- Usabilidad: la aplicación está diseñada para ser usable.
- Eficiencia: se pueden realizar las acciones con el número mínimo de pasos.
- Eficacia: la aplicación hace lo que tiene que hacer.

²¹ <https://developers.google.com/web/tools/chrome-devtools/network-performance>

A continuación, se muestran las pruebas de uso realizadas a los casos de uso más representativos.

7.5.1 Añadir clase

Flujo Normal

Con la sesión iniciada, se redirige a la página de lista de clases del usuario. Se pulsa el botón “Añadir clase”.

Nombre	Descripción	Exámenes	Actividades	Actitud	Total	Duración (mins)
1ºASIR	Gestión de bases de datos	50%	40%	10%	100%	50
2ºDAW	Desarrollo web en entorno servidor	55%	35%	10%	100%	50
4 ESO	4 eso grupos a y b	60%	30%	10%	100%	50

Aparece el formulario para introducir clase y se introducen los datos.

Curso: 2018-19

Nombre: 1ºDAW

Descripción: Entornos de desarrollo

% Exámenes: 40

% Actividades: 40

% Actitud: 20

Total: 100%

Duración (mins): 55

Curso: 18-19

Guardar Borrar Formulario Cerrar

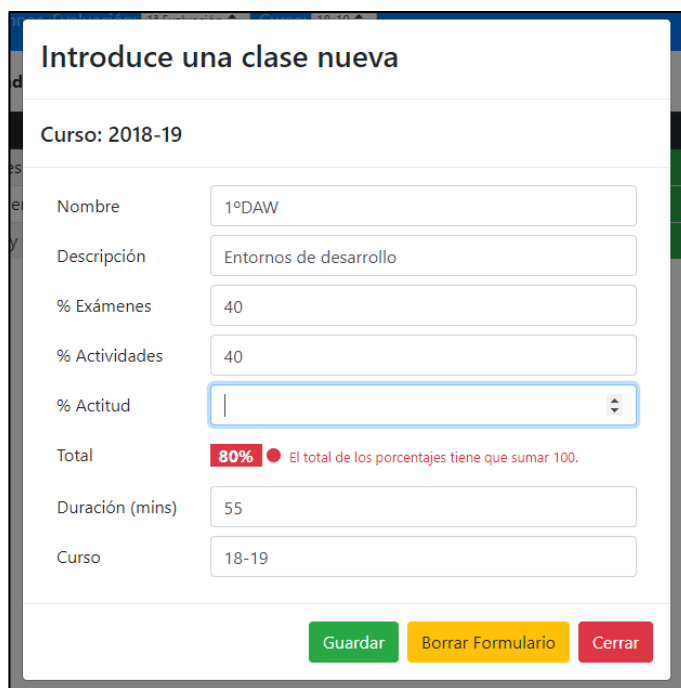
Con los datos validados se pulsa el botón “Guardar”.

Nombre	Descripción	Exámenes	Actividades	Actitud	Total	Duración (mins)
1ºASIR	Gestión de bases de datos	50%	40%	10%	100%	50
1ºDAW	Entornos de desarrollo	40%	40%	20%	100%	50
2ºDAW	Desarrollo web en entorno servidor	55%	35%	10%	100%	50
4 ESO	4 eso grupos a y b	60%	30%	10%	100%	50

La clase se añade a la lista de clases.

Excepciones

Si ocurre un error en el formulario se informa en el propio formulario.



Introduce una clase nueva

Curso: 2018-19

Nombre: 1ºDAW

Descripción: Entornos de desarrollo

% Exámenes: 40

% Actividades: 40

% Actitud: |

Total: **80%** El total de los porcentajes tiene que sumar 100.

Duración (mins): 55

Curso: 18-19

Guardar Borrar Formulario Cerrar

7.5.2 Ver clase

Flujo normal

En el listado de clases se hace clic al enlace para entrar a una clase.



	Nombre	Descripción
	1ºASIR	Gestio...
	1ºDAW	Entor...
	2ºDAW	Desar...
	4 ESO	4 eso...



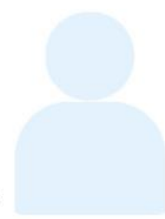
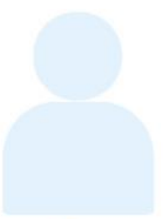
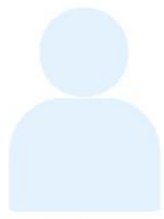





Añadir clase

Se carga la página correspondiente a la clase.

Gestión del Aula Clases Alumnos Evaluación: 1ª Evaluación Curso: 18-19 Usuario: Alejandro Torres Salir

Notas Actividades Horas Emails Porcentajes Autorregistro alumnos


Clase: 2ºDAW

 Carlos Flores Vargas Grupo: Ninguno	 Isabel García Delgado Grupo: Ninguno	 Ana Gil Manzano Grupo: Ninguno	 María López Navarro Grupo: Ninguno
 Antonio Martínez Campos Grupo: Ninguno	 Vega Ponce Nieto Grupo: Ninguno	 Juan Pozo Negro Grupo: Ninguno	 Álvaro Sala Ramírez Grupo: Ninguno
			

Excepciones

Si la clase no existe o no pertenece al usuario se muestra la página de error.

Gestión del Aula Clases Alumnos Evaluación: 1ª Evaluación Curso: 18-19 Usuario: Alejandro Torres Salir

 Lo sentimos, la sección a la que intenta acceder no existe.

Puede acceder a cualquiera de sus clases desde la siguiente lista:

Clases	
18-19	2ºDAW
	1ºASIR
	4 ESO
	1ºDAW

7.5.3 Añadir alumno

Flujo normal

En la página de la clase o en la lista de alumnos se pulsa el botón “Añadir alumno”

Gestión del Aula Clases Alumnos Evaluación: 1ª Evaluación Curso: 18-19 Usuario: Alejandro Torres Salir

Lista de alumnos del profesor: Alejandro Torres en el curso 2018-19

DNI	Nombre	1er apellido	2o apellido	Clase	Curso	grupo	
11111111H	Carlos	Flores	Vargas	2ºDAW	2 DAW	Ninguno	🗑
12345678Z	Ana	Gil	Manzano	2ºDAW	2 DAW	Ninguno	🗑
33101849L	María	López	Navarro	2ºDAW	2 DAW	Ninguno	🗑
00000000T	Antonio	Martínez	Campos	2ºDAW	2 DAW	Ninguno	🗑
66666666Q	Isabel	Martínez	Delgado	2ºDAW	2 DAW	Ninguno	🗑
77777777B	Vega	Ponce	Nieto	2ºDAW	2 DAW	Ninguno	🗑
87654321X	Juan	Pozo	Negro	2ºDAW	2 DAW	Ninguno	🗑
44444444A	Álvaro	Sala	Ramírez	2ºDAW	2 DAW	Ninguno	🗑
22577706D	Alejandro	Torres	-	2ºDAW	2 DAW	Ninguno	🗑
88888888Y	Guillermo	Torres	Fuertes	2ºDAW	2 DAW	Ninguno	🗑
99999999R	Juan	Vidal	Lozano	2ºDAW	2 DAW	Ninguno	🗑
98765432M	Marta	Barchino	García	4 ESO	4 ESO	A	🗑
33333333P	Saray	Cruz	Esteban	4 ESO	4 ESO	A	🗑
55555555K	Carmen	Pons	Montero	4 ESO	4 ESO	A	🗑
22222222J	Arturo	Roca	Navarro	4 ESO	4 ESO	B	🗑
12344321Z	Carlos	Tarraga	Peña	4 ESO	4 ESO	A	🗑

Añadir alumno

Aparece el formulario para añadir alumno y se introducen los datos.

Introduce un alumno nuevo

DNI/NIE: 33456369V

Nombre: María Del Mar

1er Apellido: Martí

2º Apellido: Agustí

No tiene 2º apellido

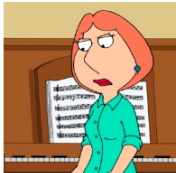
Email: mardelmar@gmail.com

Curso: 4 ESO

Grupo: B

Clase: 4 ESO

Foto: p-family-g...rstein.jpg



Con los datos validados se pulsa el botón guardar

33333333P	Saray	Cruz	Esteban	4 ESO	4 ESO	A	
33456369V	María Del Mar	Martí	Agustí	4 ESO	4 ESO	B	
55555555K	Carmen	Pons	Montero	4 ESO	4 ESO	A	

El alumno se añade a la lista de alumnos

Excepciones

Si ocurren errores en los campos del formulario se mostrarán.

Introduce un alumno nuevo ✕

DNI/NIE

● DNI/NIE con formato erróneo:
DNI: 12345678A
NIE: A1234567B

Nombre

● El nombre ha de tener al menos 2 caracteres.

1er Apellido

● El 1er apellido ha de tener al menos 2 caracteres.

2º Apellido

No tiene 2º apellido

● El 2º apellido ha de tener al menos 2 caracteres.

Email

● El email debe seguir el formato: usuario@dominio.com/es

Curso

● Se tiene que seleccionar Curso

Grupo

Clase

● Se tiene que seleccionar Clase

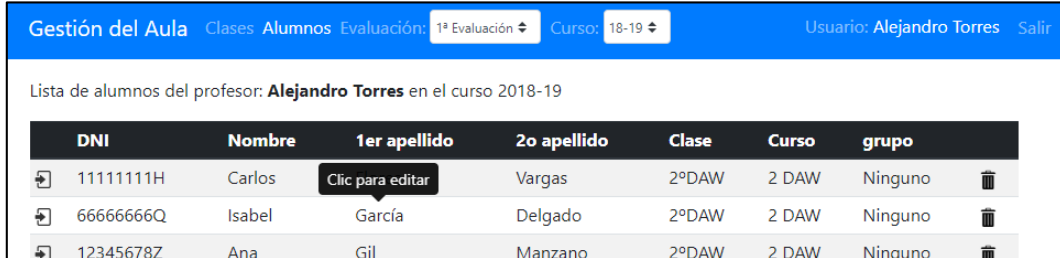
Foto Ningún archiv...seleccionado

Guardar
Borrar Formulario
Cerrar

7.5.4 Modificar los datos de un alumno

Flujo normal

En la lista de alumnos se hace clic en un dato del alumno.

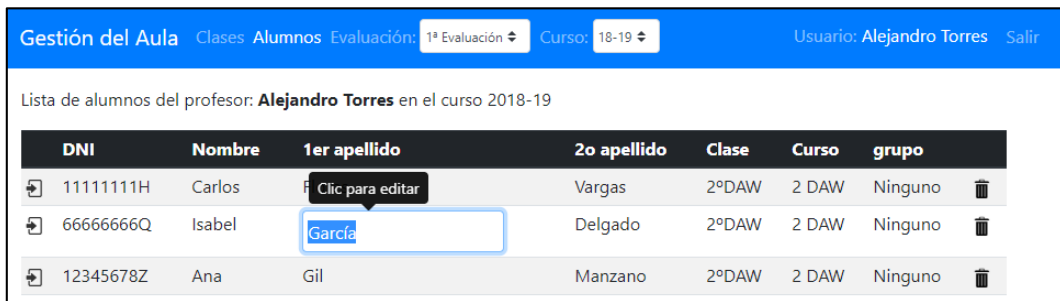


Gestión del Aula Clases Alumnos Evaluación: 1ª Evaluación Curso: 18-19 Usuario: Alejandro Torres Salir

Lista de alumnos del profesor: **Alejandro Torres** en el curso 2018-19

DNI	Nombre	1er apellido	2o apellido	Clase	Curso	grupo
11111111H	Carlos	Clic para editar	Vargas	2ºDAW	2 DAW	Ninguno
66666666Q	Isabel	García	Delgado	2ºDAW	2 DAW	Ninguno
12345678Z	Ana	Gil	Manzano	2ºDAW	2 DAW	Ninguno

El dato se convierte en un campo de formulario.



Gestión del Aula Clases Alumnos Evaluación: 1ª Evaluación Curso: 18-19 Usuario: Alejandro Torres Salir

Lista de alumnos del profesor: **Alejandro Torres** en el curso 2018-19

DNI	Nombre	1er apellido	2o apellido	Clase	Curso	grupo
11111111H	Carlos	Clic para editar	Vargas	2ºDAW	2 DAW	Ninguno
66666666Q	Isabel	García	Delgado	2ºDAW	2 DAW	Ninguno
12345678Z	Ana	Gil	Manzano	2ºDAW	2 DAW	Ninguno

Se introduce el dato nuevo y se pulsa Enter o Tabulador o se hace clic fuera del campo de formulario y se guarda el dato nuevo.



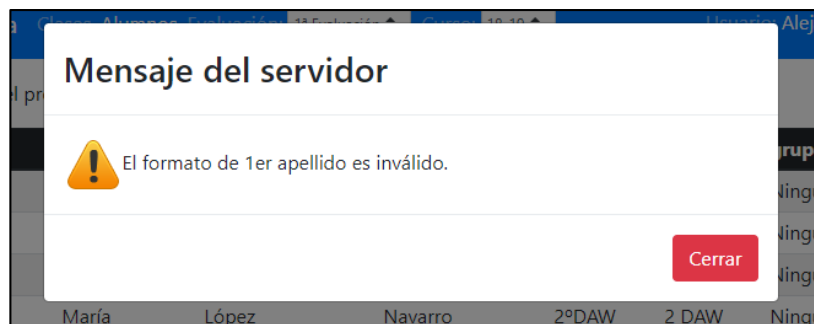
Gestión del Aula Clases Alumnos Evaluación: 1ª Evaluación Curso: 18-19 Usuario: Alejandro Torres Salir

Lista de alumnos del profesor: **Alejandro Torres** en el curso 2018-19

DNI	Nombre	1er apellido	2o apellido	Clase	Curso	grupo
11111111H	Carlos	Flores	Vargas	2ºDAW	2 DAW	Ninguno
66666666Q	Isabel	Martínez	Delgado	2ºDAW	2 DAW	Ninguno
12345678Z	Ana	Gil	Manzano	2ºDAW	2 DAW	Ninguno

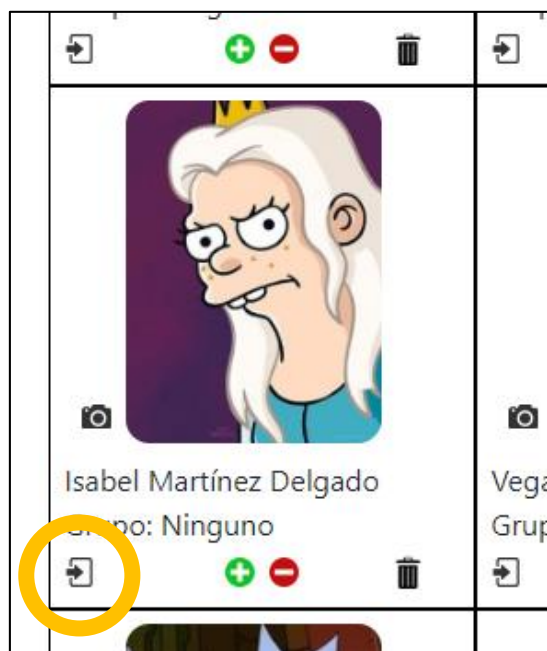
Excepciones

Si se produce un error cambiando el dato se informa al usuario

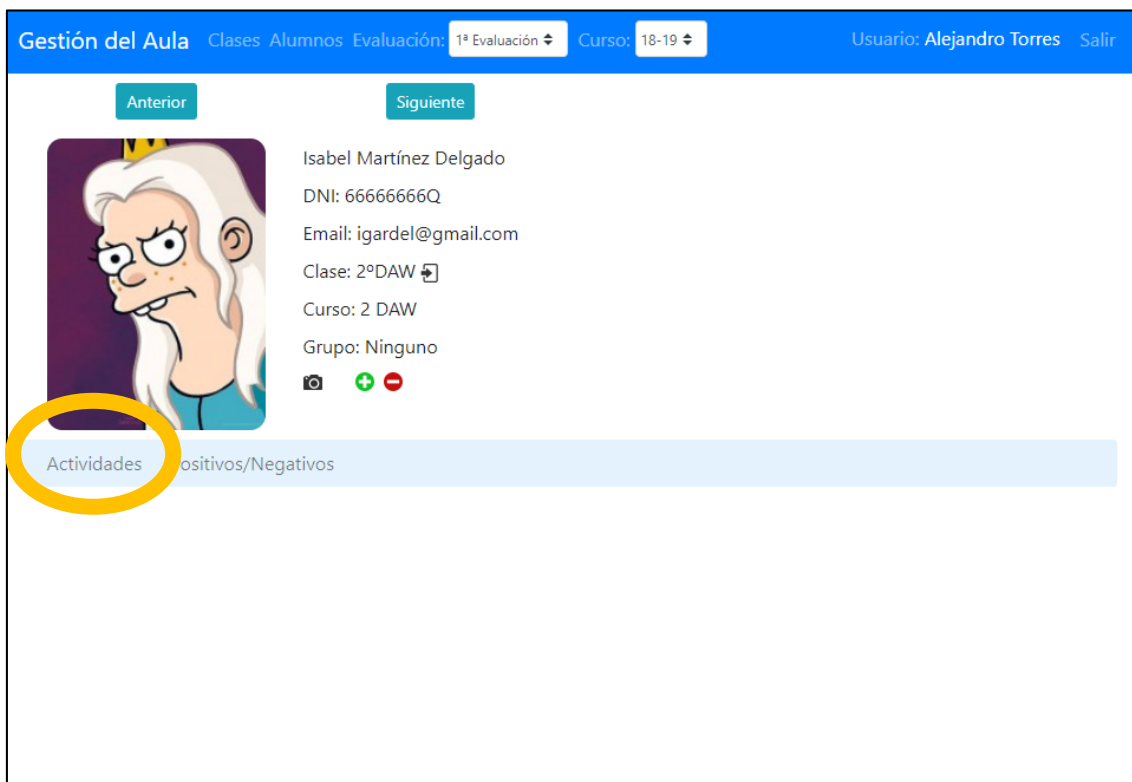


7.5.5 Consultar las actividades que tiene que realizar un alumno

En la página de la clase se pulsa el botón de entrar a la página de un alumno.




Se muestra la página del alumno. Se pulsa sobre la opción "Actividades".



Se muestran las actividades a realizar por el alumno.

Gestión del Aula Clases Alumnos Evaluación: 1ª Evaluación Curso: 18-19 Usuario: Alejandro Torres Salir

Anterior Siguiente

 Isabel Martínez Delgado
DNI: 66666666Q
Email: igardel@gmail.com
Clase: 2ºDAW
Curso: 2 DAW
Grupo: Ninguno

Actividades Positivos/Negativos

×

Todas 1ª Evaluación 2ª Evaluación 3ª Evaluación

Evaluación	Tipo	Tema	Nombre	Nota	Observaciones	Fotos
1	Práctica	Introducción	Iniciación a HTML 5	—	—	
1	Práctica	Introducción	Introducción a CSS	—	—	
1	Práctica	PHP	Introducción	—	—	
1	Práctica	PHP	Sintaxis	—	—	

7.5.6 Consultar las calificaciones obtenidas por todos los alumnos de una clase en las actividades a realizar


Flujo normal

En la página de la clase se pulsa sobre el enlace “Notas” de la barra de opciones.

Gestión del Aula Clases Alumnos Evaluación: 1ª Evaluación Curso: 18-19 Usuario: Alejandro Torres Salir

Notas Actividades Horas Emails Porcentajes Autorregistro alumnos

2ºDAW



Se muestra la página con todas las notas de todas las actividades de la clase.

Gestión del Aula					Clases		Alumnos		Evaluación: 1ª Evaluación		Curso: 18-19		Usuario: Alejandro Torres		Salir	
Notas de la clase: 2ºDAW																
1ª evaluación																
Alumno				Introducción Iniciación a HTML 5		Introducción Introducción a CSS		PHP Introducción		PHP Sintaxis						
+	Carlos	Flores	Vargas	8		—		—		—						
+	Ana	Gil	Manzano	6		—		—		—						
+	María	López	Navarro	4		—		—		—						
+	Antonio	Martínez	Campos	7.5		—		—		—						
+	Isabel	Martínez	Delgado	—		—		—		—						
+	Vega	Ponce	Nieto	—		5.7		—		—						
+	Juan	Pozo	Negro	—		—		—		—						
+	Álvaro	Sala	Ramírez	—		9		—		—						
+	Alejandro	Torres		—		—		—		—						
+	Guillermo	Torres	Fuertes	—		—		—		—						
+	Juan	Vidal	Lozano	—		—		—		—						

7.5.7 Añadir un positivo/negativo a un alumno

Flujo normal

En la página de una clase o de un alumno se pulsa el botón “Positivo”.

Anterior Siguiete



Isabel Martínez Delgado
DNI: 66666666Q
Email: igardel@gmail.com
Clase: 2ºDAW
Curso: 2ºDAW
Grupo: Ninguno

Actividades Positivos/Negativos

Aparece el formulario para introducir un positivo y se introducen los datos.

Poner positivo +

Isabel Martínez Delgado


Razón:

Guardar **Cancelar**

Una vez los datos están validados se pulsa el botón guardar.

Gestión del Aula Clases Alumnos Evaluación: 1ª Evaluación Curso: 18-19

Anterior Siguiete



Isabel Martínez Delgado
DNI: 66666666Q
Email: igardel@gmail.com
Clase: 2ºDAW
Curso: 2 DAW
Grupo: Ninguno

Actividades Positivos/Negativos

X Cantidad de positivos del alumno: 3

Razón	Día	Evaluación
Contesta a pregunta	3 septiembre 2018	1
Ejercicio en pizarra	3 septiembre 2018	1
Participa en la pizarra	8 septiembre 2018	1

Este alumno no tiene ningún negativo.

El positivo se añade a la lista de positivos del alumnos.

Excepciones

Si ocurre un error en el formulario se muestra.

Poner positivo +

Isabel Martínez Delgado

Razón

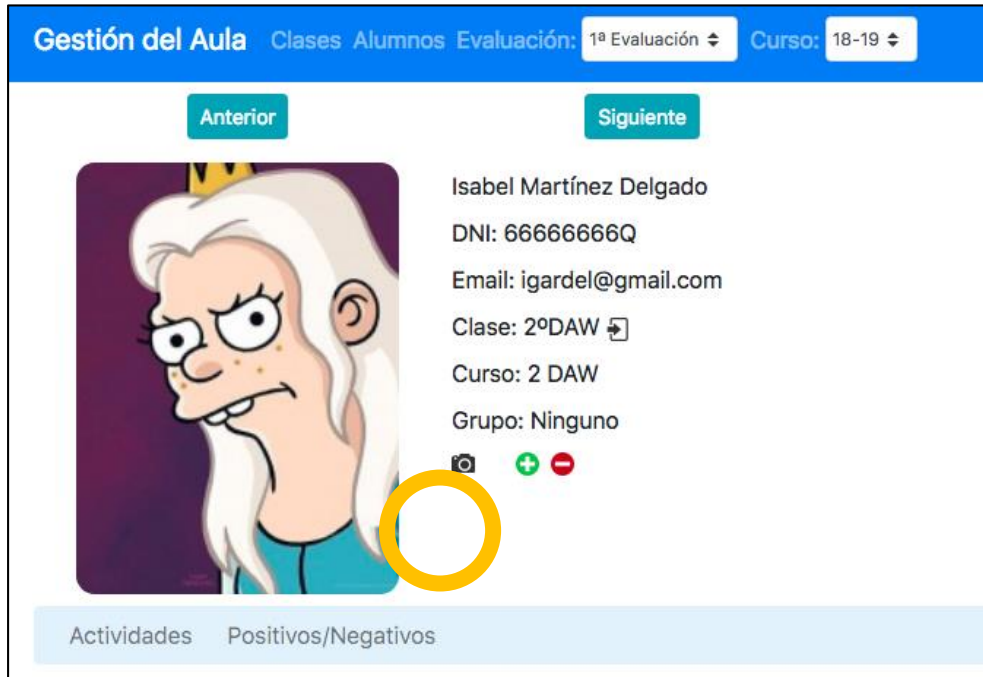
● La razón ha de tener al menos 5 caracteres.

Guardar Cancelar

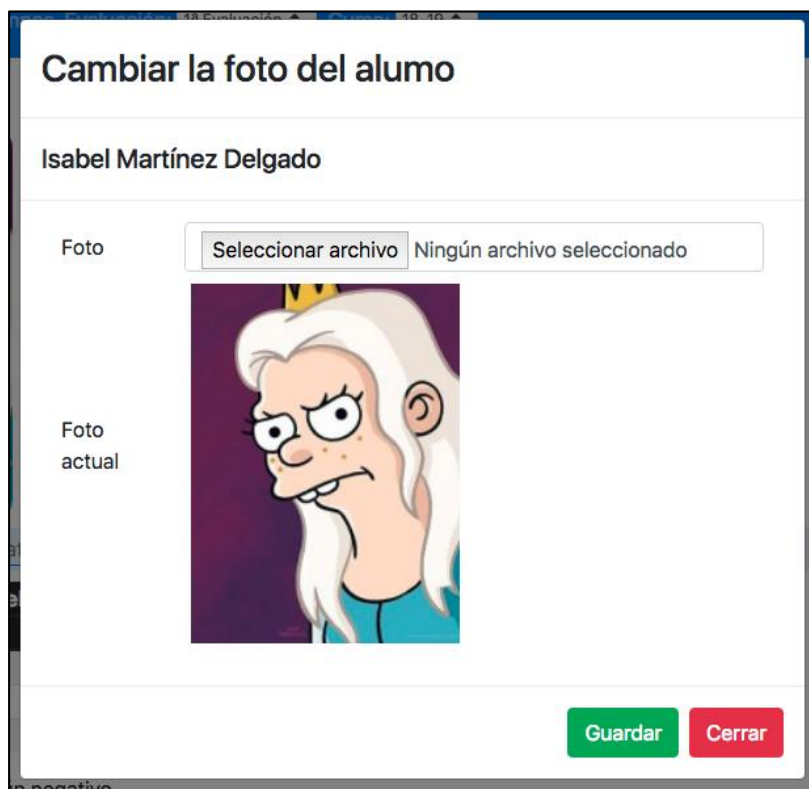
7.5.8 Modificar la foto de un alumno

Flujo normal

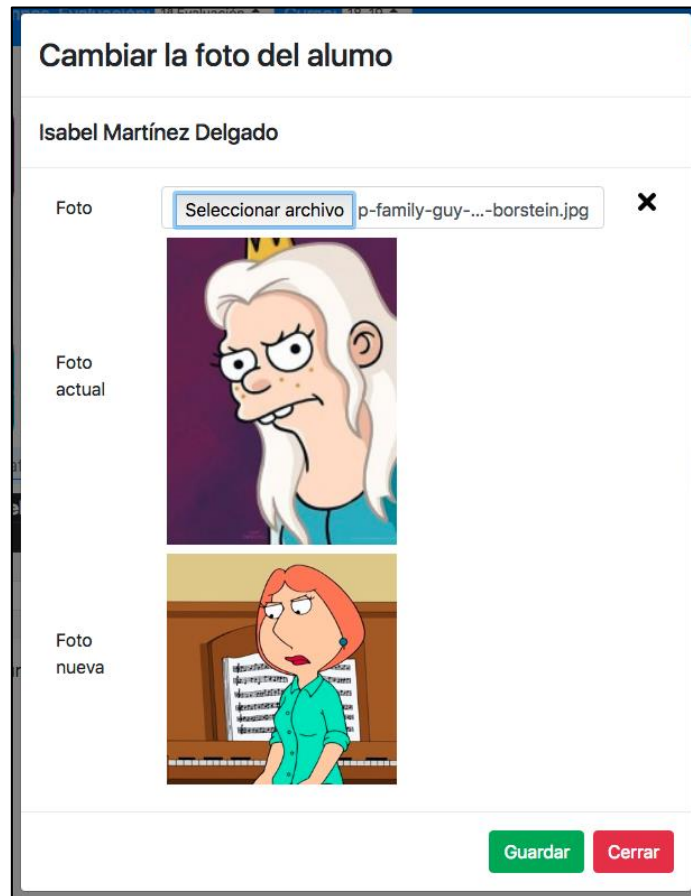
Se pulsa el icono de la cámara correspondiente al alumno.



Aparece el formulario de cambio de foto.



Se selecciona una foto nueva.



Con la foto nueva seleccionada se pulsa el botón Guardar.



La foto del alumno se actualiza.

Excepciones

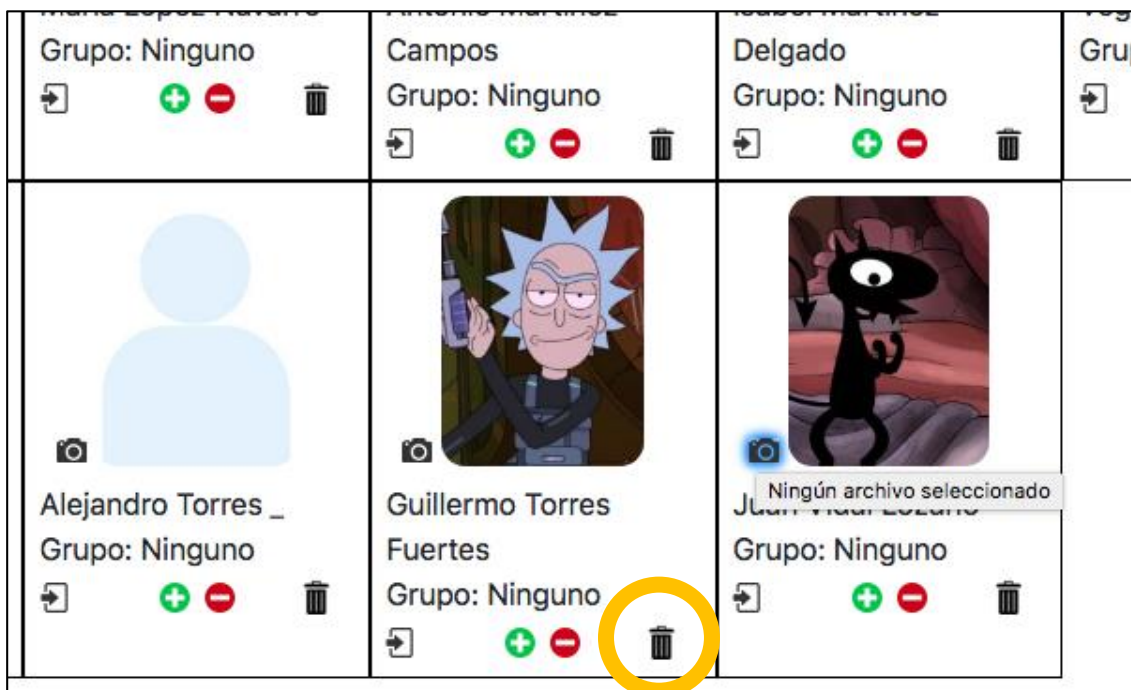
Si ocurre un error en el formulario se muestra.



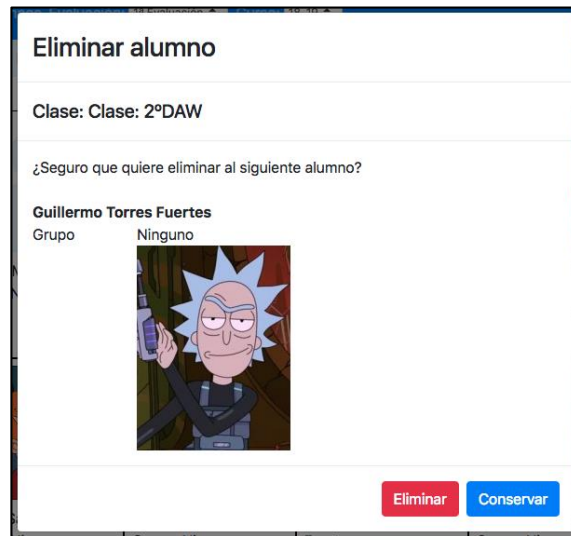
7.5.9 Eliminar un alumno

Flujo normal

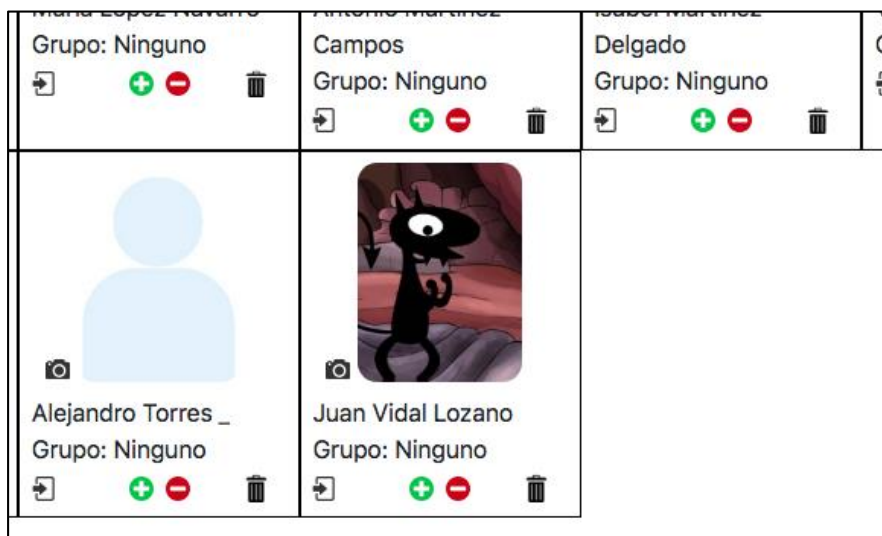
Se pulsa sobre el icono de la papelera correspondiente a un alumno desde la liasta de alumnos o la página de clase.



Aparece una ventana modal informando de la acción a realizar.



Desaparece el alumno de la lista de alumnos.



Excepciones

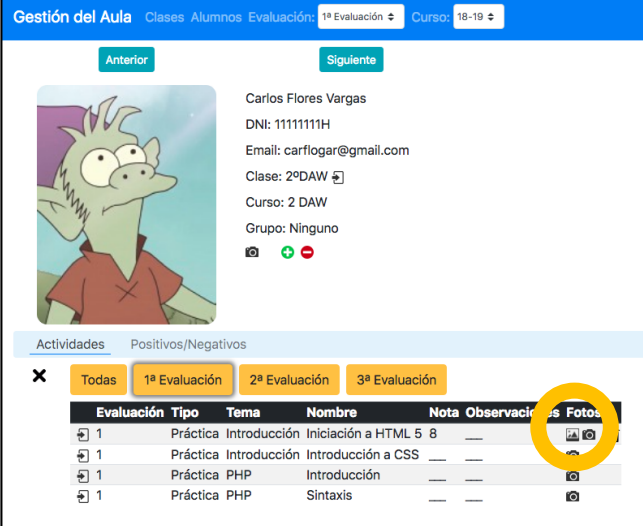
Si ocurre un error al borrar al alumno se muestra.



7.5.10 Mostrar las fotos de un alumno registradas para una actividad.

Flujo normal

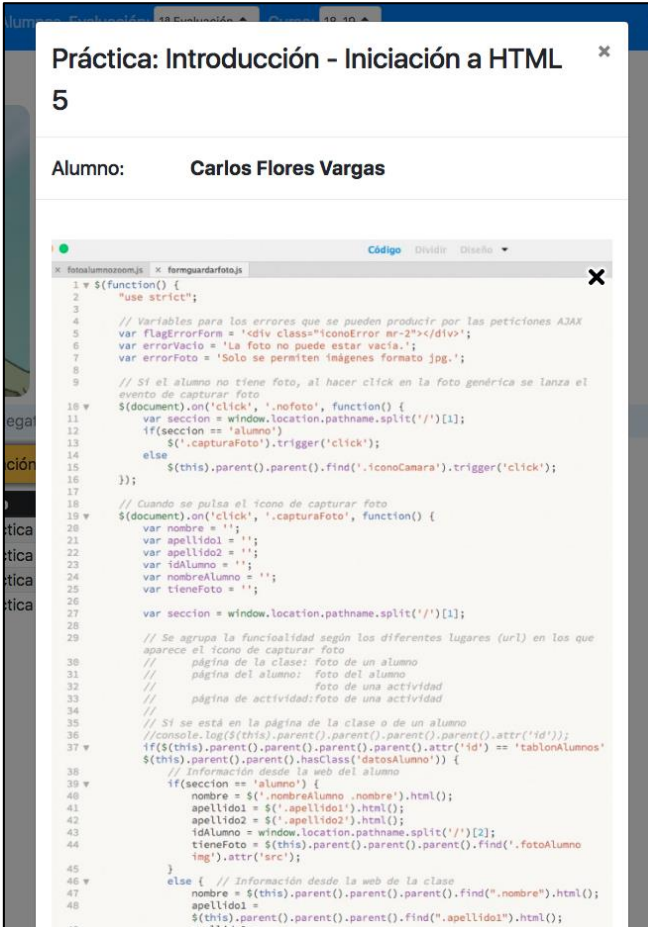
En la lista de calificaciones se pulsa sobre el icono de fotografía correspondiente al alumno.



The screenshot shows the 'Gestión del Aula' interface. At the top, there are navigation buttons for 'Anterior' and 'Siguinte'. Below that is a student profile for Carlos Flores Vargas, including his DNI, email, class, course, and group. A table below lists activities with columns for 'Evaluación', 'Tipo', 'Tema', 'Nombre', 'Nota', 'Observaciones', and 'Fotos'. A yellow circle highlights the camera icon in the 'Fotos' column for the first activity.

Evaluación	Tipo	Tema	Nombre	Nota	Observaciones	Fotos
1	Práctica	Introducción	Iniciación a HTML 5	8	---	
1	Práctica	Introducción	Introducción a CSS	---	---	
1	Práctica	PHP	Introducción	---	---	
1	Práctica	PHP	Sintaxis	---	---	

Aparece una ventana modal con las imágenes guardadas de esa actividad.



The screenshot shows a modal window titled 'Práctica: Introducción - Iniciación a HTML 5'. It displays the student's name 'Carlos Flores Vargas' and a code editor with JavaScript code for handling photo uploads. The code includes variables for error messages and functions for capturing photos and saving them to the database.

```
1 <script>$(function() {
2     "use strict";
3
4     // Variables para los errores que se pueden producir por las peticiones AJAX
5     var flagErrorForm = '<div class="iconError mr-2"></div>';
6     var errorVacio = 'La foto no puede estar vacía.';
7     var errorFoto = 'Solo se permiten imágenes formato jpg.';
8
9     // Si el alumno no tiene foto, al hacer click en la foto genérica se lanza el
10    evento de capturar foto
11    $(document).on('click', '.nofoto', function() {
12        if(seccion == 'alumno')
13            $('<div class="capturaFoto">').trigger('click');
14        else
15            $(this).parent().parent().find('<div class="iconoCamara">').trigger('click');
16    });
17
18    // Cuando se pulsa el icono de capturar foto
19    $(document).on('click', '.capturaFoto', function() {
20        var nombre = '';
21        var apellido1 = '';
22        var apellido2 = '';
23        var idAlumno = '';
24        var nombreAlumno = '';
25        var tieneFoto = '';
26
27        var seccion = window.location.pathname.split('/')[1];
28
29        // Se agrupa la funcionalidad según los diferentes lugares (url) en los que
30        // aparece el icono de capturar foto
31        // página de la clase: foto de un alumno
32        // página del alumno: foto del alumno
33        // foto de una actividad
34        // página de actividad: foto de una actividad
35
36        // Si se está en la página de la clase o de un alumno
37        //console.log($(this).parent().parent().parent().parent().attr('id'));
38        if($(this).parent().parent().parent().parent().attr('id') == 'tablonAlumnos')
39            $(this).parent().parent().hasClass('datosAlumno') {
40                // Información desde la web del alumno
41                if(seccion == 'alumno') {
42                    nombre = $('<div class="nombreAlumno">').html();
43                    apellido1 = $('<div class="apellido1">').html();
44                    apellido2 = $('<div class="apellido2">').html();
45                    idAlumno = window.location.pathname.split('/')[2];
46                    tieneFoto = $(this).parent().parent().parent().find('<div class="fotoAlumno">').attr('src');
47                }
48                // Información desde la web de la clase
49                nombre = $(this).parent().parent().parent().find("<div class="nombre">").html();
50                apellido1 = $(this).parent().parent().parent().find("<div class="apellido1">").html();
51                apellido2 = $(this).parent().parent().parent().find("<div class="apellido2">").html();
52            }
53    });
54    </script>
```


8 Conclusiones

Este proyecto final de grado ha permitido el desarrollo de una aplicación web que permite a un profesor de secundaria realizar las gestiones de las tareas diarias en la tarea docente, como son el registro de grupos, alumnos, actividades y el registro de las calificaciones obtenidas por los alumnos.

La aplicación permite su uso desde cualquier dispositivo conectado a internet y que tenga instalado un navegador web independientemente de su resolución de pantalla gracias a que el diseño de la interfaz gráfica se ha realizado para que sea un diseño responsive.

La interfaz gráfica además de adaptarse a las diferentes resoluciones de pantalla, se ha diseñado con un aspecto visual sencillo que permite que cualquier usuario sea capaz de utilizar la aplicación independientemente de su experiencia navegando por internet.

Debido a que la aplicación está destinada a un público objetivo muy determinado, los profesores de secundaria, se han desarrollado las funciones con respecto a sus necesidades por lo que la aplicación ha resultado tener un alto nivel de usabilidad.

En cuanto al diseño elegido, se ha optado por el patrón Modelo-Vista-Controlador sustentado por el framework Laravel que ha permitido el diseño en tres capas, de esta manera se ha desarrollado una aplicación fácilmente adaptable a la rápida evolución que hoy en día tienen las tecnologías.

Durante el desarrollo del proyecto han surgido algunos inconvenientes que se han solventado sin dificultad. El inconveniente más destacable ha sido que se pretendía usar una librería para jQuery que permitiera la edición de los datos en el mismo lugar donde aparece el dato (edit inPlace), la librería jEditable [14].

El problema con esta librería surgió debido a que se encuentra sin mantenimiento desde hace unos años y no es compatible con la versión 4 de Bootstrap, usada en el proyecto. La primera solución fue buscar otra librería con la misma funcionalidad, encontrando así la librería X-editable [15] que resultó tener el mismo problema de compatibilidad por la misma razón que jEditable.

Finalmente, como solución, se optó por desarrollar una función propia que realizara la funcionalidad requerida de manera que al hacer clic en la información que se pueda editar, esa información se convierta en un campo de formulario del tipo adecuado mediante el que se pueda modificar la información directamente.

Relación con los estudios cursados.

Gracias a la realización de este trabajo final de grado se han reforzado conocimientos adquiridos durante la realización primero de la Ingeniería Técnica en Informática de Sistemas, y posteriormente, del curso de adaptación al grado. De este último cabe destacar que han sido de utilidad los contenidos de las asignaturas:

- Interfaces Persona Computador (11556).
- Desarrollo centrado en el usuario (11613).
- Gestión de proyectos (11554).

Además, se han adquirido nuevos conocimientos gracias al estudio previo realizado de las tecnologías utilizadas en la implementación del proyecto como son Laravel, Eloquent y Bootstrap.

Por último, indicar que en la realización del trabajo final de grado se ha requerido el uso de las competencias transversales adquiridas en el curso de adaptación al grado. Entre todas cabe destacar las siguientes:

- CT_02 - Aplicación y pensamiento práctico. Gracias a esta competencia se ha elegido la estrategia adecuada a la hora de aplicar los conocimientos adquiridos y seguir el camino correcto para alcanzar los objetivos marcados en el desarrollo de la aplicación.
- CT_03 - Análisis y resolución de problemas. Muy importante a la hora de analizar el proyecto a desarrollar y conseguir definir elementos que constituyen la aplicación.
- CT_05 - Diseño y proyecto. Posiblemente una de las competencias más usadas debido al análisis realizado de la aplicación para obtener todos los requisitos, funcionalidades y actores que participan en la aplicación.
- CT_10 - Conocimiento de problemas contemporáneos. Gracias a esta competencia se ha sabido analizar las aplicaciones existentes en el mercado, así como las necesidades propias del usuario objetivo de la aplicación para poder desarrollar una aplicación que cumpla con las funcionalidades requeridas.
- CT_12 - Planificación y gestión del tiempo. Esta competencia ha sido indispensable para poder aprovechar y gestionar el tiempo de realización del proyecto en el periodo establecido.

9 Glosario

HTML: HyperText Markup Language. Lenguaje de marcado de hipertexto, es el lenguaje básico para el desarrollo de aplicaciones web.

CSS: Cascading Style Sheet. Hojas de estilo en cascada, es el lenguaje que permite definir las reglas de diseño de estilo para los elementos de una página web.

PHP: PHP Hypertext Preprocessor. Es el lenguaje de programación de lado del servidor para el desarrollo de aplicaciones web dinámicas.

MariaDB: sistema gestor de bases de datos. Permite tener almacenada y organizada la información para su posterior consulta.

CRUD: por sus siglas en inglés Create, Update, Read y Delete, conjunto de operaciones básicas que se pueden realizar sobre una base de datos, crear, leer, actualizar y borrar.

JavaScript: lenguaje de programación del lado del cliente para el desarrollo de aplicaciones web.

DOM: Document Object Model. Interfaz mediante la cual se puede acceder de manera organizada y jerárquica a los objetos de un documento HTML.

jQuery: biblioteca de funciones para el lenguaje JavaScript basada en los elementos del DOM de una página web.

Framework: marco de trabajo que recoge diferentes conceptos, prácticas y criterios de manera estandarizada.

Laravel: es un framework para el desarrollo de aplicaciones web con el lenguaje de programación PHP

Bootstrap: biblioteca de funciones y herramientas para facilitar el diseño de aplicaciones web.

Diseño adaptativo/responsive: técnica usada en el desarrollo web mediante la cual se pretende que una aplicación web se visualice de manera correcta en cualquier dispositivo independientemente del tamaño de su pantalla.

Mobile First: es una filosofía para el desarrollo del diseño de una aplicación web que pretende que, a la hora de desarrollar aplicaciones web adaptativa, se centre primero el diseño en los dispositivos móviles ya que estos generan actualmente la mayoría del tráfico web.

Balsamiq: aplicación para crear bocetos para todo tipo de aplicaciones. Dispone de una licencia gratuita.

Scroll: acción de mover de arriba a abajo o de izquierda a derecha, mediante el teclado o el ratón en un ordenador de escritorio y mediante el dedo en un Smartphone o Tablet con la finalidad de desplazar el contenido visible para poder ver el contenido que se encuentra fuera de los límites de la pantalla.

10 Referencias

- [1] G. Méndez, «www.fdi.ucm.es,» Universidad Complutense de Madrid, 2008 10 22. [En línea]. Disponible en: <http://www.fdi.ucm.es/profesor/Gmendez/docs/is0809/ieee830.pdf>. [Último acceso: 08 2018].
- [2] Finocam, «[Finocam.com](http://finocam.com),» [En línea]. Disponible en: <https://finocam.com/es/organizacinparalsdocentes/22531-cuaderno-agenda-del-docente.html>. [Último acceso: 8 2018].
- [3] M. d. E. y. F. Profesional, «Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado,» 2012. [En línea]. Disponible en: http://www.ite.educacion.es/formacion/materiales/55/cd/pdf/m3_aplicaciones_docentes.pdf. [Último acceso: 08 2018].
- [4] M. Fernández, «El androide libre,» 28 04 2018. [En línea]. Disponible en: <https://elandroidelibre.lespanol.com/2018/04/tamanos-de-pantalla-evolucion-en-la-historia.html>. [Último acceso: 08 2018].
- [5] StatCounter, «gs.statcounter.com,» 08 2018. [En línea]. Disponible en: <http://gs.statcounter.com/browser-market-share/desktop/worldwide>. [Último acceso: 08 2018].
- [6] W3C, «W3C,» 2018. [En línea]. Disponible en: <https://www.w3.org/standards/webofdevices/>. [Último acceso: 08 2018].
- [7] G. E. Krasner y S. T. Pope, «A cookbook for using the model-view controller user interface paradigm in Smalltalk-80,» *Journal of Object-Oriented Programming*, vol. 1, pp. 26-49, 1988.
- [8] «Wikipedia,» [En línea]. Disponible en: <https://es.wikipedia.org/wiki/Modelo%20%80%93vista%20%80%93controlador>. [Último acceso: 08 2018].
- [9] J. Vara, M. López Sanz, J. J. Jiménez Hernández y J. Verde Marín, DESARROLLO WEB EN ENTORNO SERVIDOR. CFGS., Ra-Ma, 2012.
- [10] D. K. Lapin, «web.vu.lt,» 03 2016. [En línea]. Disponible en: http://web.vu.lt/mif/k.lapin/files/2016/03/6_prototyping.pdf. [Último acceso: 08 2018].
- [11] J. E. Córcoles Tendero y F. Montero Simarro, DISEÑO DE INTERFACES WEB. CFGS., Ra-Ma, 2012.
- [12] J. Vara, M. López Sanz, J. J. Jiménez Hernández y J. Verde Marín, DESARROLLO WEB EN ENTORNO CLIENTE. CFGS., Ra-Ma, 2012.
- [13] «Wikipedia,» [En línea]. Disponible en: <https://es.wikipedia.org/wiki/Framework>. [Último acceso: 08 2018].

- [14] M. Tuupola y D. Verheul, «Appelsiini.net,» 2013. [En línea]. Disponible en: <https://appelsiini.net/projects/jeditable>. [Último acceso: 08 2018].
- [15] V. Potapov, «vitalets.github.io,» 2013. [En línea]. Disponible en: <https://vitalets.github.io/x-editable>. [Último acceso: 08 2018].

11 Bibliografía

Laravel. [En línea] - <https://laravel.com/docs/5.6/>

Imágenes con AJAX y Laravel. [En línea] - <https://programacionymas.com/blog/subir-imagen-usando-ajax-y-laravel>

Tratamiento imágenes en Laravel. [En línea] - <http://image.intervention.io/api/response>

Tratamiento de fechas en Laravel. [En línea] - <https://github.com/jenssegers/date>

Notificaciones por email en Laravel. [En línea] - <https://programacionymas.com/blog/como-enviar-mails-correos-desde-laravel>

Documentación de PHP. [En línea] - <http://be2.php.net/manual/es/>