



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior

Trabajo Fin de Grado

PLATAFORMA WEB AVANZADA DE APOYO A LA DOCENCIA DE LA INGENIERÍA GRÁFICA

Alumno: Daniel Expósito Porras

Tutor: Prof. D. Juan Ruiz de Miras
Dpto: Informática

Octubre, 2015



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Dr. D. Juan Ruíz de Miras, tutor del Proyecto Fin de Carrera titulado: Plataforma Web avanzada de apoyo a la docencia de la Informática Gráfica, que presenta D. Daniel Expósito Porras, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, octubre de 2015

El alumno:

D. Daniel Expósito Porras

Los tutores:

Dr. D. Juan Ruíz de Miras

Índice

1. INTRODUCCIÓN	5
1.1. PROPÓSITO Y MOTIVACIÓN DEL PROYECTO	5
1.2. OBJETIVOS PRINCIPALES.....	5
1.3. METODOLOGÍA A DESARROLLAR	6
1.4. RESULTADOS ESPERADOS	7
1.5. ESTRUCTURA DEL PROYECTO	7
2. REQUISITOS DE SOFTWARE	10
2.1. ESPECIFICACIÓN DE REQUERIMIENTOS	10
2.1.1. REQUERIMIENTOS FUNCIONALES	10
2.1.2. REQUERIMIENTOS NO FUNCIONALES	19
2.2. PERFIL DE USUARIOS	20
2.3. ESTUDIO DE POSIBLES SOLUCIONES.....	21
2.3.1. SISTEMAS DE HOSTING PAAS.....	21
2.3.2. ESTRUCTURA Y DESARROLLO WEB	23
2.4. CASOS DE USO DE LA APLICACIÓN	30
2.4.1. DIAGRAMAS DE CASOS DE USO	30
2.4.2. NARRATIVAS DE LOS CASOS DE USO	48
3. PLAN DEL PROYECTO	69
3.1. PLANIFICACIÓN TEMPORAL	69
3.1.1. ESTIMACIÓN DE TIEMPOS	69
3.1.2. DIAGRAMAS DE GANT	71
3.2. RECURSOS Y ESTIMACIÓN DE COSTES	73
3.2.1. DESARROLLO.....	73
3.2.2. INSTALACIÓN	77
3.2.3. MANTENIMIENTO	79
3.2.4. OTROS GASTOS	79
4. DISEÑO	81
4.1. DISEÑO COMPLETO DE LA BASE DE DATOS.....	81
4.1.1. ESQUEMA CONCEPTUAL	81
4.1.2. ESQUEMA LÓGICO	84
4.1.3. NORMALIZACIÓN	86

4.1.4.	DISEÑO DE SEGURIDAD	87
4.1.5.	DISEÑO FÍSICO	89
4.2.	DISEÑO DEL SOFTWARE	94
4.2.1.	ESTRUCTURA VISUAL DE LA INTERFAZ DE LA APLICACIÓN	94
4.2.2.	DISEÑO DE PANTALLAS.....	96
4.2.3.	STORYBOARDS.....	106
4.2.4.	DIAGRAMAS DE CLASES.....	120
4.2.5.	DIAGRAMAS DE SECUENCIA	127
5.	IMPLEMENTACIÓN	134
5.1.	TECNOLOGÍAS UTILIZADAS.....	134
5.1.1.	HTML5	134
5.1.2.	JAVASCRIPT	135
5.1.3.	CSS3	135
5.1.4.	JQUERY	136
5.1.5.	WEBGL Y THREE.JS.....	136
5.1.6.	AJAX.....	137
5.1.7.	PHP	138
5.1.8.	MYSQL	138
5.1.9.	HERRAMIENTAS DE DESARROLLO.....	139
5.2.	DETALLES DE IMPLEMENTACIÓN DEL SISTEMA WEB.....	140
5.2.1.	ALGORITMO COMPROBAR OBJETIVO	140
5.2.2.	ALGORITMO PARPADEAR OBJETO.....	143
5.2.3.	MÉTODO PARA EL CONTROL DE LA SESIÓN.....	144
5.2.4.	MÉTODO PARA EXTRAER INFORMACIÓN DE LA BASE DE DATOS	145
6.	PRUEBAS.....	147
6.1.	PRUEBAS DE VALIDACIÓN.....	147
6.1.1.	MENU DEL PROFESOR.....	148
6.1.2.	MENU DEL ALUMNO	149
6.1.3.	OTRAS CORRECCIONES.....	150
6.2.	PRUEBAS DE UNIDAD	151
6.2.1.	APPLET DE TRANSFORMACIONES GEOMÉTRICAS	151
6.2.2.	APPLET DE TÉCNICAS DE SELECCIÓN	152
6.2.3.	APPLET DE TIPOS DE LUCES	152
6.2.4.	APPLET DE LUCES Y MATERIALES.....	153
6.2.5.	APPLET DE MODELOS DE ILUMINACIÓN.....	153

6.2.6.	APPLET DE SOMBREADO	153
6.2.7.	APPLET DE TEXTURAS	153
6.3.	PRUEBAS DE INTEGRACIÓN	154
6.3.1.	PROPONER UN EJERCICIO DEL TIPO LUCES Y MATERIALES	154
6.3.2.	MODIFICAR UN EJERCICIO DE TRANSFORMACIONES GEOMÉTRICAS	156
6.3.3.	CORREGIR UN EJERCICIO DEL TIPO TEXTURAS	159
6.3.4.	REVISAR UN EJERCICIO DEL TIPO MODELOS DE ILUMINACIÓN.....	161
6.3.5.	REALIZAR UN EJERCICIO DEL TIPO SOMBREADO.....	162
6.3.6.	EDITAR UN EJERCICIO DE TIPOS DE LUCES.....	163
7.	CONCLUSIONES.....	164
7.1.	PROPUESTAS DE MEJORA	164
8.	BIBLIOGRAFÍA	166
9.	ANEXO A: MANUAL DE USUARIO	170
9.1.	ACCESO AL SISTEMA WEB	170
9.2.	INICIAR SESIÓN	171
9.3.	PROPONER EJERCICIO.....	172
9.4.	MODIFICAR EJERCICIO	175
9.5.	CORREGIR EJERCICIO	176
9.6.	REVISAR EJERCICIO	178
9.7.	CONSULTAR CALIFICACIONES.....	180
9.8.	CAMBIAR CONTRASEÑA	183
9.9.	RESTABLECER CONTRASEÑA DE ALUMNO	184
9.10.	CONFIGURAR CURSO.....	185
9.11.	AÑADIR PROFESOR	187
9.12.	SELECCIONAR CURSO ACTIVO	188
9.13.	LISTA DE EJERCICIOS DEL ALUMNO	188
9.14.	CONSULTAR CALIFICACIONES DEL ALUMNO	190
9.15.	CERRAR SESIÓN	191
10.	ANEXO B: MANUAL DE INSTALACIÓN.....	192
10.1.	CONTENIDO DEL CD	192
10.2.	INSTALACIÓN EN LOCAL USANDO XAMPP	193
10.3.	INSTALACIÓN EN EL PAAS OPENSIFT	195

1. INTRODUCCIÓN

En este capítulo se hace una introducción al proyecto que se va a realizar describiendo la motivación por la que se va a desarrollar. También se detallará un análisis de los objetivos claves del proyecto.

Partiendo de un proyecto previamente desarrollado, se van a describir también las funciones que se aprovecharán de dicho proyecto y las mejoras que se pretenden hacer sobre el mismo.

Finalmente se expondrá la estructura del proyecto de forma breve anunciando el contenido que se va a ver en cada uno de los apartados del mismo.

1.1. PROPÓSITO Y MOTIVACIÓN DEL PROYECTO

El propósito principal del proyecto se basa en desarrollar una aplicación Web para agilizar el proceso de enseñanza/aprendizaje de los principales conceptos de la Informática Gráfica a través de la realización interactiva de ejercicios previamente configurados por el profesor a través de la aplicación.

Es importante que la aplicación sea fácilmente usable, para motivar al alumno realizar los ejercicios y ayudar en su aprendizaje en la asignatura. Para ello se debe diseñar una interfaz intuitiva y usable, tanto para el alumno, como para el profesor.

También se pretende conseguir que la aplicación sea un instrumento a la hora de calificar a los alumnos, proporcionando un sistema de calificaciones que sirva de apoyo a la calificación final de la asignatura.

Además, partiendo de la base de la aplicación previa desarrollada en HTML5, Javascript, CSS y WebGL se pretende añadir la parte de servidor incluyendo en los lenguajes presentes en la aplicación PHP y MySQL para llevar la gestión de la base de datos de la aplicación.

1.2. OBJETIVOS PRINCIPALES

El objetivo principal del proyecto es dotar a la aplicación ya existente, que permite usar varios applets para hacer pruebas y comprobar el funcionamiento de

algunas técnicas de la Informática Gráfica, de una plataforma de propuesta/resolución de ejercicios para ser utilizada activamente durante el desarrollo de la asignatura.

Dicha plataforma va a permitir a los profesores proponer ejercicios referentes a los distintos applets, estableciendo los parámetros oportunos. Estos ejercicios podrán ser realizados por los alumnos, que obtendrán una calificación al ser corregidos por el profesor.

También se pretende conseguir que los profesores puedan utilizar dicho sistema de calificaciones de manera cómoda, extrayendo e incluyendo las calificaciones al sistema Web fácilmente.

Para conseguir estos objetivos principales, a continuación se plantean una serie de objetivos más específicos para la correcta realización del proyecto:

- Evaluación y selección de plataformas de hosting PaaS para albergar el sistema Web.
- Diseño e implementación del sistema de gestión de base de datos para la aplicación.
- Diseño e implementación de los menús personalizados para alumnos y profesores para poder acceder a las funciones correspondientes del sistema Web.
- Realización de pruebas para minimizar el número de errores posibles en la aplicación.
- Redacción de la memoria del proyecto junto con la documentación relacionada, manuales de instalación y de usuario.

1.3. METODOLOGÍA A DESARROLLAR

La metodología que se va a utilizar durante la elaboración de este proyecto consta de los siguientes elementos:

- Se estudiará y evaluará el sistema de hosting Web para albergar la página de este proyecto.

- Se estudiará la estructura y el diseño de la base de datos que será implementada con MySQL. Por tanto el hosting Web deberá soportarlo.
- Se implementará la aplicación basándose en el patrón Modelo-Vista-Controlador para conseguir un código desacoplado y poder garantizar una futura expansión de la aplicación de manera sencilla.
- Se realizará un proceso de implementación y prueba basado en la realimentación con alumnos pasados de la asignatura y con el profesor de la misma para facilitar el uso de la aplicación en el futuro.
- Además, será necesario realizar una búsqueda y revisión bibliográfica durante el desarrollo completo del proyecto.

1.4. RESULTADOS ESPERADOS

El resultado que se desea obtener es una plataforma que permita a los profesores proponer ejercicios referentes a los conceptos de la asignatura ejemplificados en distintos applets y que éstos puedan ser realizados por los alumnos.

También se pretende conseguir que se puedan corregir los ejercicios y los alumnos tengan su respectiva calificación en ellos.

Se va a diseñar un sistema Web que pueda ser utilizado por varios profesores y alumnos a la vez y que los alumnos puedan usarlo en distintos cursos.

Además se realizará un informe en el que se detallen los siguientes aspectos:

- Documentación detallada sobre la metodología seguida para el diseño e implementación del sistema Web de ejercicios para la Informática Gráfica.
- Pruebas de dicho sistema.
- Redacción de la memoria y manual de usuario.

1.5. ESTRUCTURA DEL PROYECTO

En este capítulo se ha hecho una breve introducción mostrando una descripción de lo que se pretende hacer, un avance de cómo se va a hacer y los resultados esperados al finalizarlo.

En el segundo capítulo se especifican los requisitos del software, funcionales y no funcionales. En este capítulo se verá el estudio de las posibles soluciones que mejor se adapten al proyecto que se va a realizar. Por último se mostrarán los casos de uso de la aplicación.

En el tercer capítulo se detallará el plan del proyecto, donde se presentará la planificación temporal, los recursos necesarios para el proyecto, tanto humanos como software y hardware. También se realizará una estimación de costos del proyecto, de desarrollo, instalación y explotación del mismo.

El cuarto capítulo se centrará en el diseño de la aplicación. Se detallará el diseño completo de la base de datos, el diseño de los menús principales de la aplicación mediante storyboards y los diagramas de secuencia.

Especificado el diseño de la aplicación, en el quinto capítulo se estudiarán aspectos de la implementación del sistema Web. Se describirán todas las tecnologías usadas para el desarrollo del mismo y su justificación. Además se explicarán algunos detalles relacionados con la implementación de la base de datos y de los menús de administración de profesores y alumnos.

En el sexto capítulo se detallarán las pruebas realizadas al sistema Web, así como de los errores detectados, y sus respectivas soluciones.

En el siguiente capítulo se explican las conclusiones propias extraídas de la realización del proyecto. Se propondrán además distintas formas de mejorarlo en un futuro.

En el octavo capítulo se expondrá la bibliografía que se ha utilizado para la realización de este proyecto, tanto para realizar la documentación, como para la implementación del mismo.

En el noveno capítulo se incluirá un anexo con el manual de usuario para ayudar al completo manejo de la aplicación sin problemas.

En el último capítulo se incluirá un anexo con el manual de instalación. En el caso de este proyecto será la dirección URL del sistema Web, donde estará todo funcionando correctamente.

2. REQUISITOS DE SOFTWARE

En este capítulo se detallarán los requerimientos que debe tener el sistema Web. Están divididos en requisitos funcionales y no funcionales, argumentando además el porqué de utilizarlos en este sistema Web.

También se realizará un estudio de las posibles soluciones para desarrollar la aplicación. Se evaluarán distintos lenguajes y se explicarán las ventajas e inconvenientes y la justificación sobre cuáles serán escogidos para desarrollar la aplicación.

Además se explicarán los perfiles de usuario existentes en nuestra aplicación y, por último, se hará un estudio sobre los casos de uso de la aplicación y las narrativas asociadas.

2.1. ESPECIFICACIÓN DE REQUERIMIENTOS

Existen dos tipos de requisitos^[1]:

- Los requisitos funcionales describen los servicios que el sistema debe proporcionar, así como las respuestas del mismo ante determinadas entradas y comportamiento en determinadas situaciones.
- Los requisitos no funcionales son las distintas restricciones para los servicios ofrecidos por el sistema: restricciones tiempo de respuesta, capacidad de almacenamiento, fiabilidad, etc.

2.1.1. REQUERIMIENTOS FUNCIONALES

Los requerimientos funcionales^[1] de un sistema software son aquellos que describen funcionalidades que debe proporcionar dicho sistema a los usuarios que vayan a usarlo.

A continuación, se detallan los distintos requerimientos funcionales que debe cumplir el sistema Web. Están divididos en requisitos generales del sistema Web, requisitos de los menús de inicio de profesor y alumno. Además se incluirán los requisitos de los distintos applets presentes en la aplicación.

1. Generales del sistema Web

RF1.1. Debe ser posible cambiar el idioma (inglés/español) de los menús de alumno y profesor de forma sencilla.

RF1.2. Debe ser posible iniciar sesión como un profesor o un alumno registrado en la base de datos.

2. Menú de inicio profesor.

El objetivo del menú es cumplimentar la funcionalidad necesaria para que el profesor pueda controlar el sistema Web de realización de ejercicios de forma óptima. Para ello se elaborará un menú donde se encontrarán las distintas funcionalidades necesarias para el profesor. También se incluirán en la página las distintas opciones de cada acción del profesor.

2.1 Proponer ejercicios

RF2.1.1. Debe ser posible proponer ejercicios nuevos sobre los distintos applets presentes en el sistema Web.

RF2.1.1. Debe ser posible realizar modificaciones sobre ejercicios propuestos anteriormente.

RF2.1.3. Debe ser posible establecer un rango de fechas de disponibilidad del ejercicio.

RF2.1.4 Debe ser posible eliminar un ejercicio propuesto.

2.2. Modificar ejercicios propuestos

RF2.2.1. Debe ser posible modificar el rango de fechas establecido para un ejercicio propuesto por el profesor conectado.

RF2.2.2. Debe ser posible modificar por completo un ejercicio previamente mandado por el profesor conectado.

2.3. Ejercicios por corregir

RF2.3.1. Debe ser posible ver los ejercicios realizados por los alumnos matriculados en el curso del profesor conectado pendientes de corregir.

RF2.3.2. Debe ser posible corregir dichos ejercicios asignándoles una calificación y, opcionalmente, observaciones sobre el ejercicio.

2.4. Revisar ejercicios corregidos

RF2.4.1. Debe ser posible revisar las calificaciones otorgadas al alumno.

RF2.4.2. Debe ser posible modificar las calificaciones otorgadas al alumno.

2.5. Consultar calificaciones

RF2.5.1. Debe ser posible consultar las calificaciones de cada alumno individualmente.

RF2.5.2. Debe ser posible consultar las calificaciones de todos los alumnos de un curso.

RF2.5.3. Debe ser posible exportar dichas calificaciones en formato excel (csv).

2.6. Cambiar contraseña

RF2.6.1. Debe ser posible cambiar la contraseña del profesor conectado.

2.7. Cambiar contraseña alumno

RF2.7.1. Debe ser posible cambiar la contraseña de cualquier alumno matriculado a un curso perteneciente al profesor conectado.

2.8. Configurar curso

RF2.8.1. Debe ser posible configurar un curso nuevo seleccionando el curso y grupo.

RF2.8.2. Debe ser posible configurar un curso nuevo importando el curso, el grupo y los alumnos desde excel(csv).

2.9. Añadir profesor

RF2.9.1. Debe ser posible añadir un nuevo profesor para los profesores que sean administradores del sistema.

2.10. Seleccionar curso activo

RF2.10.1. Debe ser posible seleccionar el curso y grupo activo para el profesor.

3. Menú de inicio alumno.

El objetivo del menú es cumplimentar la funcionalidad necesaria para que el alumno pueda controlar el sistema Web de realización de ejercicios de forma óptima. Para ello se elaborará un menú donde se encontrarán las distintas funcionalidades necesarias para el alumno. En la zona derecha de la página se expondrán las distintas opciones de cada acción del alumno. El alumno podrá seleccionar el curso activo, que por defecto será el actual, para realizar ejercicios en dicho curso. Como un alumno matriculado solo puede estar en un grupo, éste permanecerá fijo.

3.1.Lista de ejercicios

RF3.1.1. Debe ser posible ver el listado de ejercicios propuestos en el curso activo del alumno.

RF3.1.2. Debe ser posible ver cualquier ejercicio propuesto en el curso activo del alumno.

RF3.1.3. Debe ser posible realizar los ejercicios que se encuentren dentro del rango de fechas establecido en el curso activo del alumno.

RF3.1.4 Debe ser posible modificar los ejercicios realizados por el alumno si no se ha llegado a la fecha límite del ejercicio.

3.2. Consultar calificaciones

RF3.2.1. Debe ser posible consultar las calificaciones del alumno conectado en los cursos que haya estado matriculado.

3.3. Cambiar contraseña

RF3.3.1. Debe ser posible cambiar la contraseña del alumno conectado.

3.4. Seleccionar curso activo

RF3.4.1. Debe ser posible seleccionar el curso y grupo activo para el alumno.

4. Requisitos específicos de los applets.

Cada applet tiene requisitos específicos que se deben cumplir para garantizar el correcto funcionamiento de los mismos.

4.1. Applet de Transformaciones Geométricas

RF4.1.1. Debe ser posible modificar y recoger los datos de la matriz de transformación acumulada y almacenarlos en la base de datos de la aplicación.

RF4.1.2. Se debe añadir un objeto semitransparente para este applet. Dicho objeto deberá tener aplicada la matriz de transformación establecida por el profesor en el ejercicio.

RF4.1.3. El objeto semitransparente debe parpadear cuando se cumpla que la media de las distancias entre vértices sea menor que un umbral establecido.

RF4.1.4. Se debe visualizar la media de distancias entre los vértices del objeto que se está modificando y del objetivo. Mientras se encuentre por encima del umbral debe aparecer en color rojo. Una vez que la media de las distancias esté por debajo del umbral debe aparecer en color verde.

RF4.1.5. Se deben añadir los campos nota y observaciones para corregir un ejercicio.

RF4.1.6. Cuando el profesor pulse en el botón “corregir ejercicio” se deben guardar los campos nota y observaciones en la base de datos.

4.2. Applet de Técnicas de Selección

RF4.2.1. Debe ser posible modificar y recoger los datos de la posición de los objetos, el tamaño y posición del área de selección y la lista de intersecciones conseguida para almacenarlos en la base de datos.

RF4.2.2. Se debe añadir la escena configurada por el profesor para compararla con la obtenida por el alumno a la hora de corregir un ejercicio.

RF4.2.3. Cuando el alumno obtenga una media de las distancias entre las mínimas y máximas Z de los objetos inferior a un umbral establecido el rayo o los objetos deben parpadear.

RF4.2.4. Cuando se consiga bajar del umbral la media de las distancias debe aparecer en color verde. Hasta entonces debe aparecer en color rojo.

RF4.2.5. Se deben añadir los campos nota y observaciones para corregir un ejercicio.

RF4.2.6. Cuando el profesor pulse en el botón “corregir ejercicio” se deben guardar los campos nota y observaciones en la base de datos.

4.3. Applet de Técnicas de Selección

RF4.3.1. Debe ser posible modificar los parámetros de la luz y almacenarlos en la base de datos.

RF4.3.2. Debe ser posible recuperar la escena establecida por el profesor sin la posición de la luz al proponer un ejercicio y mostrarla cuando el alumno vaya a realizar un ejercicio. El alumno tendrá además su propia escena para realizar el ejercicio.

RF4.3.3. Deben cargarse las escenas obtenidas por el profesor y el alumno cuando se proceda a corregir un ejercicio. Además se deben obtener los datos del alumno y del profesor para realizar la comparación.

RF4.3.4. Se deben añadir los campos nota y observaciones para corregir un ejercicio.

RF4.3.5. Cuando el profesor pulse en el botón “corregir ejercicio” se deben guardar los campos nota y observaciones en la base de datos.

4.4. Applet de Luces y Materiales

RF4.4.1. Debe ser posible modificar los parámetros de la luz y material y almacenarlos en la base de datos.

RF4.4.2. Debe ser posible recuperar la escena establecida por el profesor sin la posición de la luz al proponer un ejercicio y mostrarla cuando el alumno vaya a realizar un ejercicio. El alumno tendrá además su propia escena para realizar el ejercicio.

RF4.4.3. Deben cargarse las escenas obtenidas por el profesor y el alumno cuando se proceda a corregir un ejercicio. Además se deben obtener los datos del alumno y profesor para realizar la comparación.

RF4.4.4. Se deben añadir los campos nota y observaciones para corregir un ejercicio.

RF4.4.5. Cuando el profesor pulse en el botón “corregir ejercicio” se deben guardar los campos nota y observaciones en la base de datos.

4.5. Applet de Modelos de Iluminación

RF4.5.1. Debe ser posible modificar los parámetros de la luz y de la cámara y almacenarlos en la base de datos.

RF4.5.2. Debe ser posible recuperar la escena establecida por el profesor y los parámetros fijados por el mismo, la intensidad de la luz, los coeficientes de los materiales y el tipo de luz seleccionada. Estos parámetros deben ser fijos y no podrán ser modificados por el alumno.

RF4.5.3. El alumno solo podrá modificar la posición de la luz y de la cámara.

RF4.5.4. Deben mostrarse los resultados del profesor cuando el alumno se disponga a realizar un ejercicio. Los resultados del alumno deberán aparecer contiguos a los del profesor y, cuando se minimice la diferencia por debajo del umbral establecido, los valores parpadearán.

RF4.5.5. Al profesor le deben aparecer la escena que configuró, la obtenida por el alumno al corregir ejercicio y los parámetros establecidos por ambos. Además se deben añadir los campos nota y observaciones.

RF4.5.6. Cuando el profesor pulse en el botón “corregir ejercicio” se deben guardar los campos nota y observaciones en la base de datos.

4.6. Applet de Sombreado

RF4.6.1. Debe ser posible modificar el número de divisiones horizontales y verticales, y el tipo de sombreado y almacenarlos en la base de datos.

RF4.6.2. El número de divisiones horizontales y verticales deben ser iguales.

RF4.6.3. Cuando un alumno se disponga a realizar un ejercicio, deben aparecerle dos canvas, uno con la escena configurada por el profesor y otro con los valores por defecto del applet.

RF4.6.4. Deber comprobarse que el tipo de sombreado del alumno y profesor sean iguales y que el número de divisiones difiera por debajo del umbral establecido. Una vez se cumplan estos requisitos la esfera parpadeará.

RF4.6.5. Al profesor le deben aparecer la escena que configuró, y la obtenida por el alumno al corregir ejercicio, así como los parámetros establecidos por los mismos. Además se deben añadir los campos nota y observaciones.

RF4.6.6. Cuando el profesor pulse en el botón “corregir ejercicio” se deben guardar los campos nota y observaciones en la base de datos.

4.7.Applet de Texturas

RF4.7.1. Debe ser posible modificar la textura seleccionada, las coordenadas de textura de los cuatro vértices, los modos de repetición, el tamaño del plano al que será aplicado la textura y los filtros de textura y almacenarlos en la base de datos.

RF4.7.2. Cuando el alumno se disponga a realizar el ejercicio le deben aparecer dos canvas, uno de ellos con la textura configurada por el profesor y el otro los parámetros por defecto del applet.

RF4.7.3. El alumno debe poder modificar las coordenadas de textura de cuatro vértices, los modos de repetición y los filtros de textura para conseguir el objetivo.

RF4.7.4. Cuando los parámetros del alumno coincidan con los del profesor, la textura deberá parpadear.

RF4.7.5. Al profesor le deben aparecer la textura que configuró, y la obtenida por el alumno al corregir ejercicio, así como los parámetros establecidos por los mismos. Además se deben añadir los campos nota y observaciones.

RF4.7.6. Cuando el profesor pulse en el botón “corregir ejercicio” se deben guardar los campos nota y observaciones en la base de datos.

2.1.2. REQUERIMIENTOS NO FUNCIONALES

Los requerimientos no funcionales^[1] son las restricciones impuestas a los requerimientos funcionales del sistema. Éstos no hacen referencia a ninguna funcionalidad, sino a propiedades como la fiabilidad, la respuesta en el tiempo o la capacidad de almacenamiento. Es decir, tienen que ver con las características que de una forma u otra puedan limitar el sistema. Describen una restricción sobre el sistema que limita nuestra elección en la construcción de una solución.

Además son tan importantes como los funcionales y pueden ser críticos para la aceptación del sistema.

A continuación se muestran los requisitos no funcionales que necesita este sistema web y su justificación:

Una restricción importante en la aplicación es la **capacidad de almacenamiento**. Debido a que la aplicación dispone de una base de datos, esta propiedad será variable y por tanto una barrera a la hora de escoger el sistema de hosting de la aplicación. Por tanto se realizará más adelante un cálculo del tamaño estimado de la aplicación para los próximos cinco años para poder cumplir este requisito. No obstante, hay que escoger un sistema de hospedaje que permita la **escalabilidad** de la aplicación en términos de almacenamiento para prolongar su vida en el tiempo.

La **facilidad de aprendizaje o de uso**, al ser un sistema interactivo es especialmente importante que los usuarios que van a usar la aplicación puedan hacerlo de una forma fácil. Para ello se necesita que la interfaz de los menús sea sencilla, de tal forma que permita a las personas que van a usar el sistema en el desarrollo de la asignatura comprendan rápidamente cómo funciona para poder resolver los ejercicios, mostrando también mensajes de error descriptivos, añadiendo texto resaltado en las funciones más importantes y con la incorporación de un apartado con las instrucciones para el manejo de cada applet.

La **robustez**, es decir, la fiabilidad del sistema o el grado en que el sistema es capaz de tolerar fallos o situaciones especiales durante la interacción. Hace

referencia a las características que ayudan a la consecución con éxito de los objetivos del usuario que emplea el sistema web,. de tal manera que hay que tener en cuenta todas las situaciones a las que puede llegar la aplicación, tanto la parte de los menús de alumno, profesor y los applets, como la parte de servidor y base de datos. Ésta ultima tiene una importancia máxima en la robustez de la aplicación.

El **tiempo de respuesta**, es importante para que el usuario sea capaz de percibir cambios ante cualquier interacción con el sistema. Debido a que la aplicación tiene parte servidor con acceso a la base de datos, el tiempo de respuesta se puede ver afectado por la carga de datos que se necesite del servidor y a la banda ancha del cliente. Se intentará que en este proyecto sea una respuesta aceptable, no obstante, si hay alguna tarea que necesite más tiempo se mostrara una pantalla de carga para que el usuario sepa que la aplicación sigue activa.

Hay que garantizar la **fiabilidad** del sistema Web y la **integridad** de los datos obtenidos de la base de datos, para un correcto funcionamiento de la aplicación.

2.2. PERFIL DE USUARIOS

En la aplicación se van a tener cuatro tipos de usuarios:

- Invitado: Puede ser cualquier usuario que tenga acceso a Internet y conozca el sistema Web. Podrá acceder a toda la funcionalidad de la aplicación previa en la que se basa este proyecto.
- Profesor: Debe tener conexión a Internet. Este tipo de usuario corresponde al profesor de la asignatura, que podrá proponer y corregir los ejercicios. Tendrá acceso toda la funcionalidad del menú de profesor, excepto las opciones de administrador.
- Profesor administrador: Debe tener conexión a Internet. Tiene un perfil de usuario similar al profesor, pero además podrá realizar algunas acciones de administrador como cambiar contraseñas, configurar un curso o añadir un nuevo profesor.
- Alumno: Debe tener conexión a Internet. Este tipo de usuario corresponde al de un alumno matriculado en la asignatura y que haya sido dado de alta por un profesor para poder realizar ejercicios en la

misma. Podrá realizar todas las acciones presentes en el menú del alumno.

2.3. ESTUDIO DE POSIBLES SOLUCIONES

Antes de realizar la implementación del sistema web, se debe realizar un estudio sobre los posibles sistemas de hosting para albergar la aplicación. Debido a que se parte de una aplicación previamente desarrollada, el lenguaje utilizado para la parte cliente permanecerá fijo. No obstante, se debe hacer un estudio de los lenguajes a utilizar para la parte servidor, así como la forma de conectar la parte cliente con la parte servidor.

Actualmente, el sistema Web se encuentra alojado en los servidores de la Universidad de Jaén [2]. Esta opción será evaluada para hospedar el nuevo sistema Web de este proyecto.

A continuación se estudiarán también algunas plataformas como un servicio para albergar la nueva aplicación.

2.3.1.SISTEMAS DE HOSTING PAAS

Las novedosas Platform as a Service (PaaS) [3]ponen a disposición del usuario plataformas completas como un servicio. Se trata de servidores web que permiten subir un sistema Web o aplicación y se encargan de hospedarla y mantenerla activa en la red.

A continuación se va a realizar un estudio de los principales sistemas PaaS actualmente.

- **OPENSIFT**

OpenShift[4] es un producto de computación en la nube de plataforma como servicio perteneciente a Red Hat. Este software funciona como un servicio que es de código abierto bajo el nombre de OpenShift Origin, y está disponible en GitHub. Los desarrolladores pueden usar Git para desplegar sus aplicaciones Web en los diferentes lenguajes de la plataforma.

OpenShift se encarga de mantener los servicios subyacentes a la aplicación y la escalabilidad de la aplicación como se necesite. Soporta los entornos de programación Web principales como PHP, Node.js, Ruby, Python, Perl o Java, así como servicios de gestión de bases de datos como MySQL o MongoDB.

Ventajas:

Permite centrarse más en la programación de la aplicación para agilizar el proceso de desarrollo. Dispone de una amplia documentación y un tutorial muy intuitivo para desplegar la aplicación. En la versión gratuita ofrece hasta 1 GB de almacenamiento. Además se pueden ampliar los recursos de la aplicación con la versión premium.

Desventajas:

El nodo europeo está colapsado de aplicaciones y no puede albergar nuevos sistemas Web.

- **AWS**

Amazon Web Services (AWS)^[5] es una colección de servicios de computación en la nube que en conjunto forman una plataforma de computación en la nube, ofrecida a través de Internet por Amazon.com. Es una de las ofertas internacionales más importantes de la computación en la nube.

Ventajas:

La aplicación web carga mucho más rápido que en todos los hosting gratuitos existentes y que muchos hosting pagados. Autoregula la cantidad de usuarios para evitar caídas.

Desventajas:

Se trata de un PaaS no gratuito y por tanto poco atractivo para desarrollos Web iniciales.

- **HEROKU**

Heroku [6] es una interfaz avanzada de gestión de recursos en la nube que soporta distintos lenguajes de programación. Principalmente se creó con el objetivo de soportar el lenguaje Ruby, pero se ha extendido a otros lenguajes de programación como PHP, Java o Node.js.

Ventajas:

Se puede escalar la aplicación en cualquier momento desde la consola de comandos. Redirige el tráfico según el estado de la red. Tiene una versión gratuita suficiente para albergar un sistema Web suficientemente complejo. Además posee un tutorial para subir la aplicación de manera sencilla.

Desventajas:

No soporta MySQL. La versión premium es bastante cara y dependiente del tráfico del sistema Web.

2.3.2. ESTRUCTURA Y DESARROLLO WEB

A continuación se expondrán algunos lenguajes de programación para la parte servidor, así como sus ventajas y desventajas. Finalmente se explicará los lenguajes elegidos para el desarrollo de la aplicación.

Debido a que la aplicación parte de un sistema Web previamente desarrollado en HTML5 y JavaScript, la estructura de la parte del cliente queda fija. Igualmente se explicarán algunas características de estas tecnologías.

- **HTML5**

HTML [7] es el lenguaje básico para describir una página Web. Se trata de un lenguaje de marcas o etiquetas que se utiliza para realizar la estructura de la misma. No es un lenguaje de programación, pero sí permite visualizar la información generada por otros lenguajes de programación Web. Está formado por una serie de elementos que definen el significado semántico del contenido o su estructura.

HTML5 es la última evolución de la norma que define HTML, incluye nuevos elementos, atributos y comportamientos, además de un conjunto más amplio de tecnologías.

Ventajas:

Es un lenguaje sencillo y estructurado que admiten todos los navegadores. El código es simple, lo que permite hacer páginas más ligeras que se cargan más rápidamente favoreciendo la usabilidad y la indexación en buscadores. Permite el uso de <canvas> que facilita la integración de gráficos 3D mediante WebGL [8].

Desventajas:

Es un lenguaje estático, el diseño es más lento y además es muy básico, por lo que se suele apoyar en el uso de otras tecnologías para el desarrollo completo de un sistema Web.

- **JavaScript**

JavaScript [9] es un lenguaje de scripting multiplataforma basado en el navegador que ejecuta el código del lado del cliente. Es un lenguaje orientado a objetos basado en prototipos, débilmente tipado y dinámico. Se utiliza principalmente en la parte del cliente, aunque es posible emplearlo en el lado del servidor y cada vez hay disponibles un mayor número de frameworks para utilizarlo en dicho lado de una forma segura y sencilla.

Puede manejar eventos según acciones que realiza el usuario. Su fácil aprendizaje potencia su demanda y, debido a la gran cantidad de frameworks existentes, se pueden hacer aplicaciones Web completas bajo este lenguaje.

Ventajas:

Es un lenguaje sencillo y liviano que utiliza muy poca memoria y es muy útil para el desarrollo de páginas web dinámicas debido a que tiene una gran cantidad de efectos visuales. Permite el desarrollo de un sistema Web completo en el lado del cliente sin necesidad del servidor.

Desventajas:

El código es visible por cualquier usuario. Los scripts tienen capacidades limitadas, por razones de seguridad. Los usuarios pueden desactivarlo en su navegador y se pueden generar errores fácilmente en aplicaciones no muy complejas.

- **JQuery**

JQuery^[10] es una biblioteca de Javascript que permite simplificar la manera de trabajar con los elementos HTML, manipular el árbol DOM, manejar eventos y agregar interacción con la técnica AJAX en el sistema Web.

Ventajas:

Contiene un alto número de plugins de fácil integración. Permite realizar llamadas AJAX para comunicar la parte cliente con la parte servidor. Provee de un fácil acceso a los elementos HTML y CSS desde Javascript.

Desventajas:

Posee un número muy elevado de versiones que obligan a actualizarla constantemente lo que puede hacer que el código tenga que adaptarse con cada actualización.

- **PHP**

PHP ^[11] es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Es uno de los lenguajes más empleados para el desarrollo de aplicaciones web.

Ventajas:

Es un lenguaje de fácil aprendizaje que soporta la orientación a objetos, clases y herencia. Es capaz de conectar con la mayoría de los manejadores de base de datos: MySQL, Oracle, SQL Sever, etc. Posee documentación en su página oficial^[11] que incluye descripción y ejemplos de cada una de sus funciones. Además el código permanece oculto al usuario ya que éste está en el lado del servidor.

Desventajas:

Se necesita instalar en el servidor Web. Todo el trabajo lo realiza en el servidor, no delega al cliente, por tanto puede ser más ineficiente a medida que las solicitudes aumenten en número.

- **ASP**

Es una tecnología del lado de servidor desarrollada por Microsoft para el desarrollo de sitios Web dinámicos. ASP^[12] no necesita ser compilado para ejecutarse. El lenguaje más utilizado para crear páginas ASP es Visual Basic Script (VBS), nativo de Microsoft. El código ASP puede ser insertado junto con el código HTML.

Ventajas:

Usa Visual Basic Script para facilitar el aprendizaje a los usuarios. Está perfectamente integrado con SQL Server.

Desventajas:

Es necesaria una gran cantidad de código para realizar funciones muy sencillas. Es una tecnología propietaria.

- **JSP**

Es un lenguaje para la creación de sitios web dinámicos, acrónimo de Java Server Pages^[13]. Está orientado a desarrollar páginas web en Java. Se ejecuta en el lado del servidor. Posee un motor de páginas basado en los servlets de Java. Para su funcionamiento necesita tener instalado un servidor Tomcat.

Ventajas:

Permite una ejecución rápida del servlets. Es multiplataforma y tiene un código muy bien estructurado. Es menos dependiente del navegador e incluso del sistema operativo gracias a la máquina virtual de Java.

Desventajas:

Para los usuarios que no conocen Java es difícil de programar. Debido a que es un lenguaje precompilado, es más lento de procesar y además tiene un espacio muy delimitado en la página donde se ejecuta, es decir, no se integra con todos los componentes de la página ni tiene acceso a ellos.

- **AJAX**

AJAX [14] no es una tecnología en sí misma, es un término que describe un nuevo modo de utilizar conjuntamente varias tecnologías existentes como HTML, CSS, JavaScript, XML, XMLHttpRequest, etc. Cuando estas tecnologías se combinan en un modelo AJAX, es posible lograr aplicaciones web capaces de actualizarse continuamente sin tener que volver a cargar la página completa debido a que la información con el servidor se procesa en un segundo plano. Esto posibilita la creación de aplicaciones más rápidas y con mejor respuesta a las acciones del usuario.

Ventajas:

La experiencia de usuario en la navegación es más rápida ya que no se refresca la página constantemente al interactuar con ella y el tiempo de espera para una petición se reduce y por tanto el tráfico al servidor. Es un método óptimo para conectar la parte cliente con la parte servidor.

Desventajas:

La utilización excesiva de código AJAX puede hacer más lenta la navegación. Para hacer aplicaciones complejas hay que limitar el uso de estas llamadas.

- **MYSQL**

MySQL [15] es un sistema de gestión de bases de datos relacional, multihilo y multiusuario muy rápido en la lectura. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en la lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Es un software de fuente abierta, por tanto es posible usarlo y modificarlo por cualquier usuario, ya que se puede bajar su código fuente.

Ventajas:

Es un lenguaje muy veloz a la hora de realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento. Tiene un bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con pocos recursos sin problema. Es fácil de configurar e instalar.

Desventajas:

Un gran porcentaje de las utilidades de MySQL no están documentadas y es un lenguaje poco intuitivo.

- **Oracle**

Oracle [16] SQL es un sistema de gestión de bases de datos relacional para la gestión de grandes cantidades de información. Es un sistema que se extiende más en grandes empresas debido a su alto coste. En desarrollo Web no está muy extendido por ese motivo.

Ventajas:

Una ventaja principal de Oracle es su estandarización y consistencia entre distintas implementaciones. Puede ejecutarse en todas las plataformas.

Desventajas:

Tiene un coste muy elevado por lo que no es una buena opción para un desarrollo Web común. También tiene un elevado coste de formación.

Conclusiones:

Una vez identificadas características, ventajas e inconvenientes sobre los lenguajes y herramientas para el desarrollo en la Web descritos anteriormente, se puede decidir cuál o cuáles de ellos utilizar para realizar el presente proyecto.

Para alojar el sistema Web se ha estudiado la posibilidad de mantener el host actual en los servidores de la Universidad de Jaén www4.ujaen.es, pero tiene algunas desventajas como la capacidad de almacenamiento establecida en 10MB y la cantidad de lenguajes de programación disponibles para la parte servidor.

Por tanto, para hospedar el sistema Web se va a emplear Openshift [4] ya que ofrece muy buenos recursos en la versión gratuita. Se va a utilizar este PaaS debido a que es un open source y permite hospedar una página Web de hasta 1GB de almacenamiento y que soporte los lenguajes necesarios para la aplicación de forma gratuita. Además tiene una gran cantidad de documentación y tutoriales para su fácil instalación y una extendida comunidad para la resolución de problemas. También permite subir mediante un repositorio Git [17] el sistema Web de manera sencilla y todo el material está muy bien documentado.

El objetivo de la nueva aplicación es crear un sistema de propuesta/resolución de ejercicios que serán almacenados en una base de datos. Para la gestión de la base de datos se va a utilizar MySQL5.5 ya que es el sistema más extendido en el desarrollo Web, es gratuito y está soportado por Openshift.

Debido a la inclusión de la parte servidor en la aplicación va a ser necesario añadir AJAX para conectarla con la parte cliente. Esto puede provocar una ralentización en la aplicación si se realizan demasiadas peticiones. Pero en este sistema Web no se estima una gran carga de transacciones, debido a que se va a utilizar para la asignatura de Informática Gráfica y no se deberían sobrepasar los 100 usuarios conectados a la vez.

Por último para completar la parte servidor se va a utilizar PHP, ya que permite conectar con la base de datos MySQL de manera sencilla, es un lenguaje muy rápido y está soportado en Openshift en la versión 5.4.

Por tanto se va a conectar la parte del cliente con la parte del servidor mediante consultas AJAX mediante JQuery, que recogerán los resultados del código PHP para ser procesado en Javascript . Mediante PHP se realizará la conexión con la base de datos MySQL y se realizarán las consultas correspondientes.

2.4. CASOS DE USO DE LA APLICACIÓN

Un modelo de casos de uso [1] describe bajo la forma de acciones y reacciones el comportamiento observable de un sistema desde el punto de vista del usuario, denominado actor y las funciones y responsabilidades se conocen con el nombre de casos de uso. Es una técnica empleada para especificar los requisitos funcionales de la aplicación y presentan ventajas sobre la descripción meramente textual.

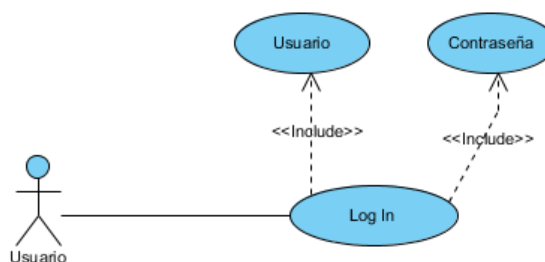
2.4.1. DIAGRAMAS DE CASOS DE USO

Un diagrama de casos de uso [1] documenta el comportamiento de un sistema desde el punto de vista del usuario. Consiste en un grafo de actores, un conjunto de casos de uso encerrados por un rectángulo y asociaciones entre los actores y los casos de uso, además pueden existir relaciones de inclusión entre casos de uso y relaciones de generalización entre actores. La principal ventaja que tienen los diagramas de casos de uso es la facilidad para interpretarlos.

A continuación se detallan los casos de uso de la aplicación Web a desarrollar en el presente proyecto. Igualmente se podrán consultar individualmente en el CD adjunto a este proyecto en la subcarpeta “Casos de Uso” de la carpeta “Documentacion”.

2.4.1.1. CASO DE USO: INICIAR SESIÓN

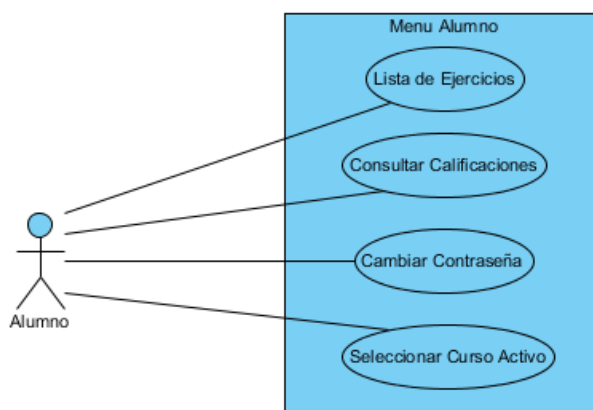
En la página principal del sistema Web se puede acceder a toda la funcionalidad del sistema Web del que ha partido este proyecto. La única modificación en esa parte es la inclusión de un sistema de inicio de sesión.



El usuario puede iniciar sesión introduciendo el usuario, que en este caso será el DNI, y la contraseña. A continuación debe pulsar el botón “Iniciar sesión” para acceder a su menú correspondiente.

2.4.1.2. CASO DE USO: MENÚ ALUMNO

Una vez se haya iniciado sesión como usuario alumno, aparecerá el menú del alumno.

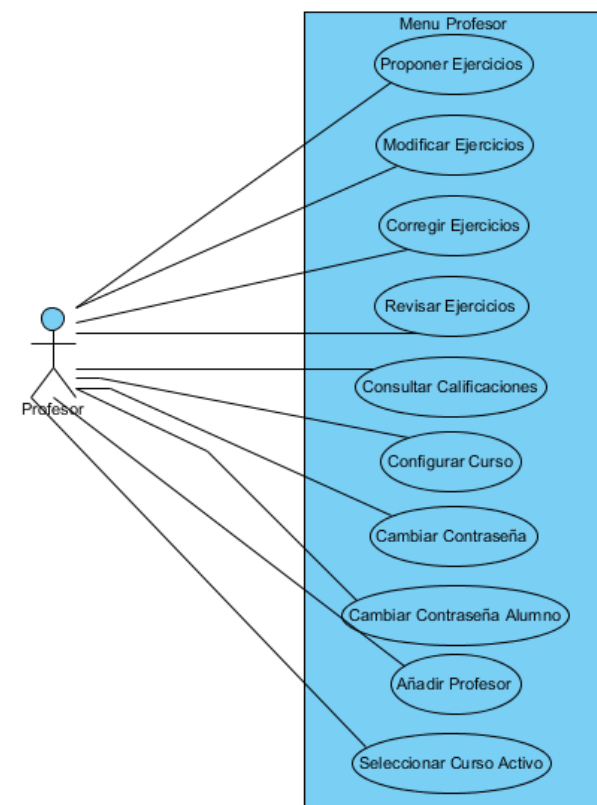


Al acceder al listado de ejercicios, el alumno podrá ver cualquier ejercicio que se haya propuesto en el curso activo. Además podrá realizarlos si se encuentra dentro del rango de fechas actuales y modificarlos cumpliendo la misma condición.

El alumno podrá consultar sus notas en los cursos que haya estado matriculado, cambiar su contraseña de acceso y podrá seleccionar el curso activo.

2.4.1.3. CASO DE USO: MENÚ PROFESOR

Una vez se haya iniciado sesión como usuario profesor, aparecerá el menú del profesor.



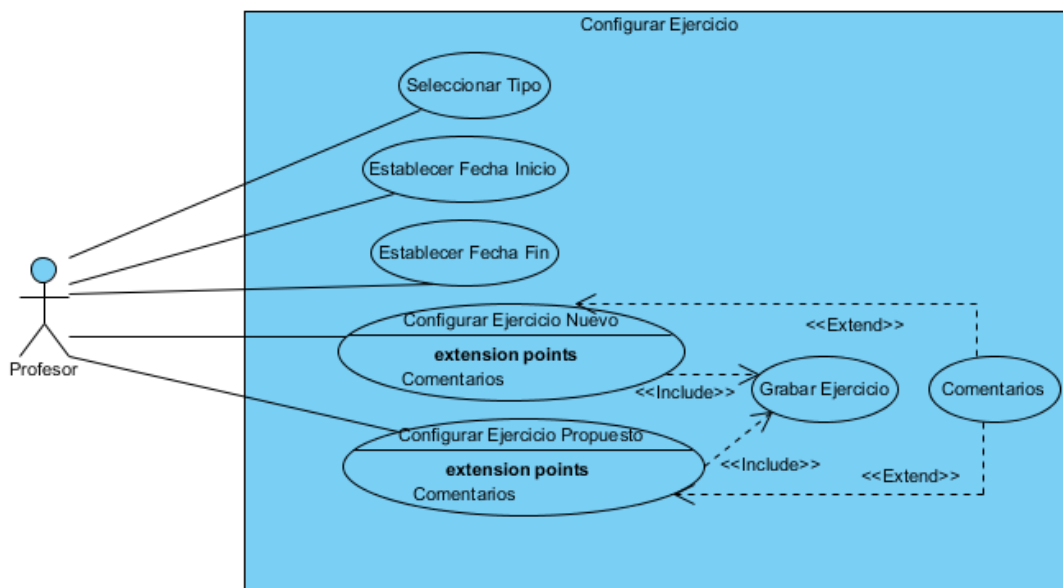
El profesor tendrá la posibilidad de proponer ejercicios de cualquier tipo de applet. Podrá modificar los ejercicios propuestos. Cuando proceda a corregir ejercicios se irán trasladando a la página de revisar ejercicios.

Será posible consultar las calificaciones por alumno y por curso. El profesor podrá también configurar un curso, añadiendo alumnos desde un fichero externo. También se podrá modificar la contraseña propia y seleccionar el curso activo.

Adicionalmente los profesores que sean administradores podrán modificar la contraseña de sus alumnos y añadir un nuevo profesor.

2.4.1.4. CASO DE USO: CONFIGURAR EJERCICIO

Cuando el profesor proceda a proponer un ejercicio tendrá que establecer el tipo, la fecha de inicio y fecha de fin del ejercicio y le aparecerán los ejercicios previamente propuestos en el sistema. Además podrá proponer un ejercicio nuevo.



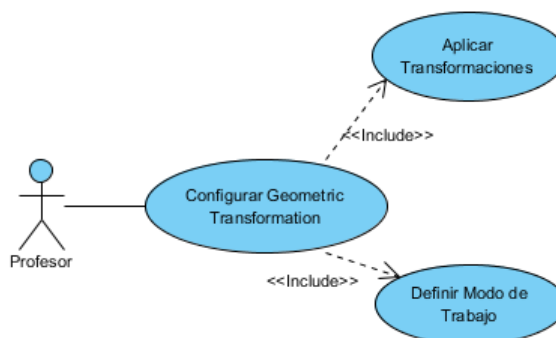
Una vez se hayan establecido el tipo de ejercicio, las fechas y seleccionado un ejercicio propuesto o configurado un ejercicio nuevo, se accederá al applet del tipo de ejercicio seleccionado.

El profesor podrá configurar el ejercicio modificando los parámetros deseados y, una vez termine, pulsará el botón “Grabar Ejercicio” para guardar el ejercicio en la base de datos. Opcionalmente, podrá añadir un comentario para identificar el ejercicio.

A continuación se expondrán los casos de uso para configurar un ejercicio de cada applet individualmente. Estos casos de uso serán subcasos de las acciones “Configurar Ejercicio Nuevo” y “Configurar Ejercicio Propuesto” por tanto mientras se esté configurando el ejercicio se tendrá acceso a la acción “Grabar Ejercicio” y “Comentarios”.

2.4.1.5. CASO DE USO: CONFIGURAR TRANSFORMACIONES GEOMÉTRICAS

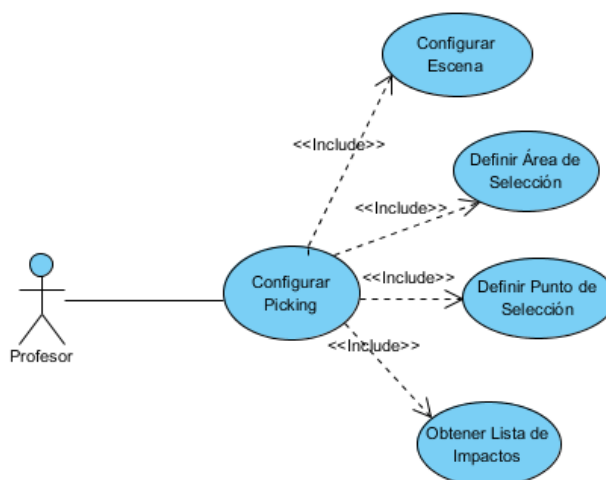
En el applet de Transformaciones geométricas se pueden realizar traslaciones, rotaciones y escalados, tanto individualmente como utilizando la matriz de transformación. También se deberá definir el modo de trabajo.



Una vez se hayan realizado las acciones se podrá guardar el ejercicio con la acción “Grabar Ejercicio” del caso de uso “Configurar ejercicio”.

2.4.1.6. CASO DE USO: CONFIGURAR TÉCNICAS DE SELECCIÓN

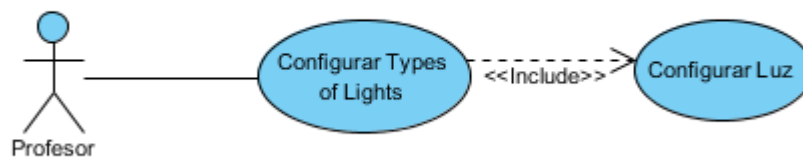
En el applet de Técnicas de selección el profesor podrá configurar una escena, definiendo el tamaño del área de selección, su posición y la posición de los objetos en la escena.



Se obtendrá una lista de impactos que será almacenada en la base de datos al guardar el ejercicio junto con la posición de los objetos y los parámetros del area de selección.

2.4.1.7. CASO DE USO: CONFIGURAR TIPOS DE LUCES

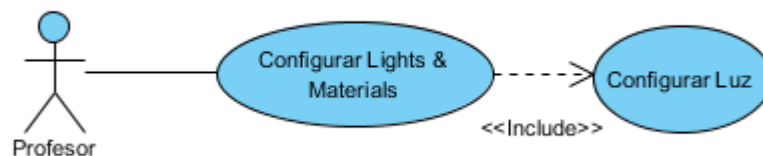
El profesor configurará una luz seleccionando el tipo de iluminación, modificando su posición, la dirección del foco y el ángulo del mismo.



Una vez configurada la luz, se guardarán los parámetros establecidos por el profesor al seleccionar el botón “guardar ejercicio”.

2.4.1.8. CASO DE USO: CONFIGURAR LUCES Y MATERIALES

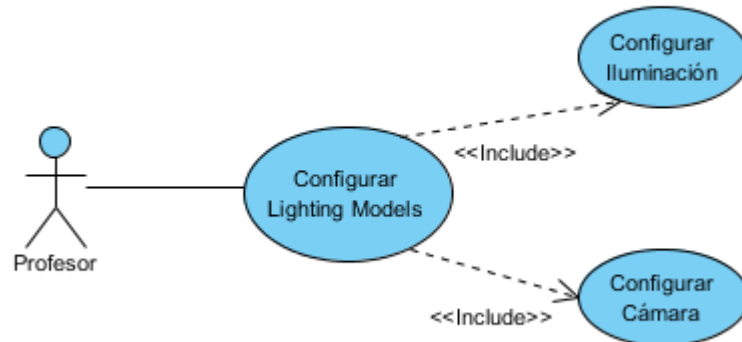
Se podrá modificar la posición, atenuación, color ambiental, difuso y especular de la luz, el coeficiente ambiental, difuso y especular del material y el exponente de Phong.



Este caso de uso es similar al anterior. Simplemente hay que configurar la luz y guardar los parámetros en la base de datos cuando el profesor guarde el ejercicio.

2.4.1.9. CASO DE USO: CONFIGURAR MODELOS DE ILUMINACIÓN

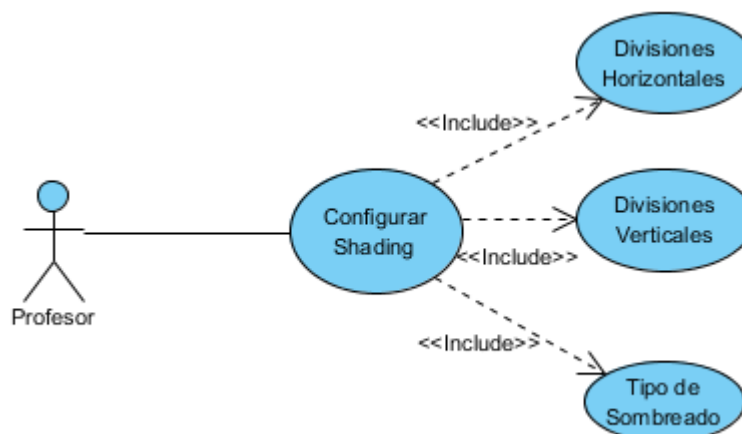
El profesor podrá elegir que tipos de iluminación desea añadir entre ambiental, difusa y especular. También podrá modificar la intensidad de la luz y los coeficientes del material.



Una vez configurada la iluminación de la escena y la cámara, el profesor procederá a guardar el ejercicio y almacenar dichos parámetros en la base de datos.

2.4.1.10. CASO DE USO: CONFIGURAR SOMBREADO

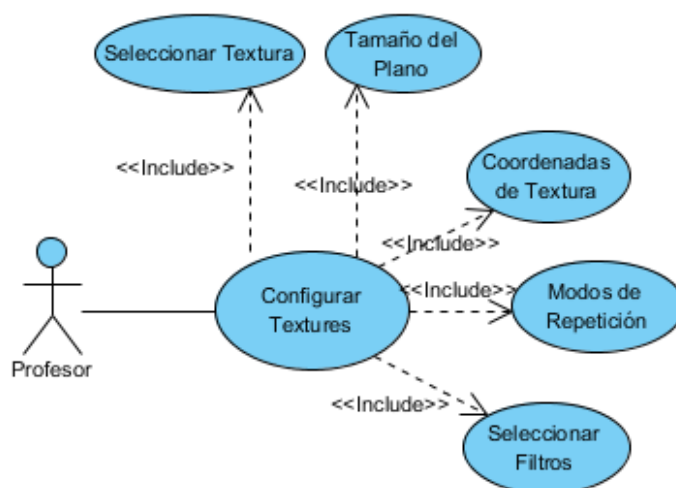
El profesor establecerá el número de divisiones horizontales y verticales, siendo éstas el mismo valor, y seleccionará el tipo de sombreado.



Una vez se hayan establecido estos parámetros se podrá guardar el ejercicio.

2.4.1.11. CASO DE USO: CONFIGURAR TEXTURAS

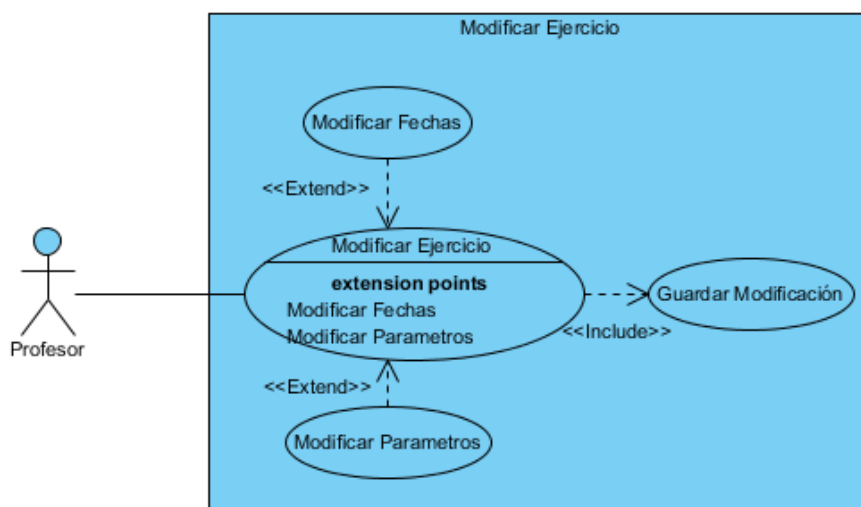
El profesor podrá seleccionar una textura, definir el tamaño del plano, establecer las coordenadas de textura de los cuatro vértices, seleccionar los modos de repetición “warpmodes” y los filtros para magnification y minification.



Una vez se hayan establecido los parámetros se guardarán en la base de datos al pulsar en “guardar ejercicio”.

2.4.1.12. CASO DE USO: MODIFICAR EJERCICIO

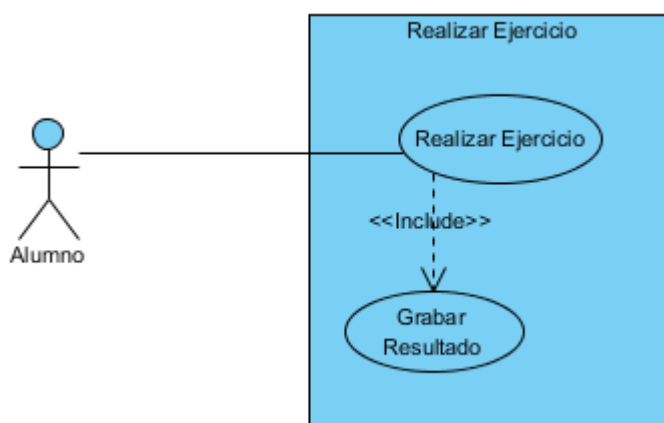
El profesor puede modificar tanto las fechas de inicio y fin, como los parámetros de un ejercicio propuesto.



Una vez se hayan modificado las fechas se accederá al applet del ejercicio que se esté modificando y se podrán modificar los parámetros y/o guardar la modificación.

2.4.1.13. CASO DE USO: REALIZAR EJERCICIO

Cuando el alumno se disponga a realizar un ejercicio activo deberá seleccionarlo y se cargarán los parámetros del ejercicio propuesto.



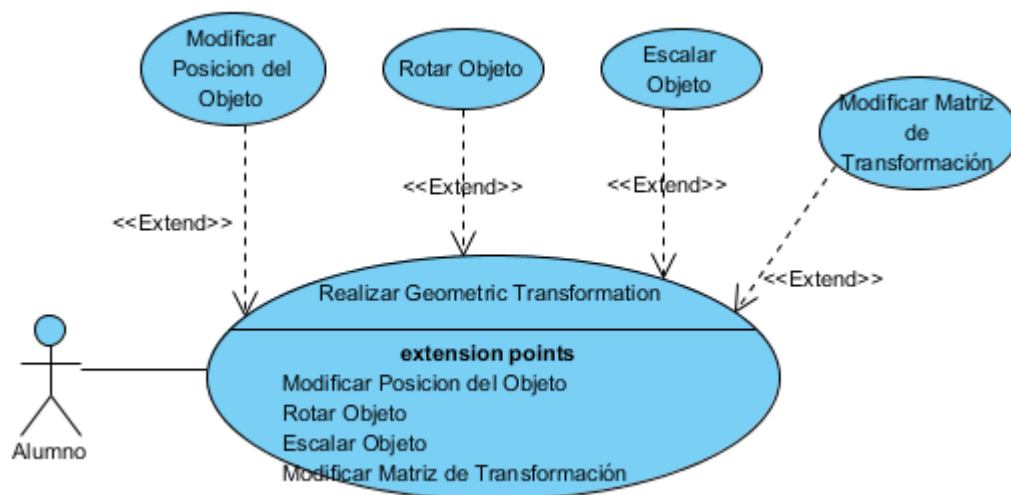
El alumno podrá realizar el ejercicio modificando los parámetros para conseguir el objetivo y, una vez termine, pulsará el botón “Grabar Resultado” para guardar el ejercicio realizado en la base de datos.

A continuación se expondrán los casos de uso para realizar un ejercicio de cada applet individualmente. Estos casos de uso serán subcasos de la acción “Realizar Ejercicio”.

2.4.1.14. CASO DE USO: REALIZAR TRANSFORMACIONES GEOMÉTRICAS

Cuando el ejercicio se muestre, el alumno podrá modificar la posición del objeto, rotarlo en los tres ejes y modificar su tamaño para conseguir el estado objetivo. Además podrá modificar la matriz de transformación y aplicarla al objeto.

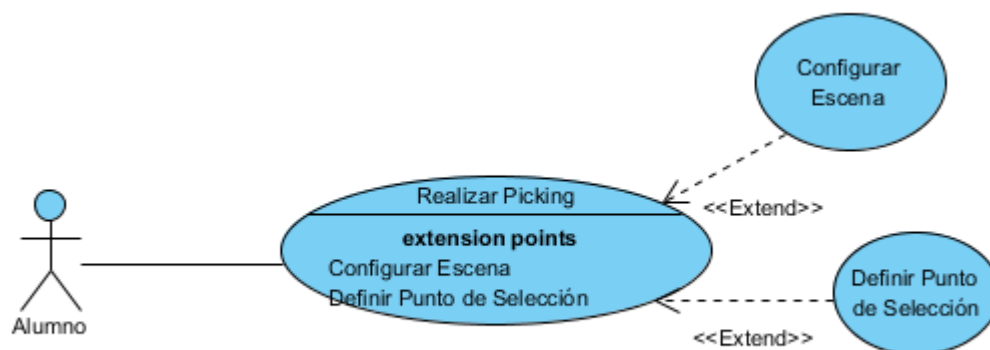
Según el modo de trabajo establecido por el profesor podrá utilizar toda la interfaz o simplemente la matriz de transformación.



Una vez se haya terminado de modificar el ejercicio, el alumno podrá guardar el resultado con la acción “Grabar Resultado” del caso de uso “Realizar Ejercicio”.

2.4.1.15. CASO DE USO: REALIZAR TÉCNICAS DE SELECCIÓN

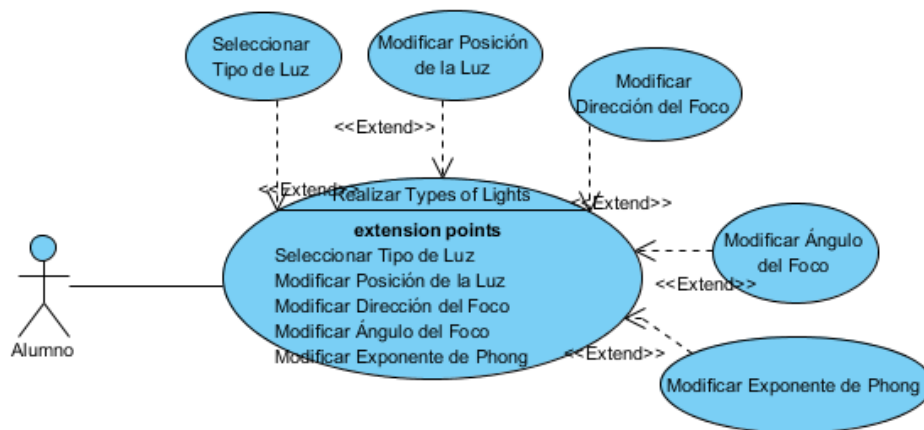
El alumno podrá configurar la posición de los objetos, y modificar la posición del área de selección para conseguir que su lista de impactos sea similar a la obtenida previamente por el profesor cuando configuró el ejercicio.



Una vez haya terminado de modificar el ejercicio podrá guardar los cambios pulsando en el botón “Grabar Resultado”.

2.4.1.16. CASO DE USO: REALIZAR TIPOS DE LUCES

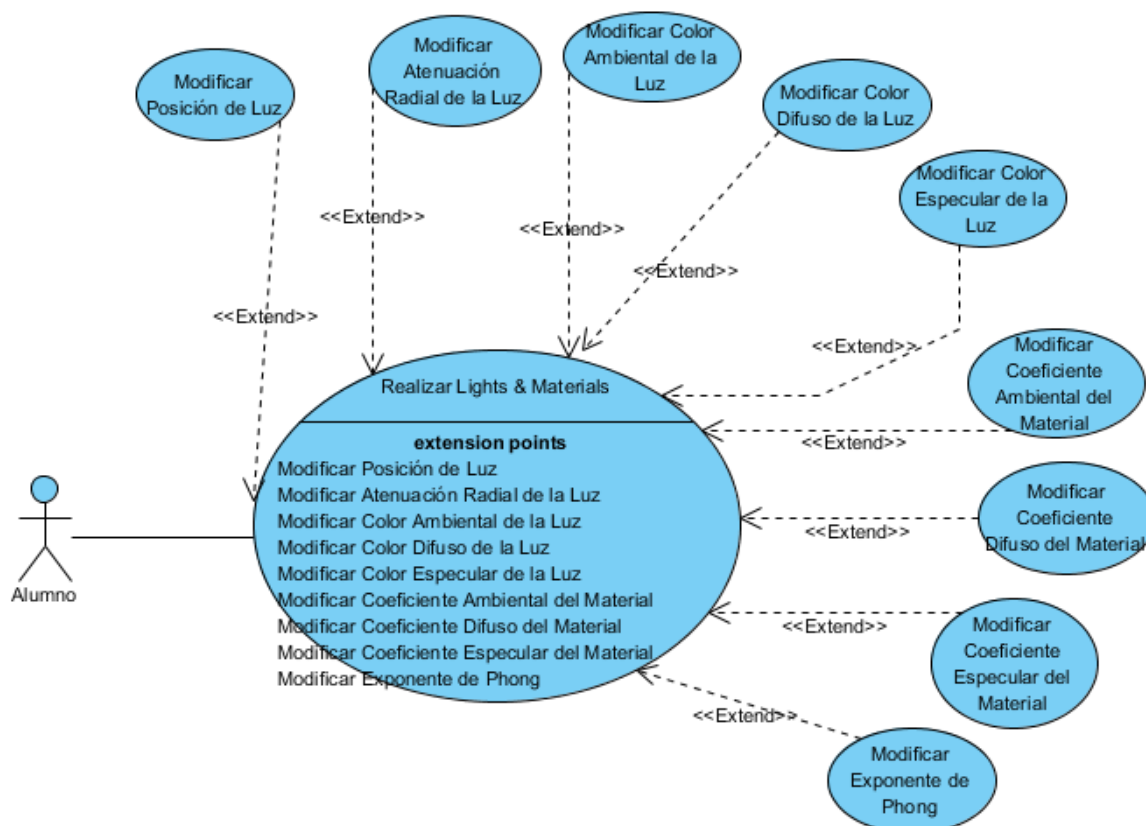
El alumno podrá seleccionar el tipo de luz y modificar su posición. Si la luz es de tipo direccional, podrá modificar su dirección. Si, por el contrario, es de tipo foco, podrá modificar también el ángulo del foco y el exponente de Phong.



Una vez se haya terminado de modificar el ejercicio se podrán guardar los cambios en la base de datos.

2.4.1.17. CASO DE USO: REALIZAR LUCES Y MATERIALES

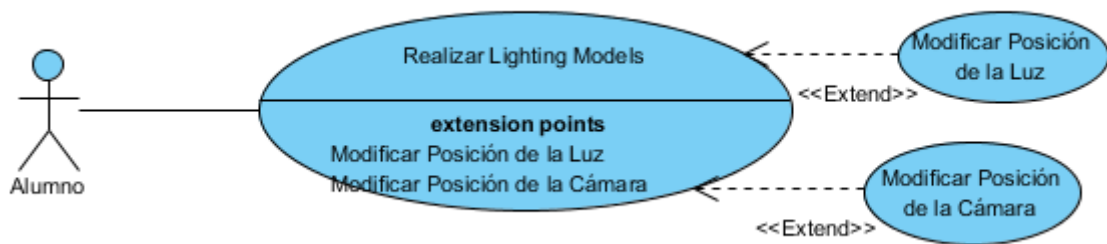
Para conseguir la iluminación objetivo marcada por el profesor, el alumno podrá modificar la posición, atenuación radial, color ambiental, color difuso y color especular de la luz. También podrá variar los coeficientes ambiental, difuso y especular del material y podrá modificar el exponente de Phong.



Una vez se haya terminado de modificar el ejercicio se podrán guardar los cambios pulsando en “Grabar Resultado”.

2.4.1.18. CASO DE USO: REALIZAR MODELOS DE ILUMINACIÓN

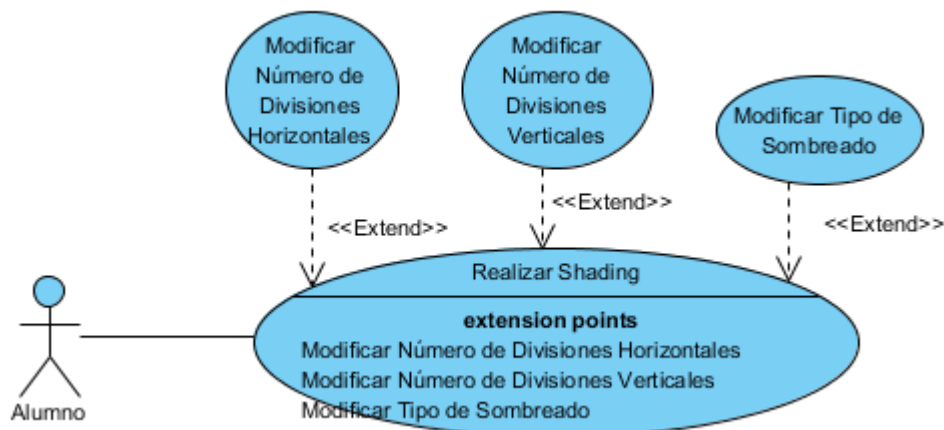
El alumno, para realizar un ejercicio de este tipo, solo podrá modificar la posición de la luz y la posición de la cámara para conseguir la iluminación configurada por el profesor.



Una vez el alumno haya terminado de modificar el ejercicio podrá guardar el resultado en la base de datos.

2.4.1.19. CASO DE USO: REALIZAR SOMBREADO

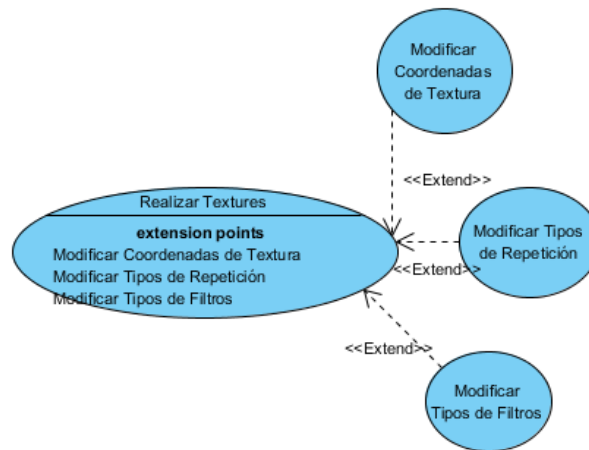
Para conseguir el sombreado objetivo, el alumno podrá cambiar el tipo de sombreado y modificar el número de divisiones horizontales y verticales. Cada una de ellas se moverá solidariamente con la otra.



Cuando se consiga el objetivo el alumno podrá guardar el resultado del ejercicio en la base de datos.

2.4.1.20. CASO DE USO: REALIZAR TEXTURAS

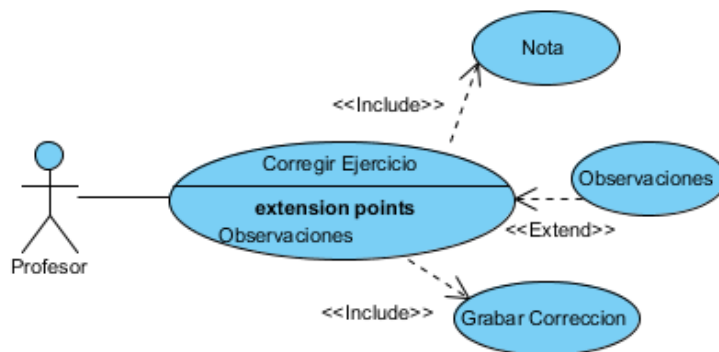
El alumno solo podrá modificar las coordenadas de textura, los tipos de repetición y los filtros de textura para conseguir el objetivo marcado por el profesor.



Cuando el alumno termine de modificar el ejercicio podrá guardar los cambios pulsando en el botón “Grabar Resultado”.

2.4.1.21. CASO DE USO: CORREGIR EJERCICIO

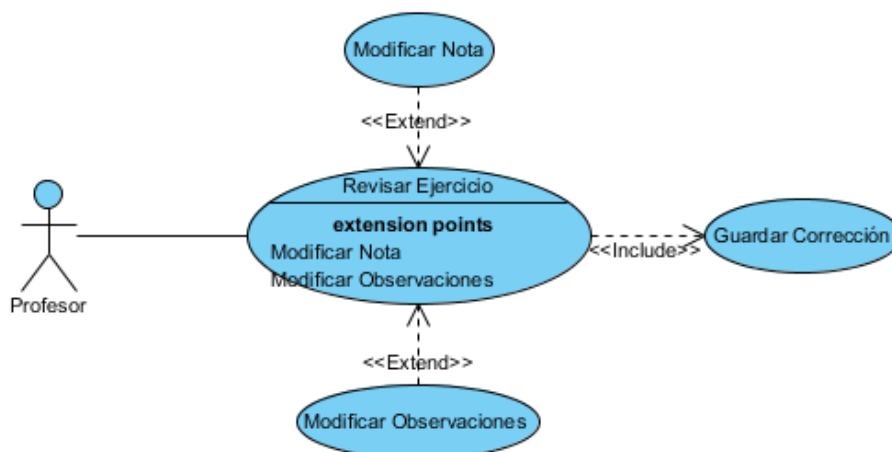
Para corregir un ejercicio, el profesor lo hará de la misma forma para cualquier tipo de applet. Simplemente escogerá el ejercicio que desea corregir y se cargará dicho ejercicio. Podrá comprobar los parámetros establecidos por el alumno y poner una calificación y, opcionalmente podrá añadir observaciones al ejercicio



Una vez se haya terminado de corregir se pulsará el botón “Guardar Corrección” y se guardará la calificación en la base de datos.

2.4.1.22. CASO DE USO: REVISAR EJERCICIO

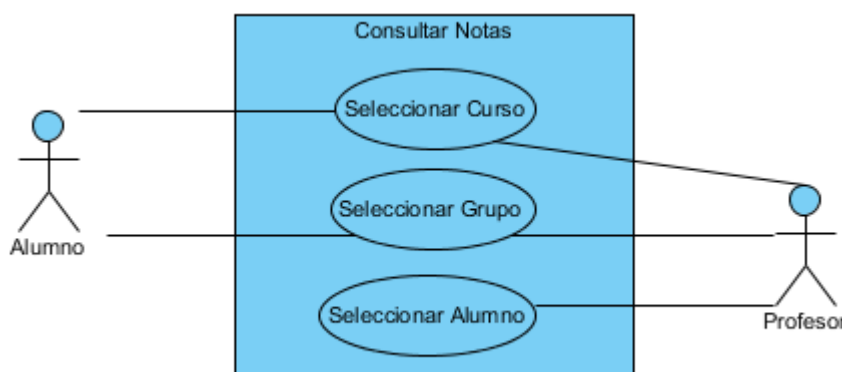
Se podrán revisar los ejercicios corregidos para poder realizar una modificación en la nota y observaciones si es necesario.



Simplemente habrá que seleccionar el ejercicio y modificar la nota y/u observaciones si se desea. Una vez realizado esto se podrá guardar la corrección.

2.4.1.23. CASO DE USO: CONSULTAR NOTAS POR ALUMNO

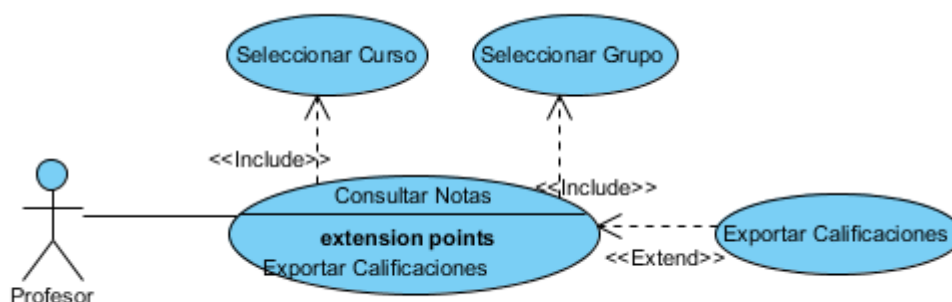
Pueden consultar las calificaciones individuales tanto los alumnos, solamente las propias, como los profesores, las de sus alumnos.



Los alumnos podrán seleccionar cualquier curso y grupo en el que estén matriculados y podrán ver sus calificaciones. El profesor deberá también seleccionar el alumno para ver sus calificaciones.

2.4.1.24. CASO DE USO: CONSULTAR NOTAS POR CURSO

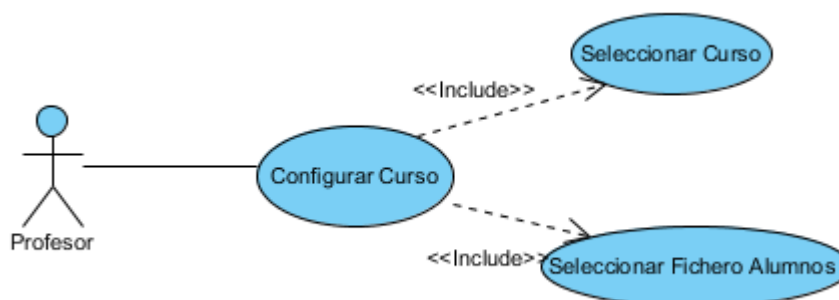
Los profesores podrán consultar todas las calificaciones de un curso para exportarlas a un fichero y facilitar su lectura.



Simplemente tienen que seleccionar un curso impartido por el profesor y su grupo y le aparecerán las calificaciones de dicho curso. Además podrá exportarlas a un fichero como se ha explicado previamente.

2.4.1.25. CASO DE USO: CONFIGURAR CURSO

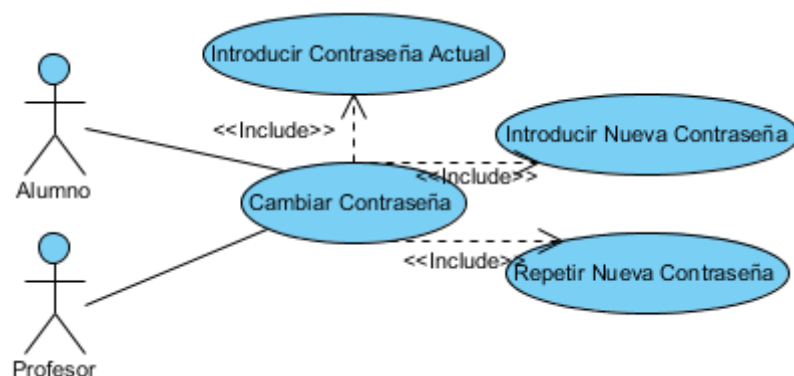
Un profesor podrá añadir un nuevo curso. Para hacerlo deberá seleccionar el curso que desea añadir y podrá añadir a los alumnos desde un fichero externo "CSV". Una vez realizado esto, podrá añadir los alumnos al curso pulsando en el botón "Configurar Curso".



Los alumnos serán creados por defecto con su contraseña igual a su DNI, pero tendrán opción a cambiar su contraseña.

2.4.1.26. CASO DE USO: CAMBIAR CONTRASEÑA PROPIA

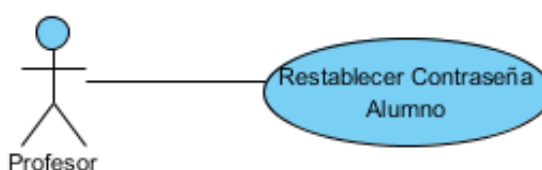
Tanto el alumno como el profesor podrán cambiar su contraseña de acceso a la aplicación desde sus respectivos menús.



Para cambiar la contraseña tendrán que introducir la contraseña actual, la nueva contraseña y repetirla para comprobar que no se haya cometido un error.

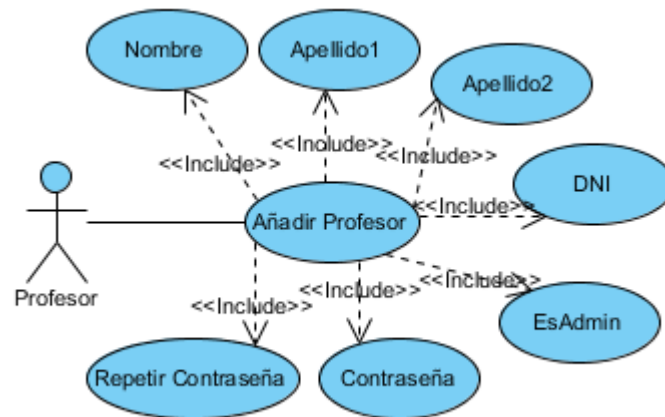
2.4.1.27. CASO DE USO: RESTABLECER CONTRASEÑA ALUMNO

Si el alumno olvida su contraseña, un profesor administrador podrá restablecer su contraseña a su DNI para que pueda volver a acceder a la aplicación.



2.4.1.28. CASO DE USO: AÑADIR PROFESOR

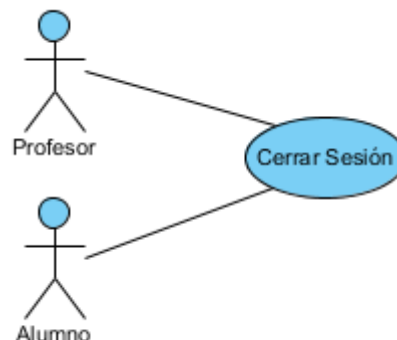
Los profesores administradores podrán añadir un nuevo profesor si lo desea. Simplemente tendrá que escribir su nombre, apellidos, DNI, decidir si será o no administrador y, por seguridad, asignarle una contraseña.



Una vez se haya terminado el nuevo profesor podrá logear sin problemas con la contraseña que le haya asignado el profesor que lo ha añadido. Se podrá cambiar la contraseña cuando se desee.

2.4.1.29. CASO DE USO: CERRAR SESIÓN

Para cerrar sesión, el usuario conectado tendrá que pulsar el botón "Cerrar Sesión" y se terminará la sesión activa.



2.4.2. NARRATIVAS DE LOS CASOS DE USO

En esta parte es donde quedan formalmente especificadas las funciones y tareas que desempeñará el nuevo sistema. Las narrativas irán enfocadas a los casos de uso de la nueva aplicación, intentando resumir la parte más específica de los applets.

2.4.2.1. CASO DE USO: INICIAR SESIÓN

Las narrativas de casos de uso asociadas a iniciar sesión son las siguientes:

Actor principal: Invitado

Precondiciones: -

Camino principal:

1. El usuario introduce el DNI.
2. El usuario introduce la contraseña.
3. El usuario pulsa el botón de “Iniciar sesión”.
- 4.a. El sistema valida que el usuario existe y se inicia sesión correctamente.

Camino alternativo:

- 4.b.1. El sistema detecta que no se ha insertado el DNI o contraseña.
- 4.b.2. Repetir desde el paso 1.
- 4.c.1. El sistema detecta que el DNI o contraseña insertados no son correctos.
- 4.c.2. Repetir desde el paso 1.

Postcondiciones:

El sistema redirige al usuario al menú de alumno o profesor según la sesión que se haya iniciado.

2.4.2.2. CASO DE USO: MENÚ ALUMNO

Las narrativas de casos de uso asociadas al menú del alumno son las siguientes:

Actor principal: Alumno

Precondiciones: Debe haber iniciado sesión un alumno

Camino principal:

- 1.a. El alumno accede a la lista de ejercicios.

Camino alternativo:

1.b.1. El alumno pulsa “Consultar calificaciones” y el sistema carga la página seleccionada.

1.b.2. El alumno selecciona “Cambiar contraseña” y el sistema carga dicha página.

1.b.3. El alumno modifica el curso activo y el sistema recarga la lista de ejercicios.

Postcondiciones:

El sistema cargará la página correspondiente a la acción que se haya realizado.

2.4.2.3. MENÚ PROFESOR

Las narrativas de casos de uso asociadas al menú del profesor son las siguientes:

Actor principal: Profesor.

Precondiciones: Debe haber iniciado sesión un profesor o un administrador.

Camino principal:

1.a. El profesor accede a la lista de ejercicios pendientes de corregir.

Camino alternativo:

1.b.1. El profesor pulsa en “Proponer ejercicio” y el sistema carga la página requerida.

1.b.2. El profesor pulsa en “Modificar ejercicio” y el sistema carga la página requerida.

1.b.3. El profesor pulsa en “Revisar ejercicio” y el sistema carga la página respectiva.

1.b.5. (Solo para administrador) El profesor pulsa en “Configurar nuevo curso” y el sistema carga la página requerida.

1.b.4. El profesor pulsa en “Modificar contraseña” y el sistema carga la página seleccionada.

1.b.5. El profesor pulsa en “Restablecer contraseña alumno” y el sistema carga la página requerida.

1.b.6. (Solo para administrador) El profesor selecciona “Añadir nuevo profesor” y el sistema carga dicha página.

1.b.7. El profesor modifica el curso activo y el sistema carga la lista de ejercicios pendientes de corregir.

Postcondiciones:

El sistema cargará la página correspondiente a la acción que se haya realizado y el profesor podrá realizar las acciones pertinentes.

2.4.2.4. CASO DE USO: CONFIGURAR EJERCICIO

Las narrativas de casos de uso asociadas a proponer un ejercicio son las siguientes:

Actor principal: Profesor

Precondiciones: Debe haber iniciado sesión un profesor o un administrador.

Camino principal:

1. El profesor selecciona el tipo de ejercicio que desea plantear.
2. El profesor introduce la fecha de inicio del ejercicio y el sistema habilita la edición de la fecha de fin.
3. El profesor introduce la fecha de fin de ejercicio y el sistema carga los datos de los ejercicios previamente propuestos del tipo seleccionado y muestra el botón para configurar un ejercicio nuevo.
- 4.a. El profesor pulsa en “Configurar nuevo ejercicio”.
5. El profesor modifica los parámetros del ejercicio.
6. El profesor escribe los comentarios para identificar el ejercicio.
7. El profesor pulsa en “Guardar ejercicio” y el sistema guardará el ejercicio en la base de datos.

Camino alternativo:

- 4.b.1. El profesor pulsa en el botón “Restablecer fechas”.
- 4.b.2. Repetir desde el paso 1.
- 4.c.1. El profesor selecciona un ejercicio de la lista de ejercicios propuestos y el sistema carga el ejercicio.

Postcondiciones:

El sistema redirige al profesor al applet del ejercicio seleccionado con los parámetros establecidos o al mismo applet con los parámetros por defecto.

2.4.2.5. CASO DE USO: CONFIGURAR TRANSFORMACIONES GEOMÉTRICAS

Las narrativas de casos de uso asociadas a configurar un ejercicio de transformaciones geométricas son las siguientes:

Actor principal: Profesor

Precondiciones: Debe haber iniciado sesión un profesor o un administrador.

Camino principal:

1. El profesor aplica las transformaciones geométricas que desee.
2. El establece el modo de trabajo del ejercicio.
3. El profesor escribe los comentarios para identificar el ejercicio.
- 4.a. El profesor pulsa en “Guardar ejercicio”, se guardará el ejercicio en la base de datos y el sistema cargará la página de inicio del profesor.

Camino alternativo:

- 4.b.1. Se produce un error al guardar el ejercicio. El sistema notifica el error y carga el menú de inicio del profesor.
- 4.b.2. Repetir desde el paso 1.

Postcondiciones:

El sistema redirige al profesor al menú de inicio notificando que el ejercicio se ha guardado correctamente.

2.4.2.6. CASO DE USO: CONFIGURAR TÉCNICAS DE SELECCIÓN

Las narrativas de casos de uso asociadas a configurar un ejercicio de técnicas de selección son las siguientes:

Actor principal: Profesor

Precondiciones: Debe haber iniciado sesión un profesor o un administrador.

Camino principal:

1. El profesor configura la escena colocando los objetos donde desee.
2. El profesor define el tamaño del área de selección.
3. El profesor define la posición del área de selección.
4. El sistema irá actualizando la lista de impactos cada vez que se realicen los pasos 1,2 y 3.
5. El profesor escribe los comentarios para identificar el ejercicio.

- 6.a. El profesor pulsa en “Guardar ejercicio”, se guardará el ejercicio en la base de datos y el sistema cargará la página de inicio del profesor.

Camino alternativo:

- 6.b.1. Se produce un error al guardar el ejercicio. El sistema notifica el error y carga el menú de inicio del profesor.
- 6.b.2. Repetir desde el paso 1.

Postcondiciones:

El sistema redirige al profesor al menú de inicio notificando que el ejercicio se ha guardado correctamente.

2.4.2.7. CASO DE USO: CONFIGURAR TIPOS DE LUCES

Las narrativas de casos de uso asociadas a configurar un ejercicio de tipos de luces son las siguientes:

Actor principal: Profesor

Precondiciones: Debe haber iniciado sesión un profesor o un administrador.

Camino principal:

1. El profesor configura una luz modificando los parámetros del applet.
2. El profesor escribe los comentarios para identificar el ejercicio.
- 3.a. El profesor pulsa en “Guardar ejercicio”, se guardará el ejercicio en la base de datos y el sistema cargará la página de inicio del profesor.

Camino alternativo:

- 3.b.1. Se produce un error al guardar el ejercicio. El sistema notifica el error y carga el menú de inicio del profesor.
- 3.b.2. Repetir desde el paso 1.

Postcondiciones:

El sistema redirige al profesor al menú de inicio notificando que el ejercicio se ha guardado correctamente.

2.4.2.8. CASO DE USO: CONFIGURAR LUCES Y MATERIALES

Las narrativas de casos de uso asociadas a configurar un ejercicio de luces y materiales son las siguientes:

Actor principal: Profesor

Precondiciones: Debe haber iniciado sesión un profesor o un administrador.

Camino principal:

1. El profesor configura una luz modificando los parámetros del applet.
2. El profesor escribe los comentarios para identificar el ejercicio.
- 3.a. El profesor pulsa en “Guardar ejercicio”, se guardará el ejercicio en la base de datos y el sistema cargará la página de inicio del profesor.

Camino alternativo:

- 3.b.1. Se produce un error al guardar el ejercicio. El sistema notifica el error y carga el menú de inicio del profesor.
- 3.b.2. Repetir desde el paso 1.

Postcondiciones:

El sistema redirige al profesor al menú de inicio notificando que el ejercicio se ha guardado correctamente.

2.4.2.9. CASO DE USO: CONFIGURAR MODELOS DE ILUMINACIÓN

Las narrativas de casos de uso asociadas a configurar un ejercicio de modelos de iluminación son las siguientes:

Actor principal: Profesor

Precondiciones: Debe haber iniciado sesión un profesor o un administrador.

Camino principal:

1. El profesor configura la iluminación de la escena.
2. El profesor configura la posición y dirección de la cámara.
3. El profesor escribe los comentarios para identificar el ejercicio.

- 4.a. El profesor pulsa en “Guardar ejercicio”, se guardará el ejercicio en la base de datos y el sistema cargará la página de inicio del profesor.

Camino alternativo:

- 4.b.1. Se produce un error al guardar el ejercicio. El sistema notifica el error y carga el menú de inicio del profesor.
- 4.b.2. Repetir desde el paso 1.

Postcondiciones:

El sistema redirige al profesor al menú de inicio notificando que el ejercicio se ha guardado correctamente.

2.4.2.10. CASO DE USO: CONFIGURAR SOMBREADO

Las narrativas de casos de uso asociadas a configurar un ejercicio de sombreado son las siguientes:

Actor principal: Profesor

Precondiciones: Debe haber iniciado sesión un profesor o un administrador.

Camino principal:

1. El profesor modifica el número de divisiones horizontales y las divisiones verticales se mueven solidariamente.
2. El profesor modifica el número de divisiones horizontales y las divisiones verticales se mueven solidariamente.
3. El profesor selecciona el tipo de sombreado y visualización de la esfera.
4. El profesor escribe los comentarios para identificar el ejercicio.
- 4.a. El profesor pulsa en “Guardar ejercicio”, se guardará el ejercicio en la base de datos y el sistema cargará la página de inicio del profesor.

Camino alternativo:

- 4.b.1. Se produce un error al guardar el ejercicio. El sistema notifica el error y carga el menú de inicio del profesor.
- 4.b.2. Repetir desde el paso 1.

Postcondiciones:

El sistema redirige al profesor al menú de inicio notificando que el ejercicio se ha guardado correctamente.

2.4.2.11. CASO DE USO: CONFIGURAR TEXTURAS

Las narrativas de casos de uso asociadas a configurar un ejercicio de texturas son las siguientes:

Actor principal: Profesor

Precondiciones: Debe haber iniciado sesión un profesor o un administrador.

Camino principal:

1. El profesor selecciona la textura.
2. El profesor especifica el tamaño del plano.
3. El profesor establece las coordenadas de textura de los cuatro vértices.
4. El profesor selecciona los modos de repetición.
5. El profesor selecciona los filtros de textura.
6. El profesor escribe los comentarios para identificar el ejercicio.
- 7.a. El profesor pulsa en “Guardar ejercicio”, se guardará el ejercicio en la base de datos y el sistema cargará la página de inicio del profesor.

Camino alternativo:

- 7.b.1. Se produce un error al guardar el ejercicio. El sistema notifica el error y carga el menú de inicio del profesor.
- 7.b.2. Repetir desde el paso 1.

Postcondiciones:

El sistema redirige al profesor al menú de inicio notificando que el ejercicio se ha guardado correctamente.

2.4.2.12. CASO DE USO: MODIFICAR EJERCICIO

Las narrativas de casos de uso asociadas a modificar un ejercicio son las siguientes:

Actor principal: Profesor

Precondiciones: Debe haber iniciado sesión un profesor o un administrador y el ejercicio debe haber sido propuesto por el actor principal.

Camino principal:

1. El profesor modifica la fecha de inicio del ejercicio.
2. El profesor modifica la fecha de fin del ejercicio.

3. El profesor pulsa en “Modificar ejercicio” y se carga el applet del ejercicio seleccionado con los parámetros configurados al proponer el ejercicio.
4. El profesor modifica los parámetros que desee en el applet del ejercicio.
5. El profesor modifica los comentarios para identificar el ejercicio.
- 6.a. El profesor pulsa en “Guardar modificación”, se modificará el ejercicio y el sistema cargará la página de inicio del profesor.

Camino alternativo:

- 6.b.1. Se produce un error al modificar el ejercicio. El sistema notifica el error y carga el menú de inicio del profesor.
- 6.b.2. Repetir desde el paso 1.

Postcondiciones:

El sistema redirige al profesor al menú de inicio notificando que el ejercicio se ha modificado correctamente.

2.4.2.13. CASO DE USO: REALIZAR EJERCICIO

Las narrativas de casos de uso asociadas a realizar un ejercicio son las siguientes:

Actor principal: Alumno

Precondiciones: Debe haber iniciado sesión un alumno. El ejercicio debe haber sido propuesto en el curso y grupo activos del alumno. La fecha actual debe estar entre el rango de fechas del ejercicio.

Camino principal:

1. El alumno selecciona el ejercicio que desea realizar y el sistema carga el applet del ejercicio seleccionado con los parámetros establecidos por el profesor para dicho ejercicio.
- 2.a. El alumno realiza las modificaciones que considere oportunas y pulsará el botón “Guardar resultado”. El sistema cargará la página de inicio del alumno.

Camino alternativo:

2.b.1. Se produce un error al guardar el resultado. El sistema notifica el error y carga el menú de inicio del alumno.

2.b.2. Repetir desde el paso 1.

Postcondiciones:

El sistema redirige al alumno al menú de inicio notificando que el resultado se ha guardado correctamente.

2.4.2.14. CASO DE USO: REALIZAR TRANSFORMACIONES GEOMÉTRICAS

Las narrativas de casos de uso asociadas a realizar un ejercicio de transformaciones geométricas son las siguientes:

Actor principal:Alumno

Precondiciones: Debe haber iniciado sesión un alumno.El ejercicio debe haber sido propuesto en el curso y grupo activos del alumno. La fecha actual debe estar entre el rango de fechas del ejercicio.

Camino principal:

1. El alumno modifica la posición del objeto. Si se consigue el objetivo el objeto parpadeará.
2. El alumno rota el objeto.
3. El alumno modifica el tamaño del objeto.
4. El alumno modifica la matriz de transformación.
5. Si se consigue el objetivo en los pasos 1,2,3 ó 4 el objeto parpadeará y la media de las distancias se mostrará de color verde.
- 6.a. El alumno pulsa en “Guardar resultado”, se guardará el resultado del ejercicio en la base de datos y el sistema cargará la página de inicio del alumno.

Camino alternativo:

6.b.1. Se produce un error al guardar el resultado. El sistema notifica el error y carga el menú de inicio del alumno.

6.b.2. Repetir desde el paso 1.

Postcondiciones:

El sistema redirige al alumno al menú de inicio notificando que el ejercicio se ha guardado correctamente.

2.4.2.15. CASO DE USO: REALIZAR TÉCNICAS DE SELECCIÓN

Las narrativas de casos de uso asociadas a realizar un ejercicio de técnicas de selección son las siguientes:

Actor principal: Alumno

Precondiciones: Debe haber iniciado sesión un alumno. El ejercicio debe haber sido propuesto en el curso y grupo activos del alumno. La fecha actual debe estar entre el rango de fechas del ejercicio.

Camino principal:

1. El alumno configura la escena colocando los objetos correctamente para conseguir el objetivo.
2. El alumno modifica la posición del area de selección.
3. Si se consigue el objetivo en los pasos 1 ó 2 la lista de impactos parpadeará.
- 4.a. El alumno pulsa en “Guardar resultado”, se guardará el resultado del ejercicio en la base de datos y el sistema cargará la página de inicio del alumno.

Camino alternativo:

- 4.b.1. Se produce un error al guardar el resultado. El sistema notifica el error y carga el menú de inicio del alumno.
- 4.b.2. Repetir desde el paso 1.

Postcondiciones:

El sistema redirige al alumno al menú de inicio notificando que el ejercicio se ha guardado correctamente.

2.4.2.16. CASO DE USO: REALIZAR TIPOS DE LUCES

Las narrativas de casos de uso asociadas a realizar un ejercicio de tipos de luces son las siguientes:

Actor principal: Alumno

Precondiciones: Debe haber iniciado sesión un alumno. El ejercicio debe haber sido propuesto en el curso y grupo activos del alumno. La fecha actual debe estar entre el rango de fechas del ejercicio.

Camino principal:

1. El alumno selecciona el tipo de luz.
2. El alumno modifica la posición de la luz.
3. El alumno modifica la dirección del foco.
4. El alumno modifica el ángulo del foco.
5. El alumno modifica el exponente de Phong.
- 6.a. El alumno pulsa en “Guardar resultado”, se guardará el resultado del ejercicio en la base de datos y el sistema cargará la página de inicio del alumno.

Camino alternativo:

- 6.b.1. Se produce un error al guardar el resultado. El sistema notifica el error y carga el menú de inicio del alumno.
- 6.b.2. Repetir desde el paso 1.

Postcondiciones:

El sistema redirige al alumno al menú de inicio notificando que el ejercicio se ha guardado correctamente.

2.4.2.17. CASO DE USO: REALIZAR LUCES Y MATERIALES

Las narrativas de casos de uso asociadas a realizar un ejercicio de luces y materiales son las siguientes:

Actor principal: Alumno

Precondiciones: Debe haber iniciado sesión un alumno. El ejercicio debe haber sido propuesto en el curso y grupo activos del alumno. La fecha actual debe estar entre el rango de fechas del ejercicio.

Camino principal:

1. El alumno modifica la posición de la luz.

2. El alumno modifica la atenuación radial de la luz.
3. El alumno modifica el color ambiental de la luz.
4. El alumno modifica el color difuso de la luz.
5. El alumno modifica el color especular de la luz.
6. El alumno modifica el coeficiente ambiental del material.
7. El alumno modifica el coeficiente difuso del material.
8. El alumno modifica el coeficiente especular del material.
9. El alumno modifica el exponente de Phong del material.
- 10.a. El alumno pulsa en “Guardar resultado”, se guardará el resultado del ejercicio en la base de datos y el sistema cargará la página de inicio del alumno.

Camino alternativo:

- 10.b.1. Se produce un error al guardar el resultado. El sistema notifica el error y carga el menú de inicio del alumno.
- 10.b.2. Repetir desde el paso 1.

Postcondiciones:

El sistema redirige al alumno al menú de inicio notificando que el ejercicio se ha guardado correctamente.

2.4.2.18. CASO DE USO: REALIZAR MODELOS DE ILUMINACIÓN

Las narrativas de casos de uso asociadas a realizar un ejercicio de modelos de iluminación son las siguientes:

Actor principal: Alumno

Precondiciones: Debe haber iniciado sesión un alumno. El ejercicio debe haber sido propuesto en el curso y grupo activos del alumno. La fecha actual debe estar entre el rango de fechas del ejercicio.

Camino principal:

1. El alumno modifica la posición de la luz.
2. El alumno modifica la posición de la cámara.
3. Si se consigue el objetivo en los pasos 1 ó 2 los valores de iluminación obtenidos del alumno y del profesor parpadearán.

- 4.a. El alumno pulsa en “Guardar resultado”, se guardará el resultado del ejercicio en la base de datos y el sistema cargará la página de inicio del alumno.

Camino alternativo:

- 4.b.1. Se produce un error al guardar el resultado. El sistema notifica el error y carga el menú de inicio del alumno.
- 4.b.2. Repetir desde el paso 1.

Postcondiciones:

El sistema redirige al alumno al menú de inicio notificando que el ejercicio se ha guardado correctamente.

2.4.2.19. CASO DE USO: REALIZAR SOMBREADO

Las narrativas de casos de uso asociadas a realizar un ejercicio de sombreado son las siguientes:

Actor principal: Alumno

Precondiciones: Debe haber iniciado sesión un alumno. El ejercicio debe haber sido propuesto en el curso y grupo activos del alumno. La fecha actual debe estar entre el rango de fechas del ejercicio.

Camino principal:

1. El alumno modifica el número de divisiones horizontales.
2. El alumno modifica el número de divisiones verticales.
3. El alumno selecciona el tipo de sombreado y visualización.
4. Si se consigue el objetivo en los pasos 1, 2 ó 3 la esfera parpadea.
- 5.a. El alumno pulsa en “Guardar resultado”, se guardará el resultado del ejercicio en la base de datos y el sistema cargará la página de inicio del alumno.

Camino alternativo:

- 5.b.1. Se produce un error al guardar el resultado. El sistema notifica el error y carga el menú de inicio del alumno.
- 5.b.2. Repetir desde el paso 1.

Postcondiciones:

El sistema redirige al alumno al menú de inicio notificando que el ejercicio se ha guardado correctamente.

2.4.2.20. CASO DE USO: REALIZAR TEXTURAS

Las narrativas de casos de uso asociadas a realizar un ejercicio de texturas son las siguientes:

Actor principal: Alumno

Precondiciones: Debe haber iniciado sesión un alumno. El ejercicio debe haber sido propuesto en el curso y grupo activos del alumno. La fecha actual debe estar entre el rango de fechas del ejercicio.

Camino principal:

1. El alumno modifica las coordenadas de textura de los cuatro vértices.
2. El alumno selecciona el tipo de repetición.
3. El alumno selecciona el filtro de textura.
4. Si se consigue el objetivo en los pasos 1, 2 ó 3 la textura parpadeará.
- 5.a. El alumno pulsa en “Guardar resultado”, se guardará el resultado del ejercicio en la base de datos y el sistema cargará la página de inicio del alumno.

Camino alternativo:

- 5.b.1. Se produce un error al guardar el resultado. El sistema notifica el error y carga el menú de inicio del alumno.
- 5.b.2. Repetir desde el paso 1.

Postcondiciones:

El sistema redirige al alumno al menú de inicio notificando que el ejercicio se ha guardado correctamente.

2.4.2.21. CASO DE USO: CORREGIR EJERCICIO

Las narrativas de casos de uso asociadas a corregir un ejercicio son las siguientes:

Actor principal: Profesor

Precondiciones: Debe haber iniciado sesión un profesor o un administrador.

El ejercicio debe haber sido propuesto por el actor principal.

El ejercicio debe haber sido realizado por un alumno del curso y grupo activo.

Camino principal:

1. El profesor selecciona el ejercicio que desea corregir.
2. El sistema carga la lista de ejercicios realizados por los alumnos para dicho ejercicio.
3. El profesor selecciona el ejercicio del alumno que desea corregir y se carga el applet del ejercicio seleccionado con los parámetros configurados al proponer el ejercicio y los parámetros establecidos por el alumno.
4. El profesor asigna una calificación al ejercicio.
5. El profesor escribe las observaciones de la corrección.
- 6.a. El profesor pulsa en “Guardar corrección”, se guardará la corrección y el sistema cargará la página de inicio del profesor.

Camino alternativo:

- 6.b.1. Se produce un error al corregir el ejercicio. El sistema notifica el error y carga el menú de inicio del profesor.
- 6.b.2. Repetir desde el paso 1.

Postcondiciones:

El sistema redirige al profesor al menú de inicio notificando que el ejercicio ha sido corregido satisfactoriamente.

2.4.2.22. CASO DE USO: REVISAR EJERCICIO

Las narrativas de casos de uso asociadas a revisar un ejercicio son las siguientes:

Actor principal: Profesor

Precondiciones: Debe haber iniciado sesión un profesor o un administrador.

El ejercicio debe haber sido propuesto por el actor principal.

El ejercicio debe haber sido realizado por un alumno del

curso y grupo activo. El ejercicio debe haber sido corregido por el profesor logueado.

Camino principal:

1. El profesor selecciona el ejercicio que desea revisar.
2. El sistema carga la lista de correcciones asignadas por el profesor para dicho ejercicio.
3. El profesor selecciona el ejercicio que desea revisar y se carga el applet del ejercicio seleccionado con los parámetros configurados al proponer el ejercicio, los parámetros establecidos por el alumno, la calificación otorgada por el profesor y las observaciones para dicho ejercicio.
4. El profesor modifica la calificación del ejercicio.
5. El profesor modifica las observaciones de la corrección.
- 6.a. El profesor pulsa en “Guardar corrección”, se guardará la corrección y el sistema cargará la página de inicio del profesor.

Camino alternativo:

- 6.b.1. Se produce un error al corregir el ejercicio. El sistema notifica el error y carga el menú de inicio del profesor.
- 6.b.2. Repetir desde el paso 1.

Postcondiciones:

El sistema redirige al profesor al menú de inicio notificando que el ejercicio ha sido corregido satisfactoriamente.

2.4.2.23. CASO DE USO: CONSULTAR NOTAS POR ALUMNO

Las narrativas de casos de uso asociadas a consultar las notas por alumno son las siguientes:

Actores principales: Profesor y alumno

Precondiciones: Debe haber una sesión iniciada.

Camino principal alumno:

1. El alumno selecciona uno de los cursos en los que está matriculado.
2. El sistema carga el grupo del curso seleccionado en el que está matriculado el alumno.

3. El sistema carga las calificaciones obtenidas por el alumno en el curso y grupo seleccionados.

Camino principal profesor:

1. El profesor selecciona uno de los cursos que imparte.
2. El sistema carga los grupos del curso seleccionado que imparta el profesor.
3. El profesor selecciona uno de los grupos que imparte.
4. El sistema carga los alumnos matriculados en el curso y grupo seleccionados.
5. El profesor selecciona uno de los alumnos.
6. El sistema carga las calificaciones obtenidas por el alumno en el curso y grupo seleccionados.

Camino alternativo: -

Postcondiciones: -

2.4.2.24. CASO DE USO: CONSULTAR NOTAS POR CURSO

Las narrativas de casos de uso asociadas a consultar las notas por alumno son las siguientes:

Actor principal: Profesor

Precondiciones: Debe haber iniciado sesión un profesor o un administrador.

Camino principal:

1. El profesor selecciona uno de los cursos que imparte.
2. El sistema carga los grupos del curso seleccionado que imparta el profesor.
3. El profesor selecciona uno de los grupos que imparte.
4. El sistema carga todas las calificaciones obtenidas en dicho curso y grupo.
5. El profesor pulsa en “Exportar a fichero CSV”.
6. El profesor selecciona el nombre y la ruta que vaya a tener el fichero.
7. El profesor guarda el fichero.

Camino alternativo: -

Postcondiciones:

El profesor podrá exportar el fichero tantas veces como sea necesario.

2.4.2.25. CASO DE USO: CONFIGURAR CURSO

Las narrativas de casos de uso asociadas a configurar un curso son las siguientes:

Actor principal: Profesor

Precondiciones: Debe haber iniciado sesión un profesor o un administrador.

Camino principal:

1. El sistema carga un listado de cursos almacenados en la base de datos.
2. El profesor selecciona uno de esos cursos.
3. El profesor selecciona un grupo 'A' o 'B'.
4. El profesor selecciona un fichero "CSV" con los alumnos que estarán matriculados en el curso.
- 5.a. El profesor pulsará en "Crear Curso" y el sistema cargará los alumnos en la base de datos y les asignará el curso y profesor correspondiente.

Camino alternativo:

- 5.b.1. Si el fichero "CSV" está vacío el sistema mostrará un error y podrá añadir otro fichero.

Postcondiciones:

Se creará el curso asignado al profesor y con los alumnos pasador por el fichero matriculados.

2.4.2.26. CASO DE USO: CAMBIAR CONTRASEÑA PROPIA

Las narrativas de casos de uso asociadas a cambiar contraseña propia son las siguientes:

Actores principales: Profesor y alumno

Precondiciones: Debe haber una sesión iniciada.

Camino principal alumno y profesor:

1. El usuario introduce la contraseña actual.
2. El usuario introduce la contraseña nueva.

3. El usuario repite la contraseña nueva.
- 4.a El usuario pulsa el botón “Cambiar contraseña”.

Camino alternativo:

- 4.b.1. Si la contraseña antigua no coincide el sistema notificará el error.
- 4.b.2. Repetir desde el paso 1
- 4.c.1. Si las contraseñas nuevas no son iguales el sistema notificará el error.
- 4.c.2. Repetir desde el paso 1

Postcondiciones:

La nueva contraseña de acceso se habrá cambiado satisfactoriamente y se vuelve al menú inicial del usuario logueado.

2.4.2.27. CASO DE USO: RESTABLECER CONTRASEÑA ALUMNO

Las narrativas de casos de uso asociadas a restablecer la contraseña de un alumno son las siguientes:

Actor principal: Profesor

Precondiciones: Debe haber iniciado sesión un profesor administrador.

Camino principal:

1. El profesor escribe el DNI del alumno que desea restablecer su contraseña.
- 2.a El profesor pulsa el botón “Restablecer contraseña”.

Camino alternativo:

- 2.b.1. Si el DNI del alumno no existe se mostrará un error.
- 2.b.2. Repetir desde el paso 1.

Postcondiciones:

La nueva contraseña de acceso para el alumno se habrá cambiado satisfactoriamente y se vuelve al menú inicial del profesor.

2.4.2.28. CASO DE USO: AÑADIR PROFESOR

Las narrativas de casos de uso asociadas a añadir un profesor son las siguientes:

Actor principal: Profesor

Precondiciones: Debe haber iniciado sesión un profesor administrador.

Camino principal:

1. El profesor escribe el DNI del nuevo profesor.
2. El profesor escribe el nombre del nuevo profesor.
3. El profesor escribe el apellido1 del nuevo profesor.
4. El profesor escribe el apellido2 del nuevo profesor.
5. El profesor selecciona si el nuevo profesor será administrador o no.
6. El profesor escribe la nueva contraseña.
7. El profesor repite la nueva contraseña.
- 8.a El profesor pulsa el botón “Añadir profesor”.

Camino alternativo:

- 8.b.1. Si alguno de los parámetros es incorrecto o si las contraseñas no coinciden se mostrará un error.
- 8.b.2. Repetir desde el paso 1.

Postcondiciones:

Se habrá añadido un nuevo profesor a la base de datos del sistema.

2.4.2.29. CASO DE USO: CERRAR SESIÓN

Las narrativas de casos de uso asociadas a cerrar la sesión son las siguientes:

Actores principales: Profesor y alumno

Precondiciones: Debe haber una sesión iniciada.

Camino principal alumno y profesor:

1. El usuario pincha en el botón “Cerrar sesión”.

Camino alternativo: -

Postcondiciones:

Se accederá a la página de inicio de demostración de la aplicación como usuario invitado.

3. PLAN DEL PROYECTO

En este capítulo se realiza la planificación temporal y la estimación de recursos necesarios para la realización del proyecto. Además se realizará un estudio de los costes de desarrollo, instalación y mantenimiento del sistema Web.

3.1. PLANIFICACIÓN TEMPORAL

En este apartado se expondrá la planificación temporal del proyecto, tanto del tipo de tareas a desarrollar como de la duración de cada una de ellas.

Debido a diversas circunstancias que han acontecido durante el desarrollo del proyecto, la planificación inicial que se realizó no se ha podido cumplir completamente. Por tanto se ha realizado también la planificación real una vez finalizado el proyecto.

3.1.1. ESTIMACIÓN DE TIEMPOS

A continuación se muestran la estimación de tiempos de las tareas a desarrollar en este proyecto. Como se ha explicado en el apartado anterior se incluyen la estimación de tiempo inicial *Tabla 3.1.* y la real una vez finalizado el proyecto *Tabla 3.2.*

Actividad	Duración estimada
Requerimientos de la aplicación	2 días
Casos de uso	2 días
Planificación temporal	2 días
Estimación de costes	1 día
Diseño de la base de datos	4 días
Storyboards	7 días
Diagramas de secuencia	4 días
Implementación de la base de datos	8 días
Implementación de la interfaz	8 días
Programación de applets	18 días
Pruebas	32 días
Redactar Memoria	76 días

Tabla 3.1. Estimación de tiempos inicial.

Finalmente el desarrollo del proyecto ha requerido un tiempo mayor para su desarrollo, lo que ha conllevado a un incremento en la duración de las tareas. Además algunas tareas no se habían estimado en la tabla inicial.

Actividad	Duración estimada
Requerimientos de la aplicación	2 días
Estudio de sistemas de hosting Web	2 días
Estudio de lenguajes de programación cliente y servidor	2 días
Casos de uso	3 días
Planificación temporal	2 días
Estimación de costes	1 día
Diseño de la base de datos	5 días
Storyboards	5 días
Diagramas de secuencia	6 días
Implementación de la base de datos	11 días
Implementación de la interfaz	15 días
Implementación de los menús de alumno y profesor	25 días
Programación de applets	36 días
Pruebas	45 días
Redactar Memoria	164 días

Tabla 3.2. Tiempos finales.

El proyecto inicialmente estaba planificado para ser entregado el día 22 de junio de 2015 pero debido a problemas en la implementación de los menús de alumno y profesor y en la integración de dichos menús con los applets para proponer, realizar y corregir ejercicios, se ha tenido que posponer la entrega hasta el día 27 de octubre de 2015.

Debido a la poca experiencia en los lenguajes de programación gráfica THREE.js y WebGL utilizados, se han tenido que probar varias opciones hasta dar con la solución adecuada para cada caso. Además se han tenido que añadir tareas que inicialmente no se consideraron, lo que ha incrementado notablemente el tiempo de desarrollo del proyecto.

3.1.2. DIAGRAMAS DE GANT

A continuación se muestran los diagramas de Gant inicial *Figura 3.1.* y final *Figura 3.2.* resultantes de la estimación de tiempos del apartado anterior.

Como se ha explicado antes se han tenido que añadir tareas que no se habían considerado inicialmente y se ha necesitado más tiempo del estimado para la implementación de la interfaz, los menús y los applets.

La estimación inicial comienza el día 9 de febrero de 2015 y la fecha de entrega del proyecto estaba prevista para el día 22 de junio del mismo año.



Figura 3.1. Diagrama de Gant inicial.

Al finalizar el proyecto se ha realizado otro diagrama de Gant en el que se puede ver que se han añadido nuevas tareas como el estudio de los sistemas de hosting Web y el estudio de los lenguajes de programación cliente y servidor.

También se ha añadido la implementación de los menús de alumno y profesor que inicialmente no se había considerado. Esta tarea ha requerido un tiempo muy elevado lo que ha conllevado a que el diagrama de Gant varíe considerablemente y, con ello, el tiempo necesario para desarrollar el proyecto.

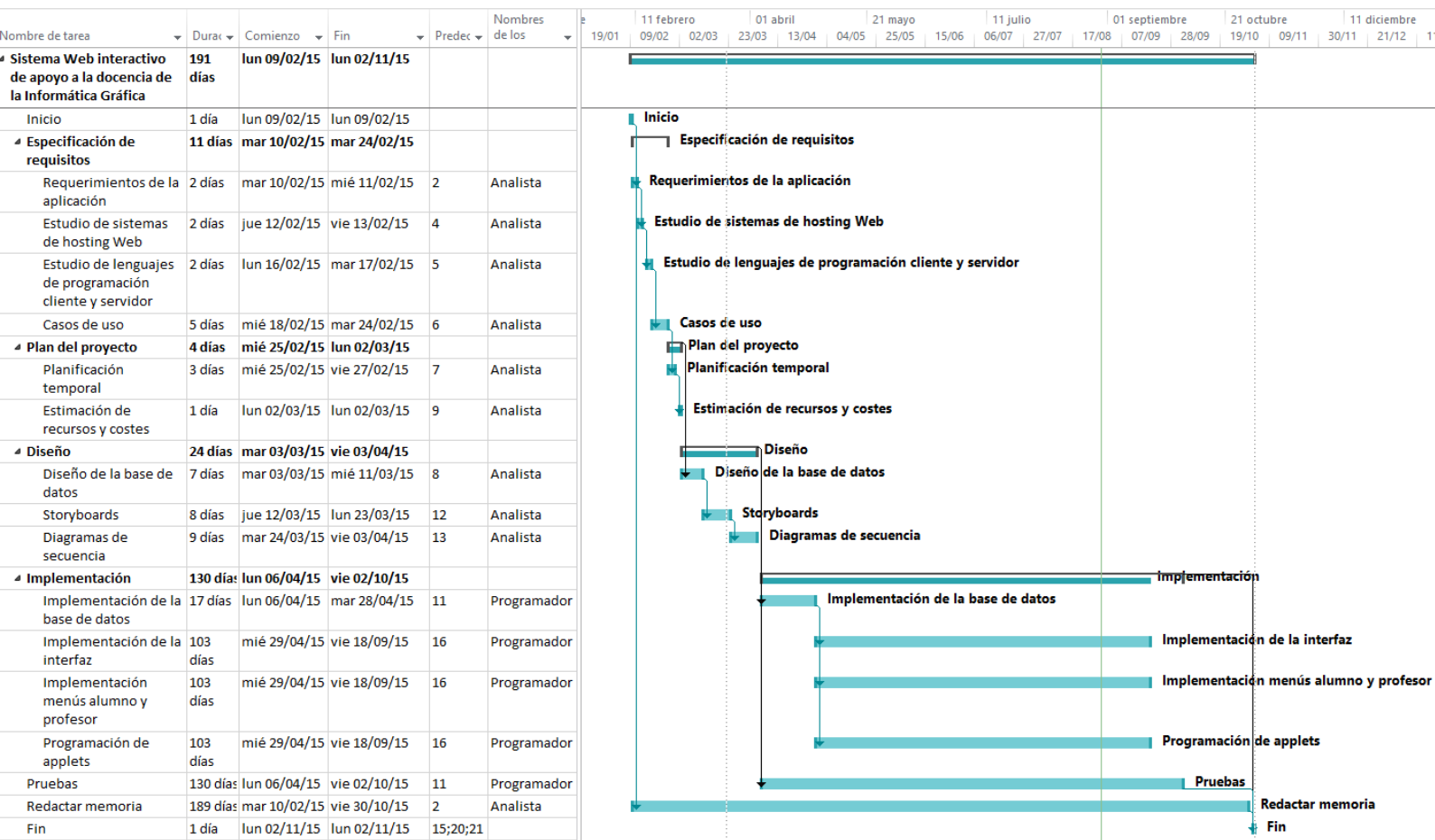


Figura 3.2. Diagrama de Gant final.

3.2. RECURSOS Y ESTIMACIÓN DE COSTES

Una vez realizada la planificación temporal del proyecto, se van a calcular los costes asociados a el desarrollo, la instalación y mantenimiento del mismo.

3.2.1. DESARROLLO

En primer lugar se realiza una estimación de los recursos y costes de desarrollo para este proyecto.

3.2.1.1. HUMANOS

Aunque es un proyecto realizado por una sola persona se tienen en cuenta los distintos roles implicados en el desarrollo del mismo. Se han calculado en base a una jornada laboral de 8 horas.

Para hacer el cálculo se muestra en primer lugar una tabla con los salarios mensuales, semanales y diarios de los distintos roles implicados en el desarrollo del proyecto y posteriormente se calcula, en función de las tareas y la duración de éstas, el costo de cada uno de ellos y el costo total de los recursos humanos en el desarrollo del proyecto.

Los datos han sido obtenidos del "XIII CONVENIO COLECTIVO DE ÁMBITO ESTATAL PARA LOS CENTROS DE EDUCACIÓN UNIVERSITARIA E INVESTIGACIÓN" realizado por el Ministerio de empleo y seguridad social [18]. De las tablas salariales para el año 2015 dentro del personal no docente se han extraído los datos del "Analista en TIC" y ,según se explica en el BOE, el salario del programador equivale al de un "Oficial de primera en administración". En base a esto se muestran los salarios en la *Tabla 3.3*.

Roles	Salario mensual	Salario semanal	Salario diario
Analista	1548,37 €	387,10 €	77,42
Programador	1074,05 €	268,51 €	53,70

Tabla 3.3. Salario de los roles implicados en el desarrollo del proyecto.

A continuación se muestra el coste inicial del proyecto durante la planificación en la *Tabla 3.4*.

Actividad	Duración estimada	Rol	Salario diario	Coste total
Requerimientos de la aplicación	2 días	Analista	77,42 €	154,84 €
Casos de uso	2 días	Analista	77,42 €	154,84 €
Planificación temporal	2 días	Analista	77,42 €	154,84 €
Estimación de costes	1 día	Analista	77,42 €	77,42 €
Diseño de la base de datos	4 días	Analista	77,42 €	309,68 €
Storyboards	7 días	Analista	77,42 €	541,94 €
Diagramas de secuencia	4 días	Analista	77,42 €	309,68 €
Implementación de la base de datos	8 días	Programador	53,70 €	429,60 €
Implementación de la interfaz	8 días	Programador	53,70 €	429,60 €
Programación de applets	18 días	Programador	53,70 €	966,60 €
Pruebas	32 días	Programador	53,70 €	1718,40€
Redactar Memoria	76 días	Analista	77,42 €	5883,92€
			Total:	11.131,36 €

Tabla 3.4. Coste estimado de los recursos humanos en el desarrollo del proyecto.

La *Tabla 3.4*. muestra el coste estimado inicial de los recursos humanos necesarios para el desarrollo del proyecto. Se ha calculado a partir de las tareas descritas en la *Tabla 3.1*. Para cada tarea se calcula su coste en función del rol implicado (Analista o Programador) y de la duración de la tarea, obteniendo un total de **11.131.36 €**.

Como se ha explicado en la planificación temporal el proyecto ha sufrido modificaciones en sus tareas y en el tiempo de las mismas, lo que ha producido un aumento de los costes que se ven reflejados en la *Tabla 3.5*.

Actividad	Duración estimada	Rol	Salario diario	Coste total
Requerimientos de la aplicación	2 días	Analista	77,42 €	154,84 €
Estudio de sistemas de hosting Web	2 días	Analista	77,42 €	154,84€
Estudio de lenguajes cliente y servidor	2 días	Analista	77,42 €	154,84€
Casos de uso	3 días	Analista	77,42 €	232,26 €
Planificación temporal	2 días	Analista	77,42 €	154,84 €
Estimación de costes	1 día	Analista	77,42 €	77,42 €
Diseño de la base de datos	5 días	Analista	77,42 €	387,10 €
Storyboards	5 días	Analista	77,42 €	387,10 €
Diagramas de secuencia	6 días	Analista	77,42 €	464,52 €
Implementación de la base de datos	11 días	Programador	53,70 €	590,70 €
Implementación de la interfaz	15 días	Programador	53,70 €	805,50 €
Implementación menús alumno y profesor	25 días	Programador	53,70 €	1.342,50 €
Programación de applets	36 días	Programador	53,70 €	1.933,20 €
Pruebas	45 días	Programador	53,70 €	2.416,50 €
Redactar Memoria	164 días	Analista	77,42 €	12.696,88 €
			Total:	21.953,04 €

Tabla 3.5. Coste final de los recursos humanos en el desarrollo del proyecto.

La *Tabla 3.5.* muestra el coste final de los recursos humanos necesarios para el desarrollo del proyecto. Se ha calculado a partir de las tareas descritas en la *Tabla 3.2.* Se ha obtenido un coste total de **21.953,04 €**. Debido a los problemas que han surgido durante el proyecto y al incremento en el número de tareas y su tiempo se ha producido un incremento en el coste final en recursos humanos del proyecto de **10.821,68 €**.

3.2.1.2. SOFTWARE

En la siguiente tabla se muestra el software que ha sido necesario para el desarrollo del proyecto, así como su coste asociado. Dicho coste depende de la amortización, por lo tanto el precio del mismo será establecido para los meses que dure el desarrollo del proyecto.

Según el Boletín Oficial del Estado del 27 de noviembre de 2014 sobre el "Impuesto de Sociedades" [19], el software tiene un periodo de amortización de 6 años por lo tanto su coste se calculará para los 9 meses que durará el proyecto.

Software	Coste Software	Coste Amortizado
Windows® 8.1 Profesional 64 bits	194,99 €	24,40 €
Microsoft® Office® 2007	90,0 €	11,25 €
Microsoft® Project 2013	769,00 €	96,12 €
Visual ParadigmCommunityEdition (Freeware)	0,00 €	0,00 €
Netbeans IDE 8.0.1 (Freeware)	0,00 €	0,00 €
Sublime Text 2.0.2 (Freeware)	0,00 €	0,00 €
AtlassianSourceTree (Git) (Freeware)	0,00 €	0,00 €
XAMPP 5.6.2 (Freeware)	0,00 €	0,00 €
Google Chrome v44.0 (Freeware)	0,00 €	0,00 €
Mozilla Firefox v40.0 (Freeware)	0,00 €	0,00 €
Internet Explorer v11.0 (Freeware)	0,00 €	0,00 €
Nero 2015	69,99 €	8,75 €
Total	1.123,98 €	140,52 €

Tabla 3.6. Coste de los recursos Software necesarios para el desarrollo del proyecto.

Como se puede observar en la *Tabla 3.6.* el coste total del Software necesario para este proyecto es de **140,52 €**. Se ha realizado el cálculo del mismo teniendo en cuenta que será utilizado durante 9 de los 72 meses de su periodo de amortización.

3.2.1.3. HARDWARE

Los costes asociados al hardware serán calculados teniendo en cuenta la amortización de acuerdo al BOE del 27 de noviembre de 2014 [19]. El Hardware tiene un periodo de amortización de 8 años, por tanto se calculará el coste en el porcentaje de tiempo que dure el desarrollo del proyecto (9 meses).

El ordenador de sobremesa con el que ha sido desarrollado este proyecto tiene las siguientes características:

Componentes	Coste Hardware	Coste Amortizado
-------------	----------------	------------------

Procesador Intel® Core™ i5-3570k 3.40 GHz	215,00 €	20,16 €
Tarjeta gráfica Nvidia GTX 660 Ti	320,00 €	30,00 €
CorsairVengeance PC3-12800 DDR3 1600 16GB 4x4GB CL9	75,00 €	7,03 €
Placa base Intel Socket 1155 Gigabyte GA-Z77-D3H	103,00 €	9,66 €
Fuente de alimentacionAntecEarthwatts EA-750W Modular	120,00 €	11,25 €
Disco duro SSD Samsung 830 Series 128GB SATA3	85,00 €	7,97 €
Disco duro Western Digital Caviar Green SATA3 2TB 64MB	72,00 €	6,75 €
Torre AntecTwelveHundred v3	209,00 €	19,60 €
Teclado Logitech k120	15,00 €	1,41 €
Ratón Logitech g550	40,00 €	3,75 €
Altavoces Trust 2.1	30,00 €	2,82 €
Monitor Samsung SyncMaster BX2235 22"	141,00 €	13,22 €
Total	1.425 €	133,62 €

Tabla 3.7. Coste de los recursos Hardware necesarios para el desarrollo del proyecto.

Concepto	Coste
Recursos humanos	21.953,04 €
Software	140,52 €
Hardware	133,62 €
Total	22.227,18 €

Tabla 3.8. Coste total asociado al desarrollo del proyecto.

Después de sumar los recursos humanos necesarios para el desarrollo del proyecto, los recursos software y los recursos hardware, obtenemos el coste total asociado al desarrollo del proyecto, siendo este de **22.217,18 €**, tal y como se puede observar en la *Tabla 3.8*.

3.2.2.INSTALACIÓN

A continuación se describen los recursos necesarios para la instalación del sistema Web. Realmente la aplicación estará hospedada en la plataforma Openshift como se explicó previamente, por tanto para instalarla solo es necesario crear una cuenta en Openshift y subir la aplicación. No obstante se detallarán los recursos Hardware y Software mínimos para la instalación del sistema Web.

3.2.2.1. SOFTWARE

En este apartado se va a especificar los requisitos mínimos del software para instalar el sistema Web.

Software
Sistema operativo CentOS
Servidor HTTP Apache
MySQL Server
PHP 5.4

Tabla 3.9. Software mínimo para la instalación del sistema Web

Para instalar esta aplicación simplemente se necesita un sistema operativo Linux, gratuito, y que ofrece unas altas prestaciones para crear servidores Web. También serán necesarios el servidor HTTP de Apache, MySQL Server para gestionar la base de datos y PHP 5.4. Todos ellos son gratuitos y fácilmente configurables en la distribución CentOS de Linux.

3.2.2.2. HARDWARE

A continuación se detallarán los requisitos mínimos Hardware necesarios para esta aplicación. Debido a que se trata de un sistema para la realización de ejercicios de la asignatura "Informática Gráfica" no debería acumularse una carga muy elevada en el sistema y, por tanto, podrá ejecutarse en un ordenador con unas prestaciones normales.

Hardware
Intel Pentium Dual Core G3240 3.1GHz
Memoria RAM 1x4GB DDR3 UDIMM
Adaptador Ethernet 332i de 1 GB y 2 puertos por controlador
Fuente de alimentación 350W

Tabla 3.10. Hardware mínimo para la instalación del sistema Web.

En la *Tabla 3.10.* se muestran los requisitos mínimos del servidor [20] para alojar el sistema Web. Estos requisitos son una referencia, pero posiblemente el sistema Web pueda estar instalado en un servidor con menos prestaciones, ya que solo debería de haber en el sistema unos 80 alumnos por curso, de los cuales se conectarán simultáneamente un número incluso menor, por tanto será suficiente con

este servidor para que todo funcione correctamente. El precio de este servidor es de **249,95 €**^[20], por tanto se trata de un servidor bastante asequible.

Para realizar la instalación del sistema, tanto el Hardware, como el Software, se necesitará un técnico en sistemas durante un periodo de tiempo de 3 horas para dejar el sistema correctamente instalado y funcionando. El coste de la hora será de 30 € por tanto tendrá que añadirse dicho coste a la instalación.

Concepto	Coste
Software	0,00 €
Hardware	249,95 €
Personal	90 €
Total	339,95 €

Tabla 3.11. Coste total asociado a la instalación del sistema Web.

3.2.3.MANTENIMIENTO

El sistema Web puede requerir un mantenimiento puntual, para solucionar un posible error en el desarrollo del mismo o para añadir cualquier funcionalidad nueva.

Desde el momento de la entrega del proyecto, una vez instalado, se dispondrá de un periodo de 2 meses de prueba en los que el mantenimiento será gratuito para la solución de errores que se puedan producir. También se podrán realizar pequeñas modificaciones en elementos que no se hayan ajustado por completo a los requisitos iniciales del proyecto.

A partir del tercer mes se establecerá una tarifa de **25 €** la hora para cualquier mantenimiento que necesite el sistema Web.

3.2.4.OTROS GASTOS

En este apartado se incluyen el resto de gastos surgidos durante el desarrollo del proyecto, tanto la conexión a Internet para resolver cualquier duda o consultar información, así como la impresora necesaria para imprimir la documentación y su coste de encuadernación. Según las tablas de amortización indicadas anteriormente ^[19], la impresora tendrá un periodo de amortización de 8 años, por tanto se calcula su coste para los 9 meses del desarrollo del proyecto.

Concepto	Coste
Cuota de alta conexión ADSL Jazztel	59,95 €
Conexión ADSL 30MB Jazztel (durante 9 meses)	332,55 €
Impresora HP Deskjet 2540	45,95 € --> 4,31 €
Encuadernación de documentación	4,20 €
Total	401,01 €

Tabla 3.12. Coste total asociado a otros gastos.

El coste total asociado a otros gastos se puede observar en la *Tabla 3.12.* y es de **401,01 €**.

Una vez hemos calculado el coste de las distintas partes del proyecto se puede calcular el coste final del proyecto indicado en la *Tabla 3.13.*

Concepto	Coste
Desarrollo	22.217,18 €
Instalación	339,95 €
Otros gastos	401,01 €
Total	22.958,14 €

Tabla 3.13. Coste total del proyecto.

Este coste final de **22.958,14 €** será independiente del coste de mantenimiento que pueda tener la aplicación pasados los 2 meses del periodo de pruebas.

4. DISEÑO

En este capítulo se expondrá el diseño de la base de datos y del software del proyecto.

4.1. DISEÑO COMPLETO DE LA BASE DE DATOS

Se detallará el diseño de la base de datos [21], desde el esquema conceptual hasta el modelo entidad-relación normalizado. También se realizará el diseño sobreseguridad de la misma y los privilegios de los usuarios. Por último se realizará un estudio de la volumetría de la base de datos con una estimación para los primeros 5 años desde la instalación del sistema Web.

4.1.1. ESQUEMA CONCEPTUAL

El diseño conceptual tiene como objetivo obtener una representación de la información con independencia de usuarios y aplicaciones en particular, y fuera de consideraciones sobre la eficiencia del sistema donde vaya a ser implementado.

Mediante la entrevista establecida con el tutor de este proyecto se han sacado algunas conclusiones a tener en cuenta a la hora de desarrollar la base de datos.

La aplicación tendrá un número elevado de alumnos, que podrán iniciar sesión en la aplicación y de los que se deberá almacenar su DNI, nombre, apellidos y una contraseña de acceso al sistema Web.

También deberán existir los profesores que impartirán la asignatura. Para ellos será necesario almacenar su DNI, nombre, apellidos, una contraseña de acceso y si se trata de un profesor administrador o no.

Tanto los alumnos como los profesores deberán pertenecer a un curso y grupo. En el caso de los alumnos, estarán matriculados en un curso y grupo, y, en el caso de los profesores, serán los encargados de impartir los mismos. Un curso y grupo solo serán impartidos por un único profesor. Pueden existir alumnos repetidores que estén matriculados en distintos cursos y grupos.

Un profesor puede proponer ejercicios con una fecha de inicio y fin. Para ello se tendrá que almacenar, además de las fechas, un identificador de ejercicio, un tipo de ejercicio, los parámetros que se vayan a almacenar y sus respectivos valores. Opcionalmente puede añadir un comentario para identificar el ejercicio visualmente.

Los alumnos podrán realizar los ejercicios configurados en el grupo en el que se encuentren matriculados.

Por último un profesor puede corregir los ejercicios, asignándoles una nota y, opcionalmente, las observaciones de su corrección.

A partir de esta información se obtiene el siguiente esquema conceptual, donde se incluye la cardinalidad entre las entidades. En dicho diseño, los rectángulos representan entidades y los rombos representan relaciones entre entidades, constando al lado de las mismas la cardinalidad de dicha relación. También se podrán ver atributos pertenecientes a las relaciones representados como elipses.

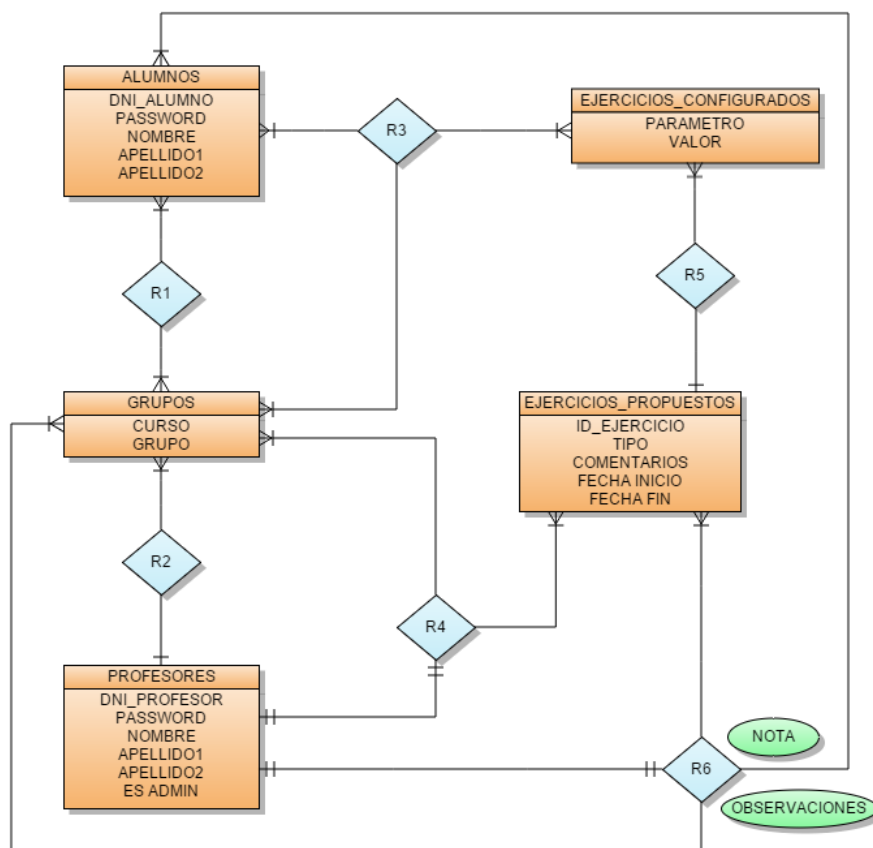


Figura 4.1. Esquema conceptual de la base de datos.

Como se puede ver en la *Figura 4.1.*, correspondiente al esquema conceptual de la base de datos, existen 5 tipos de entidades:

- **Alumnos:** contiene el DNI, contraseña, nombre, primer y segundo apellido del alumno.
- **Profesores:** contiene el DNI, contraseña, nombre, primer apellido, segundo apellido y si se trata de un profesor administrador.
- **Grupos:** contiene el curso y el identificador de grupo.
- **Ejercicios_propuestos:** contiene un identificador de ejercicio, el tipo, la fecha de inicio, la fecha de fin de ejercicio y, opcionalmente, los comentarios referentes al ejercicio propuesto.
- **Ejercicios_configurados:** contiene el parámetro y su respectivo valor.

Se puede observar también que existen 6 relaciones que serán detalladas a continuación:

- **R1:** Se trata de una relación de N/M. Un alumno puede estar matriculado en uno o más grupos en caso de ser repetidor. Un grupo tiene matriculados uno o más alumnos.
- **R2:** Se trata de una relación de 1/N. Un profesor puede impartir uno o más grupos. Un grupo solo puede ser impartido por un profesor.
- **R3:** Se trata de una relación ternaria. Un alumno puede resolver uno o varios ejercicios configurados en un grupo. A su vez, un ejercicio configurado puede ser resuelto por uno o varios alumnos en un grupo.
- **R4:** Se trata de una relación ternaria. Un profesor puede proponer uno o varios ejercicios en un grupo. A su vez, un ejercicio solo puede ser propuesto por un profesor en un grupo.
- **R5:** Se trata de una relación de 1/N. Un ejercicio propuesto tiene un número determinado de parámetros con su respectivo valor. Un ejercicio configurado solo puede ser propuesto una vez.
- **R6:** Se trata de una relación cuaternaria. Un profesor puede corregir uno o varios ejercicios propuestos en un grupo, realizador por uno o varios alumnos y establecer una nota y observaciones para dicho ejercicio.

4.1.2. ESQUEMA LÓGICO

Una vez analizado el esquema conceptual se puede desarrollar el esquema lógico, que detalla las claves de cada entidad, convirtiéndolas en tablas, y resuelve las relaciones entre entidades. La conversión del esquema conceptual en el esquema lógico, esta basado en los tres principios siguientes:

- Todo tipo de entidad del esquema conceptual se convierte en tabla en el esquema lógico.
- Todo tipo de relación entre tablas 1/N se traduce en una propagación de la clave, es decir se crea una nueva clave primaria y foránea.
- Todo tipo de relacion entre tablas N/M origina la creación de una tabla intermedia con las claves primarias de ambas tablas como claves foráneas en la nueva tabla.

Siguiendo estas reglas el esquema lógico de la base de datos queda reflejado en la *Figura 4.2*.

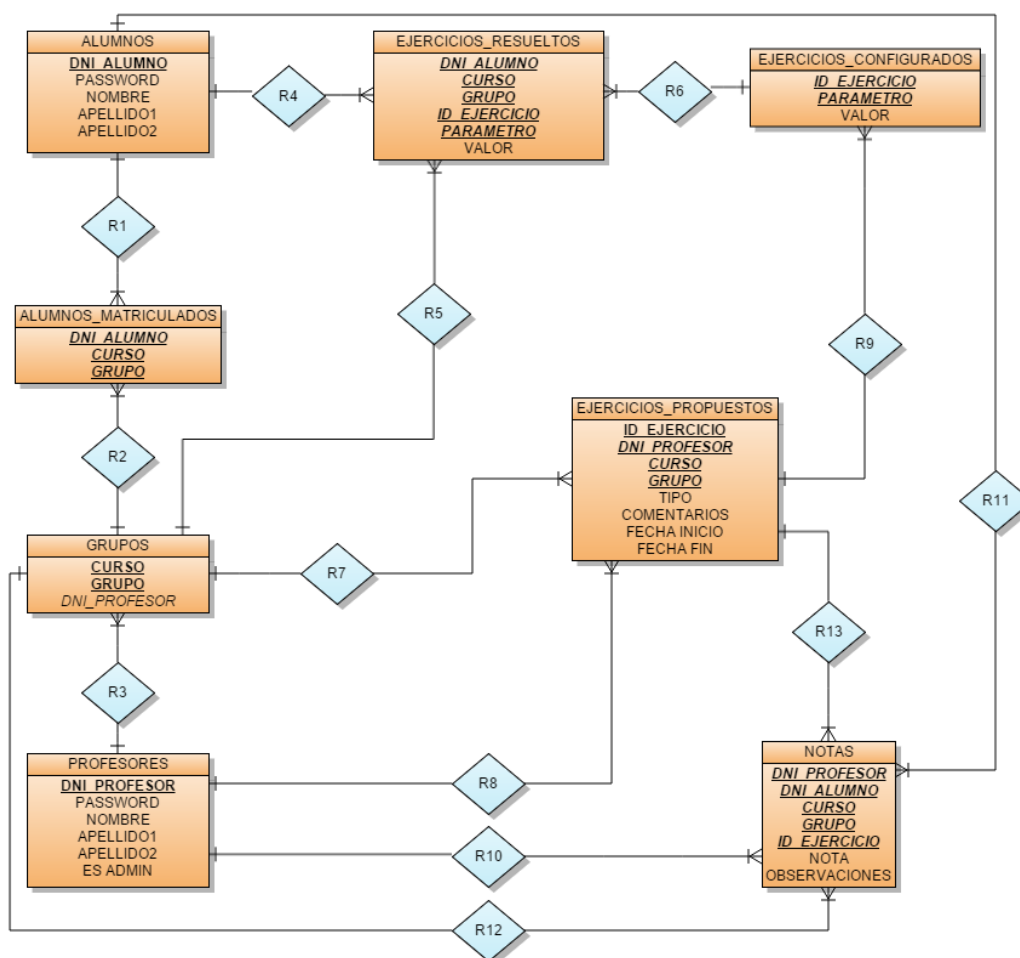


Figura 4.2. Esquema lógico de la base de datos.

Como se puede observar la base de datos tendrá 7 tablas que serán descritas a continuación:

- **Alumnos:** La clave primaria de esta tabla es el atributo "DNI_ALUMNO". También tendrá otros atributos como "NOMBRE", "APELLIDO1" y "APELLIDO2".
- **Grupos:** La clave primaria de esta tabla es "CURSO" y "GRUPO". Se ha tenido que añadir el atributo "DNI_PROFESOR" como clave foránea debido a la propagación de la clave en la relación "R3".
- **Profesores:** La clave primaria de esta tabla es el atributo "DNI_PROFESOR". También tendrá otros atributos como "NOMBRE", "APELLIDO1", "APELLIDO2", "ESADMIN".
- **Alumnos_matriculados:** Esta tabla es resultado de la relación N/M existente entre las tablas "ALUMNOS" y "GRUPOS". La clave primaria de esta tabla es "DNI_ALUMNO" (foránea de la tabla "ALUMNOS"), "CURSO" y "GRUPO" (foráneas de la tabla "GRUPOS").
- **Ejercicios_resueltos:** Esta tabla es resultado de la relación N/M existente entre las tablas "ALUMNOS", "EJERCICIOS_CONFIGURADOS" y "GRUPOS". La clave primaria de esta tabla es consecuencia de la propagación de la clave en las relaciones "R4", "R5" y "R6". También tendrá el atributo "VALOR".
- **Ejercicios_configurados:** La clave primaria de esta tabla es "PARAMETRO". Como consecuencia de la propagación de la clave en la relación "R9" también se añade a la clave primaria el atributo "ID_EJERCICIO". También tendrá el atributo "VALOR".
- **Ejercicios_propuestos:** La clave primaria de esta tabla es "ID_EJERCICIO". Como consecuencia de la propagación de la clave en la relación "R7" se añaden los atributos "CURSO" y "GRUPO" a la clave primaria. También hay que añadir a la clave primaria el atributo "DNI_PROFESOR" foránea de la tabla "PROFESORES". También tendrá los atributos "TIPO", "COMENTARIOS", "FECHA_INICIO" y "FECHA_FIN".

- **Notas:** Esta tabla es resultado de la relación cuaternaria "R6" del esquema conceptual *Figura 4.1*. La clave primaria procede de la propagación de las claves de las tablas implicadas en dicha relación. Por tanto la clave primaria de esta tabla está formada por los atributos "DNI_PROFESOR", "DNI_ALUMNO", "CURSO", "GRUPO" e "ID_EJERCICIO" propagadas de las relaciones "R10", "R11", "R12" y "R13" respectivamente. También tendrá los atributos "NOTA" y "OBSERVACIONES".

Una vez realizado el esquema lógico es necesario realizar un proceso de normalización de las tablas.

4.1.3.NORMALIZACIÓN

En el desarrollo del esquema lógico se han obtenido una serie de tablas finales que formarán la base de datos. Sin embargo, dichas tablas han sido obtenidas a partir de un esquema conceptual simple, por tanto pueden existir problemas de redundancia de información.

La teoría de la normalización [22] consiste en un conjunto de reglas formales que permiten que el esquema lógico cumpla una serie de propiedades, corrigiendo la estructura de los datos de las tablas y evitando problemas como las inconsistencias de datos, redundancia o pérdida de información.

Las reglas formales de la teoría de la normalización son conocidas con el nombre de formas normales. Existen seis formas normales de las que generalmente solo se aplican las tres primeras. Se puede decir que una base de datos está correctamente diseñada cuando se encuentra en tercera forma normal (3FN).

La primera forma normal 1FN se cumple si las claves de las tablas son correctas, es decir, que todos los atributos sean atómicos, la clave primaria sea única, no nula y que identifique al resto de campos no clave.

En el esquema lógico de la *Figura 4.2*. se puede observar que se cumple la primera forma normal, ya que las claves de todas las tablas son correctas e identifican al resto de atributos.

La 2FN se cumple si las relaciones y tablas cumplen la 1FN y si los atributos que no forman parte de ninguna clave dependen de forma completa de la clave principal. Es decir que no existen dependencias parciales, todos los atributos que no son clave principal deben depender únicamente de la clave principal.

En el esquema lógico de la *Figura 4.2.* se puede observar que la tabla "EJERCICIOS PROPUESTOS" no se encuentra en segunda forma normal. El atributo "TIPO" depende únicamente del atributo "ID_EJERCICIO" de la clave primaria. Por tanto se debería crear otra tabla "PARAMETROS_EJERCICIO_PROPUESTO" con clave primaria "ID_EJERCICIO" y el atributo "TIPO".

Debido a que no se producirá duplicidad de información en la tabla "EJERCICIOS_PROPUESTOS", se desnormalizará la base de datos deshaciendo el anterior cambio comentado, ya que la información de dicha tabla se consultará conjuntamente y no es rentable en términos de rendimiento separar dicha tabla y tener que ejecutar operaciones JOIN cada vez que se requiera la información.

Realizado esto se puede decir que la base de datos se encuentra en segunda forma normal.

La 3FN se cumple si las relaciones y tablas cumplen la 1FN, la 2FN y si no existe ninguna dependencia funcional transitiva entre los atributos que no son clave, es decir que no puede existir ningún atributo no clave que dependa de otro atributo no clave.

Todas las tablas de la base de datos cumplen la tercera forma normal y por tanto se encuentra correctamente normalizada a excepción del cambio realizado en la tabla "EJERCICIOS_PROPUESTOS".

4.1.4.DISEÑO DE SEGURIDAD

El diseño de seguridad [23] tiene como objetivo cumplir estos tres aspectos generales:

- Confidencialidad de la información: se cumple cuando sólo los usuarios autorizados pueden conocer los datos del sistema.
- La integridad de la información: consiste en que sólo los usuarios con los privilegios necesarios pueden modificar o borrar los datos.
- La disponibilidad de la información: se cumple si los usuarios autorizados pueden acceder a tiempo a la información requerida.

Las aplicaciones nunca deberían conectarse a la base de datos como su propietario o administrador, porque estos usuarios podrían ejecutar cualquier consulta, como por ejemplo, eliminar tablas o borrar su contenido por completo.

Se pueden crear distintos usuarios de una base de datos para cada aspecto de la aplicación con permisos muy limitados a los objetos de dicha base de datos. Solamente deberían otorgarse los privilegios necesarios, esto significa que si un intruso obtiene acceso a una base de datos utilizando las credenciales de la aplicación, solamente puede efectuar los cambios que la aplicación permita.

Para garantizar la seguridad en la base de datos se han creado dos tipos de usuario:

- Administrador: posee todos los privilegios del sistema de gestión de base de datos. Es el usuario que se ha utilizado para crear la estructura y tablas de la base de datos durante el desarrollo del proyecto.
- Usuario: solamente puede realizar consultas INSERT, UPDATE, SELECT y DELETE, es decir, solamente puede actuar sobre los datos de la tabla, nunca sobre su estructura. Este tipo de usuario será con el que accederá la aplicación a la base de datos, tanto alumnos, como profesores.

También se puede hacer uso de las vistas para acceder al contenido de las tablas de la base de datos desde la aplicación. En el caso de este sistema Web no es necesario el uso de vistas para mejorar la seguridad, ya que no hay una gran cantidad de atributos en las tablas, y, además es necesario que la aplicación pueda tener acceso a todos ellos.

4.1.5. DISEÑO FÍSICO

El objetivo general del diseño físico se basa en determinar estructuras de almacenamiento y acceso para que las aplicaciones que interactúen con la base de datos obtengan un buen rendimiento. Consiste en elegir las estructuras más apropiadas para las tablas de la base de datos y según el uso que se dará a los datos.

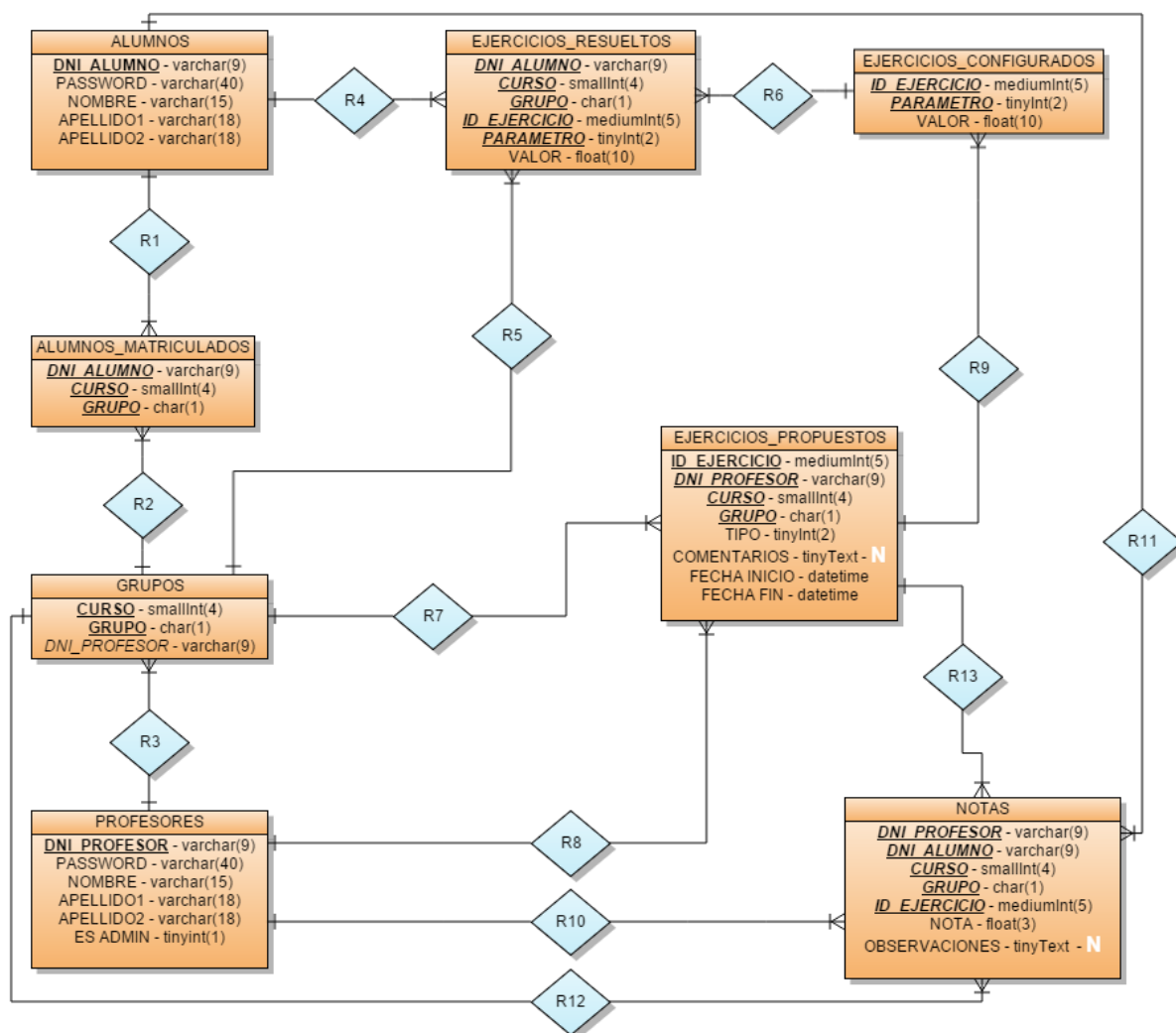


Figura 4.3. Esquema físico de la base de datos.

En la Figura 4.3. quedan definidos el tipo y tamaño de los atributos[24] de las tablas. Se han definido los tipos de atributos de acuerdo a los requerimientos de la aplicación y al diseño lógico previamente realizado de la base de datos. Se pueden ver distintos tipos de atributos:

- Varchar(x): cadena de texto de longitud máxima "x". Tendrá un tamaño de "x" + 1 bytes.
- TinyInt(1): número entero diminuto de longitud 1 con valores 0-255. Tendrá un tamaño de 1 byte.
- SmallInt(x): número entero pequeño de tamaño máximo "x". Tendrá un tamaño de 2 bytes.
- MediumInt(x): número entero mediano de tamaño máximo "x". Tendrá un tamaño de 3 bytes.
- Char(1): carácter de longitud 1 con valores A-Z. Tendrá un tamaño de 1 byte.
- Float(x): número en coma flotante de tamaño máximo "x" incluyendo decimales. Tendrá un tamaño de 4 bytes
- Datetime: fecha con formato "yyyy-mm-dd hh:ii:ss". Tendrá un tamaño de 8 bytes.
- TynyText: este tipo tiene tamaño variable, pero se trata de una cadena de texto. En el caso del atributo "OBSERVACIONES" de la tabla "NOTAS" y del atributo "COMENTARIOS" de la tabla "EJERCICIOS_PROPUESTOD" puede ser nulo si el profesor lo estima oportuno. Tendrá un tamaño igual a su longitud más 1 bytes.

En base a este esquema físico se realizará un cálculo de la volumetría de la base de datos durante un periodo de vida de los 5 próximos años. Este cálculo se podría realizar de manera automática ejecutando una consulta que estima el tamaño de almacenamiento aproximado de la base de datos, pero para realizar este proceso es necesario que la base de datos tenga un histórico de transacciones realizadas y datos almacenados.

Debido a que el diseño se ha realizado previamente a la implementación y carga de la base de datos el cálculo de la volumetría se ha realizado manualmente.

Se han obtenido los siguientes datos en la entrevista con el tutor del proyecto y profesor de la asignatura:

- Se añadirán los 5 próximos cursos con 2 grupos cada uno.

- Se matricularán 50 alumnos por grupo.
- Se añadirán 10 profesores distintos para los próximos 5 años, de los cuales solo interactuarán 2 por curso.
- Cada profesor mandará 14 ejercicios a lo largo del curso (2 de cada tipo de applet).
- Cada ejercicio tendrá como máximo 31 filas en las tablas de "EJERCICIOS_CONFIGURADOS" y "EJERCICIOS_RESUELTOS". Este valor es el máximo de atributos que son necesarios para un applet (luces y materiales).
- Todos los ejercicios deben ser realizados y corregidos para conseguir una estimación en el peor de los casos.

Teniendo en cuenta estos datos se han obtenido un número de filas para cada tabla. Todos estos datos son para el peor de los casos durante los próximos 5 años:

- La tabla "GRUPOS" tendrá un número total de filas de 5 cursos x 2 grupos = **10 grupos**.
- La tabla "PROFESORES" tendrá un número total de filas de **10 profesores**.
- La tabla "ALUMNOS" tendrá un número total de filas de 10 grupos x 50 alumnos = **500 alumnos**.
- La tabla "ALUMNOS_MATRICULADOS" tendrá un número total de filas igual a los alumnos, es decir **500 alumnos matriculados**.
- La tabla "EJERCICIOS_PROPUUESTOS" tendrá un número total de filas de 14 ejercicios x 2 grupos x 5 cursos = **140 ejercicios propuestos**.
- La tabla "EJERCICIOS_CONFIGURADOS" tendrá un número total de 140 ejercicios propuestos x 31 parámetros = **4.340 filas**.
- La tabla "EJERCICIOS_RESUELTOS" tendrá un número total de 140 ejercicios propuestos x 31 parámetros x 500 alumnos = **2.170.000 filas**.
- La tabla "NOTAS" tendrá un número total de filas de 140 ejercicios propuestos x 500 alumnos = **70.000 calificaciones**.

En total se obtendrán **2.245.500 filas** teniendo siempre en cuenta que se trata del peor de los casos para los próximos 5 años. A continuación se calculará el

tamaño de cada tabla y el tamaño total de la base de datos, añadiendo un 40% de capacidad para el tamaño de los índices. Para calcular los bytes que ocupa una tabla hay que multiplicar cada presencia de un atributo por el número de bytes que ocupa dicho tipo.

Para la tabla "GRUPOS":

CURSO	GRUPO	DNI_PROFESOR	TOTAL
10x 2 bytes	10 x 1 bytes	10 x 10 bytes	130 bytes

Tabla 4.1. Tamaño total de la tabla GRUPOS en bytes.

Para la tabla "PROFESORES":

DNI_PROFESOR	PASSWORD	NOMBRE	APELLIDO1	APELLIDO2	ESADMIN	TOTAL
10x 10 bytes	10 x 41 bytes	10 x 16 bytes	10 x 19 bytes	10 x 19 bytes	10 x 1 bytes	1060 bytes

Tabla 4.2. Tamaño total de la tabla PROFESORES en bytes.

Para la tabla "ALUMNOS":

DNI_ALUMNO	PASSWORD	NOMBRE	APELLIDO1	APELLIDO2	TOTAL
500 x 10 bytes	500 x 41 bytes	500 x 16 bytes	500 x 19 bytes	500 x 19 bytes	52.500 bytes

Tabla 4.3. Tamaño total de la tabla ALUMNOS en bytes.

Para la tabla "ALUMNOS_MATRICULADOS":

DNI_ALUMNO	CURSO	GRUPO	TOTAL
500x 10 bytes	500 x 2 bytes	500 x 1 bytes	6.500 bytes

Tabla 4.4. Tamaño total de la tabla ALUMNOS_MATRICULADOS en bytes.

Para la tabla "EJERCICIOS_PROPUUESTOS":

ID_EJERCICIO	DNI_PROF	CURSO	GRUPO	TIPO	COMENT	FECHA_INI	FECHA_FIN	TOTAL
140x 3 bytes	140 x 10 bytes	140 x 2 bytes	140 x 1 bytes	140 x 1 bytes	140 x 256 bytes	140 x 8 bytes	140 x 8 bytes	40.320 bytes

Tabla 4.5. Tamaño total de la tabla EJERCICIOS_PROPUUESTOS en bytes.

Para la tabla "EJERCICIOS_CONFIGURADOS":

ID_EJERCICIO	PARÁMETRO	VALOR	TOTAL
4340 x 3 bytes	4340 x 1 bytes	4340 x 4 bytes	34.720 bytes

Tabla 4.6. Tamaño total de la tabla EJERCICIOS_CONFIGURADOS en bytes.

Para la tabla "EJERCICIOS_RESUELTOS":

ID_EJERCICIO	DNI_ALUMNO	CURSO	GRUPO	PARAMETRO	VALOR	TOTAL
2170000 x 3 bytes	2170000 x 10 bytes	2170000 x 2 bytes	2170000 x 1 bytes	2170000 x 1 bytes	2170000 x 4 bytes	45.570.000 bytes

Tabla 4.7. Tamaño total de la tabla EJERCICIOS_RESUELTOS en bytes.

Para la tabla "NOTAS":

DNI_PROFESOR	DNI_ALUMNO	CURSO	GRUPO	ID_EJERCICIO	NOTA	OBSERV	TOTAL
70000 x 10 bytes	70000 x 10 bytes	70000 x 2 bytes	70000 x 1 bytes	70000 x 3 bytes	70000 x 4 bytes	70000 x 256 bytes	20.020.000 bytes

Tabla 4.8. Tamaño total de la tabla NOTAS en bytes.

Para calcular el tamaño total estimado de los datos en los próximos 5 años simplemente hay que sumar el tamaño requerido de los mismos para cada tabla.

Tabla	Tamaño
GRUPOS	130 bytes
PROFESORES	1060 bytes
ALUMNOS	52.500 bytes
ALUMNOS_MATRICULADOS	6.500 bytes
EJERCICIOS_PROPUUESTOS	40.320 bytes
EJERCICIOS_CONFIGURADOS	34.720 bytes
EJERCICIOS_RESUELTOS	45.570.000 bytes
NOTAS	20.020.000 bytes
Total	65.725.230 bytes
Total (MB)	62,68 MB

Tabla 4.9. Tamaño total estimado de la base de datos para los próximos 5 años.

El tamaño total estimado que tendrán los datos de la base de datos para los primeros 5 años de funcionamiento del sistema Web es **62,68 MB**. A este valor hay que añadirle un 40% más para el tamaño que puedan necesitar los índices de las tablas. Por tanto el tamaño total estimado para los primeros 5 años de funcionamiento del sistema Web es de **87,75 MB**.

Se ha conseguido una base de datos optimizada y que tiene poco tamaño, en parte, gracias a la elección de los tipos de datos adecuados y mínimos como tinyInt, smallInt, etc.

El tamaño de base de datos que permite Openshift [4] para la versión gratuita es de 1 GB, por tanto la base de datos necesaria para este sistema Web es totalmente compatible con este PaaS [3].

4.2. DISEÑO DEL SOFTWARE

Esta etapa es una de las más importantes del desarrollo del proyecto, donde se especifica el diseño que tendrá la interfaz del sistema Web, los menús y los ejercicios para los distintos applets que lo forman. También se verán los storyboards de la aplicación y los diagramas de secuencia.

4.2.1. ESTRUCTURA VISUAL DE LA INTERFAZ DE LA APLICACIÓN

Debido a que se parte de una aplicación previamente desarrollada gran parte de la estructura visual de la nueva aplicación se mantiene similar a la anterior. Por tanto esta parte se centrará en la interfaz nueva creada para los menús de alumno y profesor y en la estructura general de las acciones proponer, realizar y corregir ejercicios de cualquier tipo de applet.

Para realizar las pantallas se ha empleado papel y lápiz, ya que es más sencillo realizar las modificaciones necesarias sobre un boceto y es mucho más fácil añadir elementos especiales que algunas herramientas no pueden tener.

Cuando se acceda al sistema Web se mostrará la estructura presente en el menú inicial del sistema Web previamente desarrollado, a excepción de la opción para iniciar sesión. Se mostrará en la cabecera, *Figura 4.4*, un campo para insertar el DNI del usuario y otro para insertar la contraseña de acceso. Además se añadirá un botón para iniciar sesión.



Figura 4.4. Cabecera de la aplicación.

Una vez se haya iniciado sesión correctamente se mostrará el menú de inicio de alumno o profesor según el tipo de usuario que haya iniciado la sesión. Este menú constará en ambos casos de los tipos de acciones que puede realizar cada

usuario a la izquierda de la página. Mientras tanto, a la derecha, irán apareciendo las distintas funcionalidades que el usuario quiera aprovechar según la acción seleccionada. Debido a que el sistema Web por defecto se encuentra en inglés los elementos de las pantallas estarán en dicho idioma.

Por tanto, la estructura general de ambos menús será la siguiente:

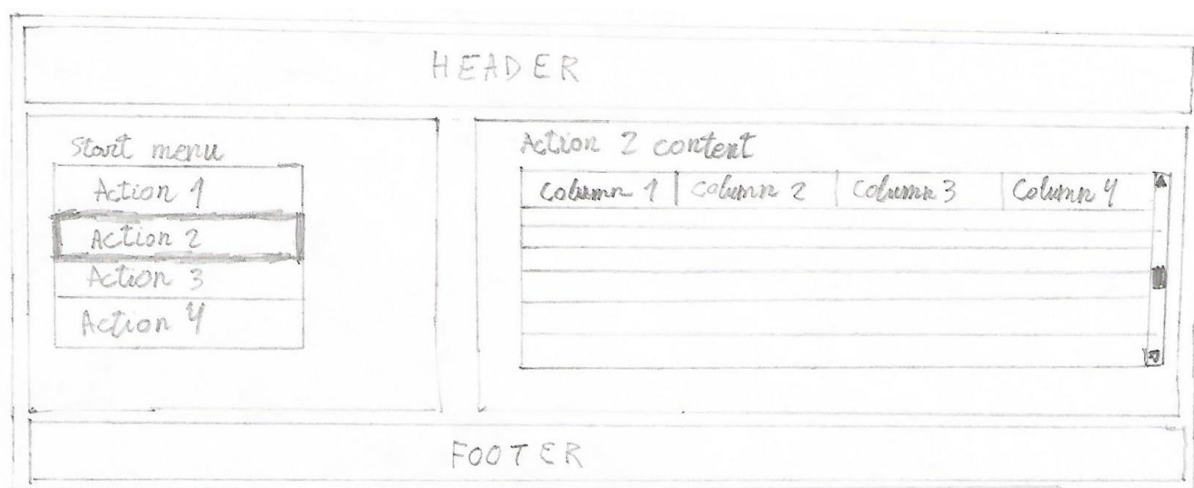


Figura 4.5. Estructura general de los menús de inicio.

Se pretende favorecer el aprendizaje del sistema Web con la estructura general de los menús de la *Figura 4.5*. Las acciones estarán en la parte izquierda del menú. El objetivo es que el usuario seleccione la acción deseada y le aparezca el contenido de dicha acción en la parte derecha de la pantalla al instante. Toda la funcionalidad de los menús será de la misma forma, por tanto se espera que el usuario se adapte rápidamente a este mecanismo.

Además, al tratarse de un diseño simple, eliminando detalles innecesarios que puedan distraer durante la interacción con la aplicación, cumple perfectamente la accesibilidad y la usabilidad de la interfaz [25].

Un cambio importante que se ha producido en la estructura del menú ha sido la inclusión en la parte izquierda de la interfaz de un selector para elegir el curso activo tanto del alumno, como del profesor.

4.2.2. DISEÑO DE PANTALLAS

En este apartado se expondrán las pantallas diseñadas para los menús de alumno y profesor con las acciones más importantes de cada uno. También se hará un diseño de las pantallas de los distintos applets para proponer, realizar y corregir ejercicios sobre los dichos applets. Igualmente se podrán consultar individualmente todas las pantallas de la aplicación en el CD adjunto a este proyecto en la subcarpeta “Pantallas” de la carpeta “Documentacion”.

Hay que tener en cuenta que el diseño se realizó, como es lógico, previamente a la implementación y que, por tanto, se han producido cambios en el diseño final del sistema Web que igualmente serán comentados.

4.2.2.1. MENÚ INICIO PROFESOR (EJERCICIOS POR CORREGIR)

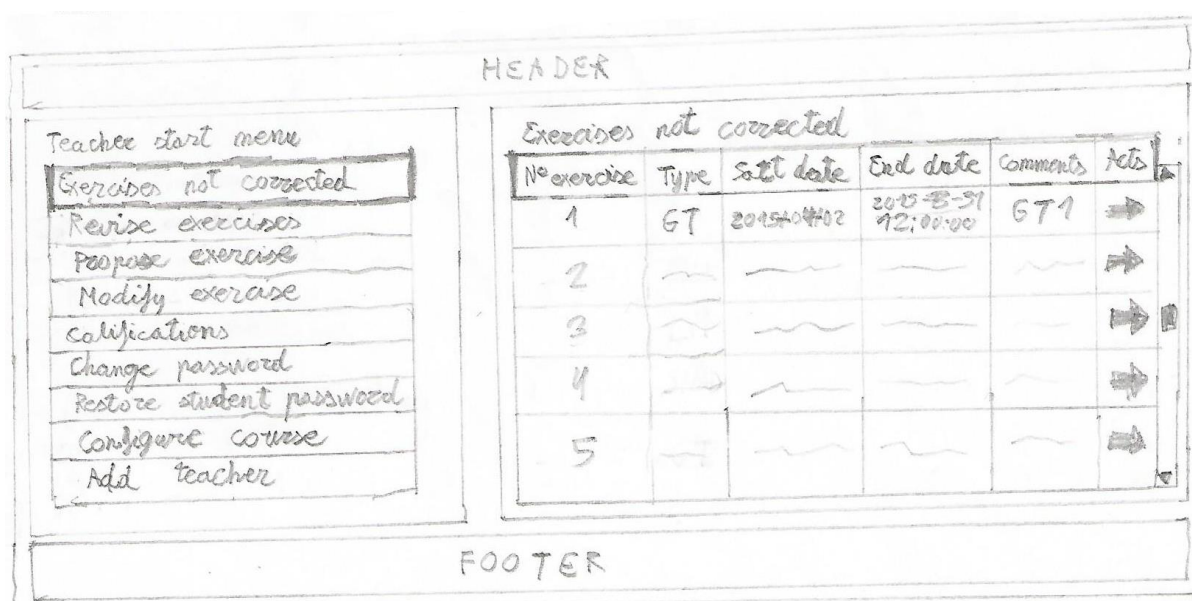


Figura 4.6. Menú de inicio del profesor.

La Figura 4.6. se corresponde con el menú de inicio del profesor. En el menú se resalta la opción correspondiente al applet seleccionado. Esta pantalla será la que verá el profesor después de iniciar sesión ya que será la acción que más realice el profesor.

Como se ha comentado previamente, en la parte izquierda se encuentra el menú con las distintas acciones que el profesor puede realizar y en la parte derecha se encuentra el contenido de dicha acción.

En esta pantalla se puede ver el contenido de la acción "Ejercicios por corregir". Deberá aparecer en la parte derecha una tabla con un scroll para ver los distintos ejercicios, ordenados descendentemente por fechas, que el profesor ha propuesto anteriormente.

Se podrá ver el número de ejercicio, el tipo de applet al que pertenece el ejercicio, la fecha de inicio de visibilidad de ejercicio, la fecha de fin de visibilidad del mismo y los comentarios asociados a dicho ejercicio para poder identificarlo.

Al final de cada fila del ejercicio se mostrará la acción correspondiente para seleccionar dicho ejercicio y mostrar los ejercicios individuales de cada alumno para dicho ejercicio propuesto como se puede observar en la *Figura 4.7*.

HEADER			
Ejercicios no corregidos			
Nº ejercicio	Type	Student	Actions
1	GT	12345678A	[Icon]
2	GT	---	[Icon]
3	GT	---	[Icon]
4	GT	---	[Icon]
5	GT	---	[Icon]

FOOTER

Figura 4.7. Listado de ejercicios por corregir resueltos por los alumnos.

La parte izquierda del menú quedará fija y sólo cambiará el contenido de la acción, donde mostrará una nueva tabla con el número de ejercicio, el tipo, el estudiante que lo ha realizado y las acciones que se pueden realizar. En este caso la acción es la de corregir dicho ejercicio.

Una vez se haya seleccionado la acción, el sistema saldrá del menú de inicio del profesor y pasará a la pantalla del applet correspondiente al ejercicio seleccionado con los datos del ejercicio realizado por el alumno.

La principal modificación que ha sufrido esta pantalla es que ha sido añadida la acción de ver un ejercicio en las columnas de acciones de la tabla de la *Figura 4.7*.

4.2.2.2. MENÚ INICIO PROFESOR (REVISAR EJERCICIOS)

Esta acción del menú de inicio del profesor es similar a la anterior. La diferencia es que en la tabla de la *Figura 4.7.* se añaden la columna "Score", que contendrá la nota que se haya asignado previamente a ese ejercicio por el profesor, y la columna "Observations" que contendrá las observaciones de la corrección que fueran escritas por el profesor al corregir ejercicio.

Una vez se pulse en la acción de modificar corrección el sistema saldrá del menú de inicio del profesor y pasará a la pantalla del applet correspondiente a la corrección realizada por el profesor del ejercicio seleccionado.

4.2.2.3. MENÚ INICIO PROFESOR (PROPONER EJERCICIO)

No exercise	Teacher	Comments	Actions
1	123456789	Picking 1	
2			
3			
4			

Figura 4.8. Menú del profesor - Proponer un ejercicio.

En la *Figura 4.8.* se puede observar que, una vez en el menú de inicio del profesor, se ha seleccionado la acción de proponer un ejercicio. Aparecerá instantáneamente el contenido de dicha acción en la parte derecha de la pantalla.

Se podrá cambiar el tipo del ejercicio que se vaya a proponer seleccionándolo en la lista desplegable con los tipos de ejercicios. Habrá que definir las fechas de inicio y fin para que aparezca la tabla con los ejercicios previamente propuestos de ese tipo.

Se pueden observar también 2 botones en la parte superior derecha del contenido. "Restore dates" limpiará los campos correspondientes a la fecha para

poder insertarlos nuevamente. El botón "Propose new exercise" aparecerá cuando se hayan insertado las fechas.

En la tabla con los ejercicios previamente propuestos se puede ver el número de ejercicio, el profesor que lo configuró, los comentarios asociados a dicho ejercicio y las acciones pertinentes, que en este caso sería seleccionar el ejercicio deseado para que el sistema cargue la pantalla del applet de dicho ejercicio y se pueda modificar si se desea para, acto seguido, proponerlo.

4.2.2.4. MENÚ INICIO PROFESOR (MODIFICAR EJERCICIO)

Teacher start menu

Modify exercise

HEADER

Type:

Nº exercise	Start date	End date	Comments	Action
1	2015-03-1 18:00:00	2015-03-8 18:00:00	Picking 1	
2	---	---	---	
3	---	---	---	
4	---	---	---	
5	---	---	---	

FOOTER

Figura 4.9. Menú del profesor - Modificar un ejercicio propuesto.

Una vez se pulse en la acción "Modify exercise" de la *Figura 4.9*, se mostrará el contenido de dicha acción. Como se puede observar aparecerá una lista desplegable para elegir el tipo de applet que se quiere modificar.

Una vez se haya elegido el tipo se mostrará la tabla con los ejercicios propuestos para ese tipo de ejercicio. Dicha tabla tendrá el número de ejercicio, las fechas de inicio y fin, los comentarios de cada ejercicio y las acciones.

En este caso la acción de cada ejercicio será un icono que, al pulsarlo, se cargará una nueva pantalla con el applet para modificar el ejercicio.

La diferencia que se ha producido respecto a esta pantalla es que al seleccionar el ejercicio el sistema carga otra pantalla para modificar las fechas del

ejercicio seleccionado si se desea. En esa pantalla también se incluirá un botón para modificar el ejercicio que hará que el sistema cargue el applet con los parámetros establecidos para ese ejercicio.

4.2.2.5. MENÚ INICIO PROFESOR (CONSULTAR NOTAS)

Teacher start menu

Califications

HEADER

Courses: 2015/2016 ▼

Groups: A ▼

Students: 12345678A ▼

No	Type	Comments	Score	Observations
1	GT	GT1	9.5	Good
2	PI	---	10	Excellent
3	NI	---	5	---
4	LYH	---	8	---

FOOTER

Figura 4.10. Menú del profesor - Consultar calificaciones.

En la *Figura 4.10.* se puede observar la acción de consultar las calificaciones de un alumno desde el menú del profesor. En la parte superior del contenido se pueden ver tres listas desplegables, una para los cursos, otra para los grupos y la última para los alumnos matriculados en el curso y grupo seleccionado.

Una vez se hayan seleccionado un elemento de las tres listas aparecerá la tabla con las calificaciones del alumno, en las cuales se puede ver el número de ejercicio, tipo, comentarios del mismo, la nota asignada y las observaciones sobre la corrección.

En esta pantalla se ha producido un cambio importante, ya que se ha tenido que incluir una funcionalidad nueva a esta acción. Dicha funcionalidad es la de exportar las calificaciones de un curso y grupo a un fichero "CSV" para facilitar al profesor la integración de las notas de los ejercicios con el resto de evaluaciones de la asignatura.

4.2.2.6. MENÚ INICIO PROFESOR (CONFIGURAR CURSO)

The wireframe shows a web interface for a teacher to configure a course. It is divided into four main sections:

- HEADER:** A horizontal bar at the top.
- Teacher start menu (Sidebar):** A vertical list on the left containing several menu items, with "Configure Course" highlighted at the bottom.
- Main Content Area:**
 - Contains the text "Configure course".
 - Below it, "Course:" followed by a dropdown menu showing "2020/2021".
 - Below that, "File with students:" followed by a "Browse" button and a text input field containing "students.csv".
- FOOTER:** A horizontal bar at the bottom.

Figura 4.11. Menú del profesor - Configurar curso.

Como en las anteriores acciones, en la parte izquierda se encuentra el menú del profesor con la acción "Configurar curso" seleccionada y en la parte derecha se encuentra el contenido de dicha acción.

En la parte derecha se encuentra una lista desplegable para seleccionar el curso en el que se desee matricular a los alumnos. Justo debajo se colocará un botón que abrirá una ventana para seleccionar el fichero "CSV" donde se encuentren los alumnos.

Cuando se seleccione el fichero, aparecerá el botón "Configurar curso". Pulsando dicho botón se matricularán a los alumnos en los cursos según el fichero seleccionado.

Las pantallas correspondientes a el resto de acciones del menú de inicio del profesor se pueden consultar en el CD adjunto al proyecto, en la carpeta "Pantallas". A continuación se detallarán las pantallas correspondientes al menú de inicio del alumno.

4.2.2.7. MENÚ INICIO ALUMNO (EJERCICIOS)

N°	Type	Start date	End date	Comments	Actions
1	GT	2015-4-8 18:00	2015-4-14 18:00	GT1	Q P
2	PICK	---	---	PCK1	Q P
3	MI	---	---	MI1	Q
4	GT	---	---	GT2	Q

Figura 4.12. Menú del alumno - Listado de ejercicios.

El menú de inicio del alumno mantiene la misma estructura que el del profesor. Las acciones se encuentran a la izquierda de la pantalla y el contenido va apareciendo instantáneamente a la derecha cuando se pulse en cada acción.

En la Figura 4.12.se puede ver como se ha seleccionado la opción "Ejercicios" que muestra el listado de ejercicios disponibles para el alumno. El contenido consta de una tabla con los ejercicios mandados para un curso.

La tabla almacena el número de ejercicio, el tipo, fecha de inicio y fin, comentarios del ejercicio y las acciones disponibles para el mismo. Entre las acciones se encontrarán "Realizar ejercicio"y "Ver ejercicio". Posteriormente se añadió la opción de "Editar ejercicio".

4.2.2.8. MENÚ INICIO ALUMNO (CALIFICACIONES)

N°	Type	Comments	Teacher	Score	Observations
1	GT	GT1	123456789	10	Good
2	PICK	PICK1	---	10	Excellent
3	MI	---	---	1	---
4	SMAD	---	---	1,4	---
5	LYM	---	---	1,5	---

Figura 4.13. Menú del alumno - Consultar calificaciones.

En la figura 4.13. se puede ver la acción de consultar calificaciones del menú del alumno. En la parte superior izquierda del contenido de la acción habrá dos listas desplegables para seleccionar el curso y el grupo. Una vez se hayan seleccionado se cargará la tabla con las notas del alumno.

La tabla almacenará el número, tipo y comentarios del ejercicio. También almacenará el profesor que lo ha corregido, la nota y las observaciones de la corrección que haya estimado oportunas.

Un gran cambio que se ha producido es la inclusión de otra columna con acciones para cada ejercicio. Esta columna contendrá una acción para que el alumno pueda ver la corrección del ejercicio.

A continuación se mostrará la estructura general de las pantallas de los applets durante la propuesta, realización y corrección de ejercicios sobre los mismos. Más adelante, en los storyboards, se verán las pantallas específicas para cada applet durante los pasos del ejercicio. Igualmente se pueden consultar las pantallas de cada applet, para cada acción, individualmente en el CD adjunto al proyecto, en la carpeta "Pantallas".

4.2.2.9. ESTRUCTURA GENERAL PARA PROPONER EJERCICIOS

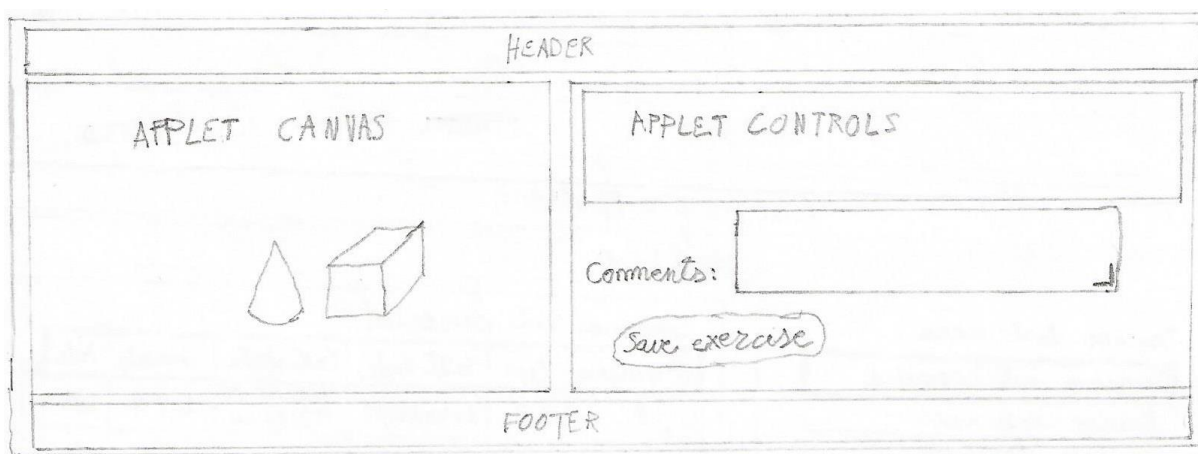


Figura 4.14. Estructura general al proponer un ejercicio de cualquier applet.

Para seguir el diseño que tiene el sistema Web previamente desarrollado durante el ciclo completo (proponer, realizar y corregir) de un ejercicio se va a

mantener dicho diseño. Éste se basa en colocar la escena en la parte izquierda de la pantalla y los controles del applet en la parte derecha.

Por tanto los elementos que sean necesarios añadir se insertarán en la parte derecha. En este caso se ha añadido un campo de texto amplio para añadir los comentarios referentes al ejercicio que se vaya a proponer. También se incluirá un botón "Guardar ejercicio" para grabar el ejercicio.

Un cambio importante que se ha producido en la estructura general de la pantalla al proponer un ejercicio es la inclusión de un apartado de instrucciones encima del pie de página donde se explicarán los pasos a seguir para proponer el ejercicio correctamente. Las instrucciones serán independientes para cada applet.

4.2.2.10. ESTRUCTURA GENERAL PARA REALIZAR EJERCICIOS

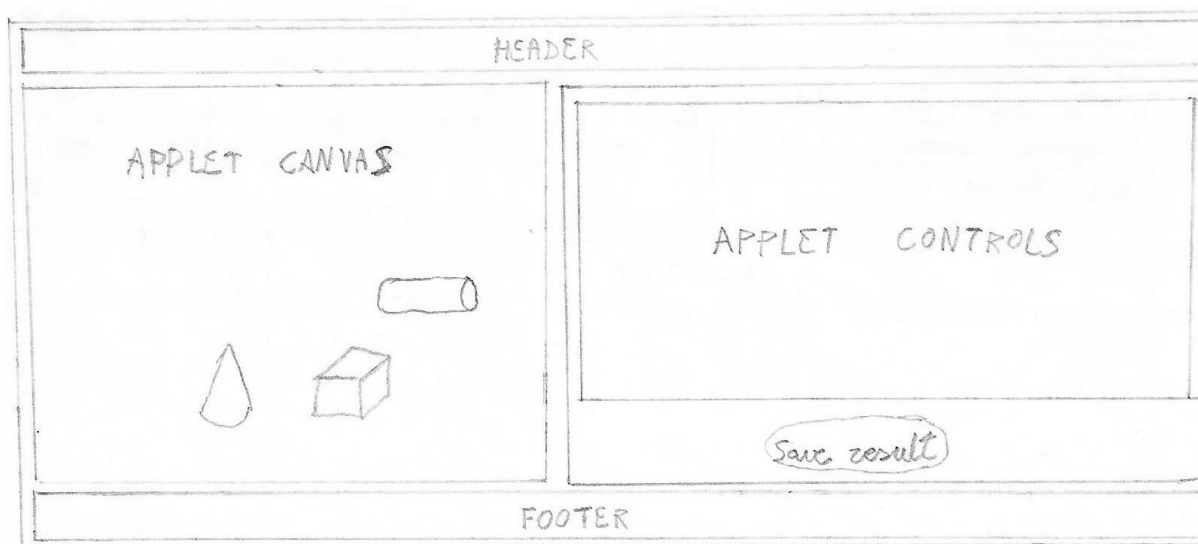


Figura 4.15. Estructura general al realizar un ejercicio de cualquier applet.

En la parte izquierda de la pantalla se colocará la escena. En algunos applet será necesario insertar dos canvas, uno con la escena correspondiente al ejercicio propuesto por el profesor, y otro con la escena que mostrará los resultados que vaya obteniendo el alumno. Si es necesario insertar dos canvas se disminuirá el tamaño de los mismos y se insertará el canvas del profesor a la izquierda, y a su derecha el canvas del alumno.

En la parte derecha se muestran los controles del applet necesarios para que el alumno pueda realizar el ejercicio correctamente. Se incluye además un botón para guardar el resultado del ejercicio.

Como se explicó en el anterior apartado se han añadido encima del "Footer" donde se explicarán, para cada applet, los pasos a seguir por el alumno para poder realizar el ejercicio correctamente.

4.2.2.11. ESTRUCTURA GENERAL PARA CORREGIR EJERCICIOS

Diagrama de la estructura general para corregir un ejercicio de cualquier applet. El diagrama muestra una interfaz dividida en tres secciones principales: un canvas a la izquierda, una tabla de parámetros y puntuaciones a la derecha, y un campo de observaciones y un botón de guardar en la parte inferior derecha.

HEADER

APPLET CANVAS

Diagrama de un applet con tres formas: un triángulo, un cubo y un círculo.

	Teacher	Student
Param1	1.3	1.2
Param2	3	3
Param3	4.5	4.5

Score: Suggested score:

Observations:

Save correction

FOOTER

Figura 4.16. Estructura general al corregir un ejercicio de cualquier applet.

Como en las anteriores pantallas el canvas (2 canvas en algunos applets) se colocará a la izquierda de la interfaz. En la parte derecha en general se incluirá una tabla con los distintos parámetros del profesor y alumno, uno al lado del otro para facilitar la corrección.

Se incluirá un campo de texto para insertar la nota y un campo de texto con la nota sugerida por el algoritmo de corrección de cada applet. Además se incluye un área de texto para que el profesor incluya las observaciones que estime oportunas sobre la corrección. Por último habrá un botón en la parte inferior para guardar la corrección del ejercicio.

En este caso no han sido incluidas las instrucciones para la corrección del applet ya que se trata de una pantalla muy intuitiva para el profesor y no es necesaria de explicar.

4.2.3. STORYBOARDS

Una vez han quedado bien definidas las pantallas de los menús y de la estructura de los ejercicios de los applets, hay que detallar las distintas interacciones dentro del sistema Web. Ésto es necesario para confirmar que el diseño realizado para las pantallas anteriores es usable o si requiere alguna modificación.

Se realizarán los storyboards para las pantallas más importantes de los menús del apartado anterior y para el ciclo completo de los ejercicios de los distintos applets. Debido a que se han producido cambios respecto a las pantallas iniciales de la aplicación, la interacción con las pantallas puede variar ligeramente de la interacción final del sistema Web.

4.2.3.1. CICLO DE EJERCICIO DE TRANSFORMACIONES GEOMÉTRICAS

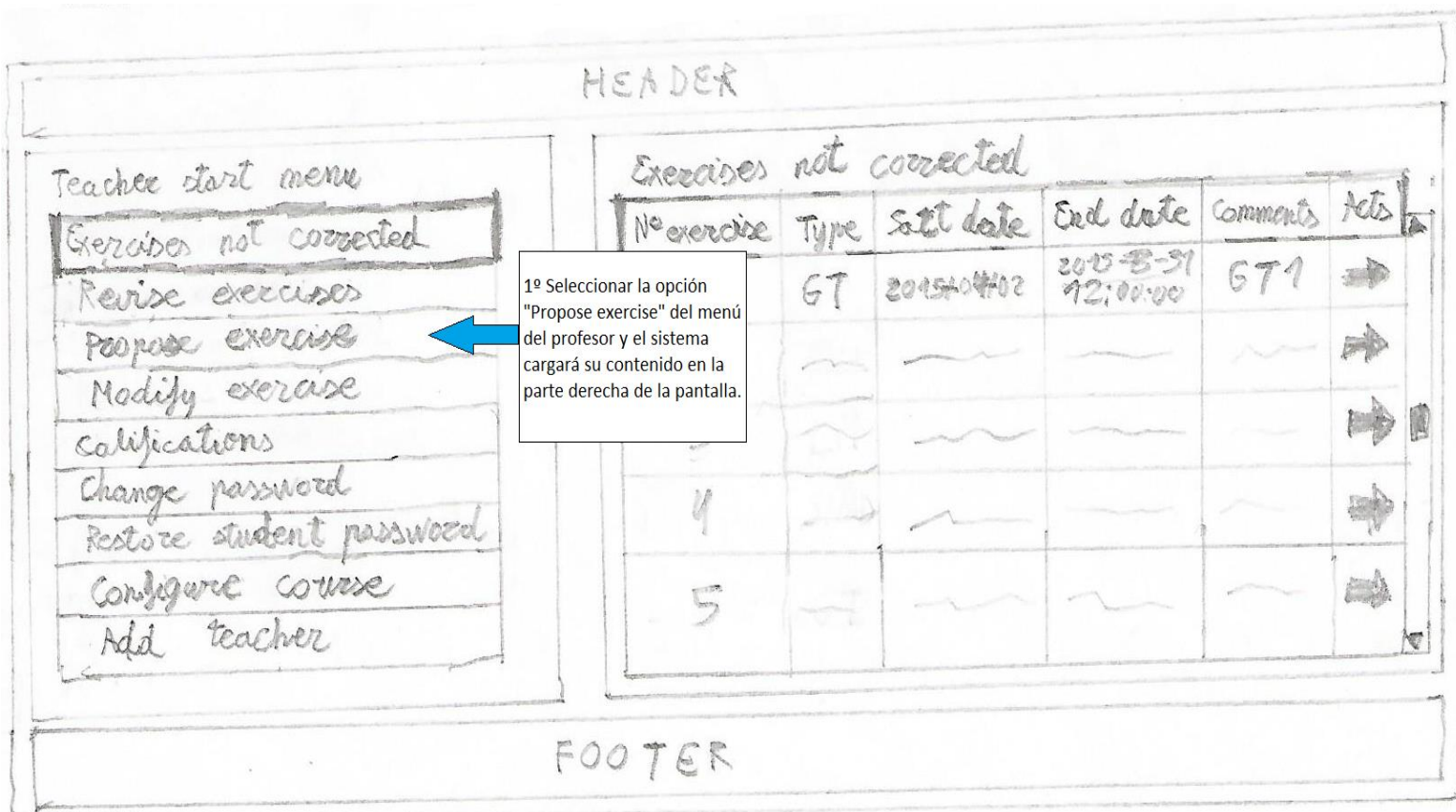


Figura 4.17. Menú Inicio profesor.

Teacher start menu

Propose exercise

Type:

Start date:

End date:

Exercises

Nº exercise	Teacher	Comments	Action
1	123456789	GT-1	
2			
3			
4			

FOOTER

Annotations:

- 1º Seleccionar el tipo de ejercicio.
- 2º Introducir la fecha de inicio.
- 3º Introducir la fecha de fin.
- 4º Pulsar en "Propose new exercise" o en la acción del ejercicio que se quiera cargar. El sistema cargará la pantalla del applet del tipo de ejercicio seleccionado.

Figura 4.18. Menú de inicio profesor. Proponer ejercicio.

Applet de transformaciones geométricas

Object:

Transformations:

- Translation: ☐ X ☐ Y ☐ Z
- Scale: ☐
- Rotation: ☐

Transformation Matrix:

☐	☐	☐	☐
☐	☐	☐	☐
☐	☐	☐	☐
☐	☐	☐	☐

Work Mode:

☐ Total ☐ Matrix

Annotations:

- 1º Aplicar al objeto las transformaciones deseadas individualmente o mediante la matriz de transformación pulsando en el botón "Send" en cualquier caso para aplicarlas.
- 2º Seleccionar el modo de trabajo "Total" o "Matrix".
- 3º Pulsar en el botón "Save exercise".

FOOTER

Figura 4.19. Applet de transformaciones geométricas - Proponer ejercicio.

Una vez el profesor haya propuesto el ejercicio (Figura 4.19), tendrá que iniciar sesión el alumno y seleccionar dicho ejercicio para poderlo realizar.

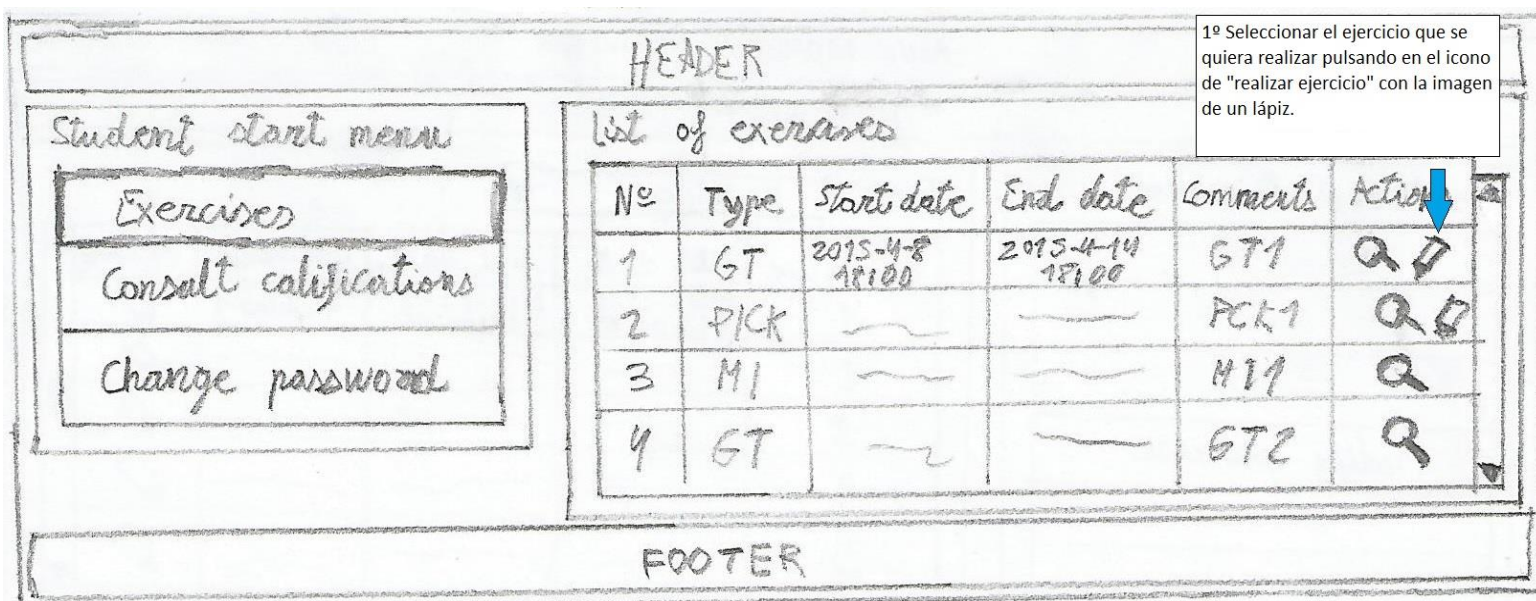


Figura 4.20. Menú de inicio alumno - Listado de ejercicios - Realizar ejercicio.

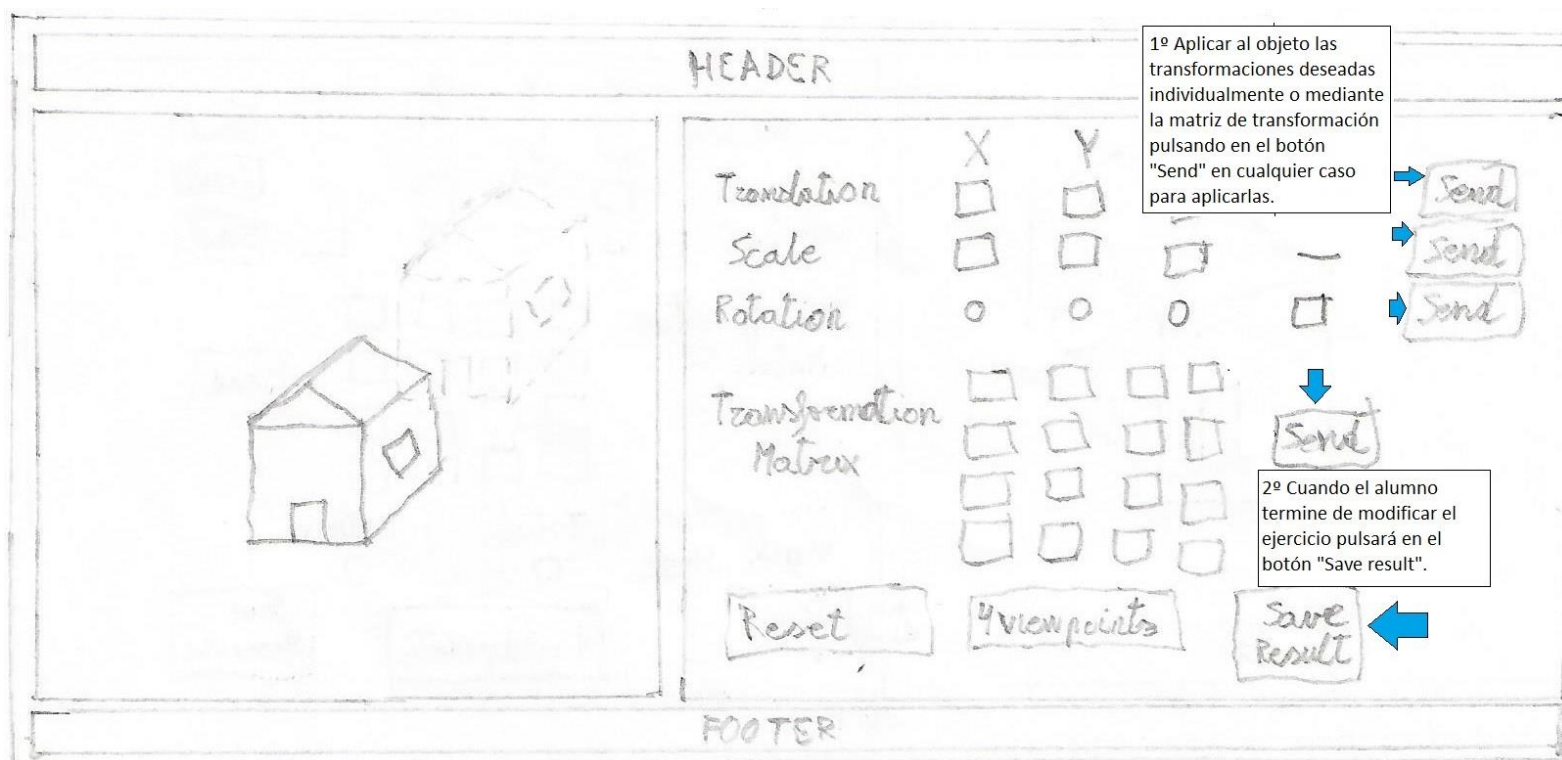


Figura 4.21. Applet de transformaciones geométricas - Realizar ejercicio.

Una vez el alumno haya resuelto el ejercicio (Figura 4.21), cuando llegue la fecha de fin del ejercicio, el profesor se dispondrá a corregirlo. Una vez iniciada la sesión le aparecerá la pantalla con los ejercicios pendientes de corrección.

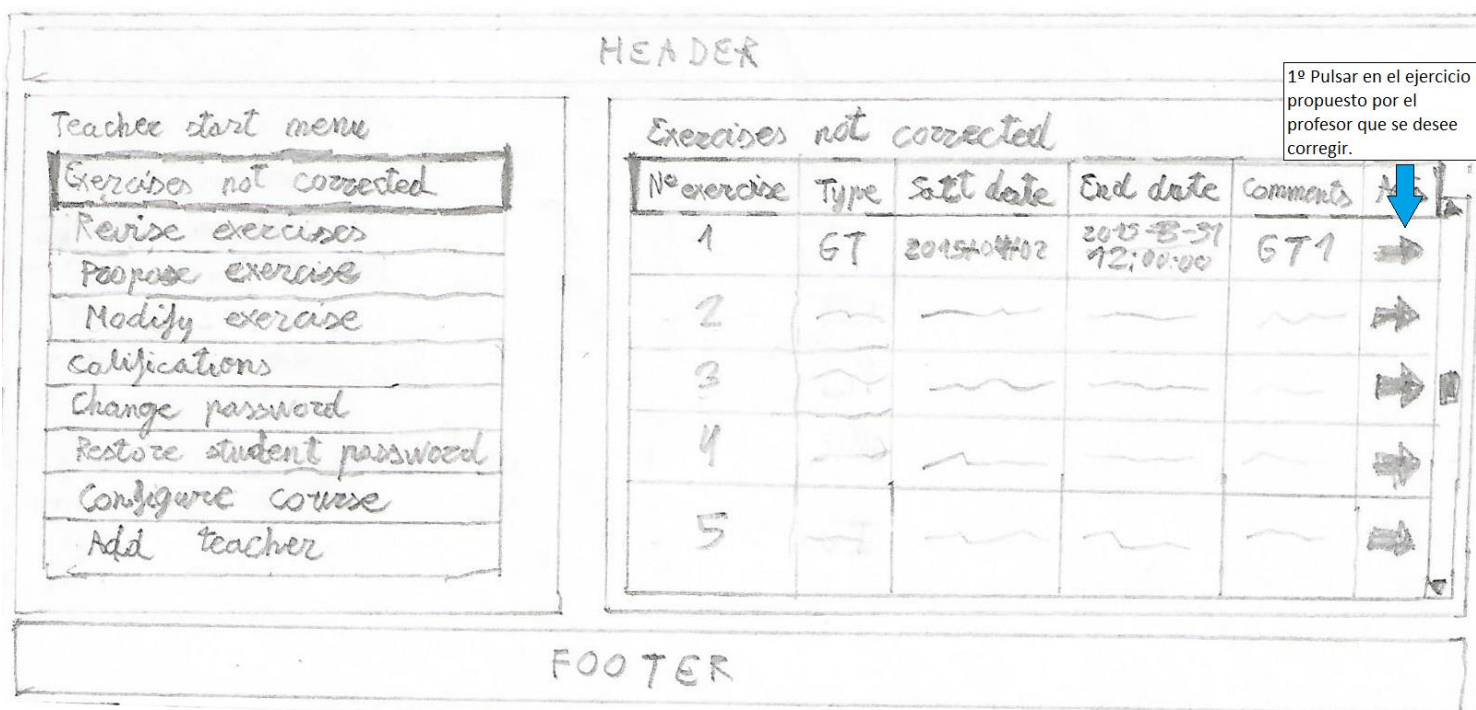


Figura 4.22. Menú inicio profesor - Ejercicios por corregir - Corregir ejercicio propuesto.

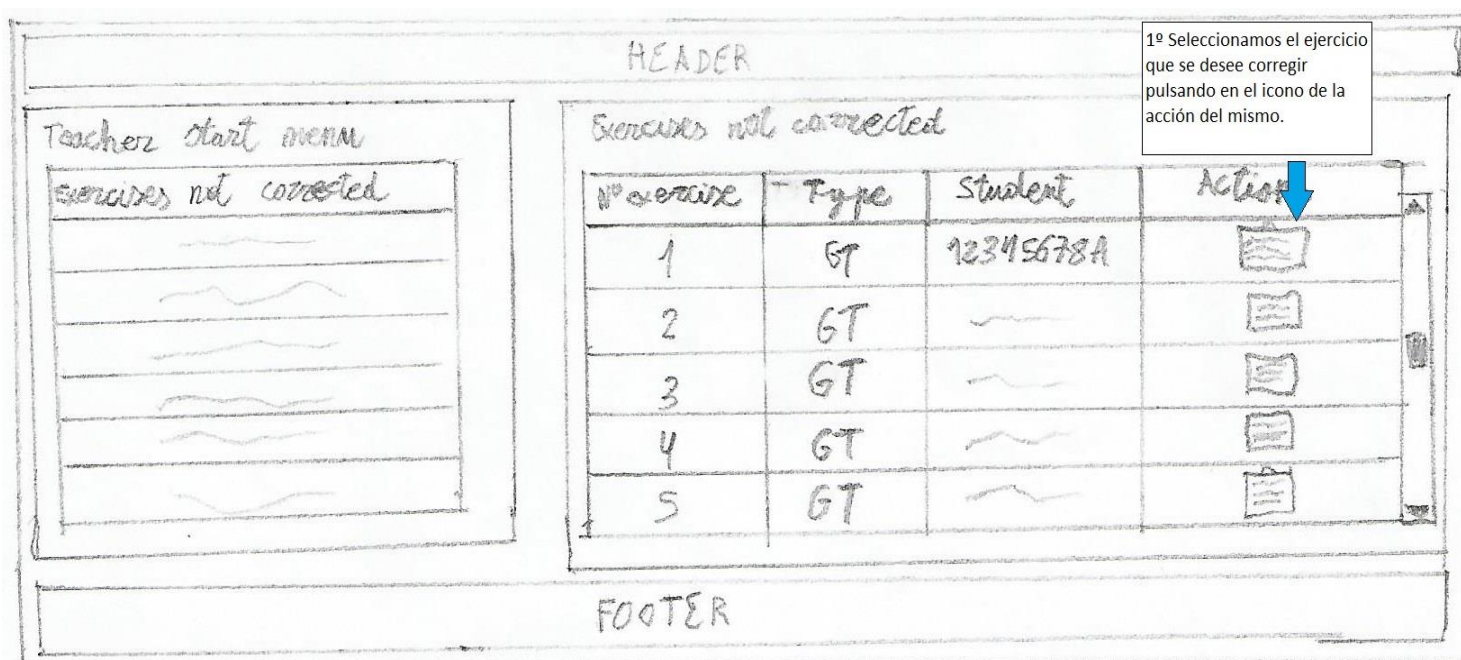


Figura 4.23. Menú inicio profesor - Ejercicios por corregir - Corregir ejercicio realizado.

Una vez el profesor haya seleccionado un ejercicio propuesto, el sistema cargará los ejercicios realizados por el alumno de dicho ejercicio. El profesor seleccionará el ejercicio que desee corregir y el sistema cargará el applet de dicho ejercicio con el contenido de la corrección.

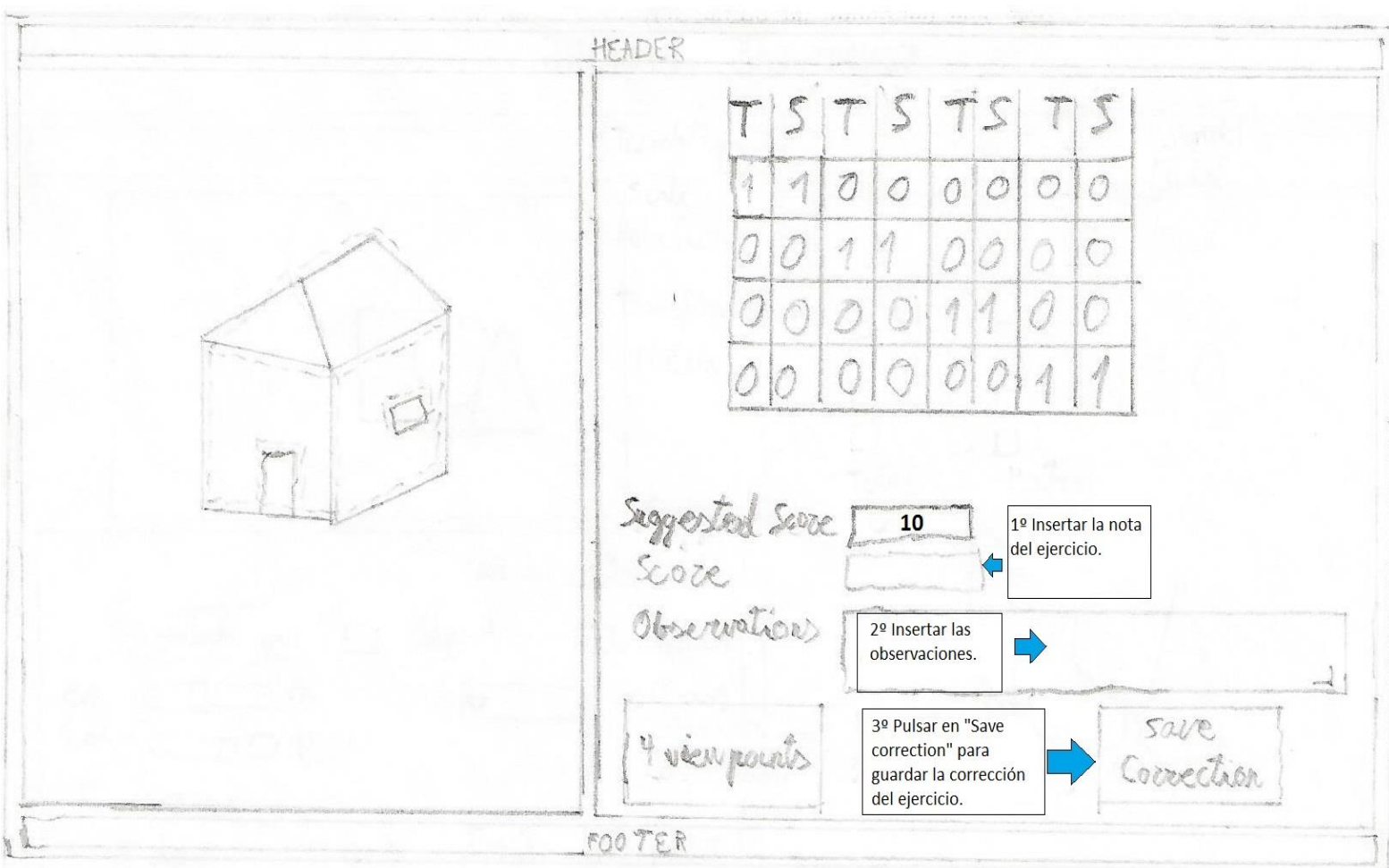


Figura 4.24. Applet de transformaciones geométricas - Corregir ejercicio.

En esta pantalla han surgido cambios considerables en el sistema Web final. Se le han añadido colores a la tabla para diferenciar las columnas del profesor y alumno. Otro cambio importante en la corrección de ejercicios en general es que se ha eliminado el campo "Nota sugerida", si es necesario calcular la nota automáticamente para un ejercicio se escribirá en el propio campo nota y el profesor podrá modificarla si lo considera oportuno.

A continuación se mostrarán el resto de storyboards para proponer, realizar y corregir ejercicios de los distintos applets. En este caso no será necesario incluir las partes de los menús de alumno y profesor, ya que esta parte permanece fija. El único cambio que se ha producido se encuentra en la selección del tipo de ejercicio.

4.2.3.2. CICLO DE EJERCICIO DE TÉCNICAS DE SELECCIÓN

En este punto el profesor se dispone a proponer un ejercicio. Ha seleccionado el tipo de ejercicio "Picking".

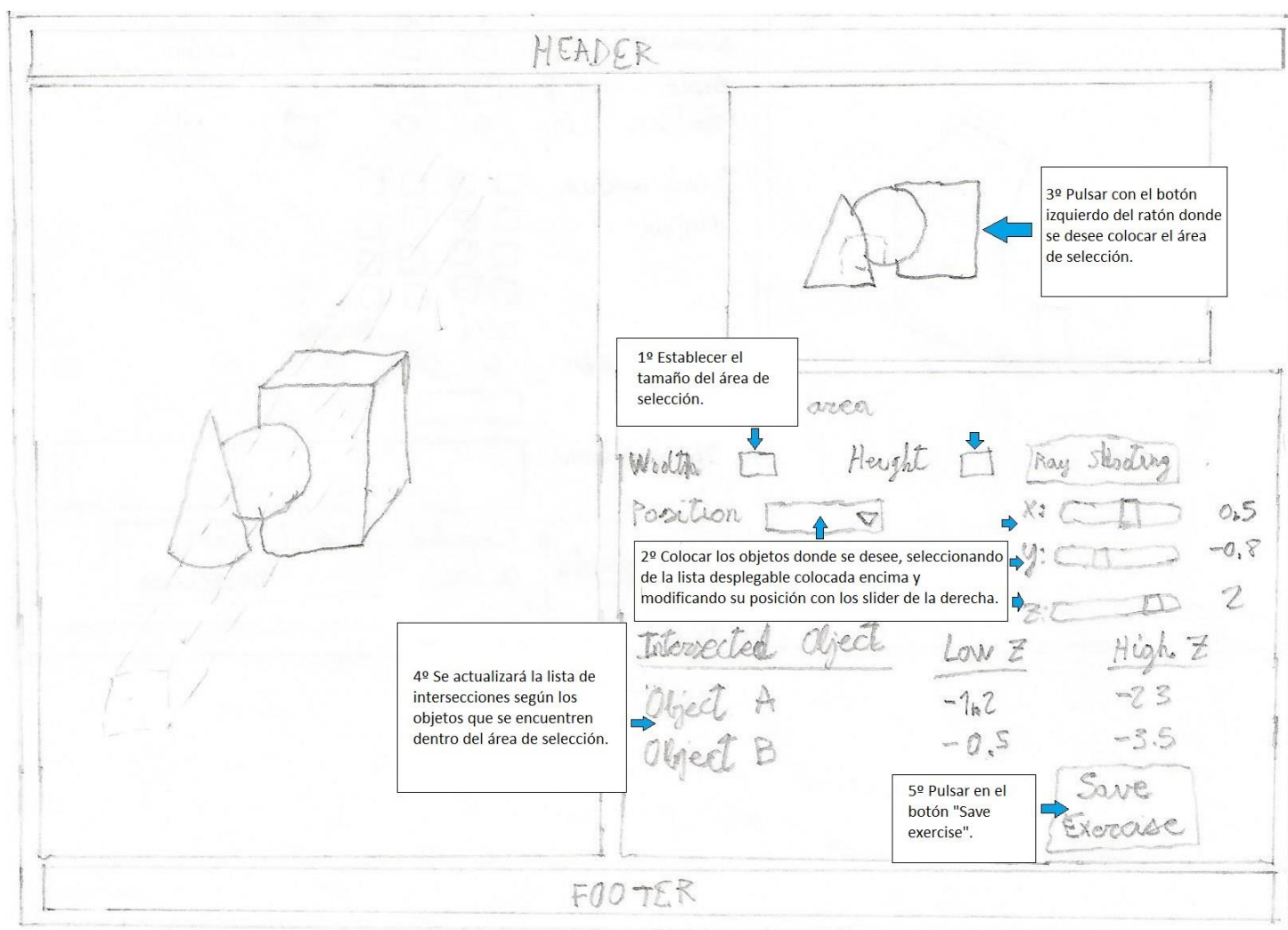


Figura 4.25. Applet de técnicas de selección - Proponer ejercicio.

El profesor configura el ejercicio y lo guarda pulsando en el botón "Save exercise" de la Figura 4.25. A continuación el alumno inicia sesión y se dispone a realizar el ejercicio. Para ello selecciona la acción del ejercicio que va a realizar y el sistema carga el applet con los parámetros del ejercicio establecidos por el profesor.

En la Figura 4.26. el alumno modifica el ejercicio intentando conseguir la lista de colisiones del profesor.

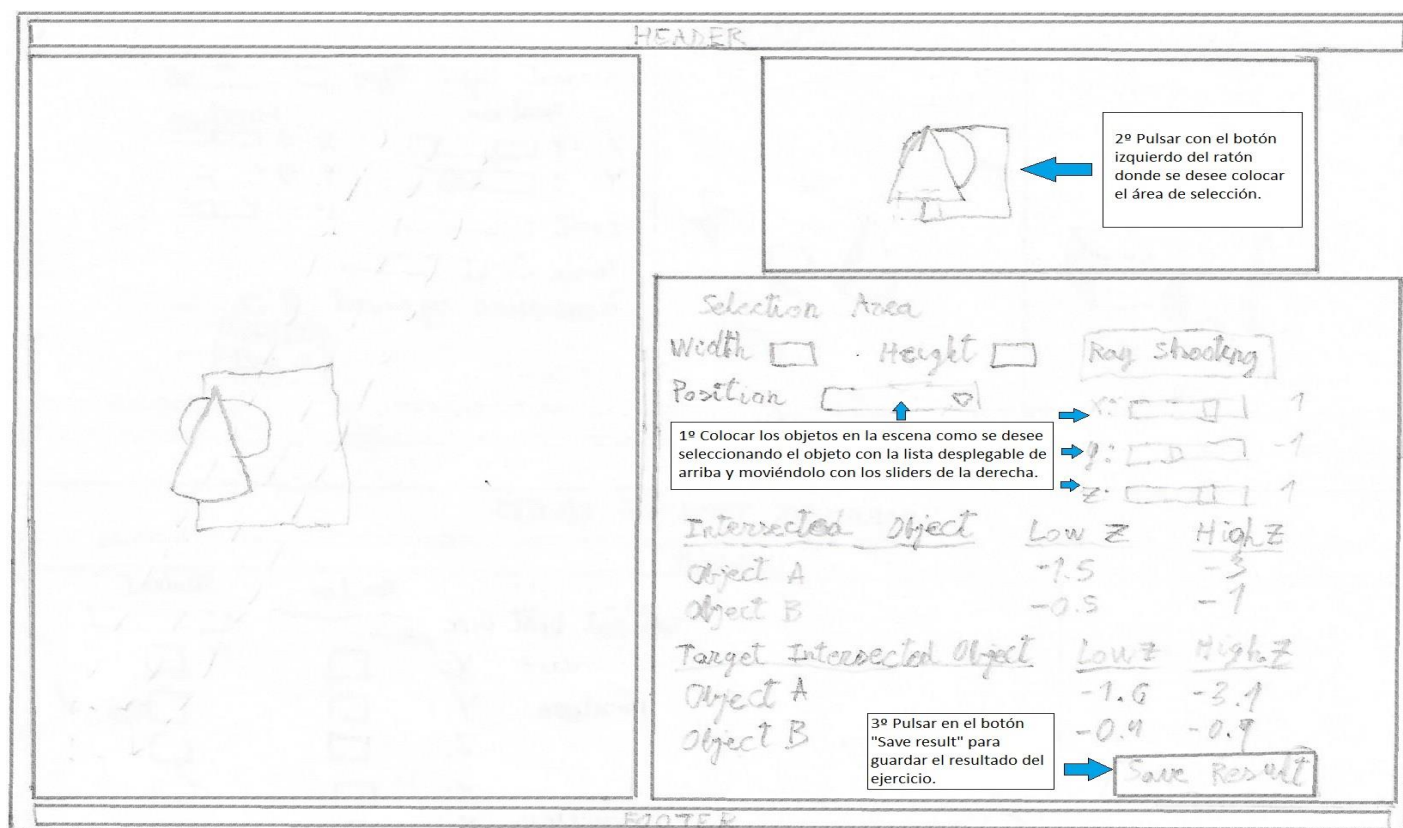


Figura 4.26. Applet de técnicas de selección - Realizar ejercicio.

Una vez el ejercicio ha sido realizado, el profesor se dispone a corregirlo. Para ello selecciona el ejercicio propuesto y el sistema cargará la lista de alumnos que lo han resuelto. El profesor seleccionará un ejercicio y se cargará la pantalla del applet.

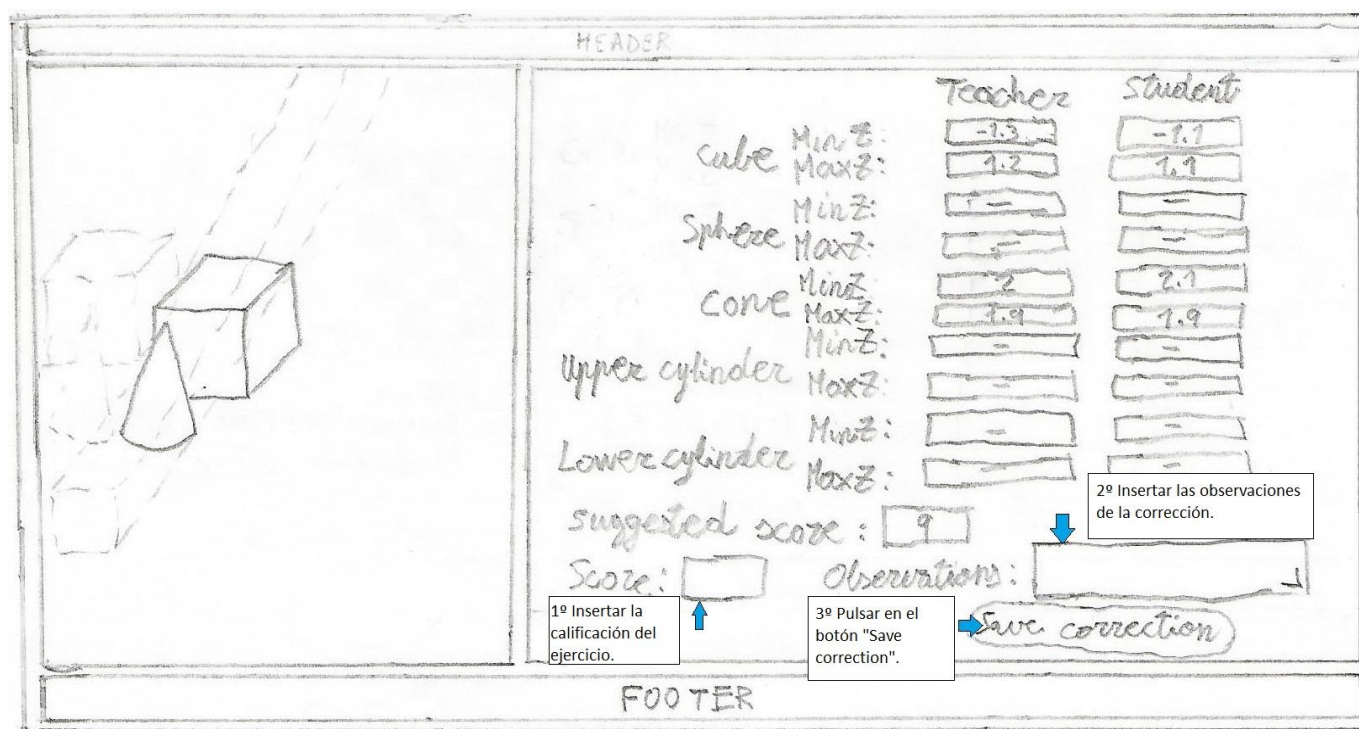


Figura 4.27. Applet de técnicas de selección - Corregir ejercicio.

4.2.3.3. CICLO DE EJERCICIO DE TIPOS DE LUCES

El profesor accede a la pantalla de la *Figura 4.28*. desde su menú de inicio.

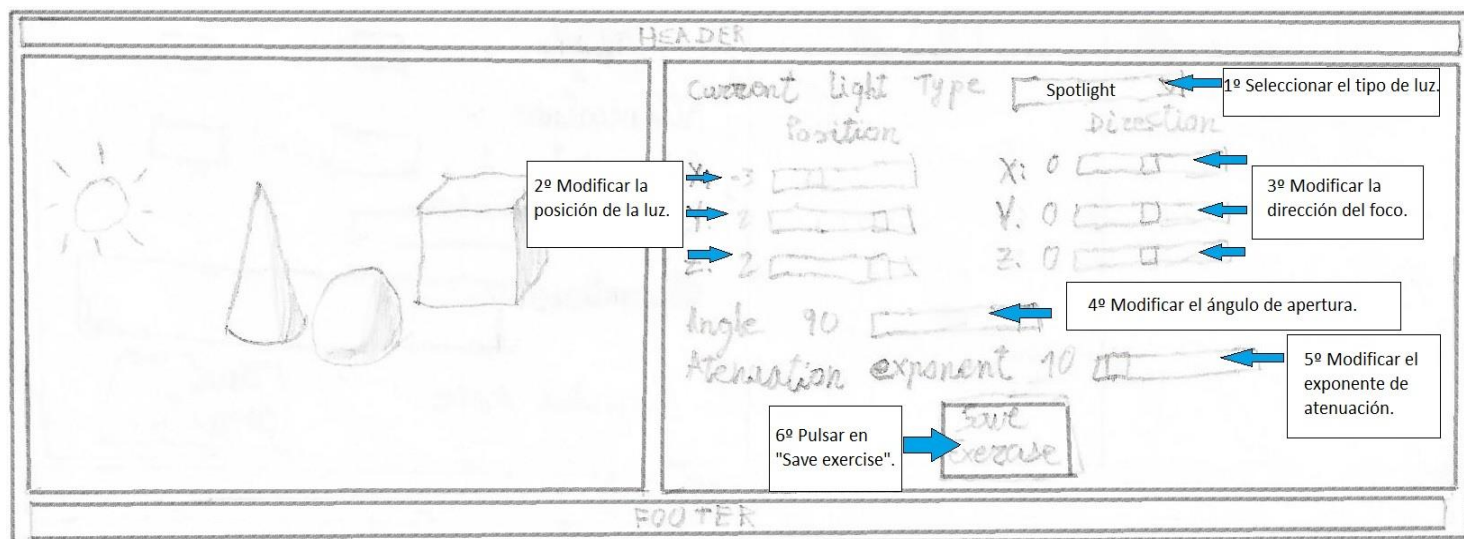


Figura 4.28. Applet de tipos de luces - Proponer ejercicio.

Una vez ha propuesto el ejercicio, el alumno accede al sistema Web y procede a realizar dicho ejercicio.

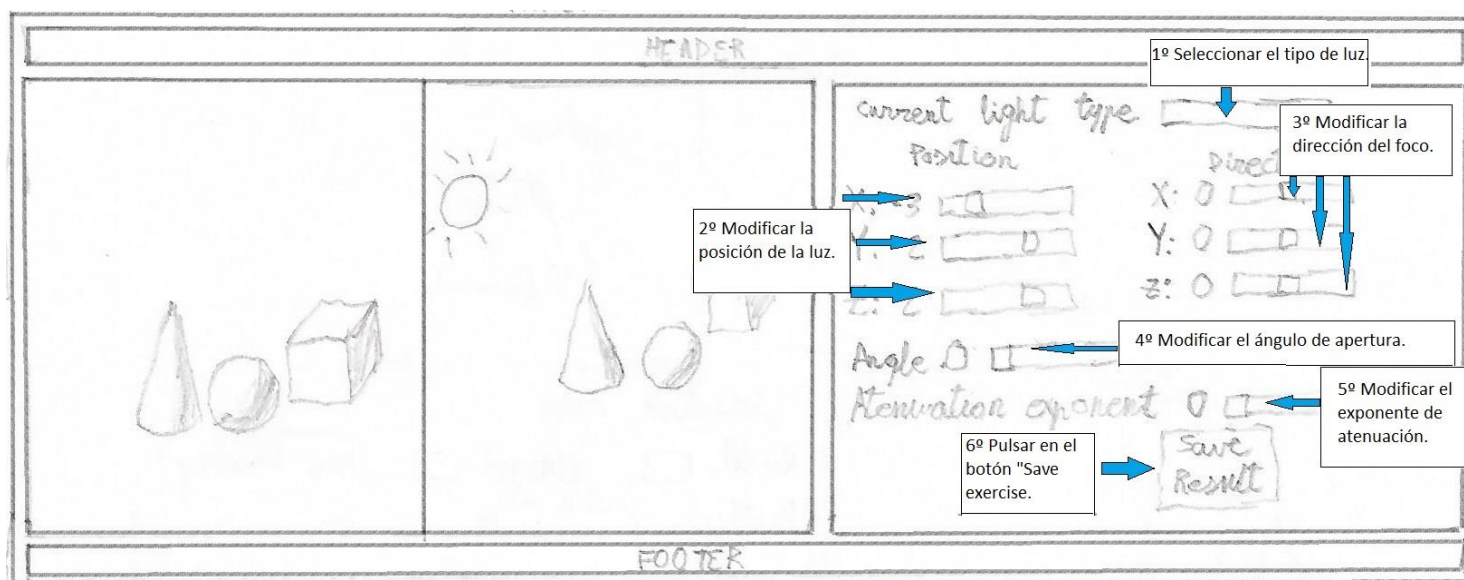


Figura 4.29. Applet de tipos de luces - Realizar ejercicio.

El alumno tiene que conseguir el objetivo mostrado en la escena izquierda modificando los parámetros de la luz. Cuando termine guardará el ejercicio y el profesor iniciará sesión para corregirlo.

	Teacher	Student
Current light type	Spotlight	Spotlight
Position	X: 0	0
	Y: 1	1
	Z: 1	1
Direction	X: 1	1
	Y: 0	0
	Z: 1	0
Angle :	0	1
Attenuation exponent :	1	1
Score	1	1
Comments	LYM1	
Suggested score :	9	

Figura 4.30. Applet de tipos de luces - Corregir ejercicio.

4.2.3.4. CICLO DE EJERCICIO DE LUCES Y MATERIALES

Light parameters	
Position	Radial attenuation
X: -1	a0: 1
Y: 1	a1: 0
Z: 1.3	a2: 0
Specular color	Diffuse Color
R: 1	R: 1
G: 1	G: 1
B: 1	B: 0
A: 1	A: 1
Material coefficients	
Ambient	Specular
R: 0.3	R: 0.5
G: 0.3	G: 0.8
B: 0.8	B: 0.1
A: 1	A: 1
Exponent :	428
Comments	LYM1

Figura 4.31. Applet de luces y materiales - Proponer ejercicio.

HEADER

Light parameters

Position: X: 1, Y: 1, Z: 1.3

Radial attenuation: a0: 1, a1: 0, a2: 0

Ambient color: R: 0.3, G: 0.3, B: 0.3, A: 1

Specular color: R: 1, G: 1, B: 1, A: 1

Diffuse Color: R: 1, G: 1, B: 0, A: 1

Material coefficients

Ambiental: R: 0.3, G: 0.3, B: 0.8, A: 1

Specular: R: 1, G: 1, B: 1, A: 1

Diffuse: R: 0.5, G: 0.8, B: 0.1, A: 1

Phong exponent: 128

Save result

FOOTER

1º Modificar los parámetros de la luz.

2º Modificar los coeficientes del material.

3º Pulsar en el botón "Save result".

Figura 4.32. Applet de luces y materiales - Realizar ejercicio.

HEADER

Position: X: 1, Y: 1, Z: 1.3

Radial attenuation: a0: 0.1, a1: 0, a2: 0.3

Ambient color: R: 0.3, G: 0.3, B: 0.3, A: 0.3

Specular color: R: 1, G: 1, B: 1, A: 1

Diffuse color: R: 1, G: 0, B: 0, A: 1

Material coefficient ambiental: R: 1, G: 1, B: 1, A: 1

Material coefficient specular: R: 0.6, G: 0.3, B: 1, A: 1

Material coefficient diffuse: R: 0.5, G: 0.8, B: 0.6, A: 1

Phong exponent: 128

Teacher

Position	Teacher	Student
X:	1	1.1
Y:	1	1.1
Z:	1.3	1.3
a0:	0.1	0.1
a1:	0	0.1
a2:	0.3	0.3
R:	0.3	0.3
G:	0.3	0.3
B:	0.3	0.3
A:	0.3	0.3
R:	1	1
G:	1	1
B:	1	1
A:	1	1
R:	1	1
G:	1	1
B:	1	1
A:	1	1
R:	0.6	0.6
G:	0.3	0.3
B:	1	1
A:	1	1
R:	0.5	0.5
G:	0.8	0.8
B:	0.6	0.6
A:	1	1
Phong exponent:	128	128

Student

1º Insertar la calificación del ejercicio.

2º Insertar las observaciones de la corrección.

3º Pulsar en el botón "Save correction".

Save correction

Figura 4.33. Applet de luces y materiales - Corregir ejercicio.

4.2.3.5. CICLO DE EJERCICIO DE MODELOS DE ILUMINACIÓN

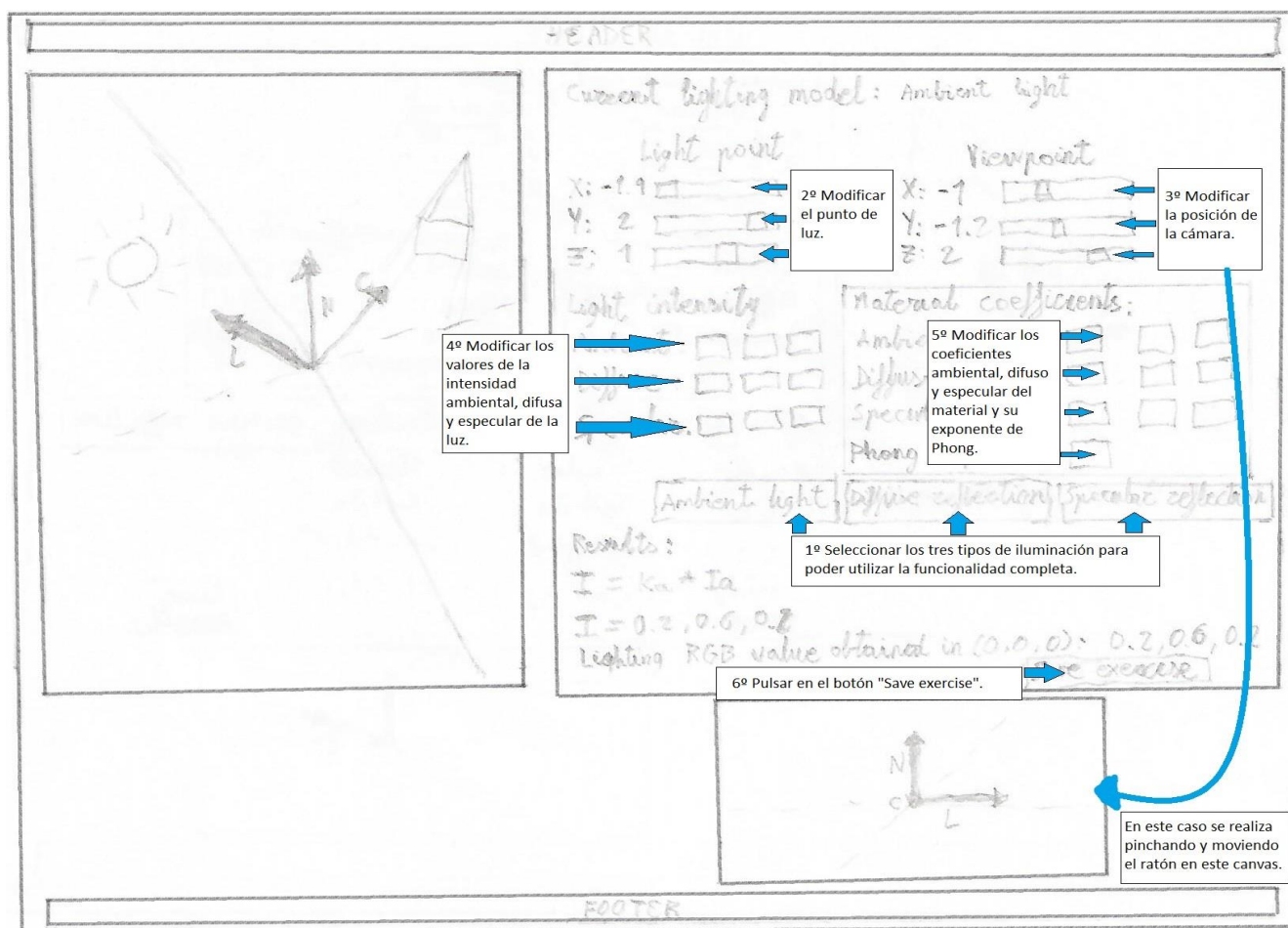


Figura 4.34. Applet de modelos de iluminación - Proponer ejercicio.

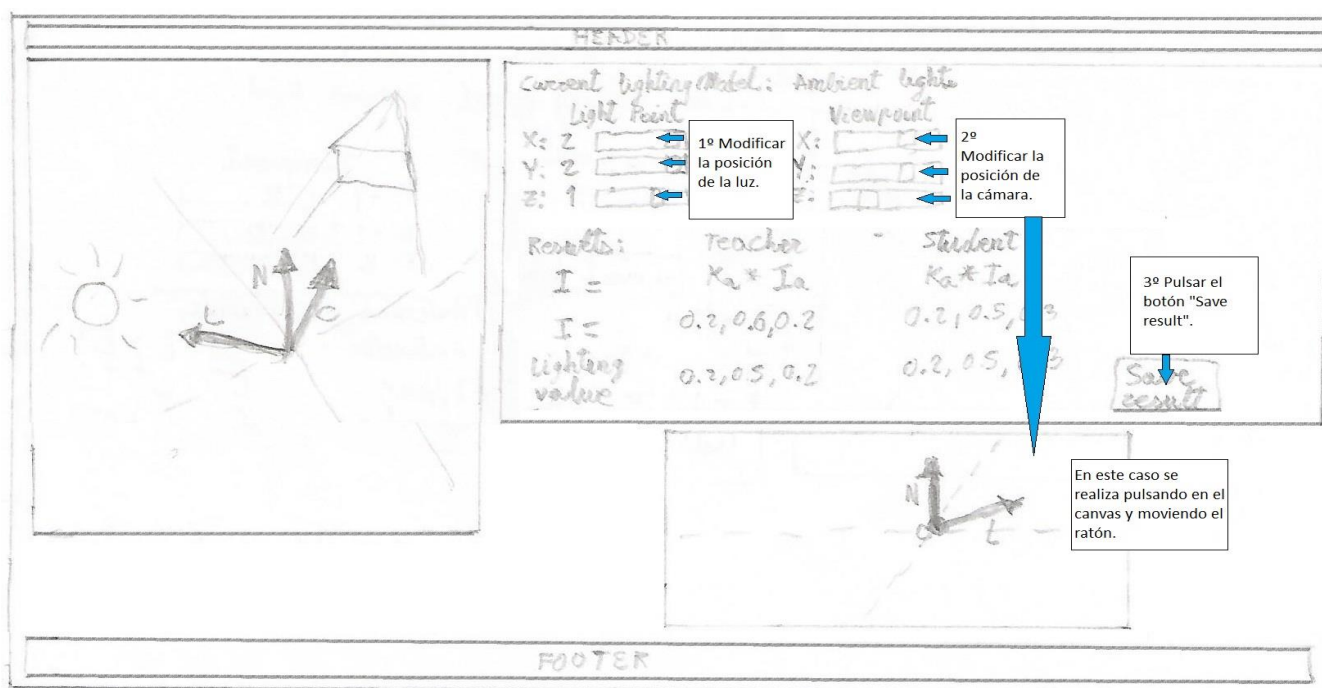


Figura 4.35. Applet de modelos de iluminación - Realizar ejercicio.

HEADER

Current lighting model: Ambient light

	Teacher	Student
Light point	X: 1.5	1.5
	Y: 1.5	1.5
	Z: 0.6	0.6
Viewpoint	X: 0.6	0.6
	Y: 0.6	1
	Z: 0.6	0.6
Results	I: $R_a * I_a$	$R_a * I_a$
	I: 0.2, 0.6, 0.2	0.2, 0.6, 0.2
Lighting value	0.2, 0.6, 0.2	0.2, 0.6, 0.2

1º Insertar la calificación del ejercicio.

2º Insertar las observaciones de la corrección.

Suggested score: 6

Save correction

3º Pulsar en el botón "Save correction".

FOOTER

Figura 4.36. Applet de modelos de iluminación - Corregir ejercicio.

4.2.3.6. CICLO DE EJERCICIO DE SOMBREADO

HEADER

Sphere parameters

Horizontal divisions: 21

Vertical divisions: 21

Display type and shading

☐ Wire
☐ Flat lighting
☐ Gouraud
☐ Phong

1º Modificar el número de divisiones horizontales de la esfera.

2º Modificar el número de divisiones verticales de la esfera.

3º Seleccionar el tipo de iluminación y sombreado para la escena.

4º Pulsar en el botón "Save exercise".

Save Exercise

FOOTER

Figura 4.37. Applet de sombreado - Proponer ejercicio.

En el applet final de las Figura 4.37, 4.38 y 4.39 se han eliminado el número de divisiones verticales y horizontales y se ha añadido un único slider con las divisiones de la esfera debido a que se producían problemas en la esfera al tener un número distinto de divisiones.

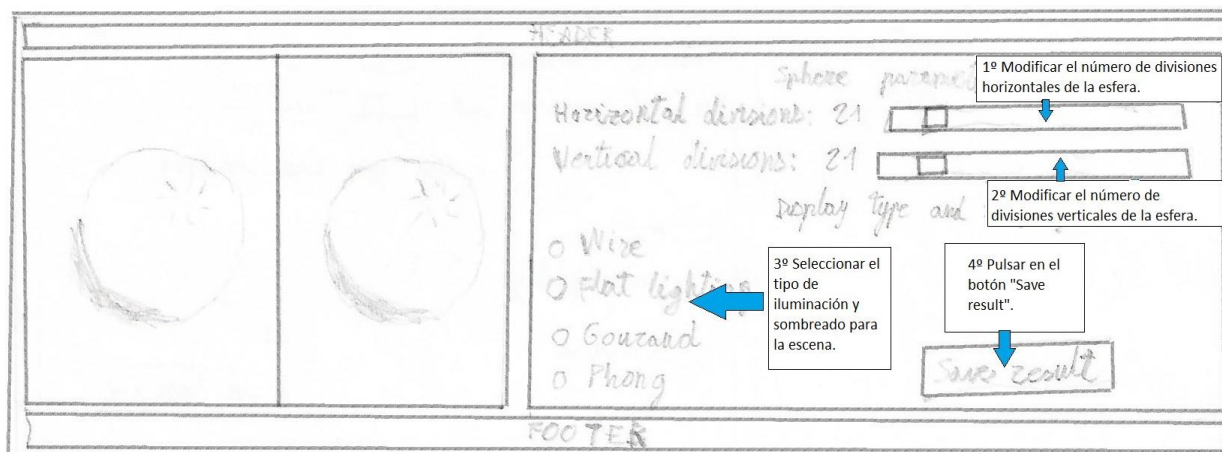


Figura 4.38. Applet de sombreado - Realizar ejercicio.

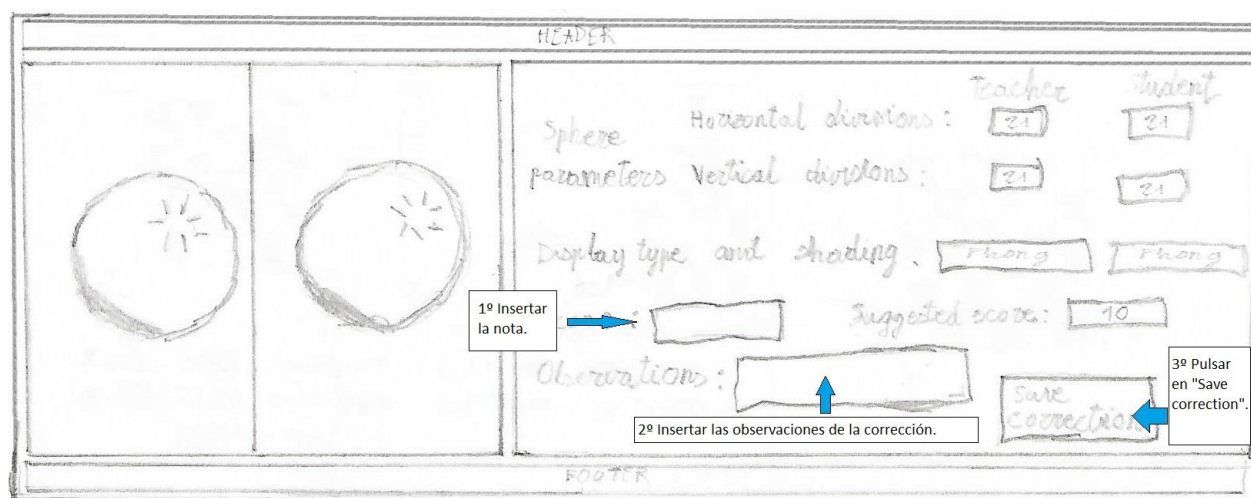


Figura 4.39. Applet de sombreado - Corregir ejercicio.

4.2.3.7. CICLO DE EJERCICIO DE TEXTURAS

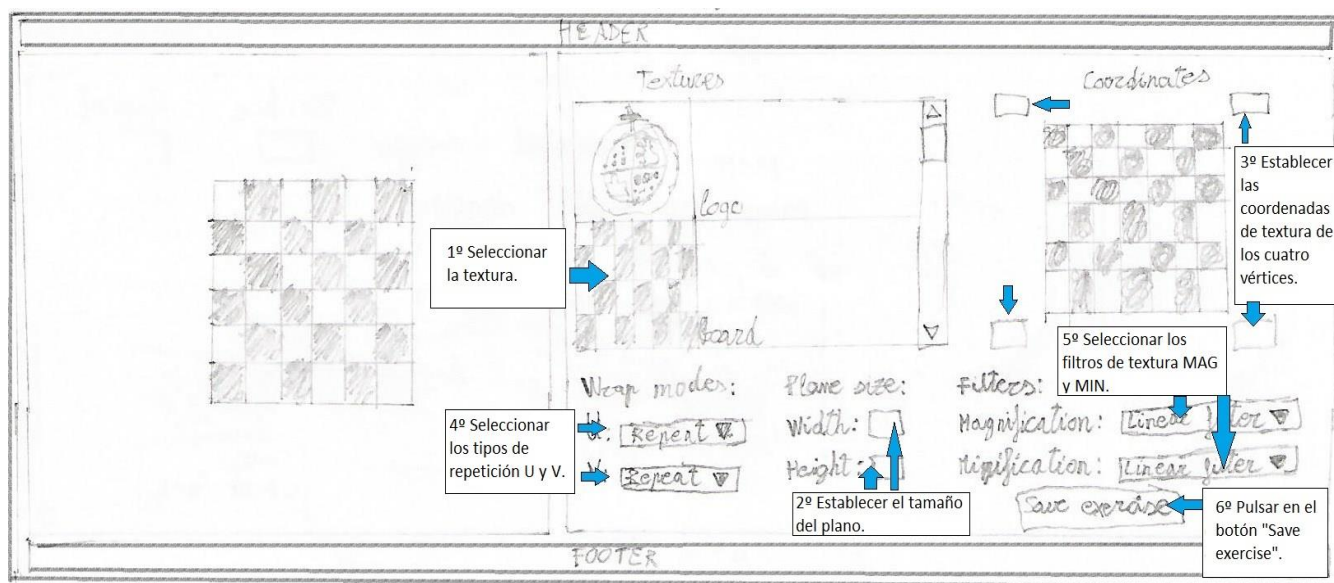


Figura 4.40. Applet de texturas - Proponer ejercicio.

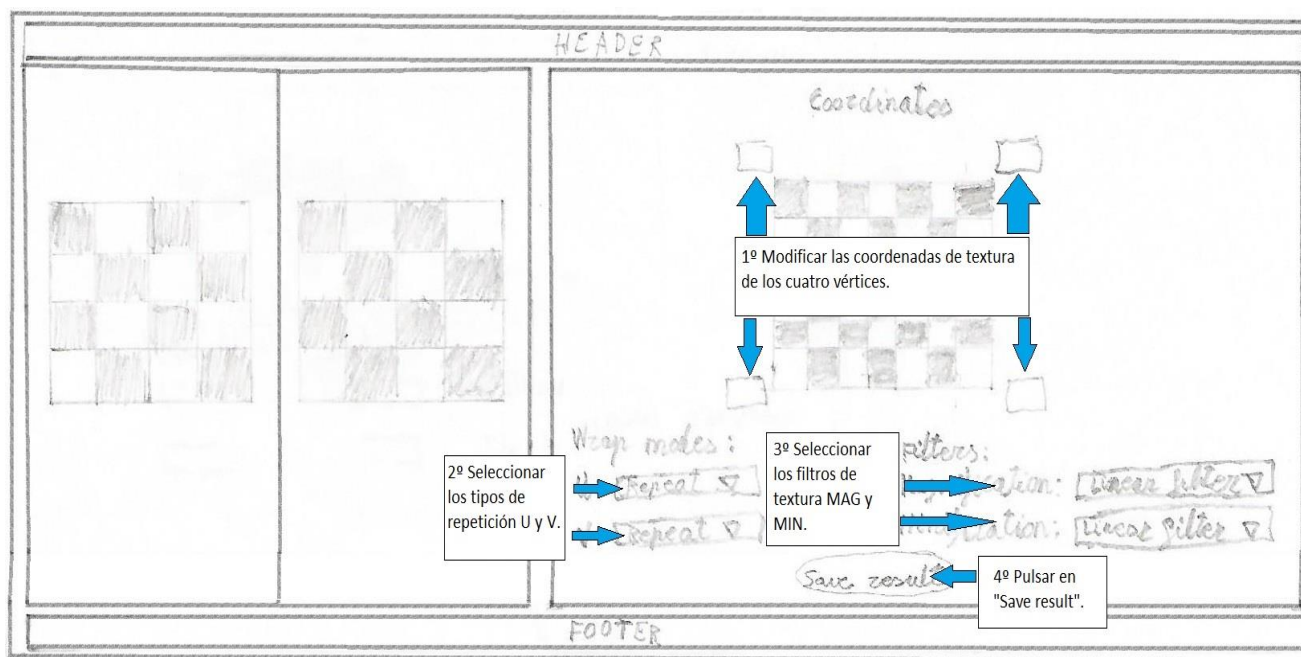


Figura 4.41. Applet de texturas - Realizar ejercicio.

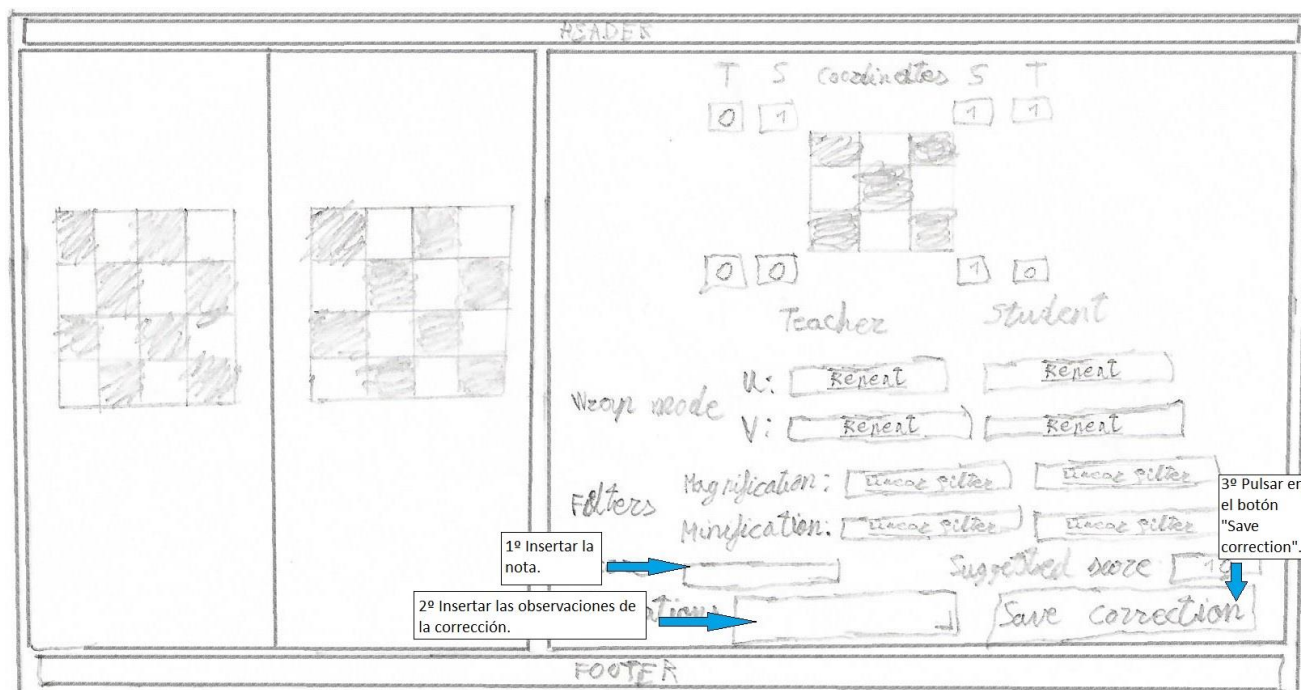


Figura 4.42. Applet de texturas - Corregir ejercicio.

En la pantalla de corregir ejercicio de la *Figura 4.42*. se ha eliminado el dibujo de la textura para mostrar las coordenadas. En su lugar se han insertado campos para mostrar por columnas los datos del alumno y profesor de igual forma que los demás parámetros mostrados como los tipos de repetición y los filtros de textura.

4.2.4. DIAGRAMAS DE CLASES

En este apartado se detallan los distintos diagramas de clases asociados a los menús de inicio de alumno y profesor. Debido a que el sistema Web desarrollado durante este proyecto debe mantener la misma funcionalidad que el sistema Web del que partía, la estructura básica del diseño de los applets no ha variado sustancialmente.

Sin embargo se han producido algunos cambios en los applets en los que se han tenido que añadir un segundo canvas para la realización o la corrección de un ejercicio sobre el mismo. Estos cambios se traducen en añadir una nueva vista y un nuevo controlador para manejar el funcionamiento de ese nuevo canvas.

Para todos los applets se han añadido funciones para gestionar la nueva funcionalidad que se ha implementado sobre los mismos. Estas funciones se han integrado en el sistema manteniendo el patrón Modelo-Vista-Controlador (MVC) y el patrón Observador que incluían el sistema Web previamente desarrollado.

Para conseguir una correcta integración de los menús en la aplicación a la hora de establecer un ejercicio con los applets, éstos han sido implementados de la misma forma que los applets de la aplicación, es decir, cada uno se trata de una capa HTML (`<div>`) que se carga en la sección **content** del fichero `index.html`.

Cada acción de los menús se ha diseñado de forma que no sea necesario cargar el contenido de la misma constantemente. Esto se ha conseguido insertando el contenido de cada acción en una capa `<div>` dentro de los ficheros `inicioAlumno.html` y `inicioProfesor.html`. Con esto se pretende que el usuario experimente una navegación fluida por los menús, ya que el navegador no tiene que recargar el menú con el contenido necesario para cada acción.

En la *Figura 4.43.* se puede ver la estructura general del sistema Web. En la figura se muestran solo los ficheros en español con la terminación "ES" pero el esquema general es equivalente para los ficheros en inglés con la terminación "EN". Como se puede observar, además de los ficheros javascript incluidos en el `index.html` se han añadido nuevos ficheros que han sido necesarios:

- Login.js: Contiene la función para iniciar sesión en el sistema. Dicha función se llama cuando el usuario pulsa el botón "Login" de la cabecera. Se realiza una llamada AJAX [13] para contactar con la parte servidor y crear la sesión del usuario conectado. Una vez se haya iniciado sesión el sistema carga el menú de inicio correspondiente.
- Logout.js: Contiene la función para cerrar sesión en el sistema. Se realiza una llamada AJAX y se elimina la sesión activa en la parte servidor. Cuando se haya destruido la sesión el sistema cargará la página de demostración de la aplicación.
- Helpers.js: Contiene varias funciones de ayuda necesarias para el correcto funcionamiento del menú y de los applets. Este fichero contiene también funciones encargadas de cargar el applet requerido desde el menú de alumno o profesor y cargar los mismos desde cada applet.
- Notificaciones.js: Contiene las notificaciones en inglés y español que se mostrarán durante la navegación de la aplicación cuando se realice una acción importante.

Debido a que se trata de un sistema Web, los diagramas de clase son diferentes a una aplicación normal de escritorio, de esta forma se muestra en primer lugar la estructura general interna del sistema web. Posteriormente se muestran los diagramas de clases asociados a los menús de alumno y profesor.

Por último se mostrará un diagrama general del funcionamiento de la parte servidor, así como de su comunicación con la parte cliente de la aplicación.

No se expondrán los diagramas de clase para cada applet en concreto ya que son muy similares respecto al sistema Web previamente desarrollado. Igualmente se adjuntan en el CD del proyecto, dentro de la subcarpeta "Diagramas de Clase" de la carpeta "Documentacion", para poder consultarlos individualmente.

4.2.4.1. DISEÑO GENERAL DEL SISTEMA WEB

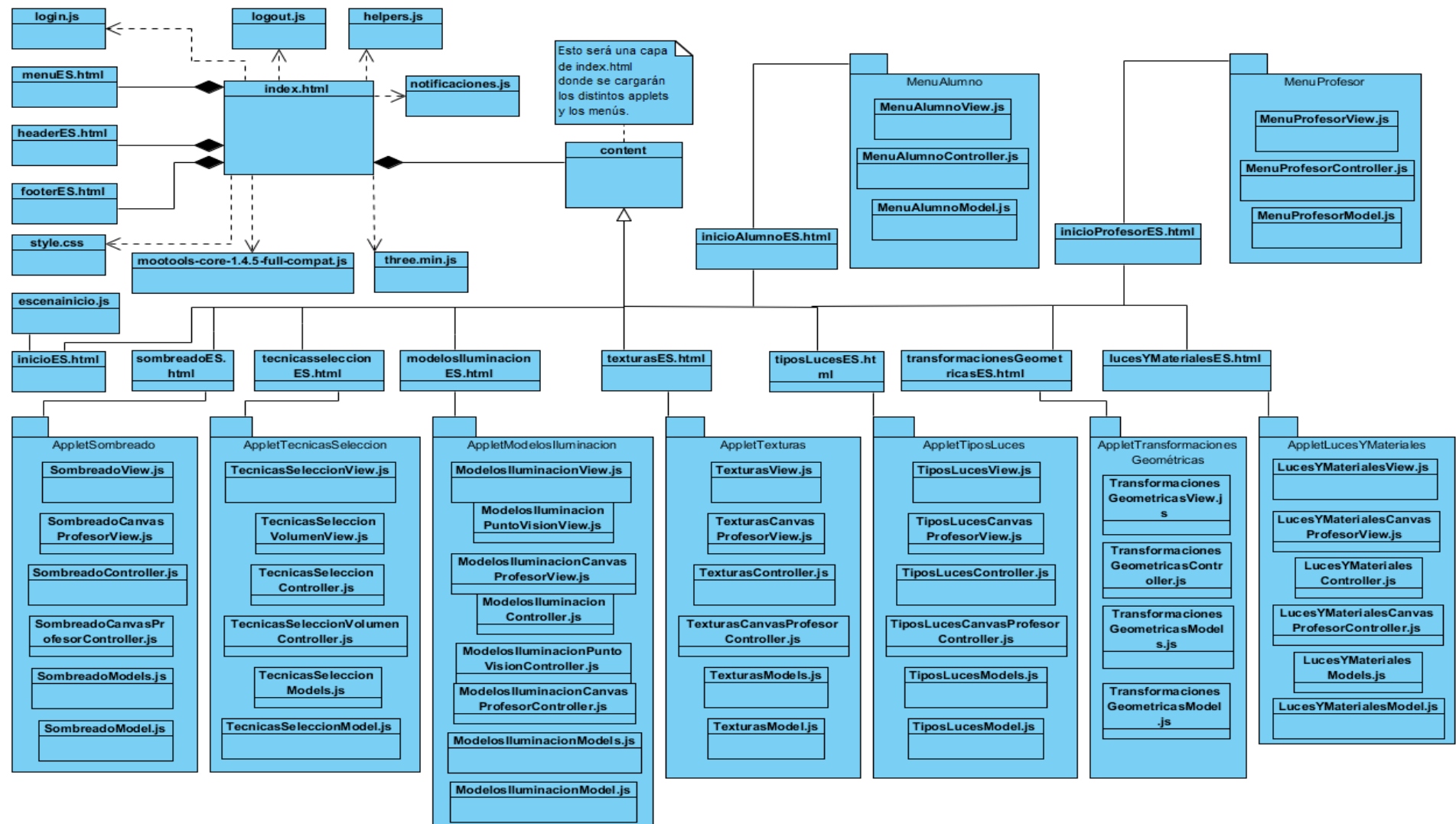


Figura 4.43. Diagrama general de la estructura del sistema Web.

4.2.4.2. MENU INICIO PROFESOR

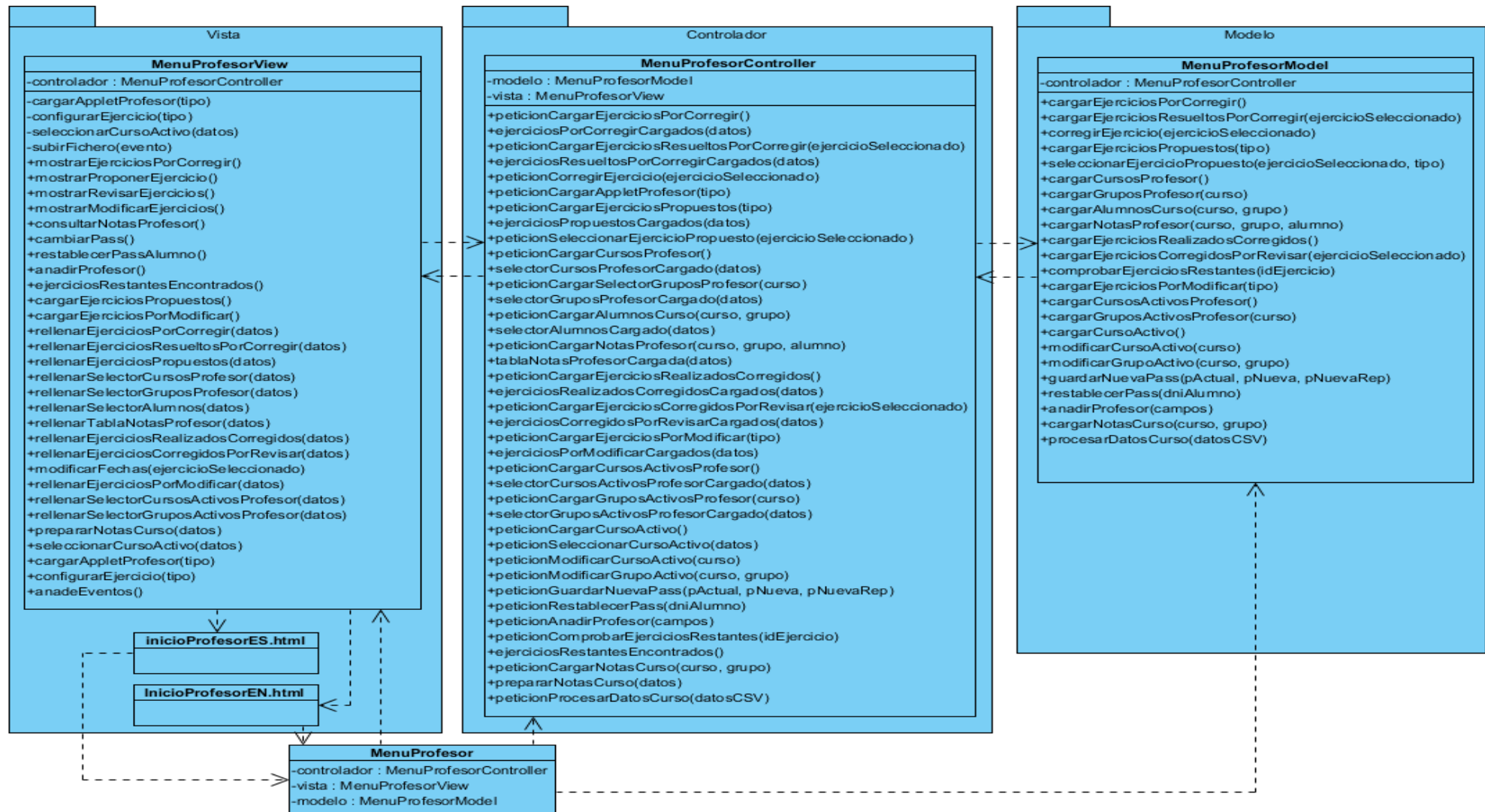


Figura 4.44. Diagrama de clases del menú de inicio del profesor.

4.2.4.3. MENU INICIO ALUMNO

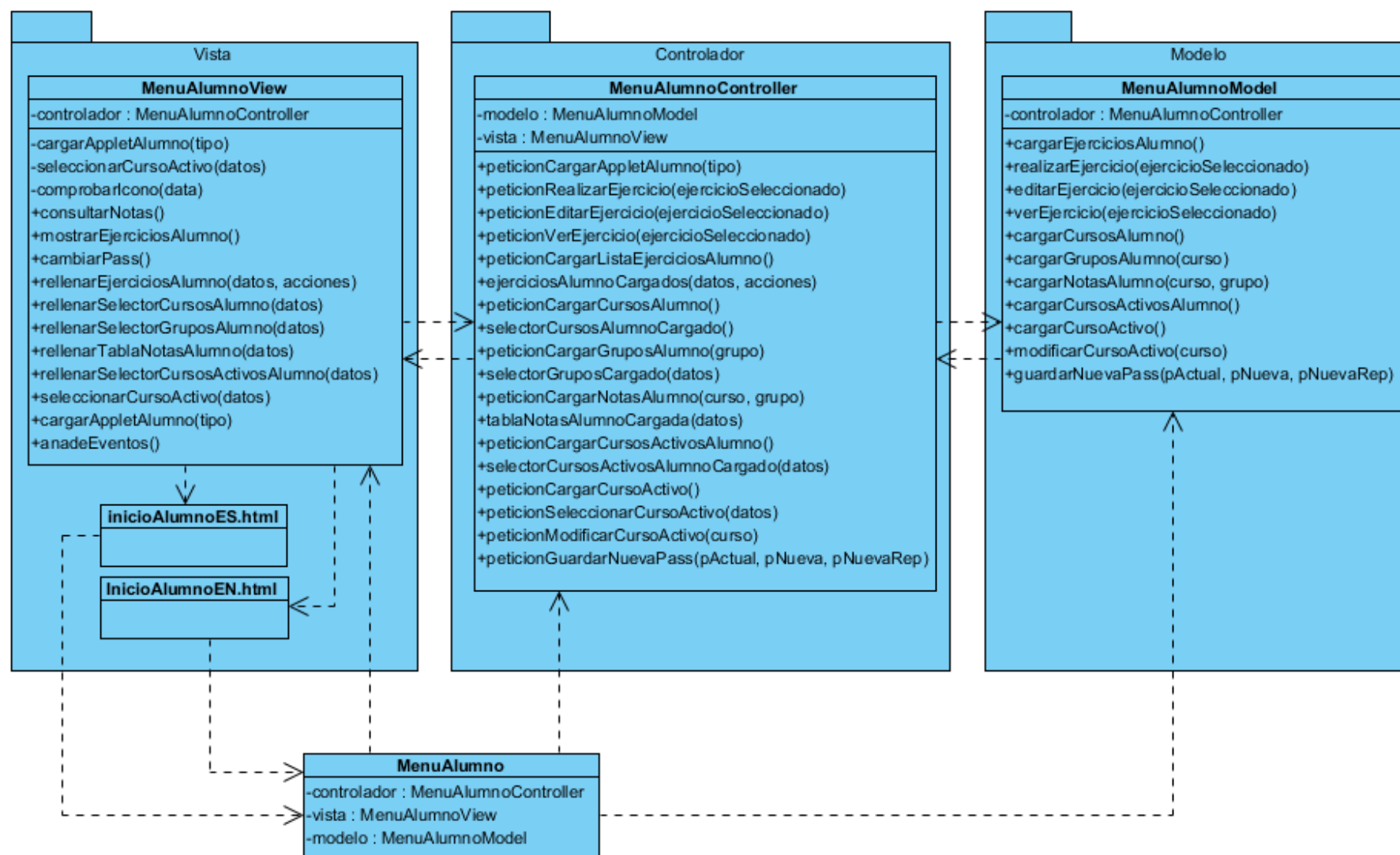


Figura 4.45. Diagrama de clases del menú de inicio del alumno.

En los anteriores diagramas (*Figura 4.44 y 4.45*) se puede observar el funcionamiento de los menús de alumno y profesor. Ambos menús tendrán un fichero "menuProfesor/Alumno.js" que será el encargado de crear toda la estructura del modelo-vista-controlador.

La vista interactuará directamente con los ficheros html de los menús, tanto en español como en inglés. También pedirá al controlador la carga de los datos necesarios de los ejercicios, alumnos, etc. Éste comunicará la petición al modelo que se encargará de realizar las peticiones AJAX [14] para comunicar la parte cliente con la parte servidor. Estas peticiones devolverán la información, normalmente, en forma de vector. El modelo comunicará los datos al controlador y éste a la vista. Ésta última será la encargada de tratar los datos y mostrarlos en el fichero html.

Cuando el usuario se disponga a proponer, realizar, modificar, ver o corregir algún ejercicio se controlará qué se desea realizar sobre qué applet a través del valor de variables globales, a las que tendrán acceso tanto los menús, como los applets, para poder identificar que se desea hacer sobre el mismo.

Para la parte servidor se expone el esquema general de su funcionamiento en la *Figura 4.46*. Para comunicar la parte cliente con la parte servidor se utilizan llamadas AJAX al fichero ".php" que tenga la función que se vaya a realizar. En este fichero se validarán los datos que se vayan a almacenar para controlar que no se inserta desde la parte cliente nada que no se deba insertar.

Una vez realizado esto se llama a la función correspondiente del fichero "accesoDB.php". Este fichero es el encargado de realizar cualquier consulta a la base de datos. Tiene incluidos el fichero "conectarDB.php" para conectarse a la base de datos de la aplicación y el fichero "validacion.php" con funciones ayuda de validación.

Cuando se termine de efectuar la operación con la base de datos se volverá al fichero ".php" que fue llamado desde AJAX, y éste se encargará de devolver la respuesta conveniente a la parte cliente para poder tratar los datos o la notificación adjunta a la respuesta.

4.2.4.4. DIAGRAMA GENERAL DE LA PARTE SERVIDOR

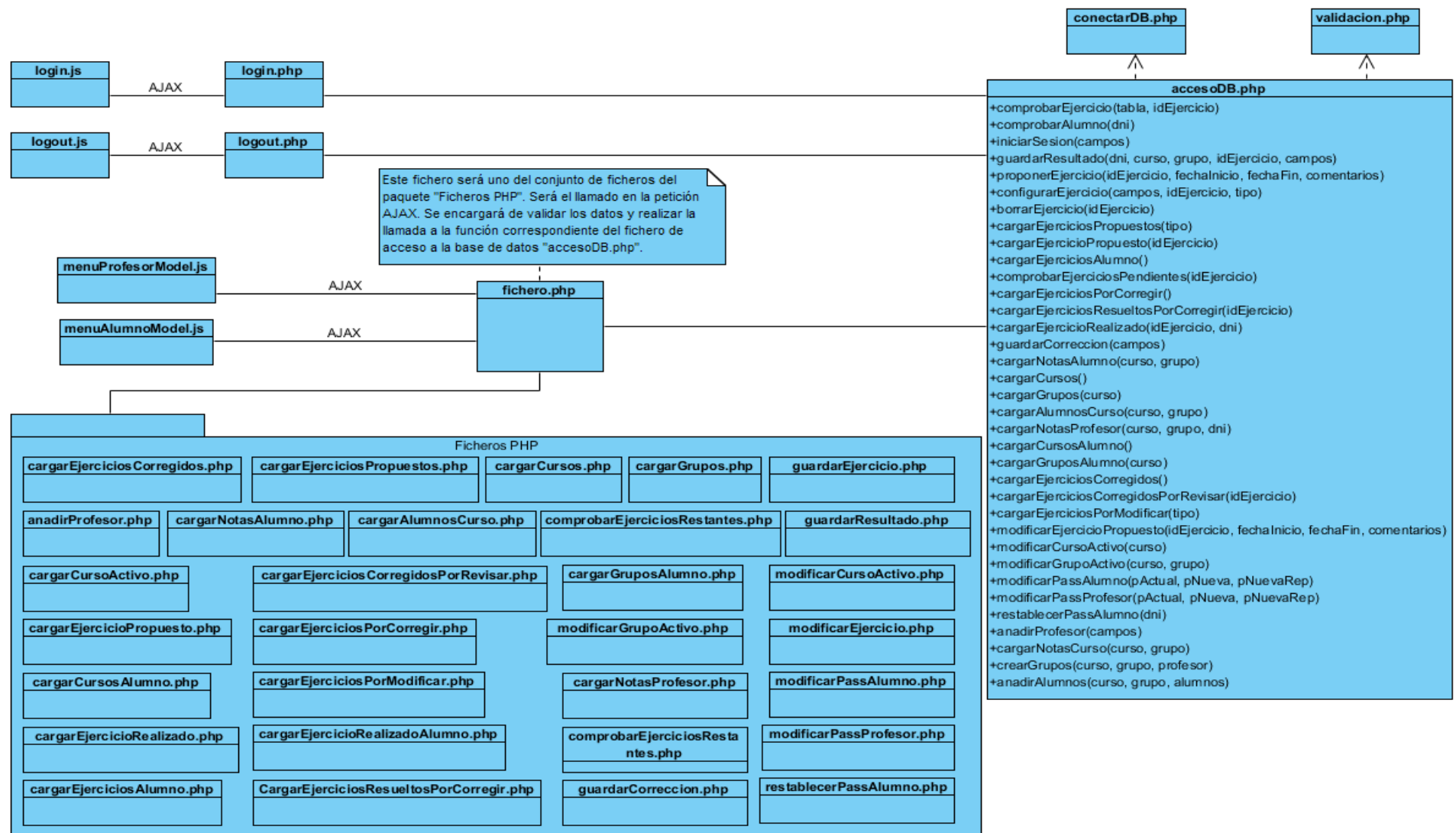


Figura 4.46. Diagrama general de la parte servidor del sistema Web.

4.2.5. DIAGRAMAS DE SECUENCIA

Los diagramas de secuencia [1] son diagramas de interacción que muestran el intercambio de mensajes entre instancias del modelo de clases organizados en una secuencia de tiempo.

Debido a que la secuencia de mensajes para los distintos applets permanece fija, es decir, el ciclo de un ejercicio (proponer, realizar y corregir) no varía según el applet, se realizarán únicamente los diagramas de secuencia del ciclo de ejercicio para el applet de "Transformaciones geométricas".

Se ha dividido el ciclo en partes. La primera parte se puede observar en la *Figura 4.47.* y corresponde a la parte del menú del profesor y la parte servidor necesaria a la hora de proponer un ejercicio.

En la *Figura 4.48.* se puede ver la interacción en el applet de transformaciones geométricas a la hora de proponer un ejercicio. Una vez realizado esto el alumno iniciará sesión en la aplicación y se dispondrá a realizar el ejercicio.

En la *Figura 4.49.* se puede ver la interacción del menú del alumno con la parte servidor cuando se disponga a realizar un ejercicio. El sistema cargará el applet del ejercicio y en la *Figura 4.50.* se puede ver la interacción dentro del applet mientras se realiza el ejercicio.

El profesor iniciará sesión en el sistema Web y se dispondrá a corregir el ejercicio. En la *Figura 4.51.* se muestra la interacción del menú del profesor con la parte servidor en el proceso de corregir el ejercicio.

Por último se muestran las interacciones necesarias para poder corregir el ejercicio correctamente en la *Figura 4.52* y la del diagrama de secuencia de iniciar sesión en la *Figura 4.53.*

Este proceso es exactamente igual con los demás applets del sistema Web, la única diferencia es el tipo de ejercicio y la cantidad de parámetros de cada applet, que es variable aunque no afecta a la interacción en dichos applets.

4.2.5.1. DIAGRAMA DE SECUENCIA DE PROPONER EJERCICIO (MENU DEL PROFESOR)

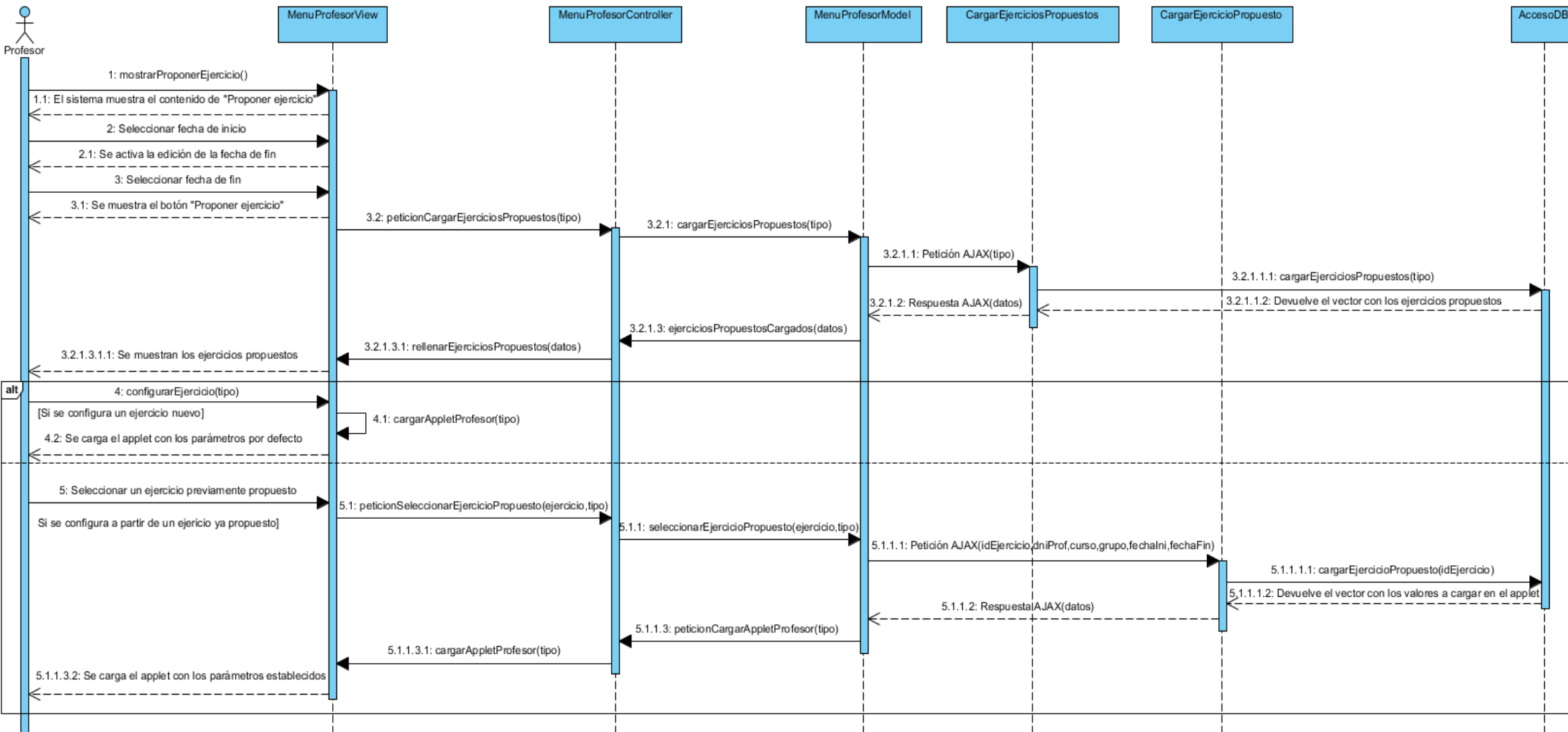


Figura 4.47. Diagrama de secuencia correspondiente a proponer un ejercicio desde el menú del profesor.

4.2.5.2. DIAGRAMA DE SECUENCIA DE PROPONER EJERCICIO (APPLET TRANSFORMACIONES GEOMÉTRICAS)

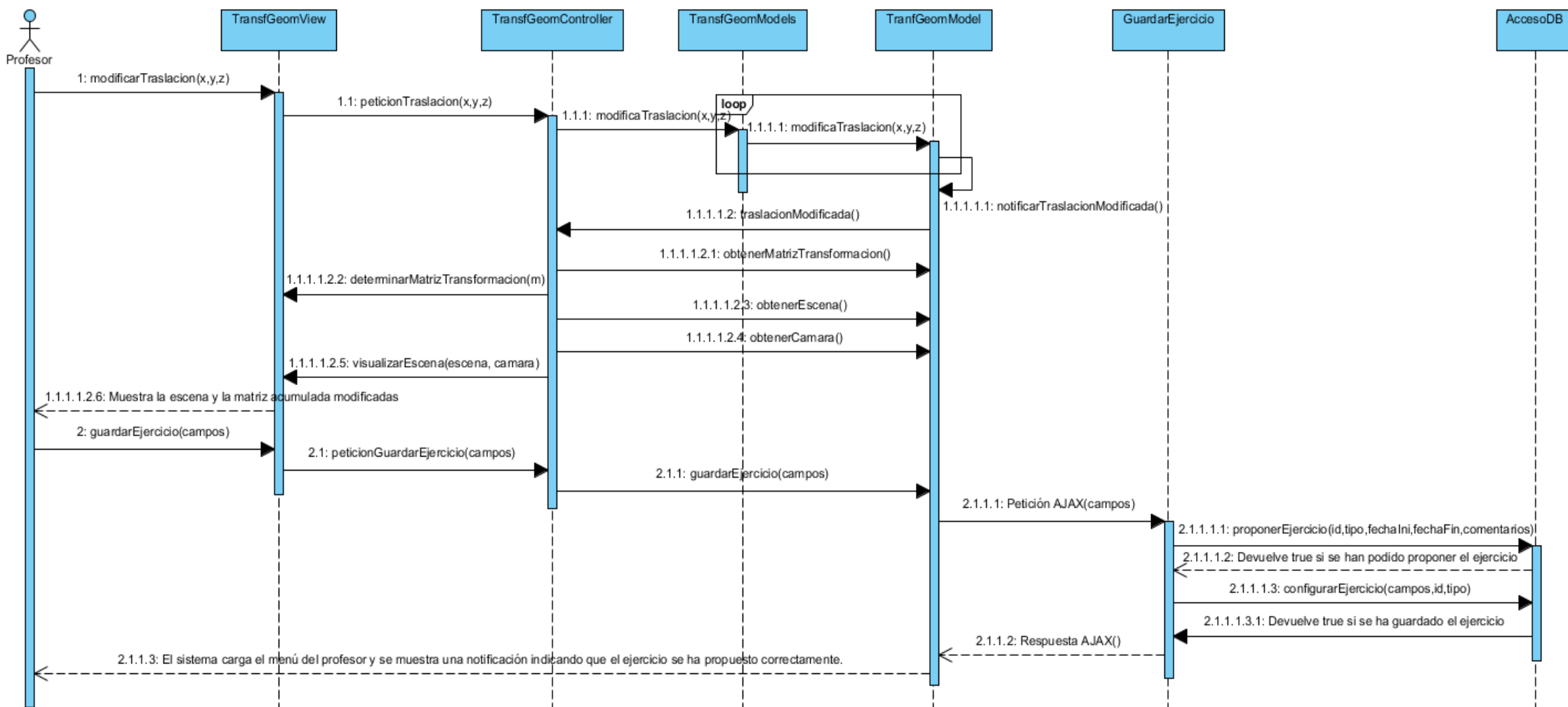


Figura 4.48. Diagrama de secuencia correspondiente a proponer un ejercicio parte del applet de transformaciones geométricas.

4.2.5.3. DIAGRAMA DE SECUENCIA DE REALIZAR EJERCICIO (MENU DEL ALUMNO)

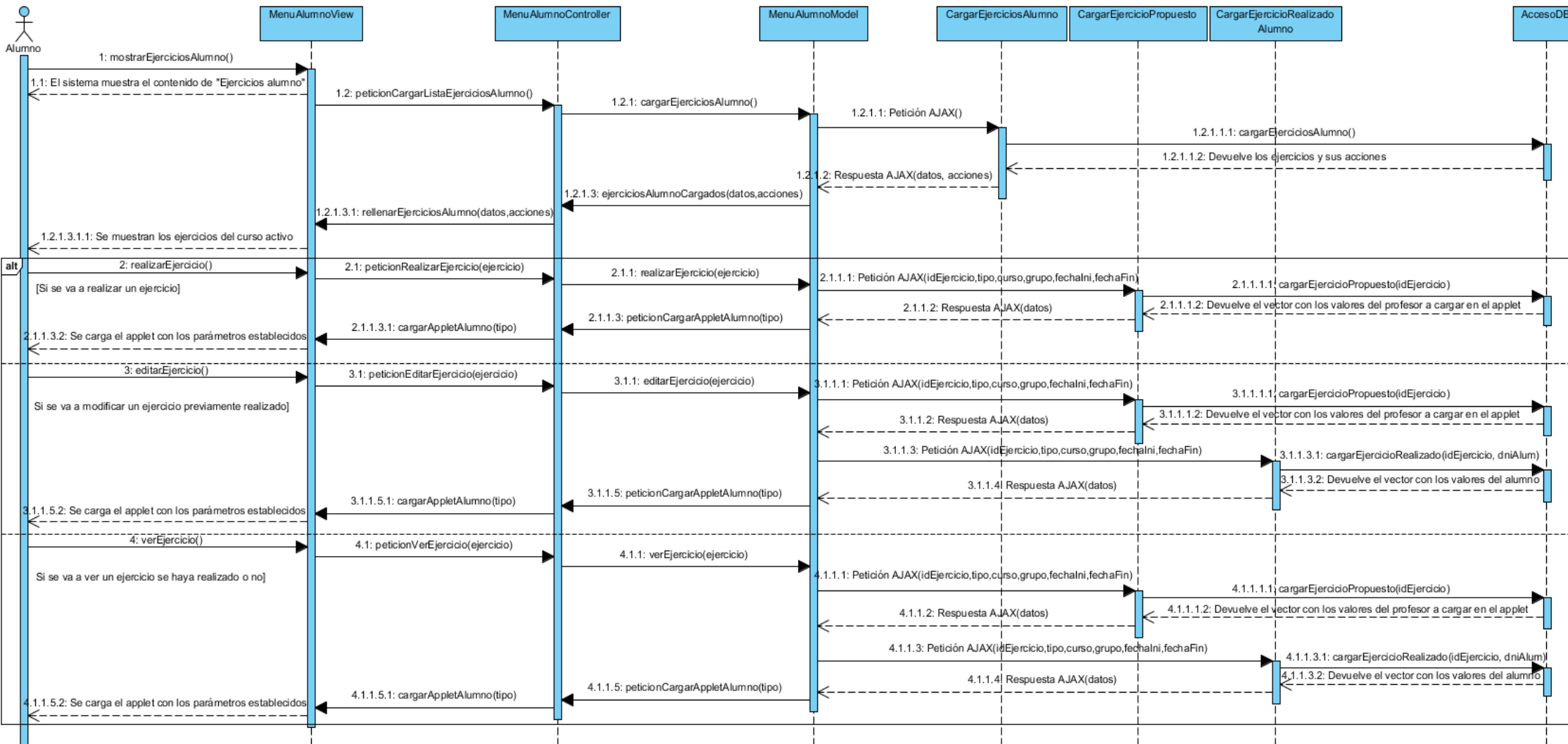


Figura 4.49. Diagrama de secuencia correspondiente a realizar un ejercicio desde el menú del alumno.

4.2.5.4. DIAGRAMA DE SECUENCIA DE REALIZAR EJERCICIO (APPLET TRANSFORMACIONES GEOMÉTRICAS)

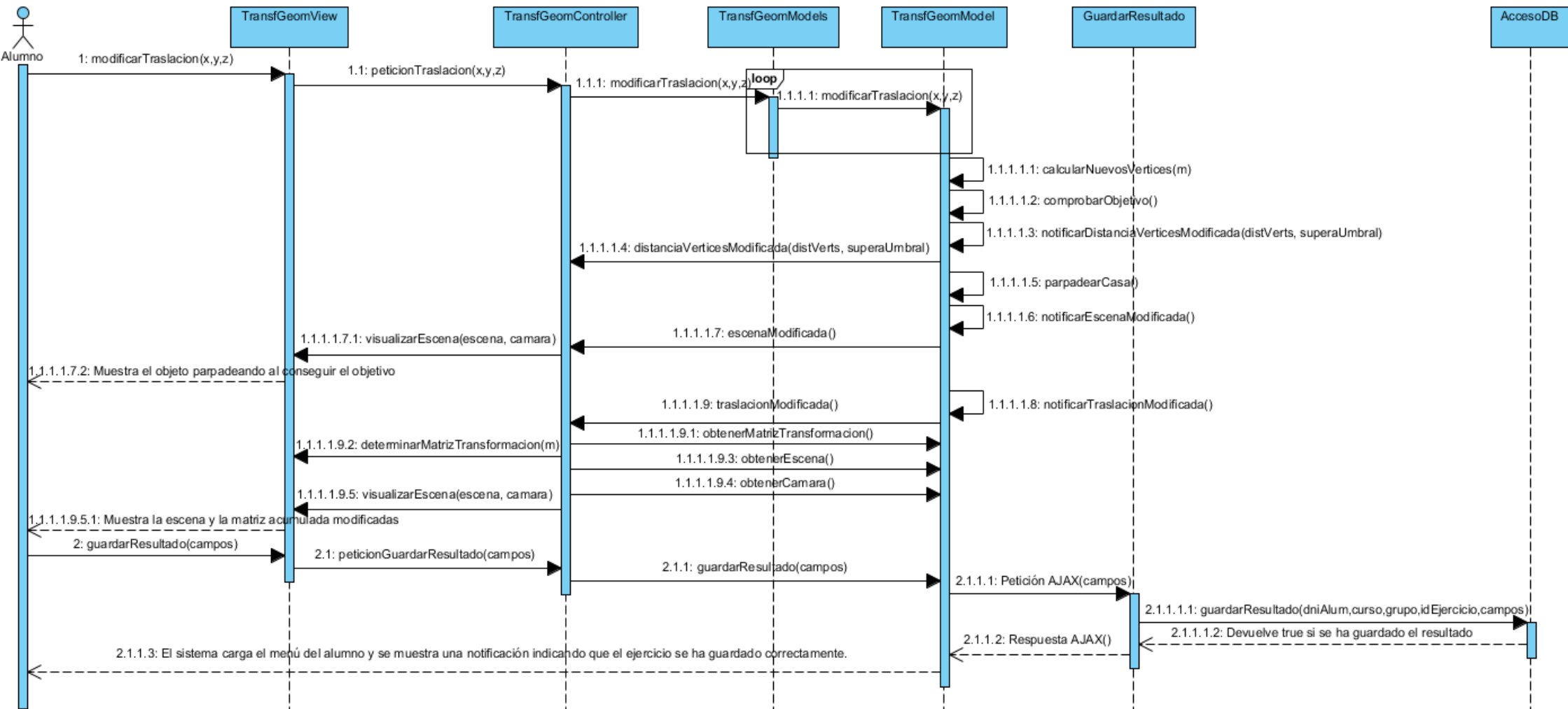


Figura 4.50. Diagrama de secuencia correspondiente a realizar un ejercicio parte del applet de transformaciones geométricas.

4.2.5.5. DIAGRAMA DE SECUENCIA DE CORREGIR EJERCICIO (MENU DEL PROFESOR)

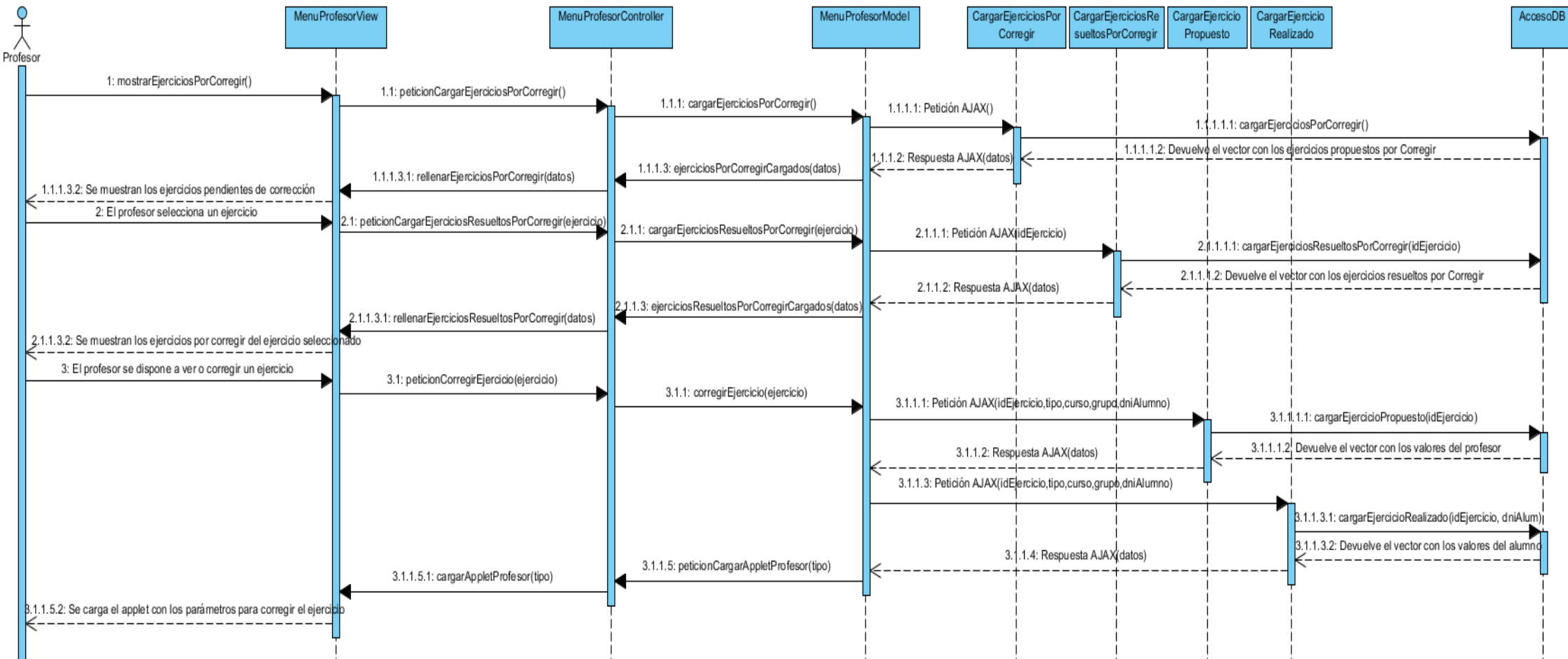


Figura 4.51. Diagrama de secuencia correspondiente a corregir un ejercicio desde el menú del profesor.

4.2.5.6. DIAGRAMA DE SECUENCIA DE CORREGIR EJERCICIO (APPLET TRANSFORMACIONES GEOMÉTRICAS)

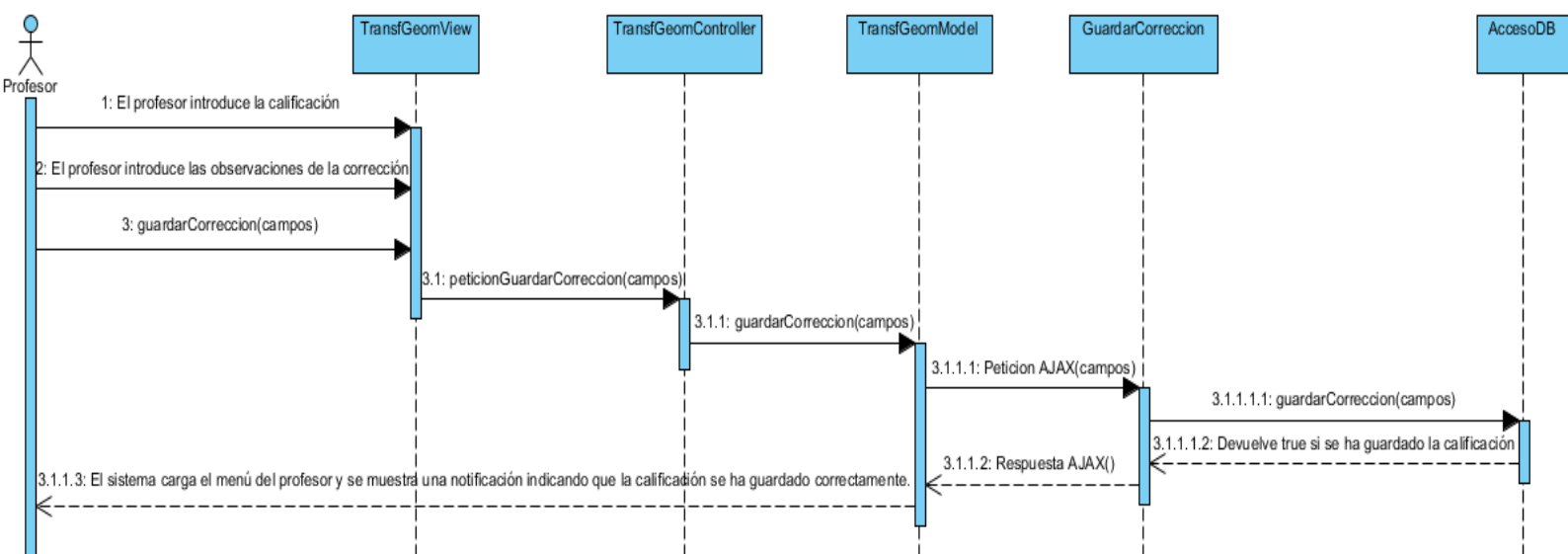


Figura 4.52. Diagrama de secuencia correspondiente a corregir un ejercicio parte del applet de transformaciones geométricas.

4.2.5.7. DIAGRAMA DE SECUENCIA DE INICIAR SESIÓN

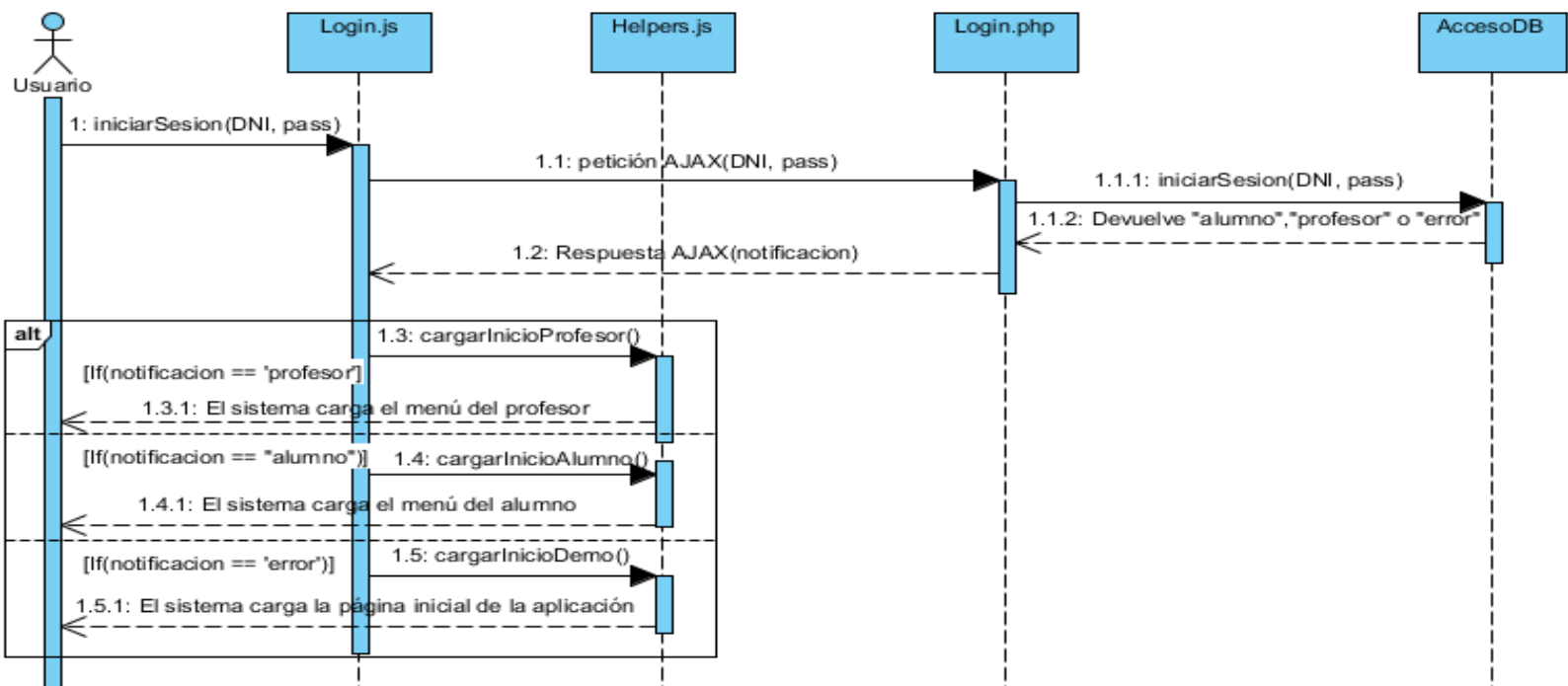


Figura 4.53. Diagrama de secuencia correspondiente a iniciar sesión.

5. IMPLEMENTACIÓN

En este capítulo se detallarán las tecnologías utilizadas durante el desarrollo del sistema Web. También se explicarán algunos detalles sobre la implementación de algunas funciones importantes de los applets, el control de la sesión de usuario y el método general para la extracción de información de la base de datos. Estos detalles forman la base del desarrollo de esta aplicación junto con la integración de los menús del alumno y profesor con los distintos applets de la aplicación que se pudo ver en la sección 4.2.5 dentro de los diagramas de secuencia.

5.1. TECNOLOGÍAS UTILIZADAS

Como se explicó en el segundo capítulo en la sección 2.3.2, para la implementación del sistema Web se han empleado las tecnologías con las que fue implementada la aplicación previamente desarrollada, base de este nuevo proyecto. Por tanto se han utilizado HTML5, CSS3, JavaScript, JQuery, WebGL y Three.js para la parte cliente de la aplicación, tanto para los menús de alumno y profesor como para los distintos applets del sistema Web.

Adicionalmente, para la parte servidor se han utilizado PHP y MySQL para trabajar con la base de datos de la aplicación. Por último, para comunicar ambas partes se ha utilizado la tecnología AJAX.

5.1.1. HTML5

HTML5^[7] ^[26] es la quinta versión importante de HTML. Esta versión provee nuevas etiquetas que añaden un significado semántico a las mismas como `<header>`, `<footer>`, `<nav>`, etc. Incorpora además etiquetas fundamentales para este proyecto como es el `<canvas>`, que permite la inserción de contenidos 2D y 3D utilizando WebGL en el sistema Web.

La estructura del el sistema Web previamente desarrollado fue creada con esta tecnología y se hizo un uso importante de la etiqueta `<canvas>` para la inserción de las escenas de los applets. Además, ha sido utilizada también, para la creación de la estructura de los menús de alumno y profesor.

5.1.2. JAVASCRIPT

JavaScript [9] [26] es un lenguaje de programación que se utiliza principalmente para crear páginas Web dinámicas, es decir, es un lenguaje idóneo para crear una página Web que incorpore efectos como animaciones, acciones que se activan al pulsar un botón, ventanas de aviso, aparición y desaparición de elementos visuales, etc.

Se trata de un lenguaje de programación interpretado, por lo que no es necesario compilarlo antes de ejecutarlo y es compatible con los navegadores más utilizados.

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java perteneciente a Oracle [16], aunque su sintaxis es muy parecida a Java o C++.

JavaScript posee la librería Three.js que ha sido usada para la implementación de los applets y es totalmente compatible por tanto es un lenguaje clave en el desarrollo de la parte cliente del sistema Web.

5.1.3. CSS3

CSS (Cascading Style Sheets) es un lenguaje usado para separar la estructura de la página Web de su presentación y estilo. Permite controlar el estilo y formato de múltiples elementos de la página Web desde el mismo lugar.

CSS3 es la última versión que se ha desarrollado para funcionar junto a HTML5. Será la tecnología encargada del diseño de la página y su presentación dejando a HTML5 únicamente la estructura de la misma. A diferencia de su predecesora CSS2, la nueva versión proporciona a los desarrolladores nuevos mecanismos para mantener un mayor control sobre el estilo con el que se muestran los elementos de las páginas Web.

En el sistema Web actual se ha utilizado principalmente para añadir detalles y mejorar la presentación de los menús de alumno y profesor. El estilo de los applets no ha variado esencialmente respecto al anterior sistema Web excepto la colocación

de algunas opciones nuevas o la inclusión de un segundo canvas que ha producido la recolocación del resto de contenedores.

5.1.4. JQUERY

JQuery^{[10] [26] [27]} es una serie de funciones y métodos en lenguaje JavaScript que permite simplificar la creación de un sistema Web moderno. Permite manipular el DOM (Document Object Model), manejar eventos, efectos y sobre todo y lo más importante proporciona una manera muy sencilla de realizar llamadas AJAX a la parte servidor de la aplicación.

Una ventaja importante de JQuery es que permite implementar una página Web independiente del navegador que se use para mostrarla. Sin embargo con Javascript habría que detectar qué navegador se estaría usando y adaptar el código para adecuarse a cada uno.

Jquery ha sido clave en el desarrollo del nuevo sistema Web, ya que ha proporcionado una forma sencilla, junto con AJAX, para conectar la parte cliente con la parte servidor. Se ha utilizado además para manipular el DOM y para añadir elementos mediante plugins que mejoran la calidad de la aplicación.

5.1.5. WEBGL Y THREE.JS

WebGL^[8] es una especificación estándar que se centra en mostrar gráficos en 2D y 3D en los distintos navegadores Web. Permite mostrar gráficos en 3D acelerados por la GPU en páginas Web, sin la necesidad de plugins en cualquier plataforma que soporte OpenGL.

Se utiliza el elemento `<canvas>` de HTML5 para la inserción de contenido gráfico en la página Web pero, debido a que se trata de una tecnología diseñada para trabajar directamente con la GPU, es difícil de codificar en comparación con otros lenguajes Web más accesibles. Por ello surgieron distintas librerías entre las cuáles la más popular y la que se ha escogido para este proyecto es Three.js.

Three.js^[28] es una librería liviana y eficiente escrita en JavaScript para crear y mostrar gráficos animados e interactivos en una página Web. Para ello aprovecha

las novedades que ofrece HTML5 para la generación de contenidos multimedia. Al igual que WebGL, esta librería también utiliza el elemento `<canvas>` para mostrar las escenas en la Web.

Uno de los principales obstáculos para el uso de Three.js en un sistema Web, es que se trata de una librería que está en constante actualización. Esto tiene aspectos positivos como que se trata de una librería que va creciendo añadiendo nuevas características y solucionando los posibles errores que pueda tener. Pero también tiene aspectos negativos, ya que las nuevas versiones no aseguran la retrocompatibilidad con las antiguas revisiones. Por tanto normalmente hay que modificar el código del sistema Web si se actualiza a la última revisión.

Debido a esto se ha mantenido la revisión 53 utilizada en el sistema Web previamente desarrollado aunque la última revisión de la librería es la 72.

Tanto WebGL como Three.js han sido utilizados para la creación de escenas en el sistema Web, ya que permiten la creación y manipulación de objetos, cámaras, luces, materiales o texturas en tiempo real.

5.1.6.AJAX

AJAX^[14] (Asynchronous JavaScript and XML) es una tecnología que se emplea en el desarrollo de páginas Web interactivas. Con el uso de esta tecnología, se pretende que las aplicaciones se ejecuten en cliente y, mientras, se mantenga una comunicación asíncrona con el servidor en segundo plano. Gracias a esto es posible realizar interacción con la parte servidor de la aplicación sin necesidad de recargar la página Web, mejorando la usabilidad, interactividad y velocidad de las mismas.

Haciendo uso de la función dedicada en la librería JQuery^[27] ^[29] a estas peticiones `"$.ajax()"` es muy sencillo conectar la parte cliente con la parte servidor. Hay que tener en cuenta que estas llamadas son asíncronas por tanto la respuesta AJAX llegará cuando se termine de trabajar con la parte servidor, aunque se pueden configurar para que sean síncronas modificando un parámetro en la llamada.

AJAX libera de parte de trabajo a la parte cliente al no tener que estar recargando la página continuamente, pero un uso excesivo de estas peticiones puede llegar a sobrecargar el servidor. Otro problema es que las páginas creadas mediante AJAX no guardan la información en el historial del navegador, por tanto no es posible volver a ellas desde el mismo.

En el desarrollo de este proyecto ha sido una tecnología muy importante, ya que, con la ayuda de JQuery, es muy sencillo realizar peticiones AJAX para la recuperación o almacén de datos en la base de datos de la aplicación, o simplemente, para ejecutar código necesario de la parte servidor. Esto ha ayudado a crear un sistema Web sin recargas de páginas en los menús de alumno y profesor lo que agiliza su rendimiento considerablemente.

5.1.7.PHP

PHP^[11] es un lenguaje de script orientado a objetos interpretado en la parte servidor utilizado para la creación de páginas Web dinámicas. Una de sus características más potentes es su soporte para un amplio abanico de bases de datos tanto NoSQL como SQL, entre la que destaca MySQL.

Es un lenguaje de fácil aprendizaje ya que posee un gran parecido con lenguajes como C o Perl. Además soporta orientación a objetos. Una ventaja muy importante de este lenguaje es la gran cantidad de funciones que posee así como la documentación existente para facilitar su aprendizaje.

En este proyecto se ha utilizado este lenguaje para desarrollar la parte servidor de la aplicación. Es muy sencillo el control de las sesiones de usuario con la variable "\$_SESSION" y combinado con AJAX es posible recoger los datos de la parte cliente con la variable "\$_POST". Ha sido elegido este lenguaje también por su sencillez a la hora de conectar e interactuar con una base de datos MySQL.

5.1.8.MYSQL

MySQL^[15] es un sistema de gestión bases de datos relacional, multihilo y multiusuario muy rápido en la lectura. Esto lo hace un lenguaje ideal para gestionar

la base de datos del sistema Web ya que la mayoría de las consultas utilizadas durante la ejecución del mismo son de lectura.

Se ha utilizado la extensión *mysqli*^[30], también conocida como *mysql* mejorada. Fue desarrollada para aprovechar las nuevas funcionalidades presentes a partir de la versión 4.1.3 de MySQL y viene incluida en PHP5 y versiones posteriores.

Esta extensión contiene numerosos beneficios respecto a la extensión *mysql*, entre los que se encuentran una interfaz orientada a objetos, soporte para procedimientos almacenados y un soporte para todas las funcionalidades presentes en MySQL a partir de la versión indicada anteriormente.

En este proyecto se ha utilizado una base de datos MySQL y por tanto se ha conectado a la misma desde PHP con una de sus APIs. Se ha utilizado *mysqli* para realizar las conexiones a la base de datos ya que aprovecha toda la funcionalidad de MySQL, mejora ligeramente el rendimiento respecto a *mysql* y es la recomendada por los desarrolladores para nuevos proyectos.

5.1.9. HERRAMIENTAS DE DESARROLLO

Para el desarrollo del sistema Web se han empleado dos tipos distintos de software. Durante las primeras fases del desarrollo se empleó el IDE NetBeans^[31] debido a que fue la herramienta de desarrollo elegida para el sistema Web del que parte este proyecto. Para facilitar el aprendizaje y la adaptación al código de la aplicación previamente desarrollada se escogió esta herramienta y, una vez se consiguió esto, el desarrollo de la aplicación continuó en Sublime Text 2.02 ^[32], un editor de código gratuito escrito en C++ y con una gran cantidad de plugins para ayudar al desarrollo de la aplicación.

Entre sus principales características en la versión 2.02 se encuentran:

- Navegación por el código mediante *Minimap*. Se trata de una pequeña vista previa del código que permite acceder rápidamente a la zona deseada.

- Soporta un gran número de lenguajes, entre los que se encuentran C/C++, C#, Java, PHP, CSS, HTML5, JavaScript, etc.
- Acceso rápido a cualquier archivo del proyecto mediante macros sencillas.
- Una gran cantidad de plugins de ayuda al desarrollador como por ejemplo: paletas de colores al seleccionar el código, autocompresor de código para crear las versiones .min del mismo o soporte para incluir un repositorio Git en el proyecto.
-

5.2. DETALLES DE IMPLEMENTACIÓN DEL SISTEMA WEB

En este apartado se describen algunos detalles de la implementación del sistema Web. Se incluyen algoritmos cuya base es similar para los distintos applets y algoritmos específicos de algunos applets. Estas funciones son importantes ya que se encargan de comprobar cuando un alumno ha realizado correctamente un ejercicio y de modificar el objeto insertado en la escena del applet. También se comentará el control de la sesión de la aplicación ya que se realiza en cualquier acceso a la parte servidor de la aplicación. Por último se explicará el método de extracción de información de la base de datos debido a que se trata de un algoritmo muy importante para la carga de las tablas y de los parámetros de los ejercicios de la aplicación.

5.2.1. ALGORITMO COMPROBAR OBJETIVO

Este algoritmo se ejecutará cuando se realice un cambio en algún elemento de la escena desde los controles del applet. En este caso será importante cuando el alumno se disponga a realizar un ejercicio.

Para el caso del applet de *Texturasse* comprueba que la diferencia entre las coordenadas de textura que configuró el profesor y las que va consiguiendo el alumno es menor que el umbral establecido para dicho applet. Además de esto se

debe cumplir que los tipos de repetición y filtros de textura deben coincidir con los configurados por el profesor.

```
ALGORITMO: comprobarObjetivoTexturas

INICIO
  SI logueado == 'ALUMNO'
    SI ABS(coord0uAlumno - coord0uProfesor) <= umbralTexturas &&
      ABS(coord0vAlumno - coord0vProfesor) <= umbralTexturas &&
      ABS(coord1uAlumno - coord1uProfesor) <= umbralTexturas &&
      ABS(coord1vAlumno - coord1vProfesor) <= umbralTexturas &&
      ABS(coord2uAlumno - coord2uProfesor) <= umbralTexturas &&
      ABS(coord2vAlumno - coord2vProfesor) <= umbralTexturas &&
      ABS(coord3uAlumno - coord3uProfesor) <= umbralTexturas &&
      ABS(coord3vAlumno - coord3vProfesor) <= umbralTexturas &&
      repeticionSAlumno == repeticionSProfesor &&
      repeticionTAlumno == repeticionTProfesor &&
      MAGnificationAlumno == MAGnificationProfesor &&
      MINificationAlumno == MINificationProfesor
      parpadearTextura()
    FIN_SI
  FIN_SI
FIN
```

Figura 5.1. Pseudocódigo algoritmo “ComprobarObjetivoTexturas”.

Para el caso del applet de *Transformaciones geométricas* se comprueba que la distancia entre cada par de vértices de ambas casas no superan el umbral establecido para este tipo de applet. Además se realiza un cálculo de la distancia media de los vértices que posteriormente será mostrada al usuario.

```
ALGORITMO: comprobarObjetivoTransformacionesGeometricas

INICIO
  SI casaGenerada && casaObjetivoGenerada
    distanciaX => 0
    distanciaY => 0
    distanciaZ => 0;
    distanciaVertices => 0;
    superaUmbral => false;

    PARA i => 0 HASTA verticesCasaTam PASO i => i + 1

      distanciaX => ABS(verticesCasa[i].x - verticesCasaObjetivo[i].x)
      distanciaVertices => distanciaVertices + distanciaX

      distanciaY => ABS(verticesCasa[i].y - verticesCasaObjetivo[i].y)
      distanciaVertices => distanciaVertices + distanciaY

      distanciaZ => ABS(verticesCasa[i].z - verticesCasaObjetivo[i].z)
      distanciaVertices => distanciaVertices + distanciaZ

      SI distanciaX >= umbralTG || distanciaY >= umbralTG || distanciaZ >= umbralTG
        superaUmbral => true
      FIN_SI

    FIN_PARA

    distanciaVertices => (distanciaVertices / verticesCasaTam)
    notificarCambio(distanciaVertices, superaUmbral);

    if(superaUmbral){
      DEVOLVER false;
    }

    DEVOLVER true;
  }

  DEVOLVER false;

FIN
```

Figura 5.2. Pseudocódigo algoritmo “ComprobarObjetivoTransformaciones Geometricas”.

En ambos casos, si se cumple el objetivo se llamará al algoritmo que hará parpadear el objeto de la escena de cada applet. Dicho algoritmo se explica a continuación.

5.2.2.ALGORITMO PARPADEAR OBJETO

Este algoritmo se encargar de modificar algún atributo del objeto al que se le están aplicando los cambios desde los controles de applet. Se aplica en los applets de *Transformaciones geométricas*, *Sombreado* y *Texturas* en el parte del modelo.

Será ejecutado únicamente cuando el alumno haya conseguido el objetivo determinado por el profesor en el ejercicio configurado.

Este algoritmo crea un intervalo de tiempo en el que se va ejecutando la acción establecida en segundo plano, es decir, se pueden seguir realizando cambios en el applet por parte del usuario y la aplicación sigue su ejecución normalmente mientras se muestran los cambios del objeto en pantalla.

A continuación se muestra el algoritmo general en pseudocódigo y algunas anotaciones sobre la sección de modificar los parámetros del objeto variable entre los distintos applets comentados anteriormente.

```
ALGORITMO: parpadearObjeto
INICIO
    CONTADOR => 0
    ESTADO_OBJETIVO => false

    inicioIntervalo(tiempoIntervalo)

    SI ESTADO_OBJETIVO == false
        modificarParametrosObjeto(estado1)
        ESTADO_OBJETIVO => true
    FIN_SI
    SINO
        modificarParametrosObjeto(estado2)
        ESTADO_OBJETIVO => false
    FIN_SINO

    notificarCambio(escena)

    CONTADOR = CONTADOR + 1;

    SI CONTADOR >= vecesIntervalo
        FinalizarIntervalo()
    FIN_SI

    finIntervalo
FIN
```

Figura 5.3. Pseudocódigo algoritmo “ParpadearObjeto”.

La sección de la función “*modificarParametrosObjeto(estado)*” varía entre los distintos applets como se comenta a continuación.

Para el caso del applet de *Transformaciones geométricas* se ha modificado cada pequeño periodo de tiempo la transparencia del objeto para simular que ambos objetos en la escena coinciden o se encuentran muy próximos a coincidir.

Para el caso del applet de *Sombreado* se ha modificado el color de la luz cada pequeña fracción de tiempo cuando coincidan el tipo de sombreado e iluminación y el número de divisiones entre el alumno y el profesor.

Para el caso del applet de *Texturas* se ha modificado el color del plano en el que se proyecta la textura cuando coincidan los tipos de filtros y repetición y las coordenadas de textura se encuentren por debajo del umbral establecido.

En definitiva lo que realiza el algoritmo es cambiar entre un estado u otro del objeto un número *vecesIntervalo* durante un determinado *tiempoIntervalo*.

5.2.3. MÉTODO PARA EL CONTROL DE LA SESIÓN

Para controlar la sesión se ha utilizado en la parte servidor la variable de sesión “S_SESSION” de PHP. En ella se almacenan el DNI, curso, grupo, tipo de usuario y la hora de la última actividad realizada.

Gracias a esta variable de sesión se controlan los permisos de cada usuario para acceder a determinadas funciones de la parte servidor. Antes de realizar cualquier función se comprueba que exista una sesión iniciada y que el tipo de usuario tiene acceso a dicha función.

También se comprueba que el tiempo de sesión no ha expirado. Esto se realiza actualizando cada vez que se accede a la parte servidor de la aplicación el tiempo de la última actividad del usuario. Si este tiempo excede al realizar la comprobación del tiempo máximo de inactividad establecido, se cerrará la sesión y se cargará la página de inicio del sistema Web.

Por último también se utiliza para almacenar el curso y grupo activos por el usuario. Por tanto se podrán modificar cuando el usuario seleccione el curso y grupo con el que pretenda realizar acciones en su menú de inicio.

5.2.4. MÉTODO PARA EXTRAER INFORMACIÓN DE LA BASE DE DATOS

Como se explicó anteriormente en la *sección 4.2.* todas las acciones que requieran consultas a la base de datos se realizan a través de funciones desde el fichero “accesoDB.php”. El resultado de dichas funciones lo recogerá el fichero PHP que fue objetivo de la petición AJAX realizada para la carga de datos o la insercción de los mismos.

Cuando se requiera información de la base de datos por ejemplo para rellenar una tabla con los ejercicios del alumno o para cargar los parámetros de un ejercicio completo, se devolverá en forma de vector con cada uno de los campos devueltos por la consulta SELECT realizada a la base de datos.

```
funcion(parametro1, parametro2)
INICIO
    validar(parametro1)
    validar(parametro2)

    sql => "SELECT * FROM tabla1"
    sql => ejecutarConsulta()

    resultado => []
    i => 0

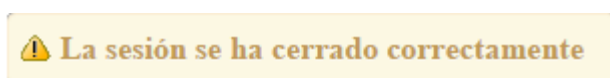
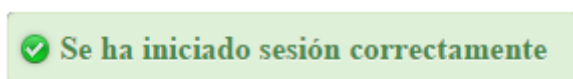
    MIENTRAS row => extraerFilaConsulta(sql)
        resultado[i++] => row['campo1']
        resultado[i++] => row['campo2']
        resultado[i++] => row['campo3']
    FIN_MIENTRAS

    DEVUELVE resultado
FIN
```

Figura 5.4. Estructura de la extracción de información de la base de datos.

El vector resultado que devuelve la función de la *Figura 5.4.* será tratado por el fichero PHP que fue llamado mediante AJAX. Éste codificará el resultado en JSON [33] y devolverá los datos y una notificación adjunta. Esta notificación es necesaria para informar de los errores que se hayan podido producir o del éxito en la operación.

Para mostrar las notificaciones se ha utilizado un plugin JQuery llamado Notify [34]. En el fichero “notificaciones.js” se ha almacenado un vector con los códigos de éxito, error o aviso que devolverán las consultas a la base de datos. En caso de que se trate de un error producido por un comportamiento anormal de la aplicación se devuelve un código con un error de validación y se muestra la notificación “Ha ocurrido un error”. En las siguientes figuras se muestra el aspecto de los tipos de notificaciones que se mostrarán en el sistema Web en la parte superior derecha.



6. PRUEBAS

En este capítulo se muestran las distintas pruebas que se han realizado a la aplicación durante su desarrollo. En concreto se han realizado pruebas de validación con el tutor del proyecto y profesor de la asignatura, pruebas de unidad utilizando el framework Jasmine [35] para JavaScript y pruebas de integración. Además se han realizado otro tipo de pruebas menores durante la implementación del sistema Web.

Las pruebas son útiles para comprobar el correcto funcionamiento del sistema Web y se trata de una parte fundamental durante su desarrollo. Ayudan a solucionar un gran número de errores que pueda tener la aplicación, sin embargo es complicado que se puedan eliminar todos los errores. Estos errores que no se solucionen durante el desarrollo del proyecto se podrán solucionar con el uso del mismo y la retroalimentación de los usuarios.

Las pruebas de validación se han ido realizando durante el desarrollo del proyecto mediante pruebas de la aplicación realizadas por el tutor del proyecto y analizando si lo existente en la aplicación se ajustaba a los requisitos iniciales del proyecto. Estos errores y modificaciones en la aplicación que se han detectado se han ido solucionando durante el desarrollo de la misma.

Las pruebas de unidad se han realizado durante el desarrollo del proyecto. Primero se realizaron las pruebas desarrolladas para el sistema Web del que parte este proyecto. También se añadieron las pruebas necesarias para la nueva funcionalidad de los applets una vez se integraron con los menús.

Por último, las pruebas de integración se han realizado durante la fase final del desarrollo de la aplicación para comprobar que las distintas partes de la misma funcionan correctamente como un conjunto.

6.1. PRUEBAS DE VALIDACIÓN

A continuación se detallan las pruebas de validación para el menú del profesor realizadas así como las decisiones tomadas para solucionar los errores cometidos.

6.1.1. MENU DEL PROFESOR

En la primera versión funcional del menú de inicio del profesor existían algunas acciones que no se ajustaban a los requerimientos de la aplicación o que se añadieron al realizar las pruebas.

6.1.1.1. EJERCICIOS PENDIENTE DE CORRECCIÓN

Inicialmente los ejercicios pendientes de corrección por el profesor se mostraban listados en la tabla una vez se había cumplido la fecha de fin de ejercicio, es decir, uno por uno los ejercicios resueltos por los alumnos. Al consultarlo con el tutor del proyecto se aclaró que debían aparecer listados los ejercicios propuestos cuya fecha de fin de ejercicio haya superado la fecha actual. Una vez se seleccione el ejercicio propuesto que se disponga a corregir, se listarán los ejercicios realizados por los alumnos de ese ejercicio propuesto.

Se añadió también la opción de ver un ejercicio sin necesidad de corregirlo, ya que constaba en los requisitos del proyecto.

En otra revisión de la aplicación se aclaró una solución para mejorar la usabilidad del sistema. Previamente, una vez el profesor terminaba de corregir un ejercicio, se cargaba el menú principal del profesor teniendo que seleccionar de nuevo el ejercicio para corregir dicho ejercicio realizado por otro alumno. El tutor recomendó que directamente el sistema accediera al ejercicio que se estaba corrigiendo mientras quedaran ejercicios pendientes de corrección del ejercicio seleccionado.

6.1.1.2. REVISAR EJERCICIOS CORREGIDOS

Inicialmente los ejercicios corregidos se mantenían en la lista de ejercicios, pero en la primera revisión de la aplicación se consideró separar los ejercicios corregidos de los ejercicios pendientes de corrección. El motivo por lo que se realizó este cambio se debía a que la revisión de notas por parte del profesor se realizaría normalmente cuando un alumno pida al profesor la revisión de alguna calificación con la que esté en desacuerdo.

Otro cambio que se ha producido en esta acción es similar al producido en la acción anterior de la sección 6.1.1.1. En concreto se mostraban los ejercicios corregidos individualmente y al realizar la modificación se muestran los ejercicios propuestos con alguna calificación asignada. Una vez se accede al ejercicio propuesto se muestran las calificaciones de los distintos ejercicios realizado para dicho ejercicio.

6.1.1.3. CONSULTAR CALIFICACIONES

Inicialmente el profesor, haciendo uso de esta acción, podía consultar las calificaciones individualmente seleccionando el curso, grupo y alumno. Tras una de las revisiones realizadas por el tutor del proyecto, se añadió la opción de exportar en formato “CSV” todas las calificaciones de los ejercicios de un curso y grupo ordenadas por alumnos, con el fin de facilitar su inclusión en las calificaciones de la asignatura.

6.1.1.4. SELECCIONAR CURSO ACTIVO

Una vez se almacenan ejercicios de distintos cursos en la aplicación, las tablas de ejercicios por corregir y por revisar tienen que almacenar demasiados ejercicios y, aunque estuvieran ordenados por fecha, la búsqueda de algún ejercicio de algún curso anterior se trataba de un trabajo tedioso de realizar.

Por tanto se añadió un selector de curso activo para el profesor. El curso seleccionado será del que aparecerán los ejercicios para corregir y revisar limpiando así dichas tablas de ejercicios pasados.

6.1.2.MENU DEL ALUMNO

A continuación se detallan las pruebas de validación para el menú del alumno realizadas así como las decisiones tomadas para solucionar los errores cometidos.

6.1.2.1. LISTADO DE EJERCICIOS

Realizando una revisión de la aplicación se detectó que el alumno, una vez había realizado un ejercicio no tenía la opción de modificarlo. Para solucionarlo se añadió esta opción.

6.1.2.2. SELECCIONAR CURSO ACTIVO

Por el mismo motivo indicado en la sección 6.1.1.4. se ha añadido esta opción para no acumular demasiados ejercicios en la lista de ejercicios del alumno si se trata de un alumno repetidor.

6.1.3.OTRAS CORRECCIONES

En una de las revisiones se detectó que al proponer un ejercicio ya configurado del tipo *Sombreado* se producía un error en algunos casos en los que la esfera parpadeaba al modificar algún parámetro. Esto no debería ocurrir ya que ese estado solo se puede producir cuando un alumno está realizando, editando o viendo un ejercicio. Simplemente se añadió la condición faltante para que no se produjera este error.

Otro error corregido se trataba de un fallo al almacenar el ángulo de apertura en el applet *Tipos de luces*. Inicialmente se almacenaba la variable con el valor del ángulo de apertura erróneo y se producía este fallo en la escena. La solución fue extraer directamente el ángulo de apertura de la luz de la escena.

En otras de las revisiones el tutor consideró oportuno mostrar al alumno en el applet *Transformaciones geométricas*, mientras realizaba un ejercicio, la media de la distancia entre los vértices del objeto modificado por el alumno y el objetivo a conseguir.

En el applet *Técnicas de selección* se detectó un error al modificar un ejercicio o proponer un ejercicio previamente configurado. Dicho error consistía en que no aparecían coloreados los objetos seleccionados en el canvas para seleccionar los objetos, mientras que en el canvas 3D de la parte izquierda de la página si lo hacían. Se solucionó el error modificando una condición cuando se cargaban los parámetros del ejercicio.

También se producía un error general en toda la aplicación en relación a las tildes y las 'ñ' en los campos de observaciones de la corrección y comentarios del ejercicio. Se solucionó cambiando la forma de almacenar el valor utilizando JQuery.

Por último, realizando las pruebas en Openshift [4], se detectó un problema en la diferencia horaria entre la hora del servidor y la actual. Se solucionó añadiendo la diferencia horaria al realizar las comprobaciones de hora en la parte servidor.

6.2. PRUEBAS DE UNIDAD

Las pruebas de unidad se han realizado para la parte cliente en los ficheros correspondiente a los modelos de los applets. Para la parte de los menús no ha sido necesaria la realización de pruebas de unidad debido a que las funciones existentes en dichos ficheros son normalmente de paso de información para almacenarla en la base de datos o para extraerla de la misma. Sin embargo se ha ido depurando la implementación de las mismas durante el desarrollo de la aplicación, siendo estas funciones muy similares entre ellas.

Para la parte servidor tampoco se han realizado las pruebas de unidad debido a que existe mucha comunicación con la base de datos y el objetivo de estas funciones es consultar o insertar datos en la misma. Sin embargo se han implementado todas las funciones de validación necesarias para evitar la posible inserción en la base de datos de cualquier información maliciosa desde la parte cliente, es decir, toda la información proveniente de la parte cliente tiene que pasar los distintos tests de validación para acceder a la base de datos. Si no se pasan estos test se devuelve el error correspondiente y el sistema mostrará una notificación acorde a dicho error.

6.2.1. APLET DE TRANSFORMACIONES GEOMÉTRICAS

En este applet se han realizado las pruebas de unidad pero no ha sido posible realizarlas correctamente debido a que el objeto de la escena se inserta con la función `JSONLoader` de `Three.js`. Se trata de una función asíncrona y a la hora de realizar las pruebas de unidad no se ha ejecutado completamente.

Igualmente las funciones que había implementadas no han sufrido modificaciones y están correctamente probadas y las funciones añadidas no tienen

parámetros o son similares a las funciones existentes y las soluciones son las mismas que para aquellas.

6.2.2. APLET DE TÉCNICAS DE SELECCIÓN

Se han realizado las mismas pruebas en este applet, ya que no se han añadido cambios sustanciales en el modelo. Solo se ha añadido una función para colocar los objetos del profesor que es similar a la función para mover los objetos de la escena. Por tanto todas las pruebas han resultado satisfactorias.

6.2.3. APLET DE TIPOS DE LUCES

Para este applet se han realizado las pruebas de unidad a la función “crearLuzGuardada(params)”. Esta función se llama para configurar la luz que guardó el profesor o el alumno en el ejercicio configurado o realizado respectivamente. Como parámetro se pasarán los parámetros del profesor o del alumno según el caso.

Al realizar las pruebas unitarias se han encontrado algunos fallos que se han solucionado y se han aplicado al resto de applets cuando se realiza la carga de los datos.

- Al realizar la carga de la luz con un tipo de luz distinto de 0, 1 ó 2 se producía un error. Se ha solucionado creando la luz Puntual por defecto del applet con posición (-2,2,2).
- Al insertar un ángulo con valor null, se asignaba un valor 0. Se ha solucionado asignando al ángulo el valor por defecto si se produce este caso. También se producía este error cuando se inserta una cadena de texto, carácter o un valor booleano y se ha solucionado añadiendo una condición para cada caso.

Este segundo punto se repetirá en el resto de applets cuando se realicen operaciones con los parámetros cargados de la base de datos.

6.2.4. APPLET DE LUCES Y MATERIALES

Para este applet se han realizado las pruebas unitarias a la función “cargarDatos(params)”. Esta función se llama para configurar el objeto de la escena con los parámetros del alumno o del profesor. Se van realizando las llamadas a las funciones para modificar los distintos parámetros del objeto y escena.

Cualquier parámetro que no sea válido lo rechazan estas funciones, por tanto para solucionar los posibles errores que puedan existir al cargar un valor null, NaN o boolean, se recorre el vector con los parámetros antes de realizar las modificaciones del objeto. Si se produce un error se cargan los parámetros por defecto del objeto.

6.2.5. APPLET DE MODELOS DE ILUMINACIÓN

Para este applet se han realizado las pruebas unitarias a la función “cargarDatos()”. Esta función se llama para establecer la escena que configuró el profesor. Para controlar que los datos para configurar la escena son válidos se aplican los mismos cambios que en los anteriores applets, es decir si alguno de los valores es null, NaN o boolean se rechazan los parámetros y se carga la luz por defecto.

6.2.6. APPLET DE SOMBREADO

En este applet se han adaptado las pruebas unitarias para realizarlas con el mismo número de divisiones horizontales y verticales. Además se ha solucionado el mismo problema indicado en las anteriores secciones en la carga de los parámetros del ejercicio desde la base de datos.

6.2.7. APPLET DE TEXTURAS

Para este applet se ha modificado la función “cargarDatos(params)”. Simplemente se han añadido las condiciones que se utilizaron en las anteriores secciones para comprobar que no se carguen valores inválidos en la configuración del applet.

6.3. PRUEBAS DE INTEGRACIÓN

A continuación se detallan las distintas pruebas de integración realizadas para las acciones principales de los menús de alumno y profesor.

6.3.1. PROPONER UN EJERCICIO DEL TIPO LUCES Y MATERIALES

- **Paso 1:** Seleccionar la opción del menú “Proponer ejercicio”.

Resultado esperado: Se muestra el contenido de la acción “Proponer ejercicio”.

Resultado: Se obtiene el resultado esperado.

- **Paso 2:** Establecer la fecha de inicio y fin de ejercicio.

Resultado esperado: Se cargan los ejercicios previamente propuestos para el tipo de ejercicio seleccionado. Por defecto *Transformaciones geométricas*.

Resultado: Se obtiene el resultado esperado en la *Figura 6.1*.

Nº exercise	Teacher	Course	Group	Start date	End date	Comments	Actions
1	12345678A	2015/2016	A	2015-09-22 10:14:00	2015-10-07 11:14:00	TG1	
2	12345678A	2015/2016	A	2015-09-22 17:33:00	2015-10-07 17:33:00	TG2	

Figura 6.1. Ejercicios propuestos del tipo Transformaciones geométricas.

- **Paso 3:** Seleccionar el tipo de applet *Luces y materiales*.

Resultado esperado: Se cargan los ejercicios previamente propuestos del tipo seleccionado.

Resultado: Se obtiene el resultado esperado.

- **Paso 4: Seleccionar el ejercicio propuesto a cargar.**

Resultado esperado: Se carga el applet del tipo seleccionado con los parámetros que fueron guardados para el mismo.

Resultado: Se obtiene el resultado esperado en la *Figura 6.2*.

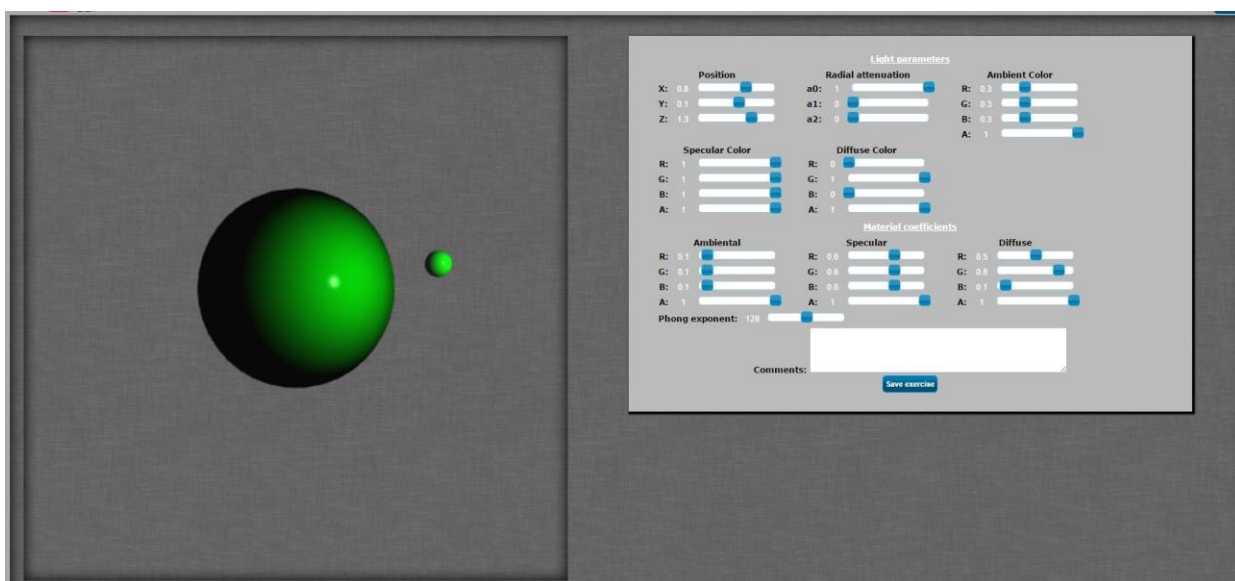


Figura 6.2. Configurar un ejercicio previamente propuesto de Luces y materiales.

- **Paso 5: Modificar los parámetros del ejercicio.**

Resultado esperado: Los parámetros de la luz y los coeficientes del material se modifican correctamente y los cambios son aplicados a la esfera insertada en la escena.

Resultado: Se obtiene el resultado esperado.

- **Paso 6: Guardar el ejercicio configurado.**

Resultado esperado: Se guarda el ejercicio correctamente, se muestra dicha notificación y se carga el menú de inicio del profesor.

Resultado: Se obtiene el resultado esperado en la *Figura 6.3*.

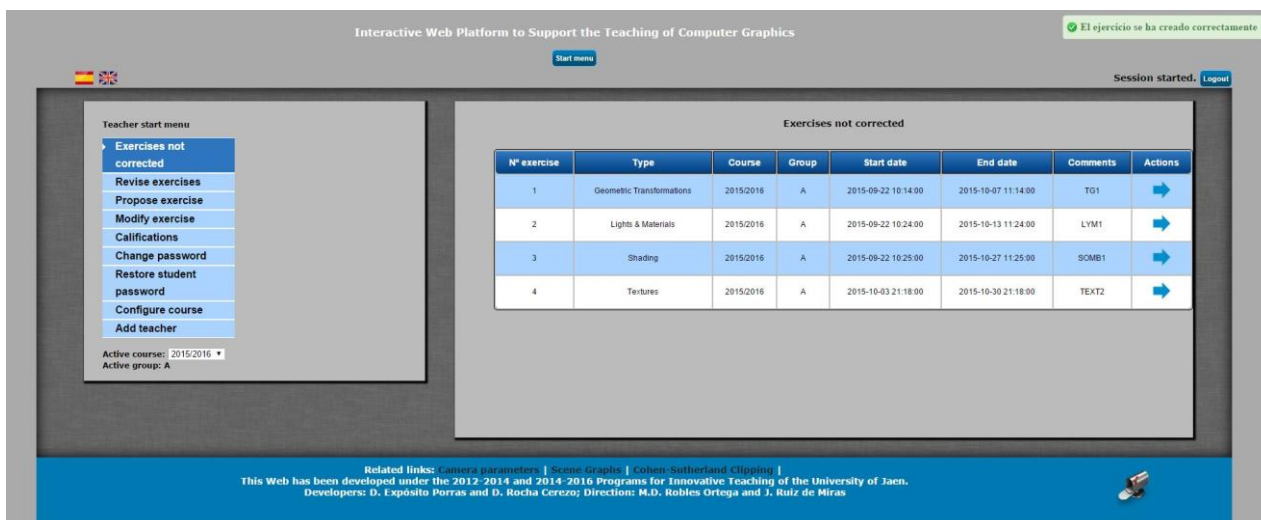


Figura 6.3. Ejercicio configurado con su notificación correspondiente.

6.3.2. MODIFICAR UN EJERCICIO DE TRANSFORMACIONES GEOMÉTRICAS

- Paso 1: Seleccionar la opción del menú “Modificar ejercicio”.

Resultado esperado: Se muestra el contenido de la acción “Modificar ejercicio”.

Resultado: Se obtiene el resultado esperado en la Figura 6.4.



Figura 6.4. Modificar ejercicio de Transformaciones geométricas.

- Paso 2: Seleccionar el ejercicio a modificar.

Resultado esperado: Se carga la pantalla para modificar las fechas de inicio y fin del ejercicio.

Resultado: Se obtiene el resultado esperado en la *Figura 6.5*.

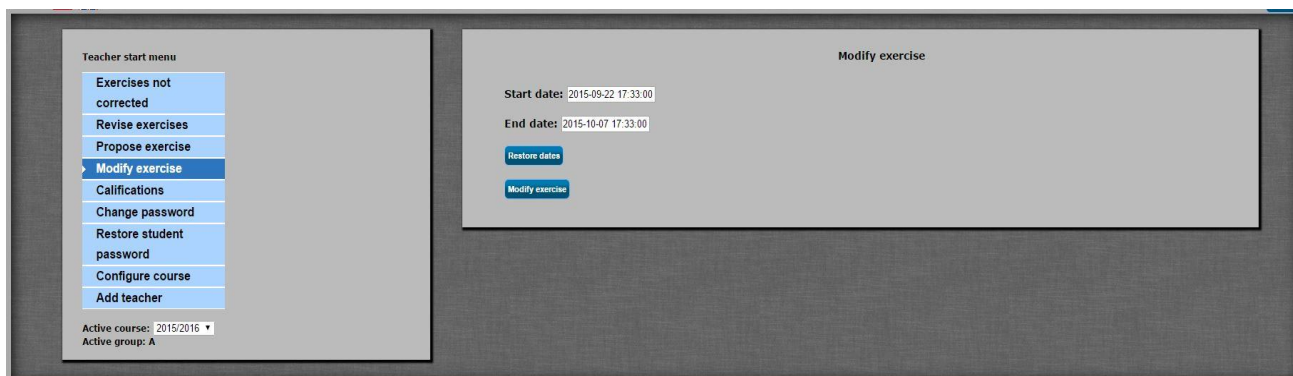


Figura 6.5. Modificar fechas del ejercicio.

- **Paso 3: Modificar la fecha de inicio o de fin del ejercicio.**

Resultado esperado: Se establecen las nuevas fechas para el ejercicio.

Resultado: Se obtiene el resultado esperado.

- **Paso 4: Pulsar el botón “Modificar ejercicio”.**

Resultado esperado: Se carga el applet del ejercicio con los parámetros que estableció el profesor para el mismo. Por tanto deberá aparecer únicamente la casa sólida en la posición en la que la colocó el profesor.

Resultado: Se produce un error. Se muestra la casa objetivo en la posición donde estableció el profesor la casa sólida y ésta última aparece en el centro de los ejes de coordenadas como se puede observar en la *Figura 6.6*.

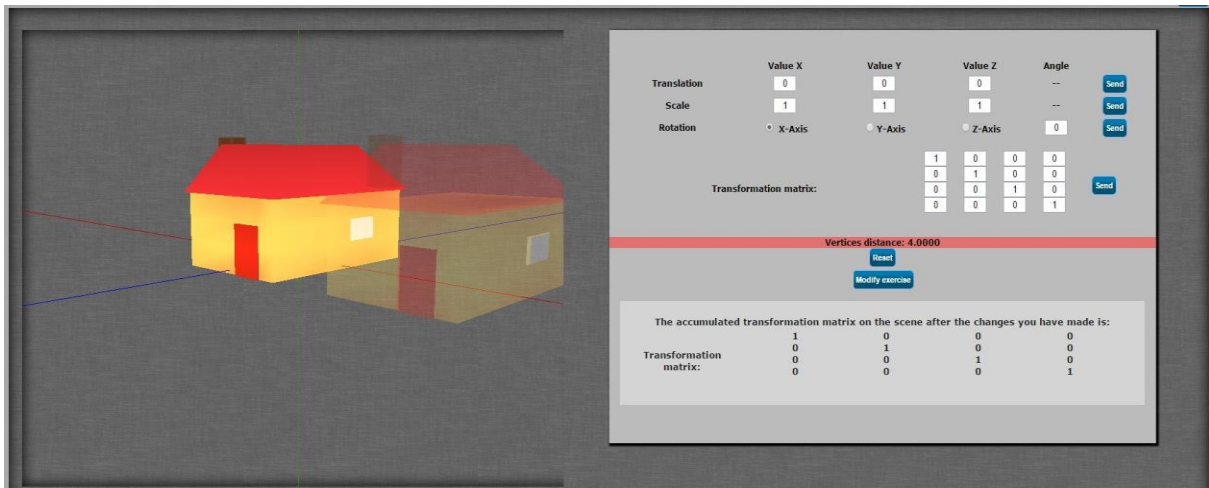


Figura 6.6. Error al modificar ejercicio de Transformaciones geométricas.

Solución: Se añade la condición para que cuando se modifique un ejercicio por parte del profesor no se cargue la casa objetivo y se apliquen las transformaciones según los parámetros que estableció el profesor como se puede observar en la *Figura 6.7*.

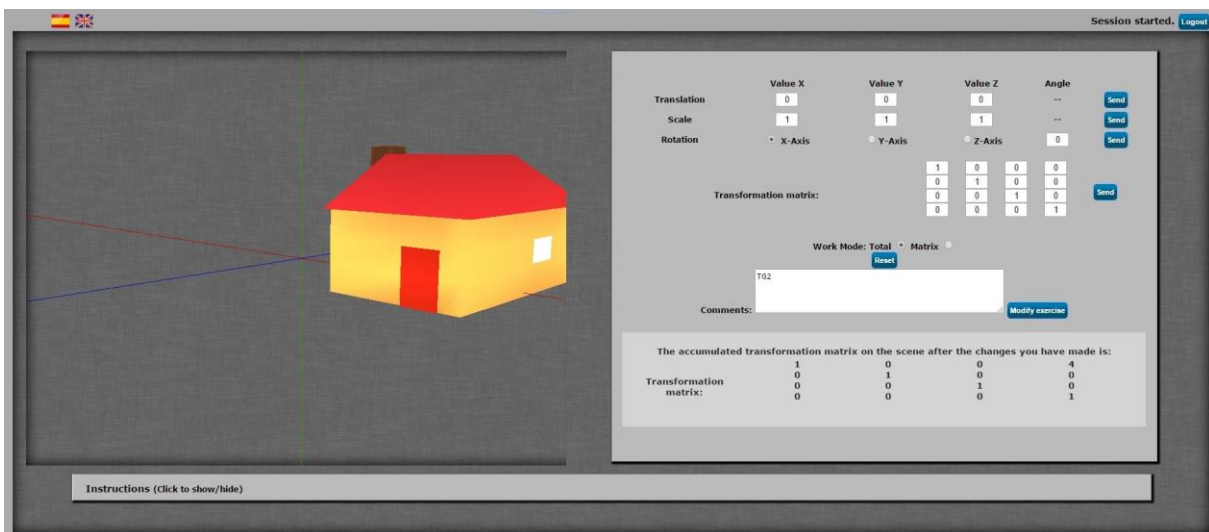


Figura 6.7. Solución tomada para modificar ejercicio de Transformaciones geométricas.

- **Paso 5: Modificar parámetros del ejercicio.**

Resultado esperado: Se aplican a la casa sólida las transformaciones individuales y de la matriz de transformación correctamente.

Resultado: Se obtiene el resultado esperado.

- **Paso 6: Guardar la modificación del ejercicio.**

Resultado esperado: Se guarda la modificación del ejercicio y se muestra la notificación correspondiente.

Resultado: Se obtiene el resultado esperado en la *Figura 6.8*.

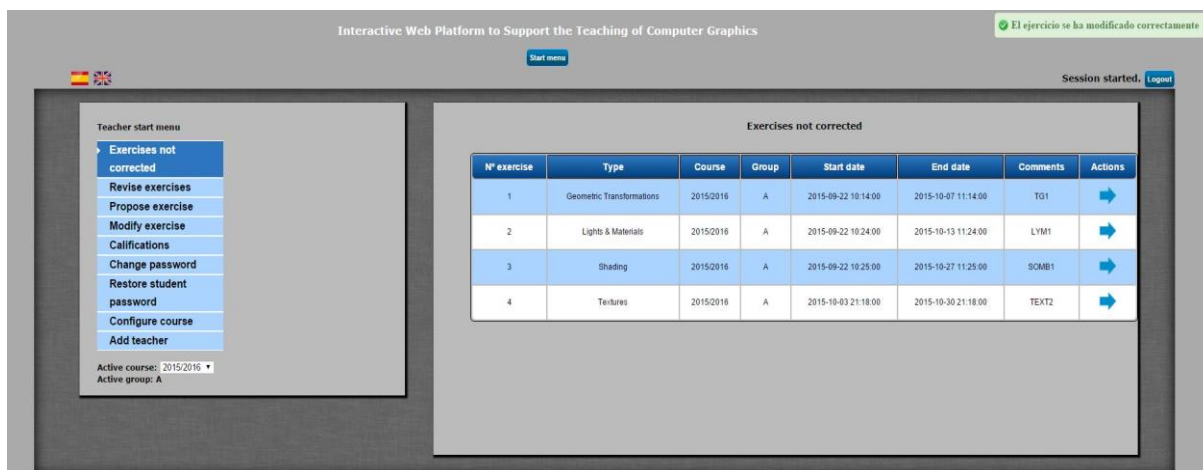


Figura 6.8. Ejercicio modificado con su notificación correspondiente.

6.3.3. CORREGIR UN EJERCICIO DEL TIPO TEXTURAS

- **Paso 1: Seleccionar el ejercicio propuesto a corregir.**

Resultado esperado: Se cargan los ejercicios realizados del ejercicio propuesto seleccionado.

Resultado: Se obtiene el resultado esperado en la *Figura 6.9*.

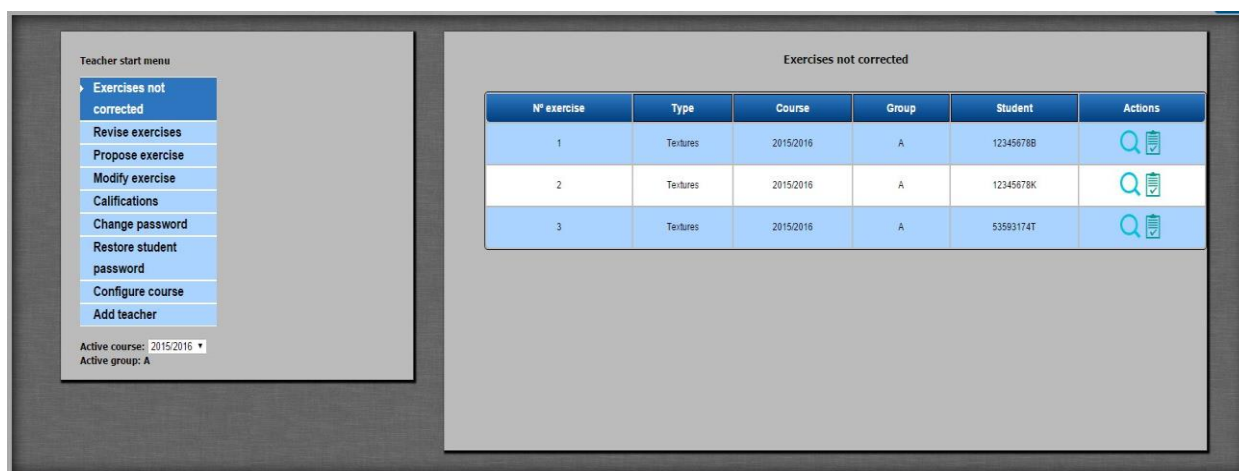


Figura 6.9. Ejercicio realizados por corregir del ejercicio propuesto seleccionado.

- **Paso 2: Seleccionar el ejercicio realizado que se desee corregir.**

Resultado esperado: Se carga el applet del ejercicio seleccionado con los datos para corregir el ejercicio. En la parte derecha se cargará la tabla con los resultados del alumno y profesor.

Resultado: Se obtiene el resultado esperado.

- **Paso 3: Insertar calificación y observaciones.**

Resultado esperado: Se insertan la calificación y las observaciones correctamente y la corrección se puede guardar cuando se desee.

Resultado: Se obtiene el resultado esperado en la *Figura 6.10*.

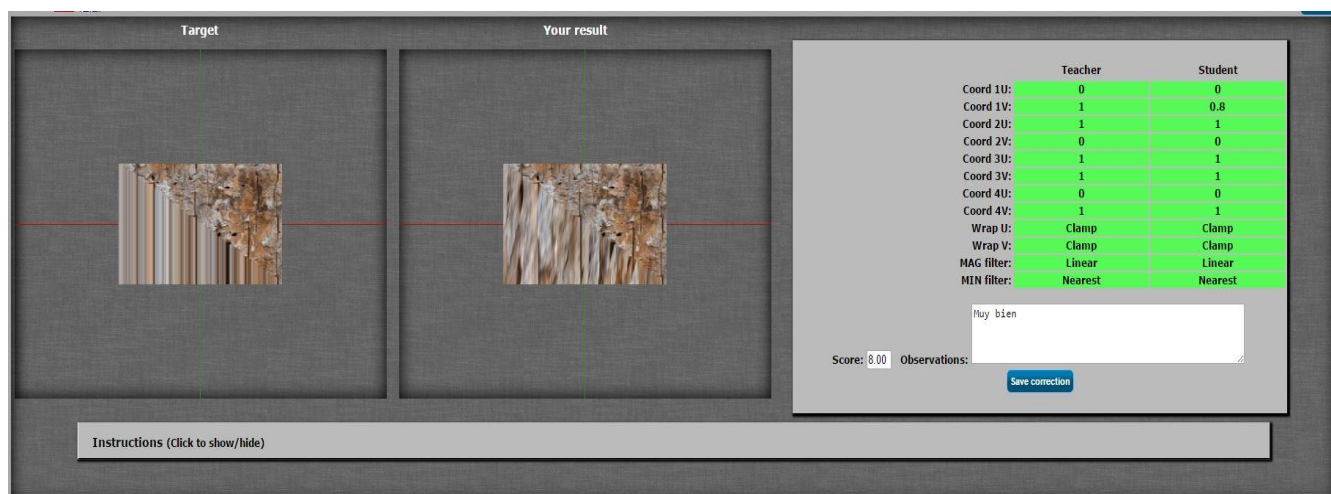


Figura 6.10. Calificaciones insertadas al corregir un ejercicio de Texturas.

- **Paso 4: Guardar corrección del ejercicio**

Resultado esperado: Se guarda la corrección del ejercicio, se carga el menú de inicio del profesor sino quedan más ejercicios por corregir del ejercicio seleccionado. Si queda algún ejercicio se carga la tabla con dichos ejercicios. Además se muestra la notificación indicando que la calificación se ha insertado correctamente. Si se inserta una calificación inválida se mostrará una notificación de error.

Resultado: Se obtiene el resultado esperado.

6.3.4. REVISAR UN EJERCICIO DEL TIPO MODELOS DE ILUMINACIÓN

- **Paso 1: Seleccionar la opción del menú “Revisar ejercicios”.**

Resultado esperado: Se muestra el contenido de la acción “Revisar ejercicios”.

Resultado: Se obtiene el resultado esperado.

- **Paso 2: Seleccionar el ejercicio propuesto que se desee revisar.**

Resultado esperado: Se cargan los ejercicios corregidos del ejercicio propuesto seleccionado.

Resultado: Se obtiene el resultado esperado.

- **Paso 3: Seleccionar el ejercicio corregido que se desee revisar.**

Resultado esperado: Se carga el applet del tipo seleccionado con los parámetros del alumno y del profesor y la pantalla de corrección. Deberá cargarse la calificación asignada por el profesor y las observaciones de la corrección.

Resultado: Se obtiene el resultado esperado exceptuando la calificación del ejercicio. Ésta última es la resultante de aplicar el algoritmo de autocorrección y no la asignada por el profesor en la corrección.

Solución: Se ejecuta el algoritmo de autocorrección únicamente cuando se esté corrigiendo el ejercicio y no cuando se esté revisando el mismo.

- **Paso 4: Guardar corrección del ejercicio.**

Resultado esperado: Se guarda la corrección correctamente, se carga el menú de inicio del profesor y se muestra la notificación correspondiente.

Resultado: Se obtiene el resultado esperado.

6.3.5. REALIZAR UN EJERCICIO DEL TIPO SOMBREADO

- **Paso 1:** Seleccionar el ejercicio que se desee realizar.

Resultado esperado: Se carga el applet del ejercicio seleccionado con los parámetros establecidos por el profesor.

Resultado: Se obtiene el resultado esperado en la *Figura 6.11*.

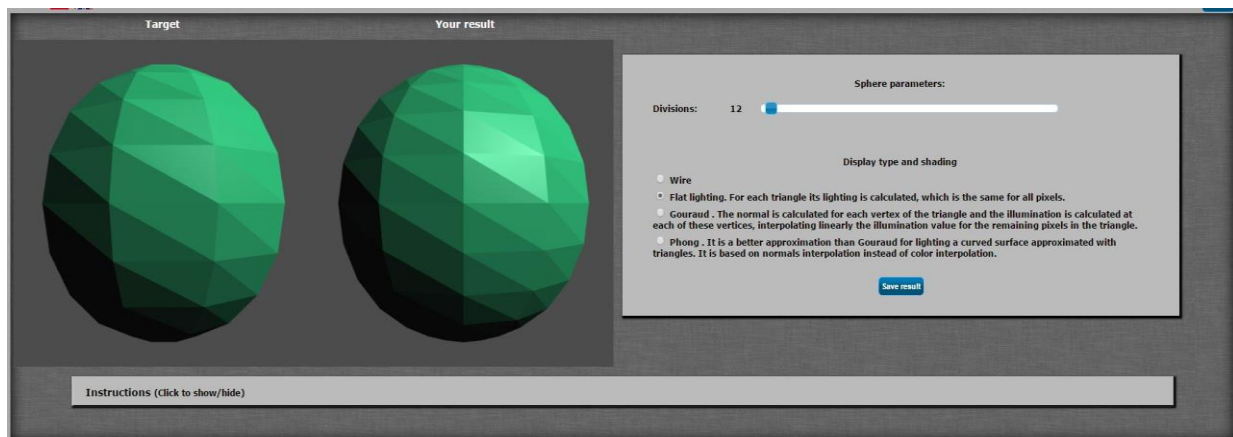


Figura 6.11. Realizar ejercicio de Sombreado.

- **Paso 2:** Modificar los parámetros del ejercicio.

Resultado esperado: Se aplican los cambios correctamente a la esfera de la escena del canvas del alumno. Si se consigue el objetivo la esfera parpadeará.

Resultado: Se obtiene el resultado esperado en la *Figura 6.12*.

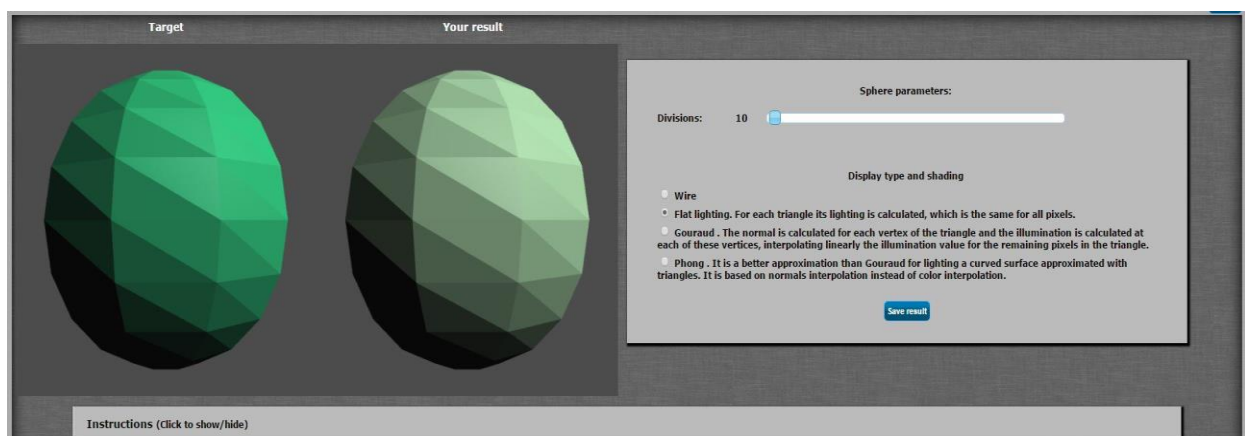


Figura 6.12. Objetivo conseguido modificando los parámetros del applet.

- **Paso 3: Guardar el resultado del ejercicio.**

Resultado esperado: Se guarda el resultado del ejercicio, se carga el menú de inicio del alumno y se muestra la notificación correspondiente.

Resultado: Se obtiene el resultado esperado.

6.3.6.EDITAR UN EJERCICIO DE TIPOS DE LUCES

- **Paso 1: Seleccionar el ejercicio que se desee editar.**

Resultado esperado: Se carga el applet del ejercicio seleccionado con los parámetros establecidos por el profesor y el resultado obtenido por el alumno.

Resultado: Se obtiene el resultado esperado.

- **Paso 2: Guardar el nuevo resultado del ejercicio.**

Resultado esperado: Se guarda el nuevo resultado del ejercicio, se carga el menú de inicio del alumno y se muestra la notificación correspondiente.

Resultado: Se obtiene el resultado esperado.

7. CONCLUSIONES

Una vez desarrollado el proyecto se puede decir que se ha cumplido el propósito principal que se planteaba en el primer capítulo de esta memoria. Se ha conseguido una plataforma Web fácilmente usable tanto para los profesores como para los alumnos para la propuesta, resolución y corrección interactiva de ejercicios de Informática Gráfica. Este hecho puede ayudar a incentivar la realización de ejercicios por parte de los alumnos para mejorar sus conocimientos en Informática Gráfica, así como mejorar las calificaciones obtenidas en esta asignatura.

La principal dificultad que ha planteado el proyecto se trata de la adaptación a la implementación realizada para el sistema Web previamente desarrollado. Esta adaptación ha requerido un tiempo considerable a lo largo del desarrollo del proyecto, esto ha sido una razón fundamental para los retrasos ocasionados en la entrega del mismo.

La integración del sistema de menús de alumno y profesor con los applets de la aplicación ha supuesto un gran reto, debido a que cada applet, aunque mantenga la misma estructura, tiene sus características individuales, lo que provoca que sea un trabajo complicado integrar estos applets para proponer, modificar, realizar, ver y corregir ejercicios.

Por último, se ha conseguido integrar correctamente la parte servidor del sistema Web con la parte cliente gracias a la tecnología AJAX, que ha sido clave en el desarrollo de este proyecto.

7.1. PROPUESTAS DE MEJORA

Este proyecto ha surgido de una propuesta de mejora del sistema Web previamente desarrollado. Para este proyecto se ha seguido la estructura de la implementación existente, por lo que es sencillo la inclusión de nuevo módulos.

Una propuesta de mejora puede ser añadir nuevos applets a la aplicación que cubran otros aspectos relacionados con la Informática Gráfica. Esta propuesta sería sencilla de integrar a la aplicación aunque haría falta integrar los menús con los nuevos applets para poder plantear ejercicios de los mismos correctamente.

Otra propuesta de mejora puede ser la optimización del sistema Web para ser visualizado en dispositivos móviles. Esta mejora supondría un gran avance en la aplicación y se puede realizar de manera sencilla gracias al empleo del patrón Modelo-Vista-Controlador, teniendo en cuenta como principal dificultad la adaptación de la vista al tamaño de la pantalla del dispositivo móvil.

Por último, otra posible mejora sería la modificación en tiempo real de los parámetros de la escena mediante la interacción con el ratón en el canvas de cada applet. Gracias a esta mejora se puede conseguir que los canvas de los applets sean más grandes, ya que se podrían eliminar los controles de algunos applets o integrarlos en el propio canvas, lo que supondría una ventaja también si se adapta la aplicación para visualizarla en dispositivos móviles.

8. BIBLIOGRAFÍA

[1] - Apuntes de la asignatura Fundamentos de Ingeniería del Software - Luis Alfonso Ureña López y Jose Ignacio Gómez Espinola

[2] - Servicio de alojamiento de servidores de la Universidad de Jaén

<http://www10.ujaen.es/conocenos/servicios-unidades/sinformatica/servicioweb>

[3] - Platform as a Service - Wikipedia

https://en.wikipedia.org/wiki/Platform_as_a_service

[4] - Openshift by Red Hat

<https://www.openshift.com/>

[5] - Amazon Web Services (AWS) - Wikipedia

https://es.wikipedia.org/wiki/Amazon_Web_Services

[6] - Heroku - Wikipedia

<https://es.wikipedia.org/wiki/Heroku>

[7] - Guía HTML para desarrolladores - Mozilla Developers Network (MDN).

<https://developer.mozilla.org/es/docs/Web/Guide/HTML>

[8] - Especificación WebGL Versión 1.0.2 - Grupo Khronos

<https://www.khronos.org/registry/webgl/specs/1.0/#1>

[9] - Guía Javascript - Mozilla Developers Network (MDN)

<https://developer.mozilla.org/es/docs/Web/JavaScript/Guide>

[10] - API de JQuery

<http://api.jquery.com/>

[11] - Manual de PHP

<https://secure.php.net/manual/es/index.php>

[12] - ASP.NET - Wikipedia

<https://es.wikipedia.org/wiki/ASP.NET>

[13] - JavaServer Pages (JSP) - Wikipedia

https://es.wikipedia.org/wiki/JavaServer_Pages

[14] - Guía AJAX para desarrolladores - Mozilla Developers Network (MDN)

<https://developer.mozilla.org/es/docs/AJAX>

[15] - MySQL Developer zone - Reference manuals

<http://dev.mysql.com/doc/>

[16] - Oracle - Página Web oficial

<http://www.oracle.com/es/index.html>

[17] - Git Book - Documentación

<https://git-scm.com/book/es/v1>

[18] - BOE - 30 de enero de 2015 - tablas salariales

<https://www.boe.es/boe/dias/2015/02/11/pdfs/BOE-A-2015-1343.pdf>

[19] - BOE - 27 de noviembre de 2014 - Impuesto sobre Sociedades

https://www.boe.es/diario_boe/txt.php?id=BOE-A-2014-12328

[20] - Servidor HP Proliant ML10 V2

http://www.pccomponentes.com/hp_proliant_ml10_v2_g3240_4gb_usb_3_0.html

[21] Apuntes de Diseño de Bases de Datos de la Universidad de Valencia

<http://informatica.uv.es/estguia/ATD/apuntes/teoria/documentos/DisenioBD.pdf>

[22] Normalización de Bases de Datos

https://es.wikipedia.org/wiki/Normalizaci%C3%B3n_de_bases_de_datos

[23] Seguridad de la información

[https://es.wikipedia.org/wiki/Seguridad_de_la_información](https://es.wikipedia.org/wiki/Seguridad_de_la_informaci%C3%B3n)

[24] Tipos de datos en MySQL

<http://www.desarrolloweb.com/articulos/1054.php>

[25] Usabilidad y accesibilidad

<http://www.desarrolloweb.com/articulos/usabilidad-y-accesibilidad.html>

[26] El gran libro de HTML5, CSS3 y Javascript. Juan Diego Gauchat.
MARCOMBO, S.A.

[27] Manual de JQuery. Miguel Ángel Álvarez

<http://www.desarrolloweb.com/manuales/manual-jquery.html>

[28] Documentación oficial Three.js

<http://threejs.org/docs/>

[29] Funciones para AJAX. Librería JQuery

http://librosweb.es/libro/ajax/capitulo_10/la_libreria_jquery.html

[30] Manual PHP. Extensión mysqli mejorada

<http://php.net/manual/es/book.mysqli.php>

[31] Página oficial NetBeans IDE

<https://netbeans.org/>

[32] Página oficial Sublime Text

<http://www.sublimetext.com/>

[33] Introducción a JSON – JavaScript Object Notation

<http://www.json.org/>

[34] Notify.js – Plugin JQuery para notificaciones

<http://notifyjs.com/>

[35] Framework Jasmine para realizar pruebas unitarias en Javascript

<http://jasmine.github.io/>

9. ANEXO A: MANUAL DE USUARIO

En este anexo se explicará el funcionamiento del sistema Web, centrándose en la nueva funcionalidad incluida, es decir, lo referente a la funcionalidad incluida en los menús de alumnos y profesor así como su integración con los distintos applets.

9.1. ACCESO AL SISTEMA WEB

El sistema Web se encuentra alojado en la PaaS Openshift. Para acceder al mismo hay que introducir en el navegador la siguiente URL:

<http://igvapp-ujatfg.rhcloud.com/#home>

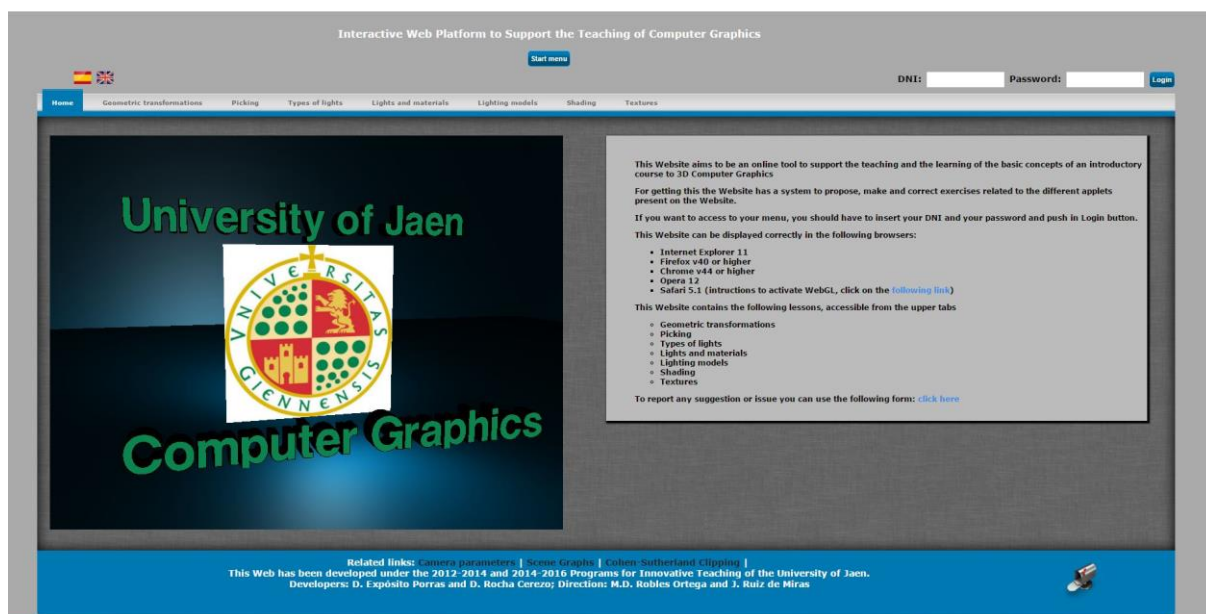


Figura 9.1. Pantalla de inicio del sistema Web.

Una vez se accede al sistema Web se puede ver la pantalla de inicio (Figura 9.1). En la parte izquierda se puede ver una escena de ejemplo, mientras en la parte derecha se encuentran algunos comentarios explicando el funcionamiento del sistema Web.

El sistema Web se encuentra actualmente en dos idiomas, español e inglés. Para cambiar de idioma basta con pulsar en la bandera del idioma deseado y el sistema cambiará de idioma.

9.2. INICIAR SESIÓN

Para iniciar sesión el usuario tendrá que introducir su DNI y contraseña en la parte superior derecha de la página. Cuando haya introducido ambos campos podrá pulsar en el botón “Login” para iniciar sesión.

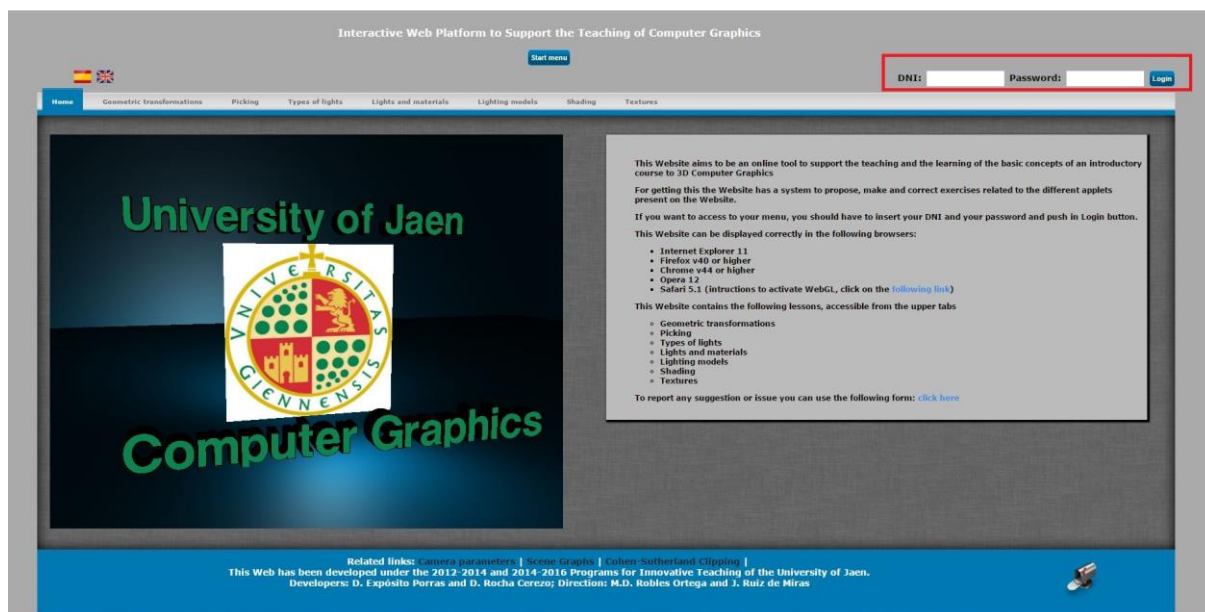


Figura 9.2. Pantalla de inicio del sistema Web antes de iniciar sesión.

En el caso de que haya entrado en el sistema un profesor, se cargará el menú de inicio del profesor como se puede ver en la *Figura 9.3*. Si, por el contrario el que ha iniciado sesión es un alumno se cargará el menú correspondiente, el cual se puede ver en la *Figura 9.4*.

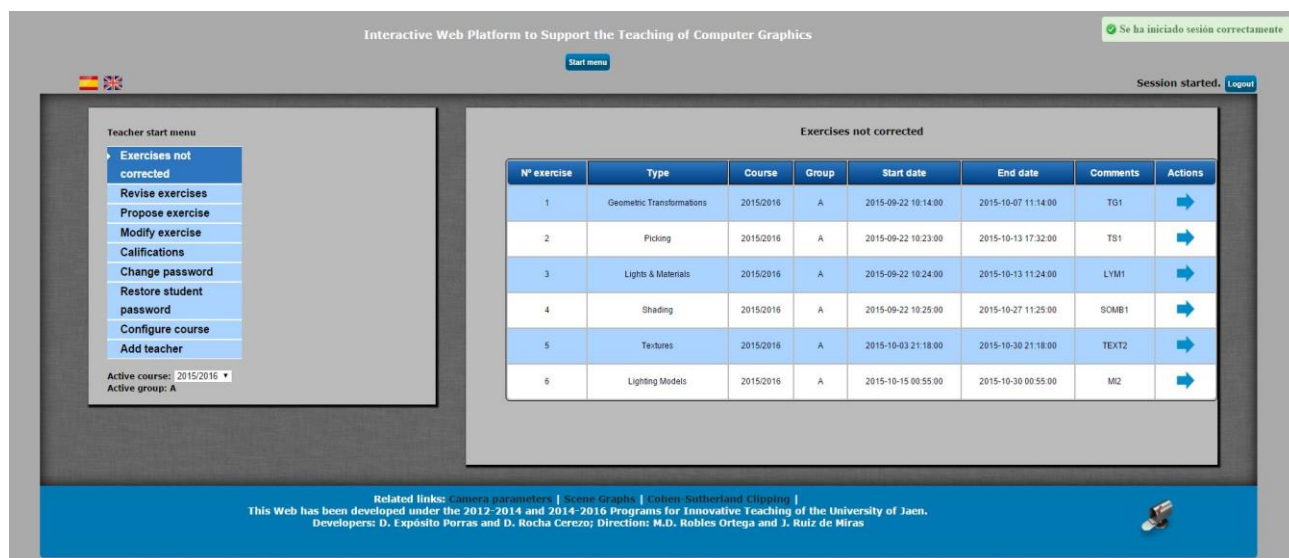


Figura 9.3. Pantalla del menu del profesor una vez se ha iniciado sesión.

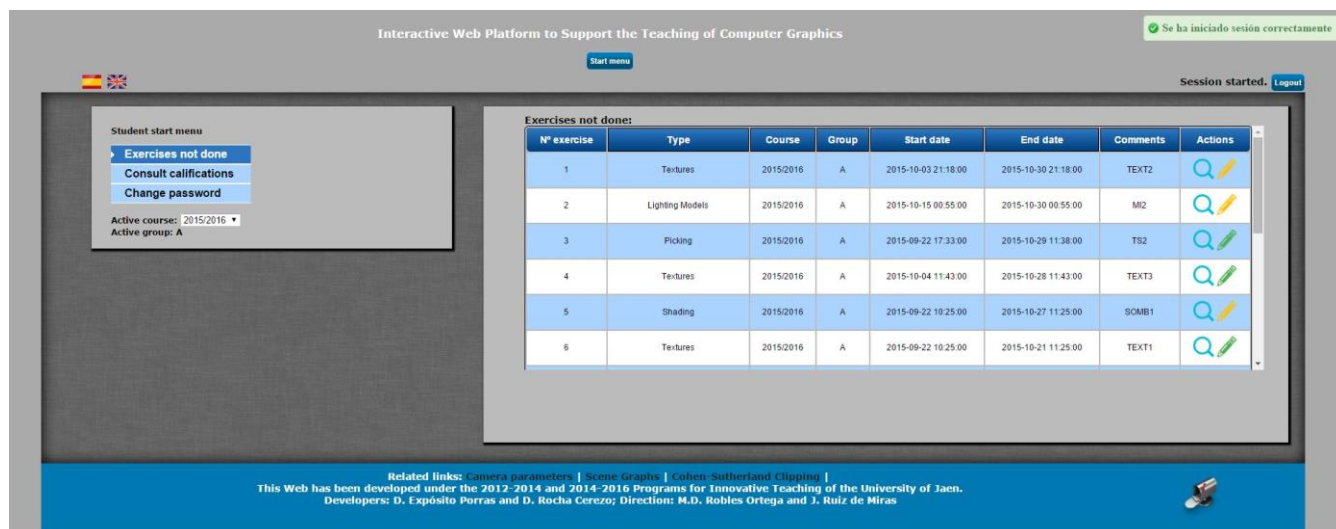


Figura 9.4. Pantalla del menu del alumno una vez se ha iniciado sesión.

9.3. PROPONER EJERCICIO

Para proponer un ejercicio, el profesor deberá acceder al sistema y seleccionar la acción del menú correspondiente (Figura 9.5).

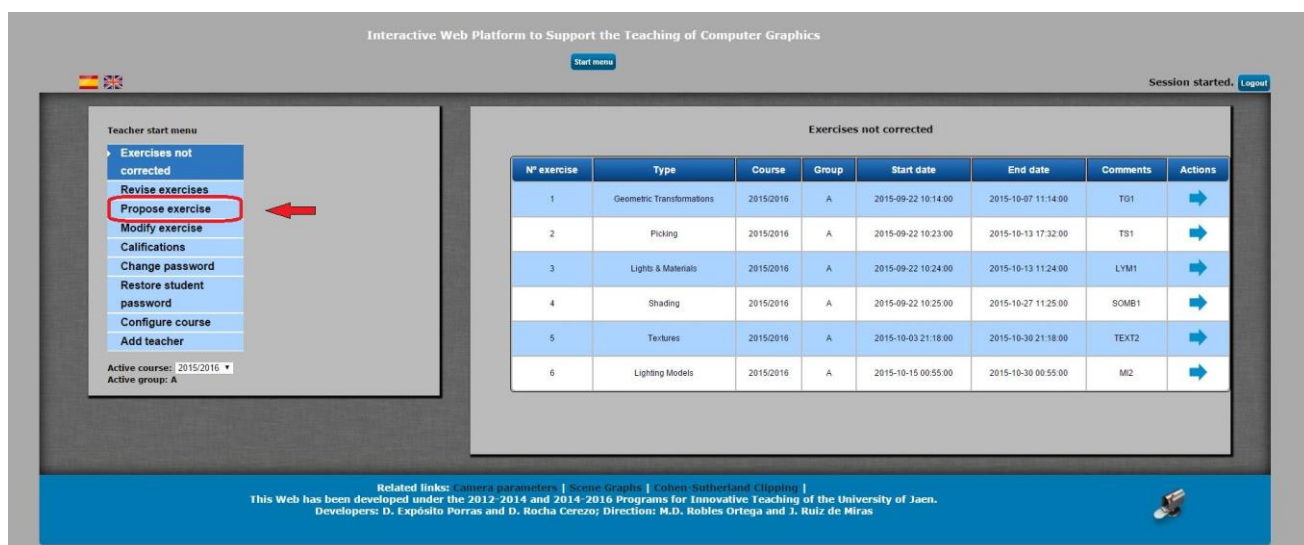


Figura 9.5. Pantalla del menu del profesor cuando se dispone a proponer un ejercicio.

El sistema cargará el contenido de dicha acción (Figura 9.6). El profesor podrá seleccionar el tipo de ejercicio que desea configurar y las fechas de inicio y fin del ejercicio. Una vez realizado esto el sistema cargará los ejercicios previamente propuestos del tipo seleccionado y el botón para proponer un nuevo ejercicio partiendo de los parámetros por defecto como se puede ver en la Figura 9.7.

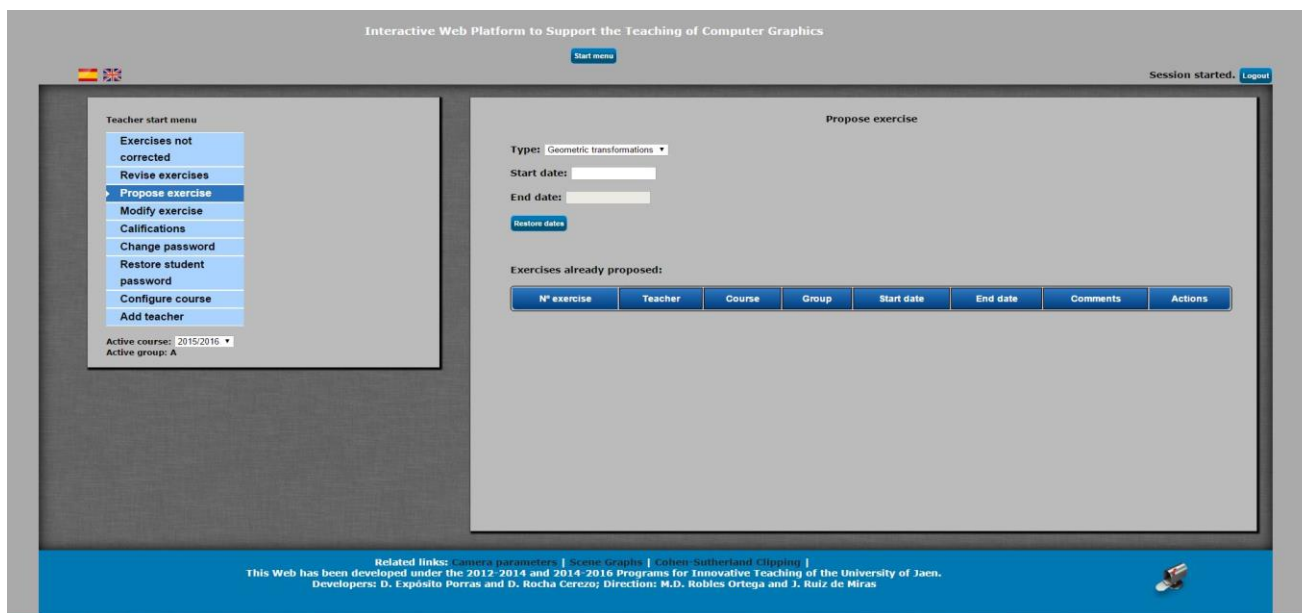


Figura 9.6. Contenido de la acción de proponer un ejercicio.

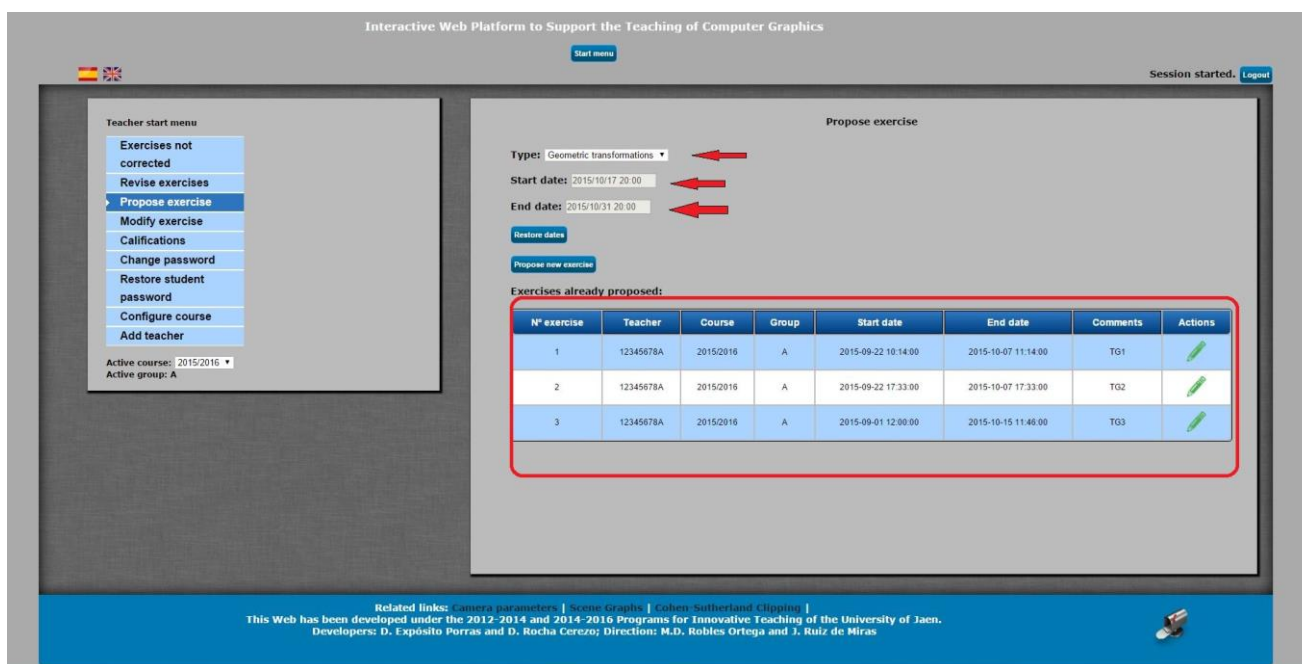


Figura 9.7. Tipo de ejercicio seleccionado y fechas de inicio y fin introducidas.

Una vez se haya realizado esto, habrá dos opciones para proponer un ejercicio (Figura 9.8). La primera sería pulsando en el botón “Proponer nuevo ejercicio”. El sistema cargaría el applet del tipo seleccionado con los parámetros por defecto. La segunda opción sería proponer un ejercicio a partir de un ejercicio previamente configurado del tipo seleccionado. Para realizar esto, el profesor tendrá que seleccionar el ejercicio deseado pulsando en la acción del mismo. A continuación el

sistema cargará el applet del tipo de ejercicio seleccionado con los parámetros que fueron configurados para dicho ejercicio.

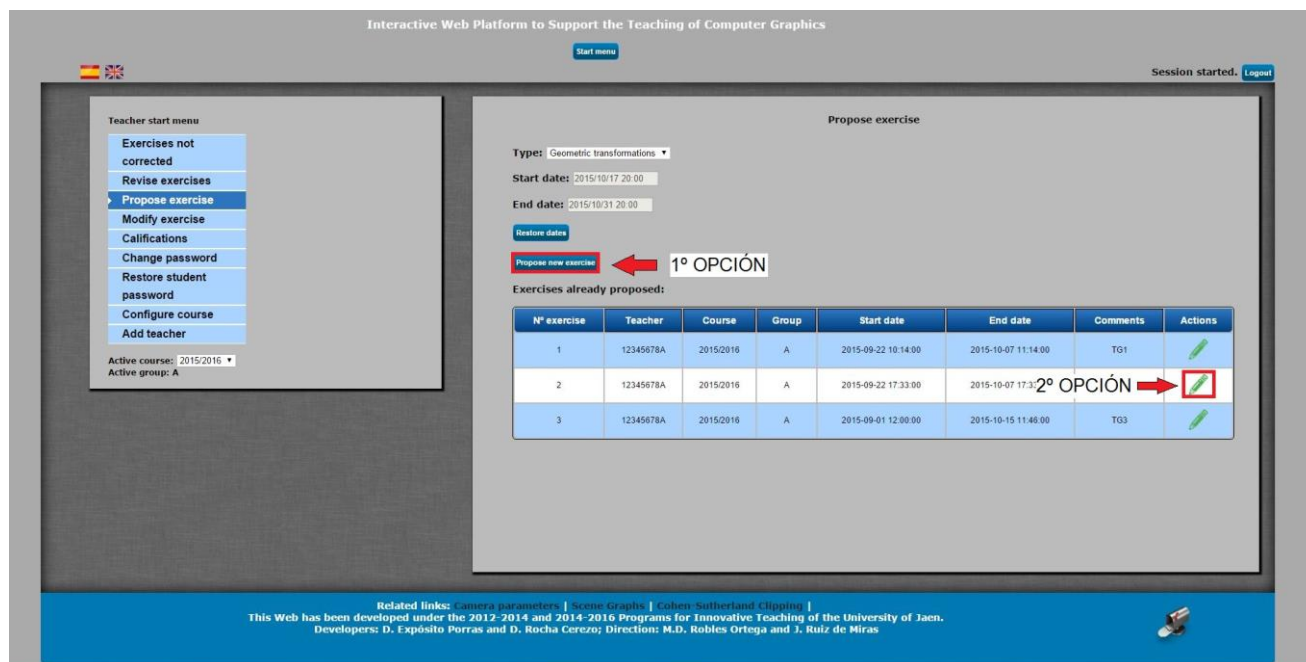


Figura 9.8. Distintas opciones para proponer un ejercicio.

En la Figura 9.9 se puede observar que el sistema ha cargado el applet del tipo seleccionado que en este caso es de *Transformaciones Geométricas*. Para proponer un ejercicio basta con configurar los parámetros del applet y pulsar el botón de “Guardar ejercicio”. Opcionalmente se pueden añadir comentarios en el área de texto indicada para poder identificar el ejercicio. Una vez se haya guardado el ejercicio, el sistema cargará el menú de inicio del profesor.

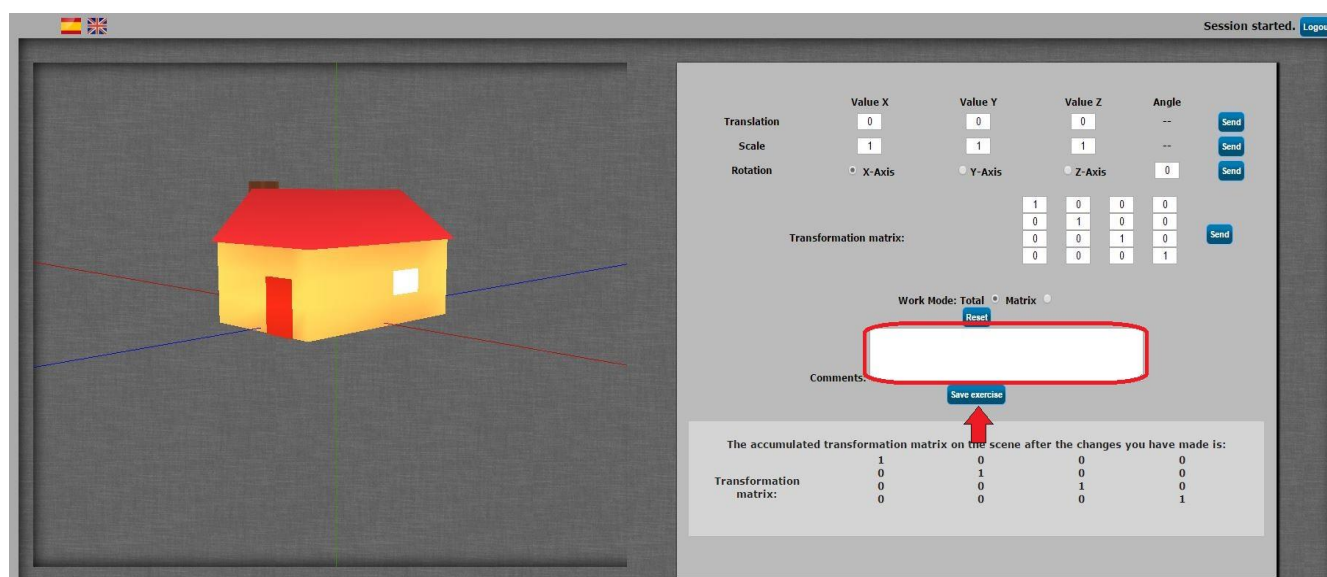


Figura 9.9. Configurar ejercicio y almacenarlo en la base de datos.

9.4. MODIFICAR EJERCICIO

Para modificar un ejercicio, el profesor deberá acceder al sistema y seleccionar la acción del menú correspondiente (*Figura 9.10*).

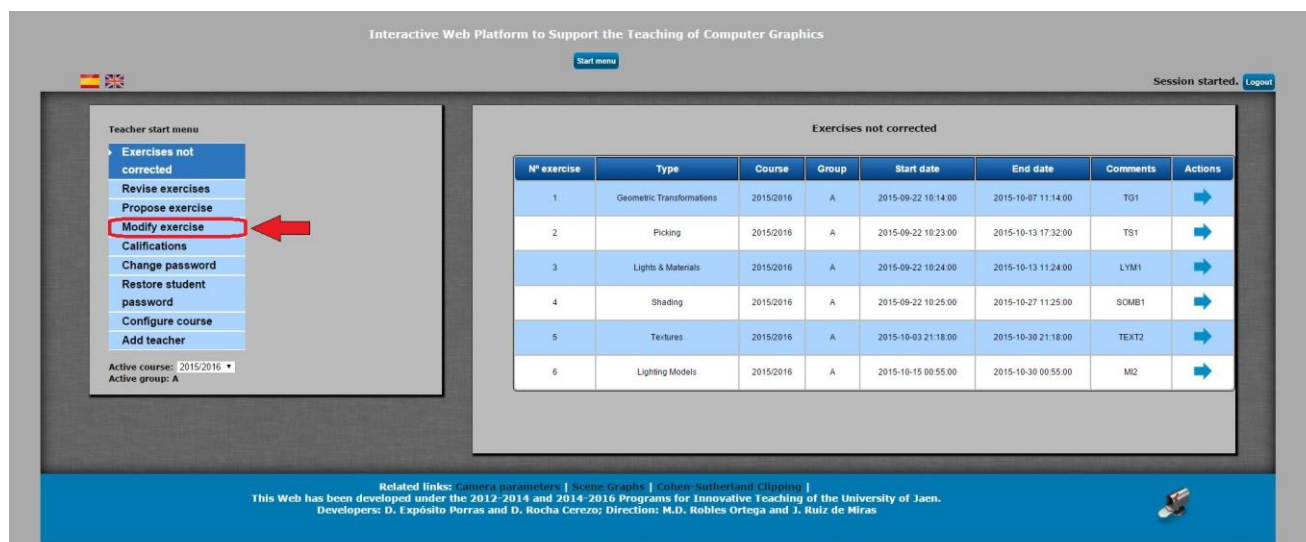


Figura 9.10. Pantalla del menú del profesor cuando se dispone a modificar un ejercicio.

El sistema cargará el contenido de la acción seleccionada (*Figura 9.11*). Para seleccionar el tipo de ejercicio que se desea modificar bastará con escogerlo en el selector de tipos.

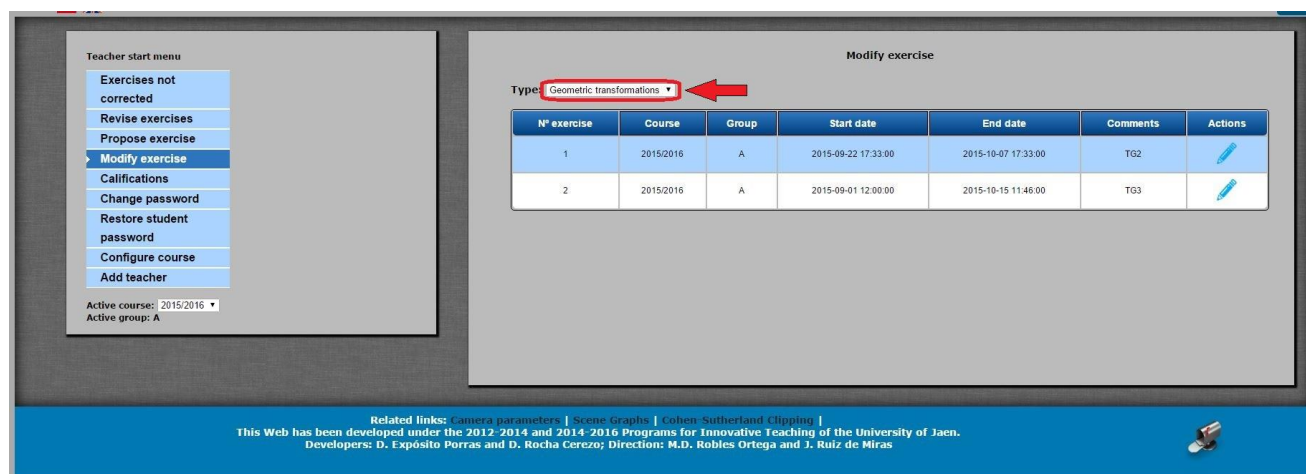


Figura 9.11. Pantalla para seleccionar el tipo de ejercicio que se desea modificar.

El profesor podrá seleccionar el ejercicio deseado pulsando en la acción del mismo y el sistema cargará la pantalla (*Figura 9.12*) para modificar las fechas del ejercicio si se considera oportuno. Para acceder al ejercicio que se desea modificar

habrá que pulsar el botón “Modificar ejercicio” y el sistema cargará el applet del tipo seleccionado con los parámetros establecidos para el ejercicio.

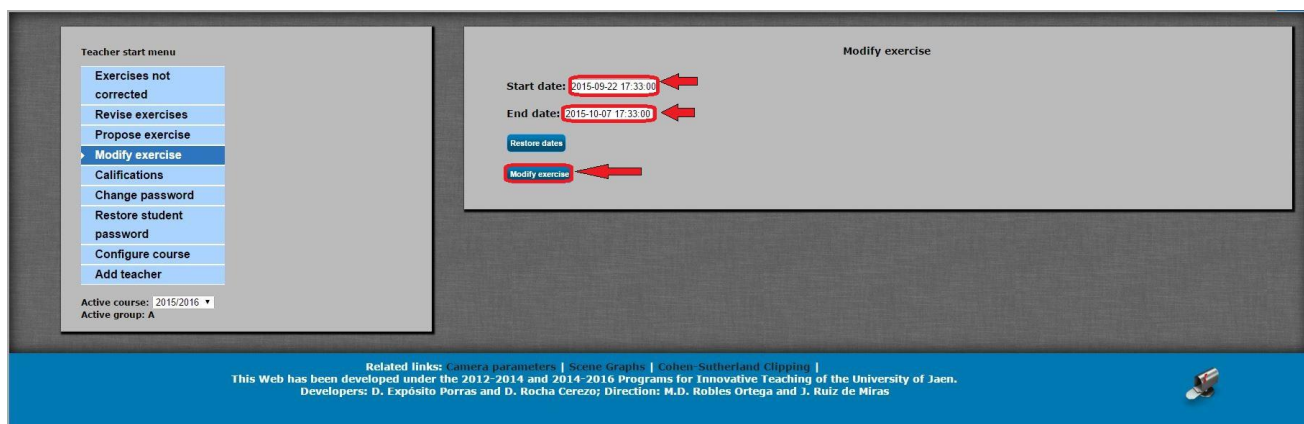


Figura 9.12. Modificar las fechas del ejercicio y pulsar el botón para modificarlo.

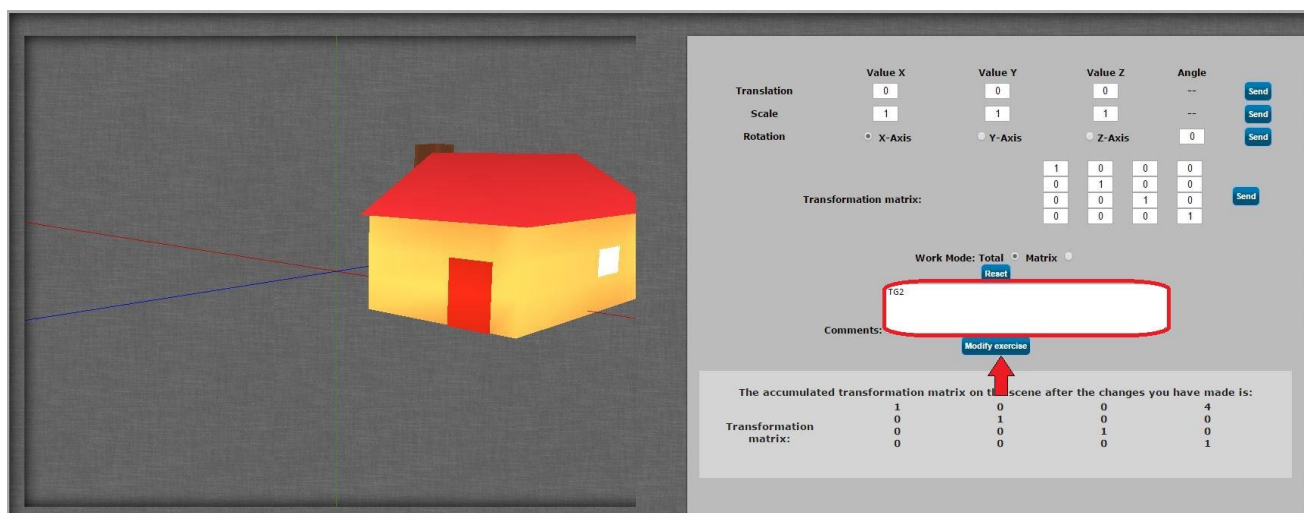


Figura 9.13. Modificar los comentarios del ejercicio y pulsar en “Modificar ejercicio”.

El profesor podrá modificar los comentarios del ejercicio si lo estima oportuno y podrá pulsar en el botón “Modificar ejercicio” para guardar los cambios en el ejercicio configurado como se puede observar en la *Figura 9.13*.

9.5. CORREGIR EJERCICIO

Para corregir un ejercicio, el profesor deberá acceder al sistema y seleccionar la acción del menú correspondiente (*Figura 9.14*). El profesor podrá pulsar en la acción del ejercicio propuesto que desee corregir y el sistema cargará los ejercicios realizados por los alumnos de dicho ejercicio (*Figura 9.15*).

Teacher start menu

- Exercises not corrected
- Revise exercises
- Propose exercise
- Modify exercise
- Califications
- Change password
- Restore student password
- Configure course
- Add teacher

Active course: 2015/2016
Active group: A

Exercises not corrected

N° exercise	Type	Course	Group	Start date	End date	Comments	Actions
1	Geometric Transformations	2015/2016	A	2015-09-22 10:14:00	2015-10-07 11:14:00	TG1	➡
2	Picking	2015/2016	A	2015-09-22 10:23:00	2015-10-13 17:32:00	TS1	➡
3	Lights & Materials	2015/2016	A	2015-09-22 10:24:00	2015-10-13 11:24:00	LYM	➡
4	Shading	2015/2016	A	2015-09-22 10:25:00	2015-10-27 11:25:00	SCMB1	➡
5	Textures	2015/2016	A	2015-10-03 21:18:00	2015-10-30 21:18:00	TEXT2	➡
6	Lighting Models	2015/2016	A	2015-10-15 00:55:00	2015-10-30 00:55:00	MI2	➡

Related links: Camera parameters | Scene Graphs | Cohen-Sutherland Clipping |
This Web has been developed under the 2012-2014 and 2014-2016 Programs for Innovative Teaching of the University of Jaen.
Developers: D. Expósito Porras and D. Rocha Cerezo; Direction: M.D. Robles Ortega and J. Ruiz de Miras

Figura 9.14. Ejercicios propuestos pendientes de corrección.

Teacher start menu

- Exercises not corrected
- Revise exercises
- Propose exercise
- Modify exercise
- Califications
- Change password
- Restore student password
- Configure course
- Add teacher

Active course: 2015/2016
Active group: A

Exercises not corrected

N° exercise	Type	Course	Group	Student	Actions
1	Lights & Materials	2015/2016	A	12345678B	🔍 📄
2	Lights & Materials	2015/2016	A	53593174T	🔍 📄

Related links: Camera parameters | Scene Graphs | Cohen-Sutherland Clipping |
This Web has been developed under the 2012-2014 and 2014-2016 Programs for Innovative Teaching of the University of Jaen.
Developers: D. Expósito Porras and D. Rocha Cerezo; Direction: M.D. Robles Ortega and J. Ruiz de Miras

Figura 9.15. Ejercicios realizados del ejercicio propuesto seleccionado.

El profesor tendrá dos opciones para realizar en este punto. Podrá corregir el ejercicio o simplemente verlo. Pulsando en ambas acciones el sistema cargará el applet del ejercicio seleccionado con la pantalla para corregir o ver el ejercicio (Figura 9.16).

El profesor podrá tomar como referencia la calificación indicada por el sistema después de realizar el algoritmo de autocorrección. Opcionalmente podrá asignar las observaciones de la corrección que considere oportunas. Finalmente deberá pulsar en el botón “Guardar corrección”, se guardará la corrección y el sistema cargará, si existen, el resto de ejercicios realizados del ejercicio propuesto que se está

corrigiendo o el menú de inicio del profesor si se estaba corrigiendo el último ejercicio realizado del ejercicio propuesto seleccionado.

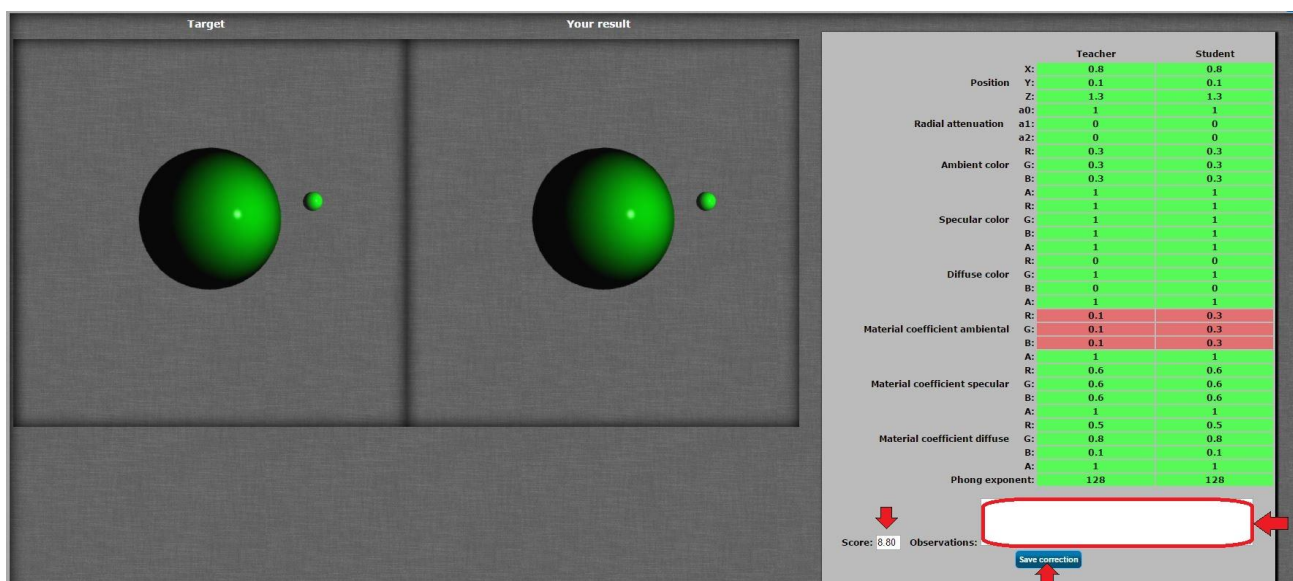


Figura 9.16. Corregir un ejercicio del tipo Luzes y materiales.

9.6. REVISAR EJERCICIO

Para revisar un ejercicio, el profesor deberá acceder al sistema y seleccionar la acción del menú correspondiente y el sistema cargará dicho contenido (Figura 9.17).



Figura 9.17. Pantalla del menú del profesor cuando se dispone a revisar un ejercicio.

El profesor podrá pulsar en la acción del ejercicio propuesto que desee revisar (Figura 9.18) y el sistema cargará las calificaciones asignadas para los ejercicios realizados de dicho ejercicio propuesto.

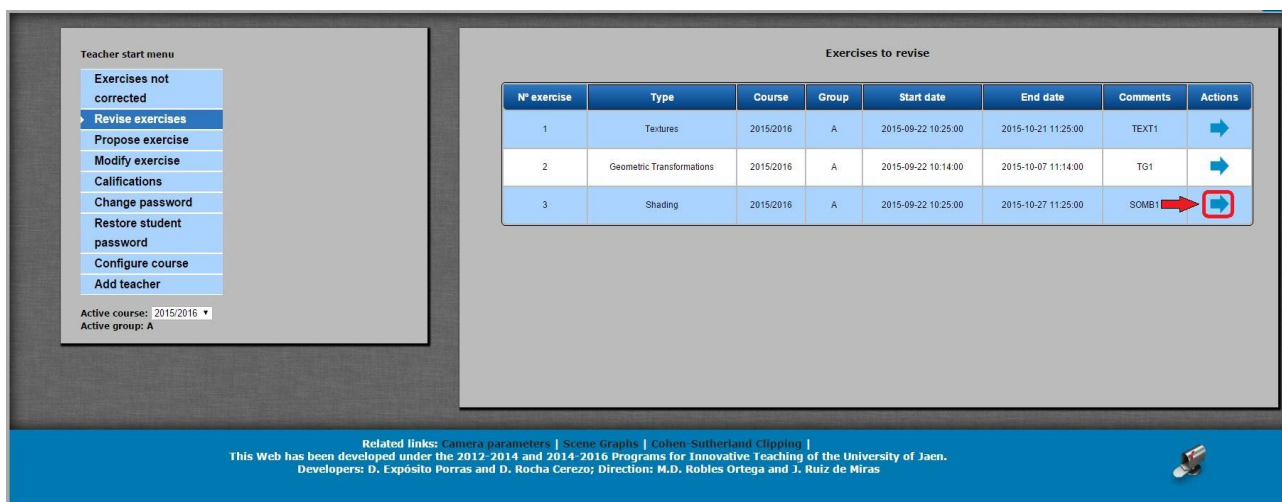


Figura 9.18. Seleccionar el ejercicio propuesto que se desee revisar.

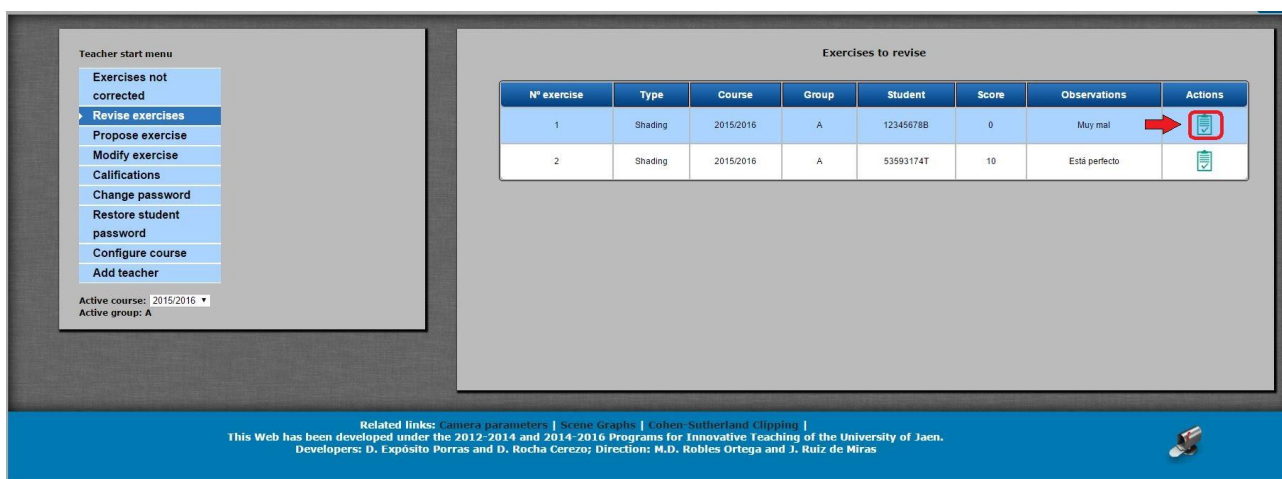


Figura 9.19. Seleccionar el ejercicio corregido que se desee revisar.

El profesor podrá pulsar en la acción del ejercicio que desee revisar y el sistema cargará el applet de dicho ejercicio con la pantalla de corrección del mismo.

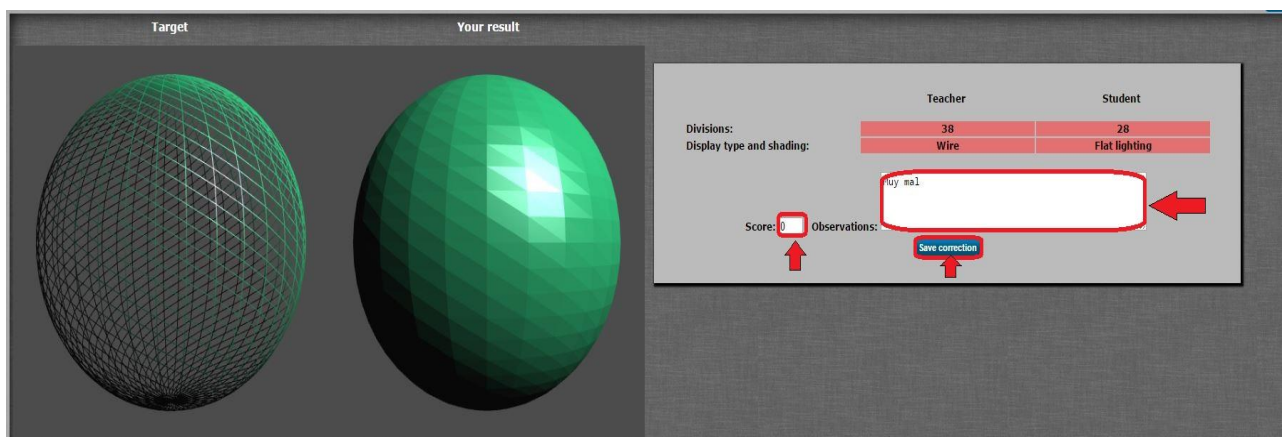


Figura 9.20. Modificar la calificación y observaciones de la corrección.

El profesor podrá modificar la calificación y las observaciones de la corrección si lo considera oportuno (*Figura 9.20*). A continuación deberá pulsar en el botón “Guardar correccion” para guardar la revisión del ejercicio.

9.7. CONSULTAR CALIFICACIONES

Para consultar las calificaciones, el profesor deberá acceder al sistema y seleccionar la acción del menú correspondiente y el sistema cargará dicho contenido (*Figura 9.21*).

Interactive Web Platform to Support the Teaching of Computer Graphics

Start menu

Session started. Logout

Teacher start menu

- Exercises not corrected
- Revise exercises
- Propose exercise
- Modify exercise
- Califications**
- Change password
- Restore student password
- Configure course
- Add teacher

Active course: 2015/2016
Active group: A

Exercises not corrected

Nº exercise	Type	Course	Group	Start date	End date	Comments	Actions
1	Geometric Transformations	2015/2016	A	2015-09-22 10:14:00	2015-10-07 11:14:00	TG1	➡
2	Picking	2015/2016	A	2015-09-22 10:23:00	2015-10-13 17:32:00	TS1	➡
3	Lights & Materials	2015/2016	A	2015-09-22 10:24:00	2015-10-13 11:24:00	LYM1	➡
4	Shading	2015/2016	A	2015-09-22 10:25:00	2015-10-27 11:25:00	SCMB1	➡
5	Textures	2015/2016	A	2015-10-03 21:18:00	2015-10-30 21:18:00	TEXT2	➡
6	Lighting Models	2015/2016	A	2015-10-15 00:55:00	2015-10-30 00:55:00	MI2	➡

Related links: camera parameters | scene graphs | colour management | lighting

This Web has been developed under the 2012-2014 and 2014-2016 Programs for Innovative Teaching of the University of Jaén.
Developers: D. Expósito Porras and D. Rocha Cerezo; Direction: M.D. Robles Ortega and J. Ruiz de Miras

Figura 9.21. Pantalla del menú del profesor cuando se dispone a ver las calificaciones.

En esta acción el profesor podrá consultar las calificaciones individuales de cada alumno y/o podrá exportar las calificaciones de un curso y grupo en un fichero “CSV”.

Para realizar la primera acción (*Figura 9.22*) el profesor tendrá que seleccionar el curso impartido del cual pretenda consultar las calificaciones. A continuación podrá seleccionar el grupo. En este punto se cargarán los alumnos del curso y grupo seleccionado y se podrán ir consultando las calificaciones individuales seleccionando el alumno.

Para realizar la segunda acción (*Figura 9.23*) simplemente habrá que seleccionar el curso y grupo que se deseen exportar y pulsar en el botón “Exportar calificaciones”.

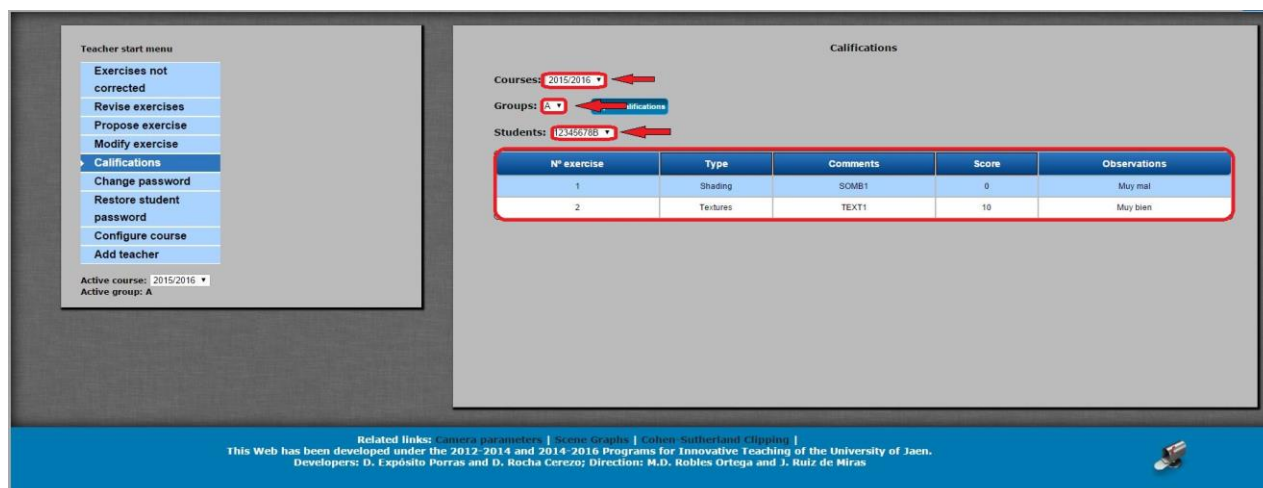


Figura 9.22. Consultar calificaciones por alumno.

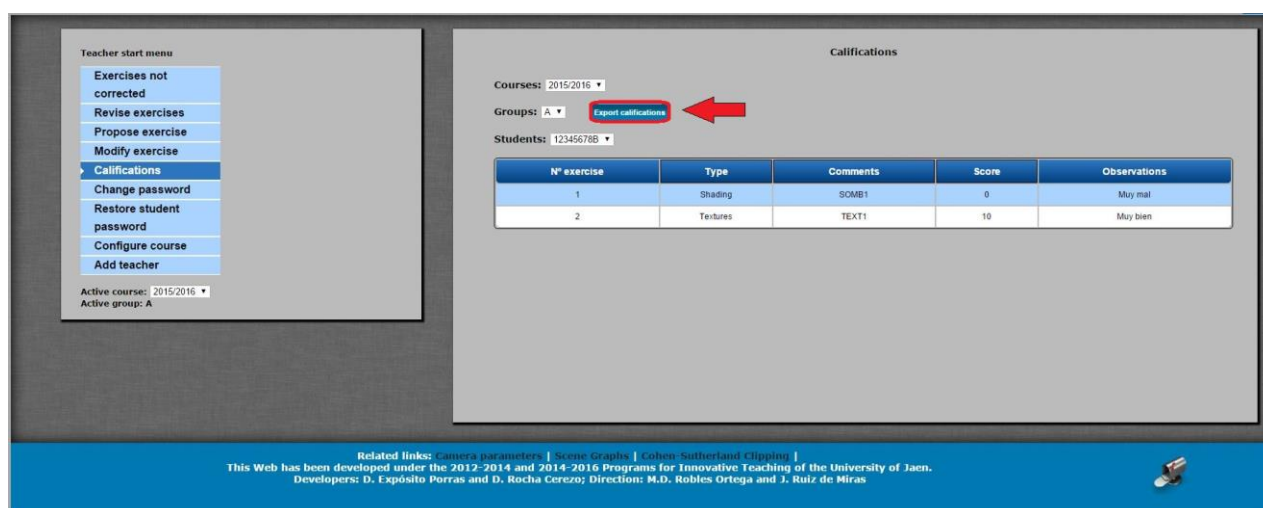


Figura 9.23. Exportar calificaciones del grupo.

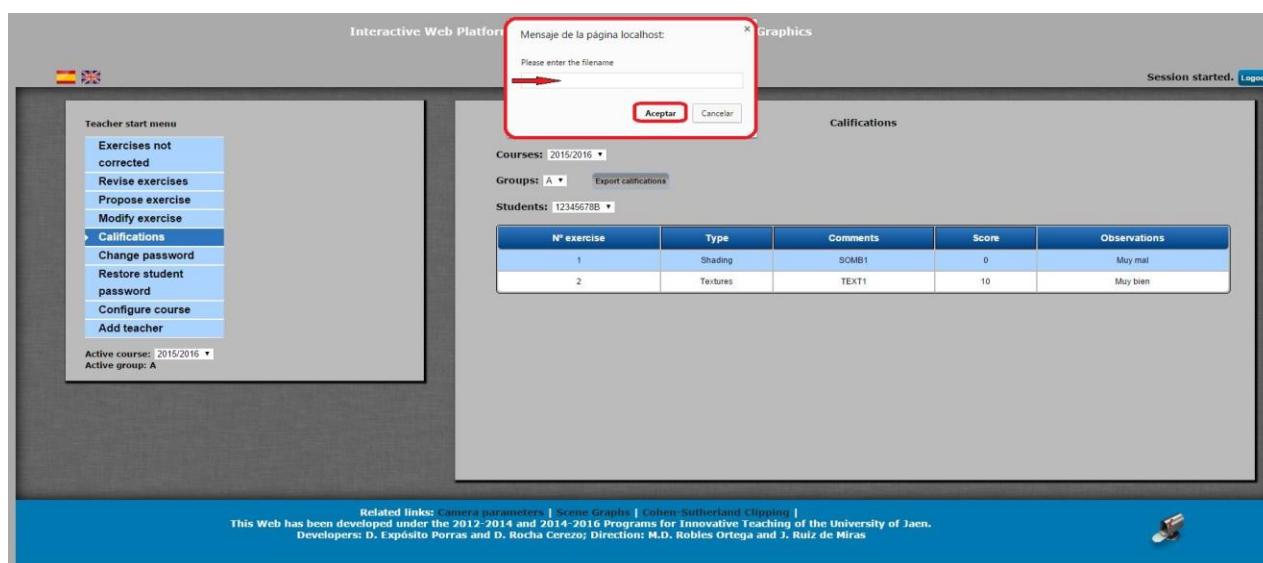


Figura 9.24. Escribir el nombre del fichero.

Una vez se haya pulsado el botón de “Exportar calificaciones”, aparecerá en la ventana un mensaje para escribir el nombre del fichero “CSV” que se desea exportar (Figura 9.24). A continuación habrá que pulsar en “Aceptar” para proceder a guardar el fichero.

Aparecerá otra ventana del sistema operativo para seleccionar la ruta donde desea guardar el fichero (Figura 9.25). Habrá que pulsar en “Guardar” para guardar el fichero.

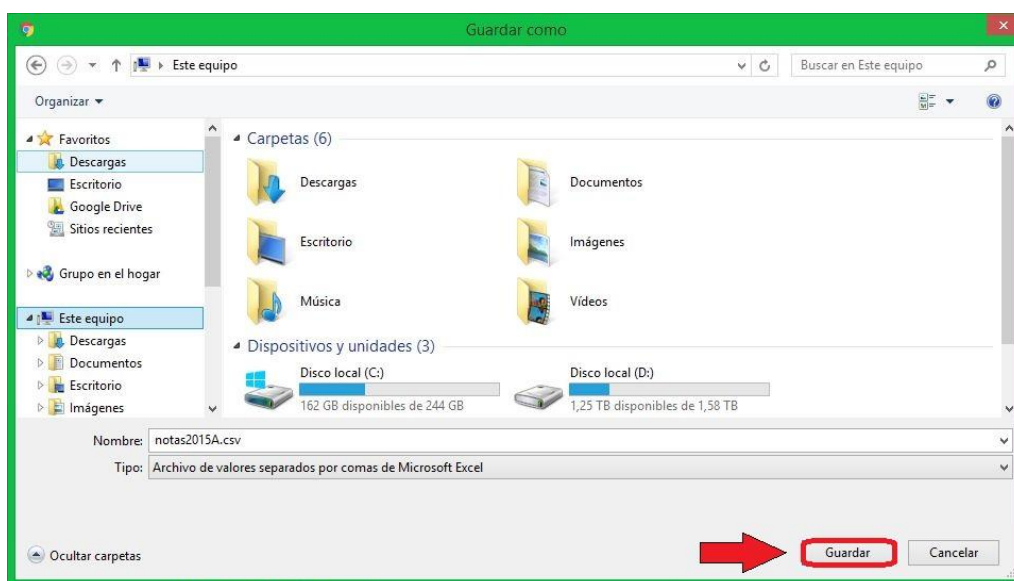


Figura 9.25. Exportar calificaciones del grupo.

La estructura del fichero quedará de la siguiente forma (Figura 9.26).

	A	B	C	D	E	F	G	H
1	DNI_ALUMNO	TIPO_EJERCICIO	COMENTARIOS	NOTA	OBSERVACIONES			
2	12345678B	Shading	SOMB1	0	Muy mal			
3	12345678B	Textures	TEXT1	10	Muy bien			
4	53593174T	Geometric Transformations	TG1	9,6	Falta un poco de rotación			
5	53593174T	Shading	SOMB1	10	Está perfecto			
6								
7								
8								
9								

Figura 9.26. Fichero con las calificaciones exportadas.

9.8. CAMBIAR CONTRASEÑA

Para cambiar la contraseña de usuario, tanto el alumno como el profesor deberán seleccionar la acción de su menú correspondiente (*Figura 9.27*).

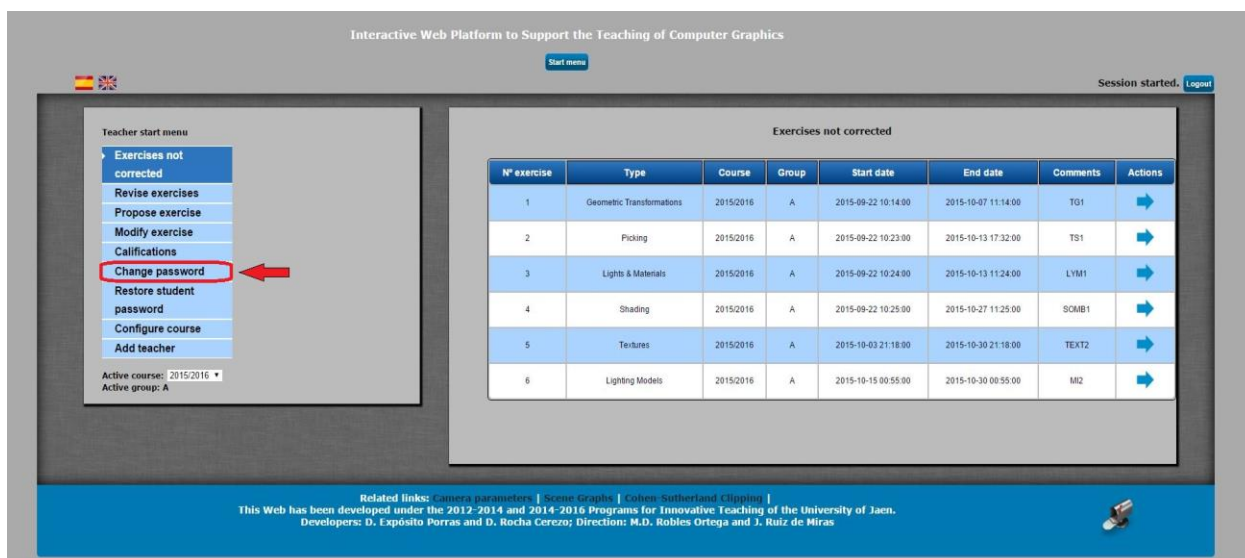


Figura 9.27. Pantalla del menú del profesor cuando se dispone a cambiar su contraseña.

Para realizar el cambio de contraseña satisfactoriamente el usuario tendrá que escribir su contraseña actual (por defecto su DNI), escribir la nueva contraseña y repetirla (*Figura 9.28*). A continuación podrá pulsar en el botón “Guardar contraseña”.

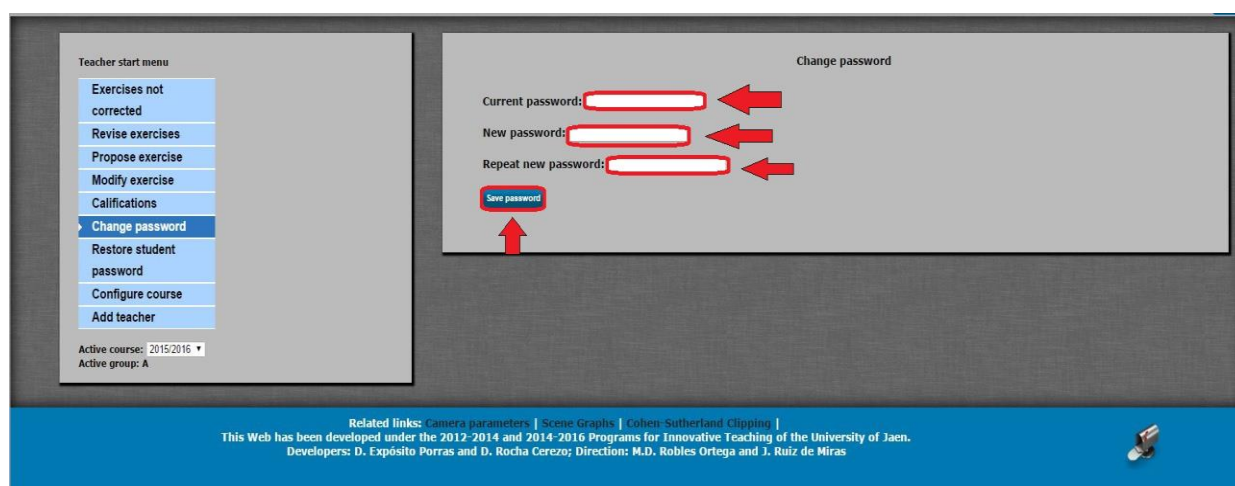


Figura 9.28. Cambiar contraseña de usuario.

9.9. RESTABLECER CONTRASEÑA DE ALUMNO

Esta acción sólo la podrán realizar los profesores administradores. Para restablecer la contraseña de un alumno a su DNI, el profesor tendrá que pulsar en la acción del menú correspondiente (*Figura 9.29*).

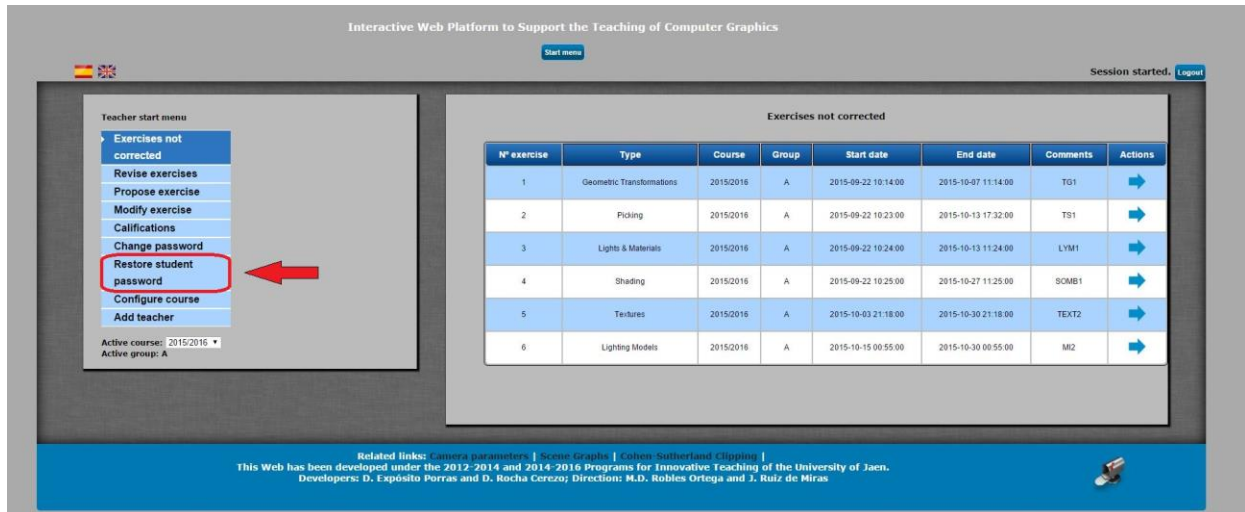


Figura 9.29. Pantalla del menú del profesor cuando se dispone a restablecer la contraseña de un alumno.

Para completar la acción simplemente tendrá que introducir el DNI del alumno y pulsar en el botón “Restablecer contraseña” (*Figura 9.30*).

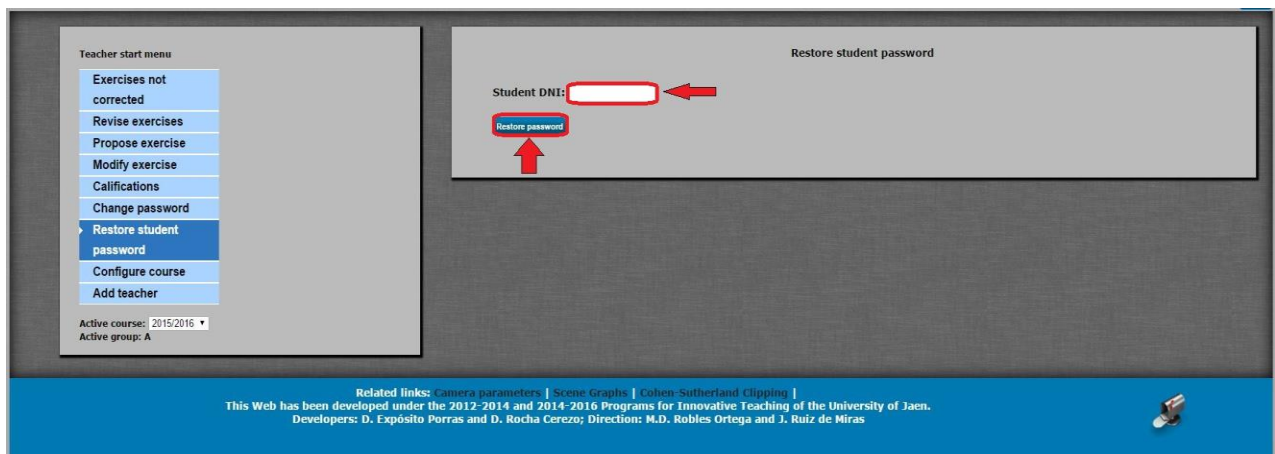


Figura 9.30. Restablecer contraseña de un alumno.

9.10. CONFIGURAR CURSO

Esta acción sólo la podrán realizar los profesores administradores. Para configurar un nuevo curso, el profesor tendrá que pulsar en la acción del menú correspondiente (*Figura 9.31*).



Figura 9.31. Pantalla del menú del profesor cuando se dispone a configurar un curso.

A continuación deberá pulsar en el botón para seleccionar el fichero “CSV” que desea importar (*Figura 9.32*). Se abrirá una ventana del sistema para seleccionar el fichero con el curso, grupo y los alumnos que se deseen insertar (*Figura 9.33*).

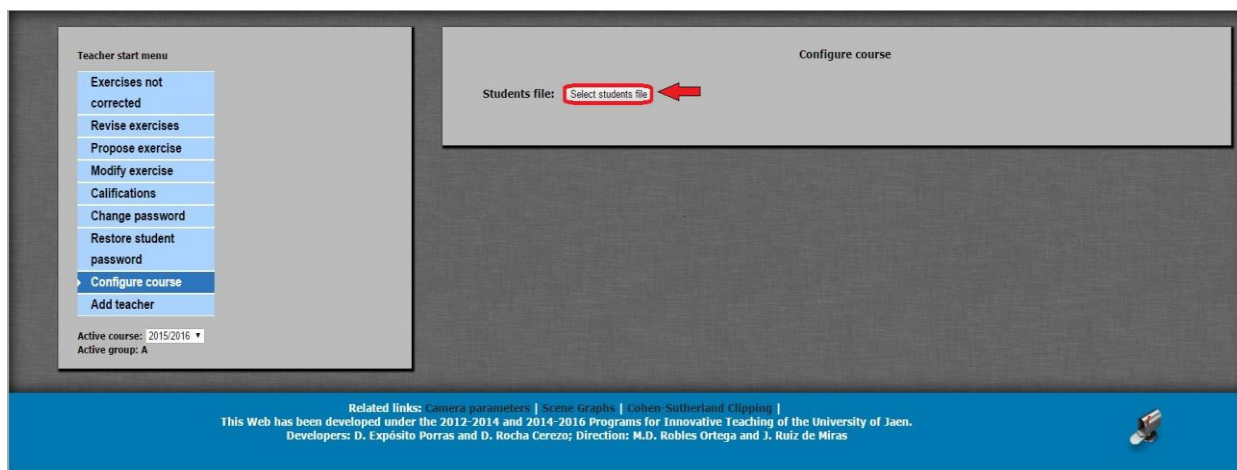


Figura 9.32. Seleccionar fichero con los alumnos del grupo.

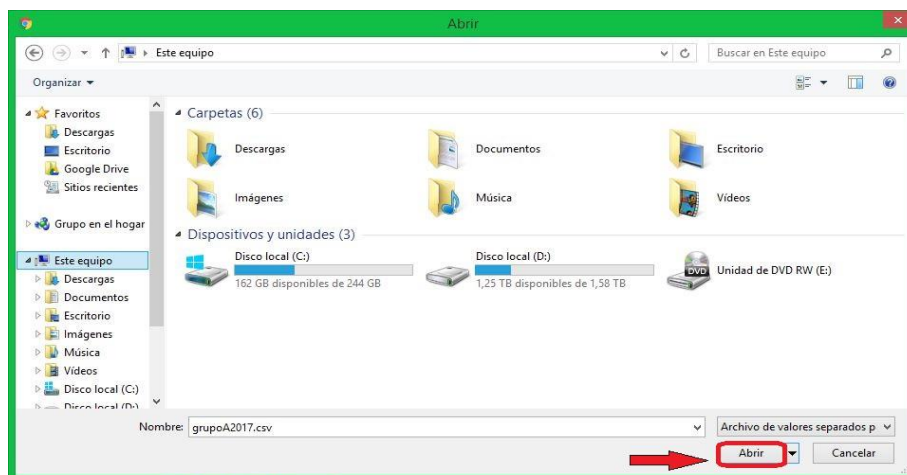


Figura 9.33. Abrir fichero con los alumnos del grupo.

Este fichero “CSV” deberá tener la siguiente estructura para poder realizar la importación de la información correctamente (Figura 9.34).

	DNI PROFESOR		NUMERO DE ALUMNOS		
CURSO	2017/2018				
GRUPO	A	12345678A	3		
ALUMNOS	53593174T	Daniel	Expósito	Porras	
	12345678P	Juan	González	Sánchez	
	12345678B	Miguel	García	Pérez	
	12345678D	Adrián	Expósito	Porras	

Figura 9.34. Estructura del fichero con el curso para configurar.

Por último el profesor tendrá que pulsar en el botón “Guardar curso” para almacenar el curso, grupo y los alumnos en el sistema (Figura 9.35).

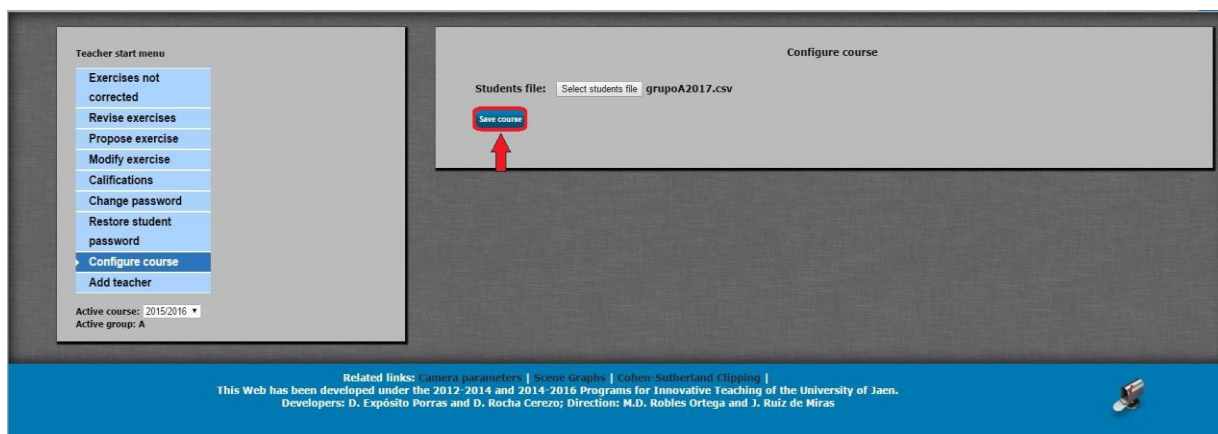


Figura 9.35. Guardar curso configurado en el sistema.

9.11. AÑADIR PROFESOR

Esta acción sólo la podrán realizar los profesores administradores. Para añadir un nuevo profesor, el profesor tendrá que pulsar en la acción del menú correspondiente (*Figura 9.36*).

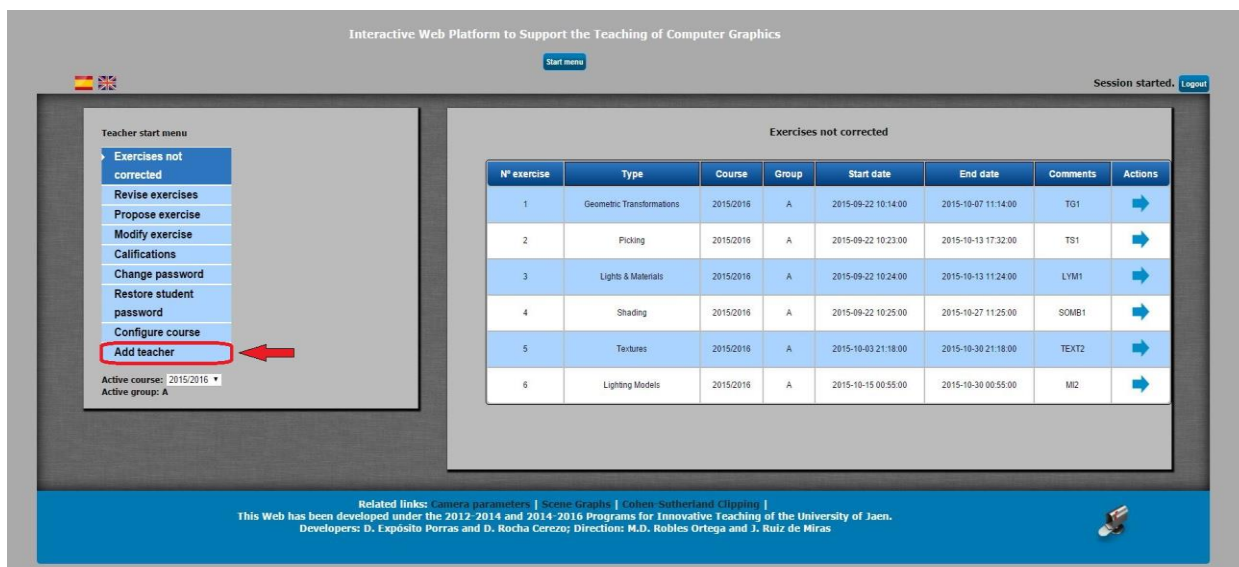


Figura 9.36. Pantalla del menú del profesor cuando se dispone a añadir un profesor.

Para añadir un nuevo profesor se tendrán que insertar el DNI, contraseña, contraseña repetida, nombre, primer apellido, segundo apellido e indicar si se trata de un profesor administrador. A continuación el profesor deberá pulsar el botón “Añadir profesor” para guardar el nuevo profesor en el sistema (*Figura 9.37*).

The screenshot shows the 'Add teacher' form. It includes input fields for DNI, Password, Repeat password, Name, Last name 1, and Last name 2. There is a radio button for 'Is admin' with options 'No' and 'Yes'. A red arrow points to the 'Add teacher' button. The 'Teacher start menu' is visible on the left, with 'Add teacher' highlighted.

Figura 9.37. Añadir nuevo profesor.

9.12. SELECCIONAR CURSO ACTIVO

Tanto el profesor como el alumno deberá tener un curso activo para poder interactuar con el sistema sin recibir la carga excesiva de ejercicios de los cursos previos. Para seleccionar el curso en el que el usuario vaya a realizar la interacción con el sistema, simplemente habrá que seleccionarlo en el selector correspondiente (*Figura 9.38*).

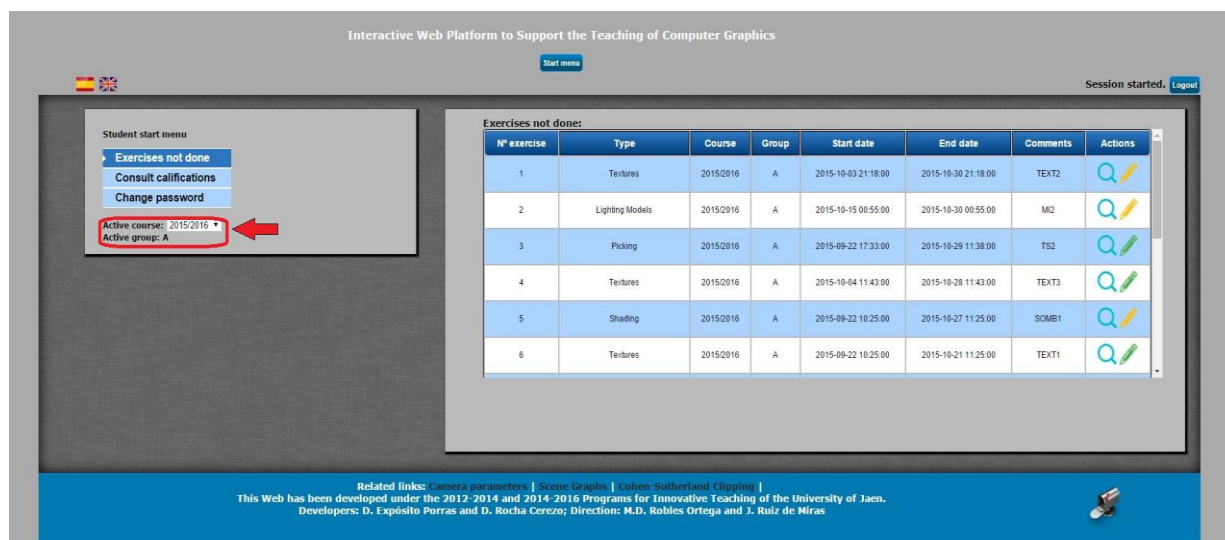


Figura 9.38. Seleccionar curso activo.

9.13. LISTA DE EJERCICIOS DEL ALUMNO

El alumno podrá realizar tres acciones distintas para cada ejercicio:

- Podrá realizar un ejercicio siempre que la fecha actual se encuentre entre la fecha de inicio y la fecha de fin del ejercicio y no haya sido previamente realizado.
- Podrá editar un ejercicio siempre que la fecha actual se encuentre entre la fecha de inicio y la fecha de fin del ejercicio y haya sido previamente realizado.
- Podrá ver el ejercicio siempre que la fecha actual sea posterior a la fecha de inicio del ejercicio.

Las acciones para cada ejercicio se encuentran a la derecha de la tabla y se podrá seleccionar una u otra opción para cada ejercicio y el sistema realizará la acción pertinente (*Figura 9.39*).



Figura 9.39. Listado de ejercicios del alumno.

En este caso se mostrará como realizar un ejercicio. Simplemente habrá que seleccionar el ejercicio y pulsar en la acción correspondiente. El sistema cargará el applet del ejercicio seleccionado con los parámetros establecidos por el profesor. El alumno tendrá que modificar los parámetros del applet para conseguir obtener un resultado similar al profesor (Figura 9.40).

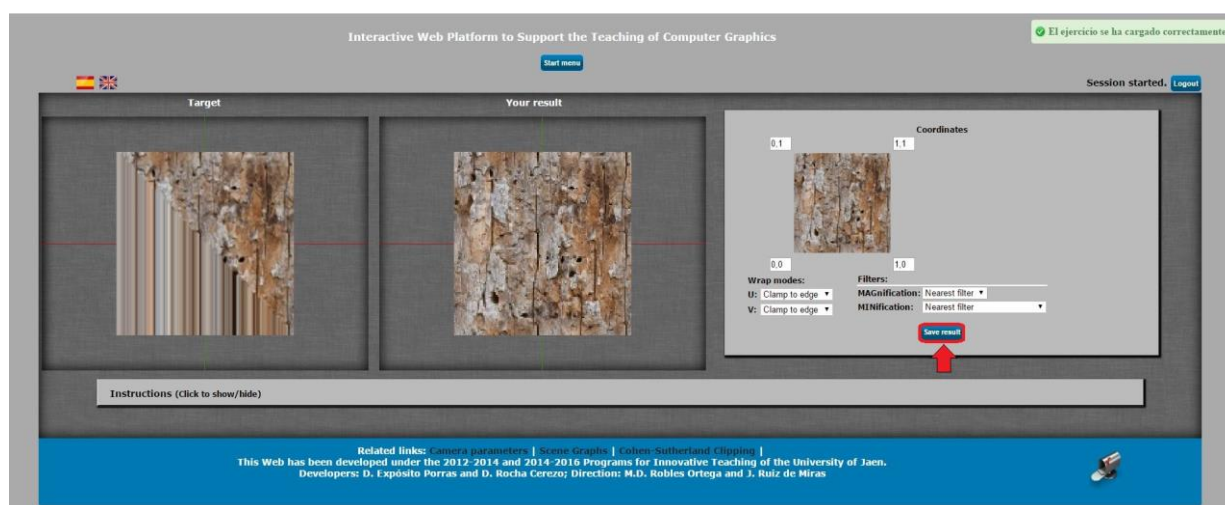


Figura 9.40. Realizar ejercicio.

En todos los applets existe algún indicador de ayuda para que el alumno sepa si el ejercicio está correcto. En este caso la textura parpadeará cuando consiga un resultado parecido al resultado obtenido por el profesor. Una vez haya terminado de modificar el ejercicio podrá pulsar en el botón “Guardar resultado” para almacenar los cambios en el sistema y terminar de realizar el ejercicio.

Para editar un ejercicio el proceso sería el mismo que para realizarlo, excepto que en el applet se cargaría la configuración que guardó el alumno además de los parámetros del profesor.

Por último para ver un ejercicio simplemente habría que pulsar en la acción correspondiente y el sistema cargará el applet del ejercicio configurado por el profesor sino se ha realizado y con los parámetros del profesor y del alumno si ya ha sido realizado.

9.14. CONSULTAR CALIFICACIONES DEL ALUMNO

Para consultar sus calificaciones, el alumno tendrá que pulsar en la acción del menú correspondiente (Figura 9.41).

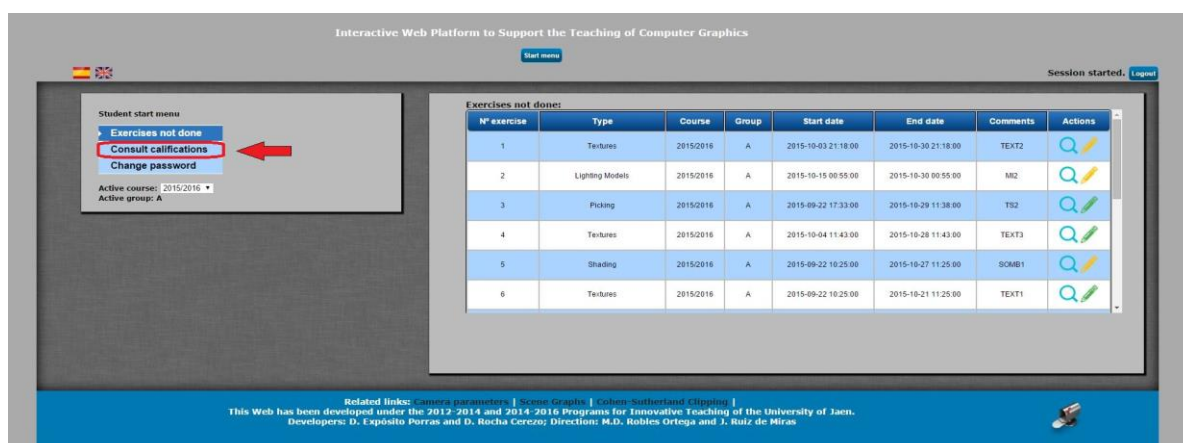


Figura 9.41. Pantalla del menú del alumno cuando se dispone a ver sus calificaciones.

A continuación podrá seleccionar el curso y grupo de los que desea consultar las calificaciones y el sistema rellenará la tabla con dichas calificaciones (Figura 9.42).

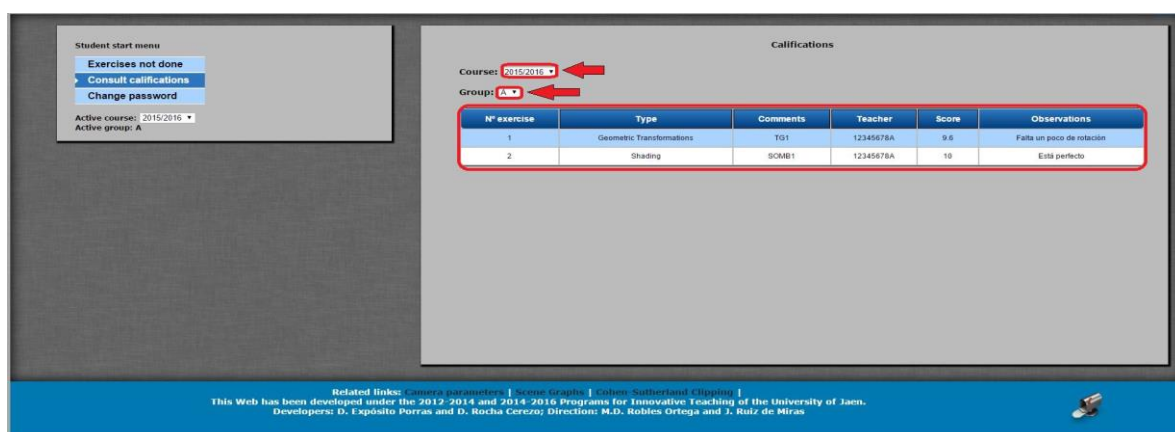


Figura 9.42. Consultar calificaciones del alumno.

9.15. CERRAR SESIÓN

Para cerrar la sesión actual simplemente habrá que pulsar en el botón “Logout” (Figura 9.43).

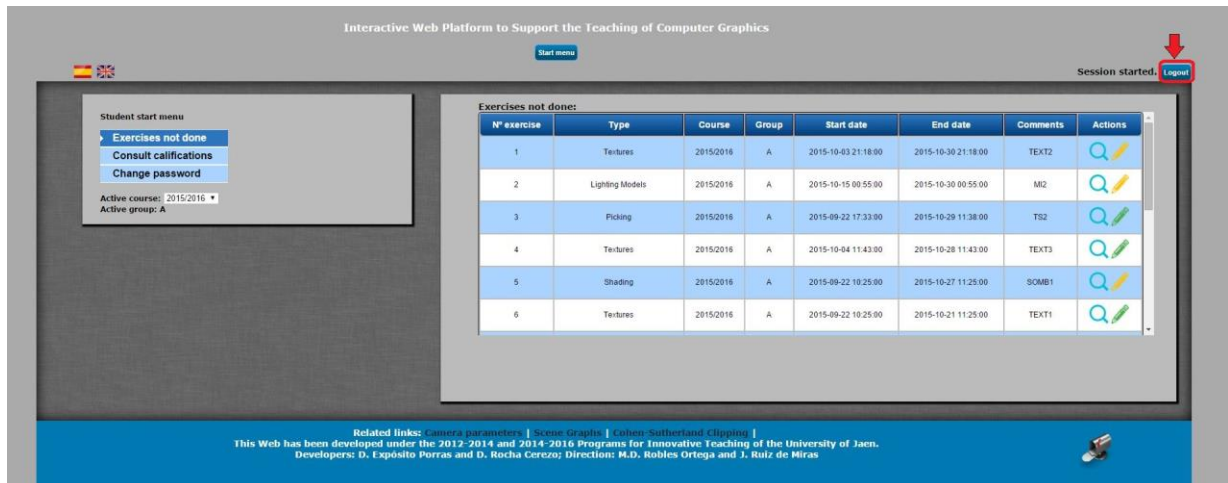


Figura 9.43. Cerrar sesión desde el menú del alumno.

Una vez cerrada la sesión el sistema cargará el menú de inicio de demostración (Figura 9.44) y se mostrará la notificación indicando que se ha cerrado la sesión.

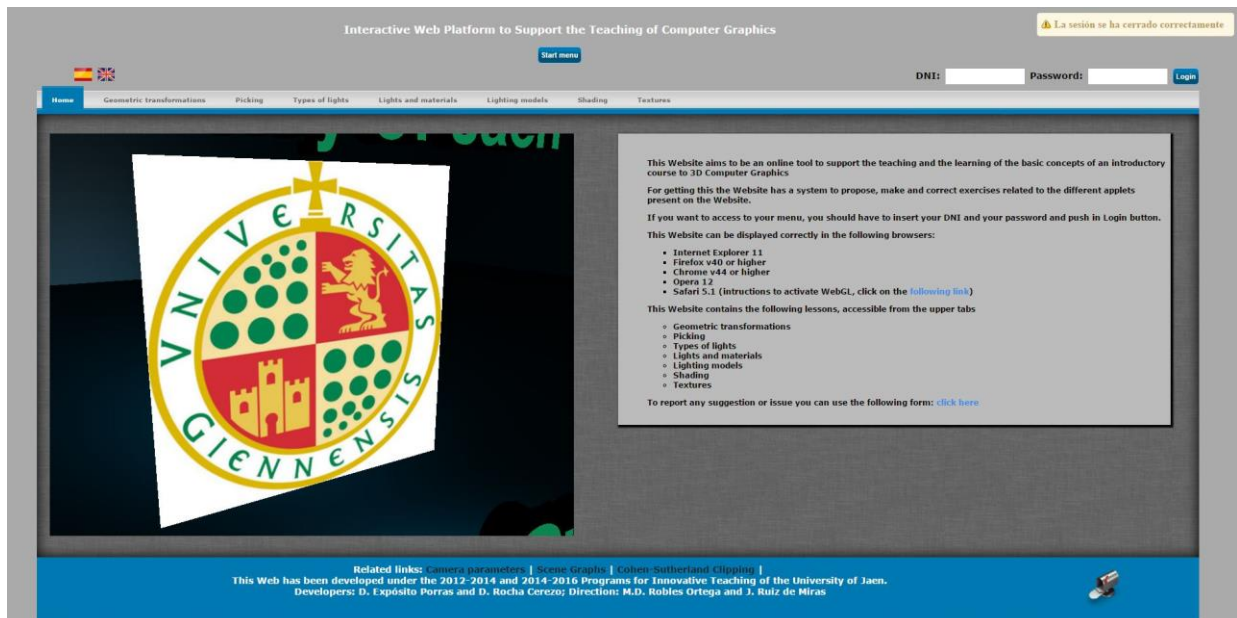


Figura 9.44. Sesión cerrada correctamente.

10. ANEXO B: MANUAL DE INSTALACIÓN

En este anexo se detalla el contenido incluido en el CD adjunto al proyecto y todo lo necesario para que el sistema Web funcione correctamente en la URL donde se encuentra alojado. Además se indicarán los pasos para ejecutar el sistema Web de forma local.

10.1. CONTENIDO DEL CD

- Carpeta **“Documentación”**: contiene la memoria del proyecto en formato PDF, con el nombre “Memoria.pdf”. Se incluyen también varias carpetas:
 - La carpeta **“Casos de Uso”**: contiene todos los casos de uso de la aplicación que fueron explicados en la sección 2.4.1.
 - La carpeta **“Diagramas de Clases”**: incluye los diagramas de clase de la aplicación que fueron incluidos en la sección 4.2.4 de esta memoria y los distintos diagramas de los applets.
 - La carpeta **“Diagramas de Secuencia”**: incluye los diagramas de secuencia que se expusieron en la sección 4.2.5. para poder consultarlos individualmente.
 - La carpeta **“Diseño de la Base de Datos”**: contiene las imágenes del diseño conceptual, lógico y físico de la base de datos que fue explicado en la sección 4.1.
 - La carpeta **“Pantallas”**: incluye las pantallas de la aplicación mostradas en la sección 4.2.2 de este documento y las pantallas que no se pudieron mostrar en la misma.
 - La carpeta **“Storyboards”**: contiene los storyboards explicados en la sección 4.2.3 de esta memoria para poder consultarlos individualmente.
- Carpeta **“Desarrollo”**: incluye la subcarpeta **“SistemaApoyoDocente”** donde se encuentra el código fuente del sistema Web. Esta subcarpeta contiene:
 - Los ficheros HTML de la aplicación.

- En la carpeta “js” se encuentran los ficheros JavaScript de la misma. Los scripts del MVC de los applets y los menús del alumno y profesor se encuentran en esta carpeta.
- En la carpeta “filesPHP” se encuentran los ficheros PHP de la parte servidor de la aplicación.
- En la carpeta “css” se encuentran los ficheros CSS de la aplicación.
- La carpeta “.openshift” contiene la configuración necesaria para alojar el sistema Web en el PaaS Openshift.
- La carpeta “.git” contiene la configuración del repositorio Git utilizado durante el desarrollo de la aplicación con SourceTree.

Además se incluye otra subcarpeta “**Base de Datos**” dentro de la carpeta “**Desarrollo**” que contiene el fichero de creación de la base de datos con la información necesaria para testear la aplicación.

10.2. INSTALACIÓN EN LOCAL USANDO XAMPP

La aplicación se ha desarrollado con el editor de código Sublime Text y se ha utilizado Xampp para ejecutar en local la aplicación y realizar las pruebas.

Se puede descargar Xampp en la siguiente dirección URL:

<https://www.apachefriends.org/es/index.html>

Una vez instalado, simplemente habría que insertar la carpeta “**SistemaApoyoDocente**” indicada en el contenido del CD en la subcarpeta “htdocs” de la carpeta raíz del programa Xampp.

Para ejecutar la aplicación en local simplemente habría que abrir el programa Xampp e iniciar los servicios Apache y MySQL como se puede ver en la *Figura 10.1*.

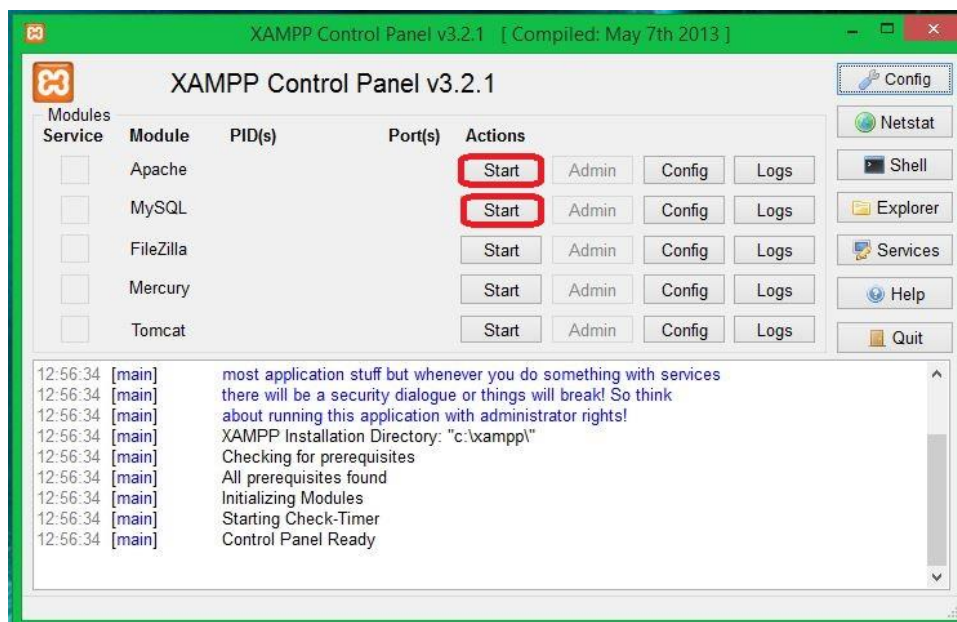


Figura 10.1. Iniciar Xampp.

Una vez se hayan iniciado ambos servicios el panel de control de Xampp debería quedar igual que en la *Figura 10.2*.

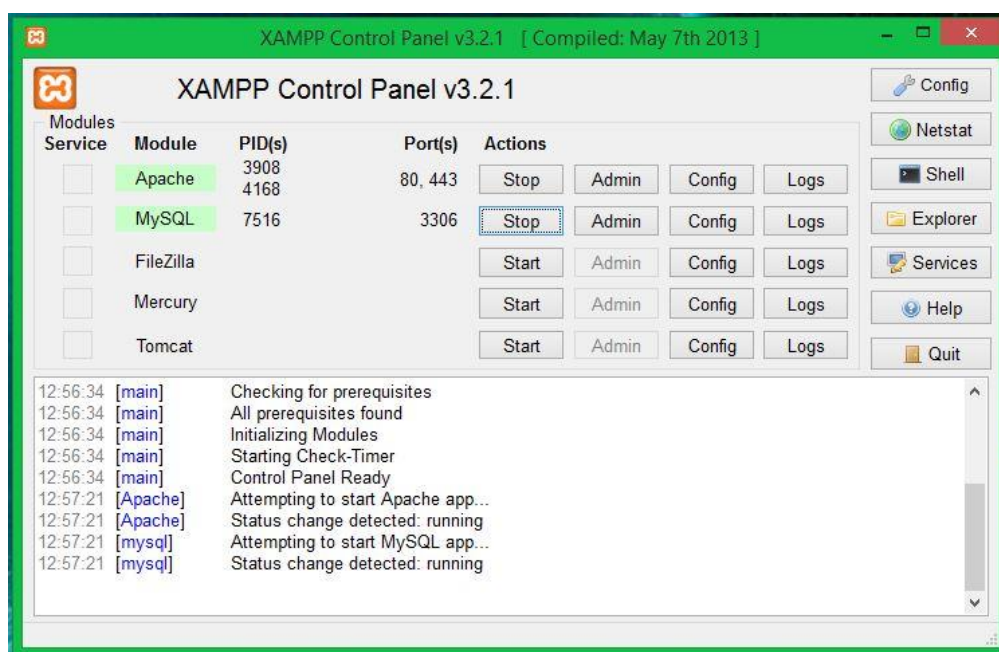


Figura 10.2. Servicios Apache y MySQL iniciados.

Una vez realizado esto el sistema Web estará funcionando en la URL "localhost/SistemaApoyoDocente". Sin embargo la aplicación requiere de una base de datos por tanto para poderla utilizar hace falta configurarla. Para ello hay que insertar en el navegador la URL "localhost/phpmyadmin" e importar el fichero

incluido en la carpeta “Base de Datos” del CD adjunto como se puede ver en la *Figura 10.3*.

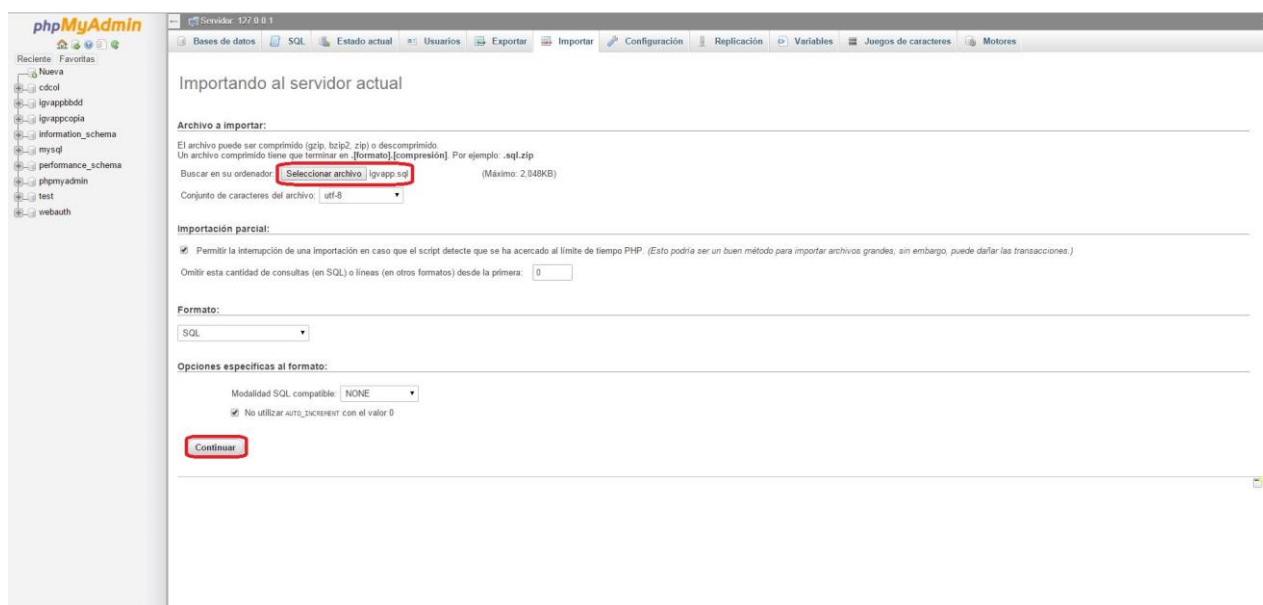


Figura 10.3. Importar base de datos con PHPmyadmin.

La base de datos se creará con un profesor para poder acceder a la aplicación cuyo DNI es 12345678A y contraseña “123”. Con este profesor se podrá añadir nuevos profesores y nuevos alumnos desde la aplicación.

10.3. INSTALACIÓN EN EL PAAS OPENSIFT

Actualmente la aplicación se encuentra funcionando en la siguiente URL:

<http://igvapp-ujatfg.rhcloud.com/>

Sin embargo se ha tenido que realizar una instalación previa y aplicar una configuración predeterminada en la cuenta de Openshift para alojar correctamente la aplicación.

Antes de nada hay que crearse una cuenta en la página Web oficial de Openshift y acceder a ella. Habrá que crear una nueva aplicación desde dicha página como se muestra en la *Figura 10.4*.

Based On	PHP 5.4 Cartridge PHP is a general-purpose server-side scripting language originally designed for Web development to produce dynamic Web pages. Popular development frameworks include CakePHP, Zend, Symfony, and Code Igniter. http://www.php.net ☆ OpenShift maintained 🛡️ Receives automatic security updates
Public URL	http:// SistemaApoyoDocente -ujatfg.rhcloud.com OpenShift will automatically register this domain name for your application. You can add your own domain name later.
Source Code	Optional URL to a Git repository Branch/tag We'll create a Git code repository in the cloud, and populate it with a set of reasonable defaults. If you provide a Git URL, your application will start with an exact copy of the code and configuration provided in this Git repository.
Gears	small Gears are the application containers running your code. For most applications, the small gear size provides plenty of resources. You can also upgrade your plan to get access to more gear sizes.
Cartridges	PHP 5.4 Applications are composed of cartridges - each of which exposes a service or capability to your code. All applications must have a web cartridge.
Scaling	Scale with web traffic OpenShift automatically routes web requests to your web gear. If you allow your application to scale, we'll set up a load balancer and allocate more gears to handle traffic as you need it.

Figura 10.4. Crear la aplicación en Openshift.

Hay que seleccionar los complementos PHP 5.4 y MySQL 5.5 para el correcto funcionamiento de la aplicación. A continuación es necesario instalar Ruby y Git siguiendo los pasos indicados en el tutorial de instalación en la documentación de Openshift:

<https://developers.openshift.com/en/getting-started-windows.html>

A continuación es necesario importar el fichero de creación de la base de datos incluido en el CD como se ha explicado en la sección anterior de este anexo. En este caso habrá que insertar la clave pública desde la Web de Openshift como se puede ver en la *Figura 10.5*.

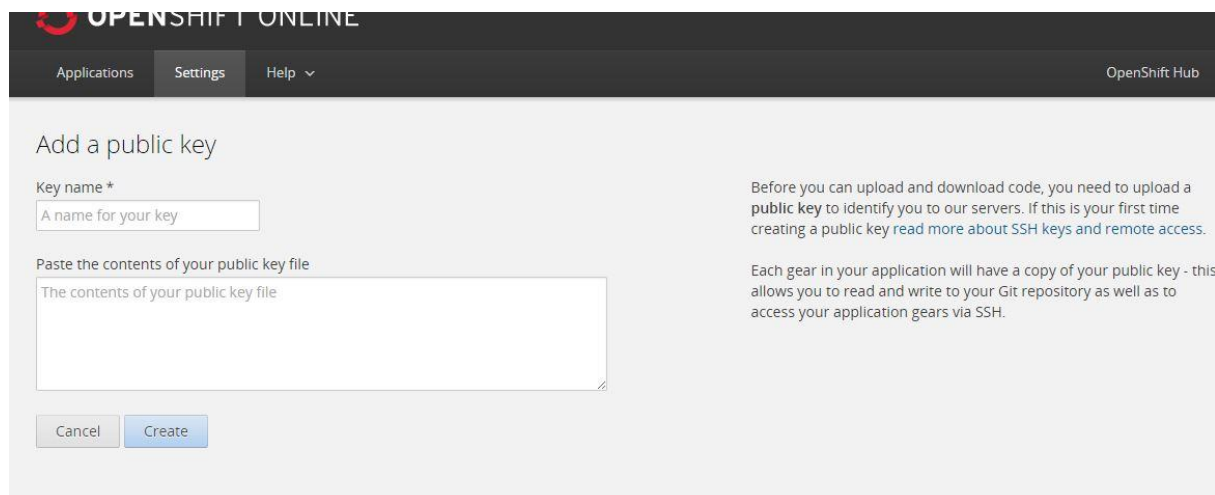


Figura 10.5. Añadir la clave pública a la cuenta de Openshift.

Una vez realizado esto habrá que acceder a la consola de ruby e instalar la “gem rhc” tal y como viene indicado en el tutorial indicado previamente. Desde la consola se podrá acceder a el complemento MySQL de la aplicación y se podrá importar el archivo que contiene la base de datos incluido en la ruta “Desarrollo\Base de Datos\” del CD adjunto a este proyecto.

Por último será necesario subir el repositorio de la aplicación incluido en la carpeta “Desarrollo” mediante el uso de Git como viene indicado en el tutorial.

Siguiendo estos pasos el sistema Web estará correctamente alojado en la dirección URL indicada en la creación de la aplicación como se puede ver en la *Figura 10.4*.