

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERÍA DEL SOFTWARE

**DESARROLLO DE COMPONENTES WEB DE EJERCICIOS DE
LECTOESCRITURA**

(DEVELOPMENT OF WEB-BASED EXERCISES OF LITERACY)

Realizado por

JUAN JOSÉ CARRERA DÍEZ

Tutorizado por

EDUARDO GUZMÁN DE LOS RISCOS

Departamento

LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN

UNIVERSIDAD DE MÁLAGA

MÁLAGA, JUNIO 2017

Resumen

El desarrollo de la aplicación web “ActiLine” tiene como principal objetivo la facilitación de una herramienta con la que tanto profesores como alumnos puedan ejercitar su sistema educativo. Así entonces, como se puede deducir, la aplicación contará con dos perfiles claramente diferenciados, el de profesor y el del alumno.

El profesor podrá elaborar ejercicios que están disponibles en la propia plataforma, estos ejercicios serán generados a partir de entradas sencillas introducidas por el mismo profesor, a su vez, este mismo podrá visualizar las soluciones a estos ejercicios o incluso eliminar cualquier ejercicio con sus respectivas respuestas por parte del alumnado.

El alumno, podrá visualizar los ejercicios disponibles pudiendo seleccionar cualquiera de ellos y realizar cualquiera que elija.

Todo lo descrito anteriormente podrá ser visualizado tanto por el profesor como el alumno en cualquier dispositivo, ya sea ordenador de sobremesa, ordenador portátil o tablet.

En resumidas cuentas, el desarrollo de esta aplicación se ha centralizado en la gestión de las tareas educativas tanto del lado del profesor como del alumno y a diferencia de aplicaciones actuales existentes, las soluciones generadas por el sistema se centran en dar una explicación detallada paso a paso de cada ejercicio.

Palabras clave

JQuery, Javascript, JSP, Servlet, Diseño, Responsive

Summary

The main object in the development of the web application “ActiLine” has been to supply an educational tool with two different profiles: the teacher profile and the student profile.

On the one hand, in this application a teacher can add an exercise. This type of exercise requires simple inputs as two numbers for example. Besides, teacher can view exercise solutions or deletes an exercise, if a teacher deletes an exercise, all the solutions of that exercise will be deleted too.

On the other hand, a student can visualize all the exercises that are created by teachers, and they can also solve these exercises.

All users can view the functionality of the web application in devices as desktop PC, laptop or tablet (mobile devices with supported resolution).

In comparison to other educational environments, the solution generated by student are explained step by steps in detail.

Keywords

Javascript, JQuery, JSP, Servlet, Design, Responsive

ÍNDICE

INTRODUCCIÓN	9
1.1. Objetivos	9
1.2. Contenido de la memoria en capítulos	10
TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS	11
2.1. Resumen de tecnologías empleadas	11
2.2. Bootstrap	14
2.3. JQuery	15
2.4. GitHub	15
ANÁLISIS DE REQUISITOS	17
3.1. Análisis	17
3.2. Requisitos	18
DISEÑO	23
4.1. Descripción de los casos de uso	23
4.2. Modelo de clases	29
IMPLEMENTACIÓN	33
5.1. Primera iteración : Elementos Comunes.	34
5.1.1. Estructura global, pie de página y cuerpo	34
5.1.2. Menús profesor y alumno	35
5.1.3. Creación de mensajes informativos	35
5.1.4. Pruebas y corrección de errores	36
5.2. Segunda iteración : Bloque profesor	36
5.2.1 Bloque profesor	38
5.2.2. Bloque alumno	44
CONCLUSIONES	47
6.1. Resultado final y conclusiones.	47
6.2. Futuras mejoras	48
BIBLIOGRAFÍA	49
Anexo : Manual de Usuario	51
Introducción	51
Perfil Profesor	51
Pantalla principal	51
Acciones disponibles para perfil Profesor	52

Acciones detalladas para perfil Profesor	53
Perfil Alumno	57
Pantalla principal	57
Acciones disponibles para el perfil Alumno	58
Acciones detalladas para perfil Alumno	59

INTRODUCCIÓN

El desarrollo de esta aplicación viene a responder a necesidades que pueden surgirles a algunos centros educativos respecto tanto de la enseñanzas de sus alumnos como a la tarea de sus profesores en cuanto a su enseñanza, adecuando estos dos aspectos a su vez a los tiempos actuales.

He aquí una enumeración de necesidades encontradas:

- Alumnos que necesiten realizar ejercicios fuera de su horario de clase, incluidos aquí a los que propiamente no puedan asistir a esta debido a incapacidad o a enfermedad temporal.
- Alumnos que deseen reforzar su aprendizaje con ejercicios adicionales por parte del profesorado.
- Profesores que deseen cambiar su método de asignación de tareas, teniendo como ventaja adicional el ahorro de materiales para la realización de ejercicios, al realizarse todo sobre la plataforma.

A diferencia de algunas otras aplicaciones estudiadas como portales desarrollados con JOOMLA o tecnologías similares, los ejercicios tendrán más detalles para la comprensión de estos tanto por parte del alumno como del profesor.

1.1. Objetivos

La finalidad que persigue esta aplicación viene a resolver las necesidades anteriormente mencionadas en la introducción.

Para ello se va a dividir en un conjunto de tareas el desarrollo del mismo, tareas que pasarán desde el desarrollo de los controladores encargados de consultar la información necesaria en la aplicación sobre los ejercicios posibles a realizar al alumnado, hasta el desarrollo mismo de la interfaz que será lo que visualice tanto profesores como alumnos.

Estas tareas se dividirán en las siguientes:

Preparación del entorno de trabajo y creación de la máquina virtual y configuración de la plataforma de desarrollo GitHub para copias de seguridad.

Creación de la capa de datos para el alojamiento de información como los datos de los usuarios, así como sus permisos.

Creación de la capa de control, la cual se encargará de tratar la información que se reciba desde la capa de datos de la interfaz de usuario, además del uso de sesiones en la misma para la identificación del usuario.

Creación de la interfaz de usuario, la cual además de contener los estilos de la misma, contendrá también el tratamiento de los datos introducidos por el usuario evitando que sean estos erróneos o insuficientes.

Revisión de todas las tareas anteriores para su optimización.

1.2. Contenido de la memoria en capítulos

A continuación se detallan los capítulos en los que se divide la presente memoria, sin mencionar el actual. La razón por la que ha sido estructurada del siguiente modo se debe a la forma en la que se ha realizado el proyecto, es decir, pasando desde el nivel más abstracto y alto nivel (como es la obtención de requisitos) hasta el bajo nivel donde podemos encontrar especificaciones de programación y pruebas.

1.2.1. Tecnologías utilizadas

En este capítulo, primeramente, se resumirá buena parte de las tecnologías y herramientas que se han utilizado y, posteriormente, se especificará con más detalle algunas de ellas por su especial importancia para resaltar así aspectos que puedan resultar útiles al conocimiento del usuario.

Cabe destacar ante todo *Bootstrap*, por la facilidad que nos brinda para realizar aplicaciones *responsive*, *JQuery*, por la sencillez que demuestra en la manipulación de elementos DOM y datos introducidos por el usuario, *Java Servlets*, por la capacidad de manipulación de datos que nos ofrece, y por último, *GitHub*, por su plataforma de desarrollo.

1.2.2. Análisis de requisitos

En esta segunda sección primeramente se va a establecer mediante la toma de requisitos las necesidades que debería de cubrir el proyecto, estos se han ido especificando mediante las reuniones realizadas con el tutor.

Así entonces, una vez obtenidos claramente estos requisitos, se procedió a la definición de los casos de uso, con sus respectivas pre-condiciones y post-condiciones y sus actores, todo ello vendrá agrupados en los diferentes escenarios que puedan darse.

Una vez terminado lo anterior, se procederá con el diagrama de clases, y con ello se habrá acabado con la parte “abstracta” para dar paso a la implementación.

1.2.3. Implementación y pruebas.

Una vez acabada la fase de análisis de requisitos, ya tenemos una definición detallada de todo lo que debe hacer el sistema.

Durante esta fase de desarrollo se detalla también los problemas que se han encontrado y el método que se ha seguido para la resolución de estos junto a sus pruebas.

TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS

En este apartado se explican primero la mayoría de las herramientas que se han utilizado en el desarrollo del proyecto, posteriormente, se describen con más detalles las más importantes.

2.1. Resumen de tecnologías empleadas

A continuación se va a enumerar dividido en tres secciones las herramientas usadas para el desarrollo de la aplicación web, cabe destacar que este proyecto ha sido desarrollado bajo licencias de software libre o gratuito, por ello todas las herramientas que se mencionan pueden obtenerse sin coste alguno.

Preparación del entorno



VMWare Player

VMware Workstation Pro y VMware Workstation Player son el estándar del sector para la ejecución de varios sistemas operativos como máquinas virtuales en un único PC. Miles de profesionales de TI, desarrolladores y empresas utilizan Workstation Pro y Workstation Player para mejorar su agilidad, productividad y seguridad todos los días.

Para evitar posibles problemas con un sistema operativo nativo en un PC se ha optado por desarrollar en una máquina virtual, realizando copias de seguridad de esta periódicamente. En lo referente al sistema operativo, este carece de importancia dado que la herramienta que se usa en el sistema operativo es multiplataforma, por tanto se podría haber desarrollado tanto en Windows como en Linux, MacOS etc.

Software



Netbeans

NetBeans es un entorno de desarrollo integrado libre, concebido principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo.

Cabe mencionar que este entorno fue el usado para el desarrollo de aplicaciones en varias asignaturas por lo que es un entorno ya familiar.



GitHub

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git.

Este se ha usado fundamentalmente debido a que almacenar versiones de código posee bastantes ventajas, entre ellas la seguridad de no perder este mismo y ante errores poder recuperar una versión correcta y comenzar desde esta.



Creately es una herramienta online que funciona desde cualquier navegador web (ya que usa flash), para crear cualquier tipo de diagramas (en este caso de tipos UML).



Notepad++

Notepad++ es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación.

Este ha sido usado como apoyo al entorno netbeans en varios casos como por ejemplo la reutilización de código.



H2 Database administration

H2 Database administration es un administrador de base de datos para la propia base de datos que usamos en el proyecto.

Este ha sido un apoyo debido a la rapidez en que se pueden tratar los datos almacenados en modo visual y no en sentencias SQL.

Servidor



GlassFish

GlassFish es un servidor de aplicaciones de software libre desarrollado por Sun Microsystems, que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación.

Está integrado con el entorno Netbeans, sin embargo, es relevante mencionarlo debido a su importancia en el proyecto.

Librerías



La librería Java Servlets (servlet-api.jar) nos aporta todo lo necesario para trabajar con esta tecnología en cuanto a nuestro modelo, específicamente en la parte del controlador, los cuales se encargan de recoger las peticiones que hacen los usuarios desde la vista, tratar los datos y devolver una respuesta acorde.



jQuery es una biblioteca multiplataforma de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, gestionar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web.



Bootstrap

Bootstrap es un framework o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales.

2.2. Bootstrap

Bootstrap es básicamente un conjunto de hojas de estilos que vienen a implementar una variedad de componentes; esto nos hace tener la posibilidad asimismo de poder modificar las librerías propias de Bootstrap para seleccionar únicamente los componentes que deseemos y así adaptarlo a nuestras necesidades.

Estos componentes que nos brinda Bootstrap son modificaciones de los propios de HTML5 como pueden ser inputs, botones, menús, etc ...

Entre las ventajas que se han identificado por el uso de esta herramienta cabe destacar:

- Selección de componentes que deseemos, ahorrando así tiempo de carga y espacio en el proyecto.
- Una vez se familiariza con la herramienta, se hace sencillo el desarrollo de una aplicación responsive, lo cual a veces puede resultar tedioso, sin embargo, Bootstrap con su metodología de clases nos facilita bastante la tarea.
- Estructura sencilla, con su sistema organizado de columnas.
- Ventanas modales fáciles de implementar, que junto a jQuery también hacen sencilla esta tarea de informar al usuario.
- Fácil modificación de los componentes de HTML5 para adaptarlos a Bootstrap, con su mencionada implementación de clases en estos.

Por todo ello Bootstrap nos ofrece la posibilidad de tener una herramienta que nos facilite unas buenas prácticas conforme a la forma de implementar nuestro proyecto, evitando algunos problemas como puede ser por ejemplo discrepancias en la forma en la que trabaja cada uno (hablando teóricamente en un grupo de trabajo).

Actualmente dicha herramienta puede encontrarse en multitud de aplicaciones web, si bien no por sí misma, aplicada junto a otra tecnología como puede ser Wordpress.

2.3. JQuery

JQuery es una librería que permite interactuar con los elementos HTML de la página (pudiendo cambiar su aspecto, comportamiento...). Esta tecnología emplea como base JavaScript, por lo cual son totalmente compatibles.

En el desarrollo de la aplicación se ha implementado esta librería para el manejo de los elementos junto al propio JavaScript nativo para algunos casos.

La principal ventaja reside en la forma en la que se manejan los elementos, su facilidad para aplicar lo que queramos mediante sus *selectores* lo cual facilita más la tarea de cómo lo hace JavaScript.

Actualmente JQuery se encuentra implementado en la mayoría de aplicaciones web por lo que se puede localizar en prácticamente cualquier sitio web.

2.4. GitHub

Se menciona así mismo también GitHub debido a su plataforma para alojar cualquier proyecto web compatible como es nuestro caso.

GitHub tiene una serie de ventajas importantes, las cuales se enumeran a continuación:

Control de versiones: con esto podemos volver a una versión anterior o comparar con una versión correcta para apoyarnos en esta misma.

Control de usuarios que han subido una versión: por cada fichero queda registrado quién lo ha modificado y qué ha hecho.

Herramienta para control de trabajo colaborativo: con ello los usuarios no se “molestan” a la hora de realizar su implementación, ya que la herramienta detectará las discrepancias que se puedan suceder.

Actualmente, Github forma parte mayormente en proyectos de software libre.

ANÁLISIS DE REQUISITOS

3.1. Análisis

Primeramente, se va a describir el entorno en que se enmarca la aplicación y la finalidad que se persigue con el desarrollo de este proyecto.

En los centros educativos, por diversas razones, no se pueden cubrir totalmente los contenidos que se pretenden tanto por el profesor como por el alumno.

Entre estos casos podemos encontrarnos con los siguientes:

- Profesores que pretendan reforzar el aprendizaje de sus alumnos con ejercicios en los que haya todo tipo de detalles o que para ahorrar tiempo en las aulas para emplearlo en otras labores educativas puedan crear estas tareas en un mínimo de tiempo además del ahorro en material que ello supone.
- Alumnos que deseen realizar ejercicios pero no puedan por alguna incapacidad o enfermedad temporal.
- Alumnos que no hayan comprendido o no hayan podido asistir a clase de cómo se realiza alguna tarea, por lo cual deseen reforzar su aprendizaje

Por todo ello y con los datos recogidos en las reuniones con el tutor se ha recogido las siguientes necesidades:

Debemos tener claramente diferenciados el perfil de profesor y alumno en la aplicación.

Para el perfil del profesor en cuanto a las tareas, deben de generarse a partir de simples entradas, con lo cual se deja a generar por parte del sistema las explicaciones y los pasos que debe seguir en la resolución de esta tarea.

También para el perfil del profesor, debe poder visualizar el ejercicio que ha introducido como las soluciones de cada uno de sus alumnos a la vez que posibilitar la opción de eliminar cualquiera de estos ejercicios como sus soluciones asociadas.

Para el perfil del alumno, debe poder realizar cualquier ejercicio que haya introducido anteriormente el profesor, así como poder visualizar cualquier consejo que pueda servirle de ayuda mediante un cuadro de diálogo.

La aplicación se debe poder ver correctamente tanto en un ordenador personal

como en un dispositivo portátil como una tablet.

Así entonces a partir de todas estas especificaciones se ha deducido el listado de requisitos.

3.2. Requisitos

3.2.1. Requisitos funcionales

En esta sección se detallan los requisitos funcionales, los cuales se han generado a partir de la información que hemos ido obteniendo de las reuniones anteriormente mencionadas. Estos requisitos nos son útiles para identificar de forma clara cuáles son las acciones que debe poder realizar cada usuario en el sistema; por ello se ha dividido estos en dos perfiles, el de profesor y el de alumno.

Perfil	ID	Nombre	Descripción
Profesor	P01	Visualizar ejercicios disponibles	Un profesor podrá visualizar los ejercicios que se encuentran disponibles en la plataforma para poder elegir entre ellos.
	P02	Crear tarea	Un profesor podrá generar un ejercicio que haya sido previamente seleccionado.
	P03	Visualizar tareas creadas	Un profesor puede visualizar las tareas que ha generado anteriormente.
	P04	Eliminar tarea generada	Un profesor podrá eliminar una tarea generada de un ejercicio con las respectivas soluciones de usuarios.
	P05	Visualizar tareas-soluciones	Un profesor podrá visualizar el listado de las tareas de soluciones generadas por alumnos.
	P06	Visualizar solución tarea	Un profesor podrá visualizar una solución de una tarea que haya completado un alumno con la corrección sugerida por el sistema.

Perfil	ID	Nombre	Descripción
Alumno	A01	Visualizar tareas disponibles	Un alumno podrá visualizar las tareas previamente generadas por el profesor.
	A02	Realizar tarea	Un alumno podrá realizar una tarea que haya sido generada previamente por el profesor.

3.2.2 Requisitos no funcionales

Los requisitos que a continuación se detallan tratan acerca de condiciones y restricciones que deben de respetarse en el sistema, en las cuales puede ser posible que se informe al usuario cuando no se cumpla alguna de ellas o simplemente deben ser controladas por este mismo.

Categoría	ID	Descripción
Seguridad	RN01	El sistema debe de diferenciarse claramente entre el perfil de profesor y perfil de alumno.
Usabilidad	RN02	El sistema informará del resultado de la acción que realice el usuario a través de mensajes de diálogos al usuario.
Usabilidad	RN03	El sistema informará al usuario de cualquier dato erróneo introducido a través de mensajes de diálogos al usuario.
Usabilidad	RN04	El sistema controlará que en las tareas generadas no se puedan introducir datos erróneos por parte del alumno.
Usabilidad	RN05	El sistema controlará que el alumno no pueda empezar a realizar un ejercicio hasta que se haya efectuado la adecuación a la misma tarea correctamente.
Interfaz	RN06	El sistema tendrá las opciones diferentes de acceso para diferenciar las diferentes tareas disponibles.
Interfaz	RN07	El sistema mostrará las opciones diferentes de tipos de ejercicios disponibles para cada perfil bien diferenciados.
Accesibilidad	RN08	El sistema podrá usarse en cualquier navegador con las versiones más recientes.
Accesibilidad	RN09	El sistema debe poderse visualizar en cualquier dispositivo ya sea tableta o pc.

3.3. Actores

Se especifican a continuación los actores que intervienen en la aplicación y su función en este:

- Profesor: representa el usuario que puede generar las tareas para los alumnos, es decir, tendrá permisos de escritura sobre este mismo listado, además de poder visualizar soluciones o eliminar cualquier ejercicio dado de alta previamente.
- Alumno: representa el usuario con menos privilegios, este puede realizar las tareas que previamente ha introducido el profesor.
- Sistema: representa la máquina en donde tenemos albergada la aplicación.

3.4. Diagramas de casos de uso

A continuación se describen los diagramas de casos de uso, en los cuales se detallan las conexiones que tienen los actores con las correspondientes acciones que pueden realizar.

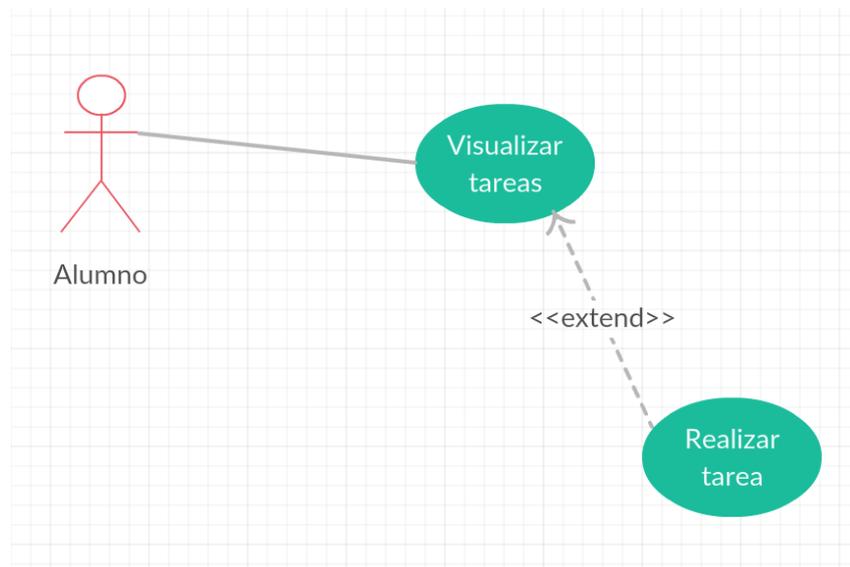


Diagrama correspondiente al actor con perfil "Alumno"

En este se puede observar cómo el usuario con perfil alumno tiene la posibilidad de visualizar las tareas disponibles con posibilidad de realizar una de ellas, por ello, como se refleja en el diagrama, es necesario visualizar estas tareas antes de poder realizar una de ellas.

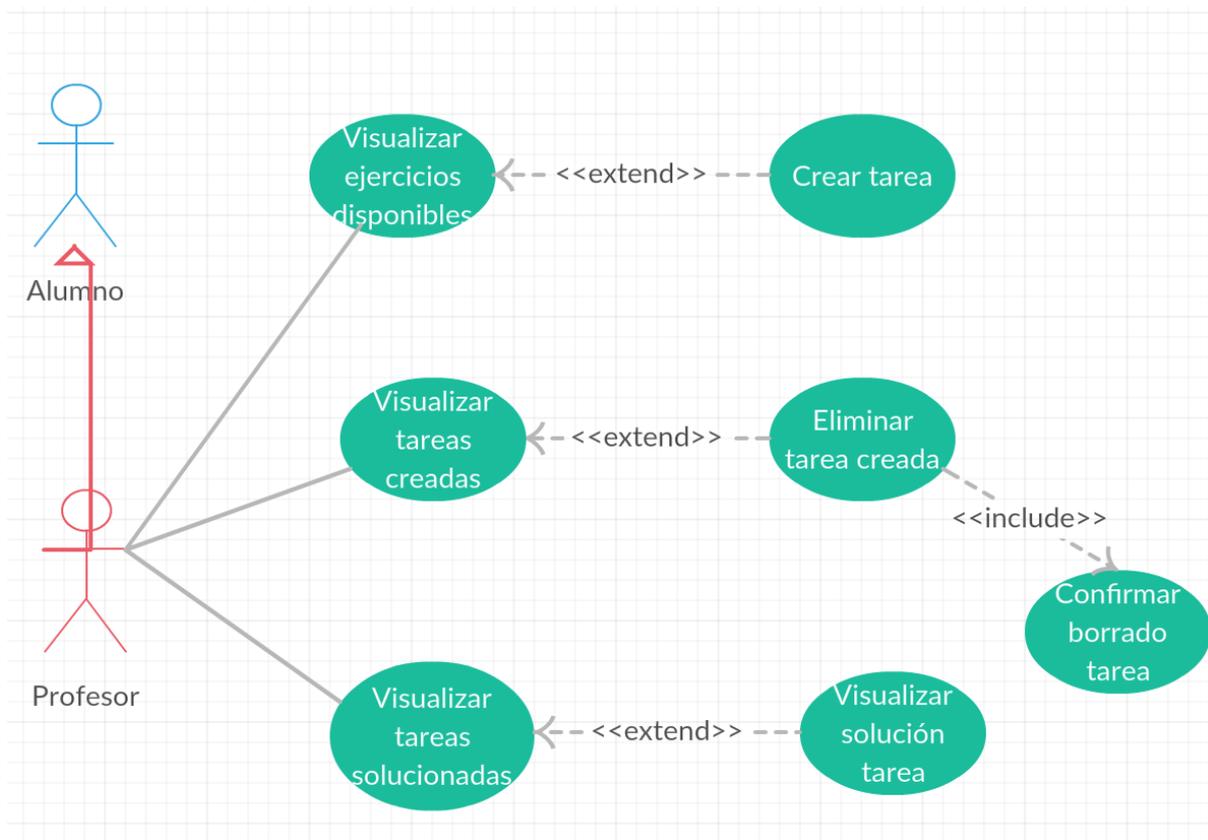


Diagrama correspondiente al actor Profesor

En este se puede observar que un profesor puede realizar tareas como un alumno, esto se ha realizado de este modo por si un profesor quiere verificar su propia tarea generada.

Por otra parte, también se puede ver en este diagrama cómo un profesor puede realizar otras acciones como son crear una tarea previamente habiendo visualizado los ejercicios modelo disponibles, eliminar una tarea previamente habiendo visualizado las tareas creadas anteriormente y ver la solución de una tarea generada por un alumno previamente habiendo visualizado las soluciones de tareas solucionadas.

Por último, existe un caso de uso particular, el de eliminar tarea creada, el cual necesita de la confirmación del profesor, ya que es una acción que no puede ser revertida una vez realizada.

DISEÑO

4.1. Descripción de los casos de uso

En esta sección se va a detallar cada uno de los casos de uso en sus respectivos escenarios, como se ha hecho anteriormente estos se van a dividir en dos según su perfil; Profesor o alumno.

Esto nos va a servir para entender el comportamiento correcto que debe tener la aplicación en cada uno de los casos, dado el caso en el que no se haya respetado algún procedimiento por el sistema se ha debido de corregir.

Como recordatorio se presenta en la siguiente tabla los requisitos funcionales pertenecientes al perfil alumno:

Perfil	ID	Nombre
Alumno	A01	Visualizar tareas disponibles
	A02	Realizar tarea

Partiendo de estos requisitos se han generado los siguientes casos de uso:

ID	CU01A
Caso de uso	Visualizar tareas
Actores	Alumno, Sistema
Descripción	Alumno consulta las tareas disponibles para realizar
Precondiciones	
Escenario principal	<ol style="list-style-type: none">1. El alumno se encuentra en la pantalla principal.2. El alumno hace clic en una de las opciones de diferentes tipos de tareas que tiene disponibles.3. El alumno hace clic en el subtipo de tarea que quiere realizar.4. El sistema genera el listado de tareas disponibles que quiere realizar y lo muestra al usuario.
Escenario alternativo	<ol style="list-style-type: none">2. El sistema muestra que la sección no está aún disponible.
Postcondiciones	
Requisito asociado	A01

ID	CU02A
Caso de uso	Realizar tarea
Actores	Alumno, Sistema
Descripción	Alumno realiza una tarea
Precondiciones	
Escenario principal	<ol style="list-style-type: none"> 1. El alumno se encuentra en la pantalla principal. 2. El alumno hace clic sobre la opción de menú "Realizar ejercicio". 3. El alumno hace clic en una de las opciones de diferentes tipo de tareas que tiene disponibles. 4. El alumno hace clic en el subtipo de tarea que quiere realizar. 5. El sistema genera el listado de tareas disponibles que quiere realizar y lo muestra al usuario. 6. El usuario hace clic sobre una tarea del listado. 7. El sistema genera la tarea con las entradas necesarias para completar la tarea y con un mensaje informativo. 8. El usuario introduce las entradas y hace clic en corregir. 9. El sistema valida las entradas. 10. El usuario puede elegir entre corregir y volver al paso 7 o directamente enviar el resultado haciendo clic en enviar.
Escenario alternativo	<ol style="list-style-type: none"> 7. El usuario no introduce todas las entradas. 8. El sistema valida las entradas. 9. El sistema muestra un mensaje informativo para que el usuario introduzca las entradas necesarias.
Postcondiciones	El sistema registra la solución a la tarea.
Requisito asociado	A02

Como recordatorio se presenta en la siguiente tabla los requisitos funcionales pertenecientes al perfil profesor:

Perfil	ID	Nombre	Descripción
Profesor	P01	Visualizar ejercicios disponibles	Un profesor podrá visualizar los ejercicios que se encuentran disponibles en la plataforma para poder elegir entre ellos.
	P02	Crear tarea	Un profesor podrá generar un ejercicio que haya sido previamente seleccionado.

	P03	Visualizar tareas creadas	Un profesor puede visualizar las tareas que ha generado anteriormente.
	P04	Eliminar tarea generada	Un profesor podrá eliminar una tarea generada de un ejercicio con las respectivas soluciones de usuarios.
	P05	Visualizar tareas-soluciones	Un profesor podrá visualizar el listado de las tareas de soluciones generadas por alumnos.
	P06	Visualizar solución tarea	Un profesor podrá visualizar una solución de una tarea que haya completado un alumno con la corrección sugerida por el sistema.

Partiendo de estos requisitos se han generado los siguientes casos de uso:

ID	CU01P
Caso de uso	Visualizar ejercicios disponibles
Actores	Profesor, Sistema
Descripción	Profesor consulta los ejercicios disponibles para realizar tareas.
Precondiciones	
Escenario principal	<ol style="list-style-type: none"> 1. El profesor se encuentra en la pantalla principal. 2. El profesor hace clic en la opción de menú "Introducir tarea". 3. El profesor hace clic en una de las opciones de diferentes tipos de ejercicio que tiene disponibles. 4. El profesor hace clic en el subtipo de ejercicio que quiere realizar como tarea para los alumnos. 5. El sistema genera el listado de ejercicio disponibles que quiere realizar y lo muestra al usuario.
Escenario alternativo	2. El sistema muestra que la sección no está aún disponible.
Postcondiciones	
Requisito asociado	P01

ID	CU02P
Caso de uso	Crear tarea

Actores	Profesor, Sistema
Descripción	Profesor realiza una tarea a partir de un ejercicio disponible.
Precondiciones	
Escenario principal	<ol style="list-style-type: none"> 1. El profesor se encuentra en la pantalla principal. 2. El profesor hace clic en la opción de menú "Introducir tarea". 3. El profesor hace clic en una de las opciones de diferentes tipo de ejercicio que tiene disponibles. 4. El profesor hace clic en el subtipo de ejercicio que quiere realizar como tarea para los alumnos. 5. El sistema genera el listado de ejercicio disponibles que quiere realizar y lo muestra al usuario. 6. El profesor hace clic sobre uno de los ejercicios disponibles. 7. El sistema genera el formulario con las entradas necesarias para el ejercicio seleccionado. 8. El profesor introduce las entradas. 9. El profesor hace clic en enviar. 10. El sistema valida las entradas necesarias. 11. El sistema muestra la confirmación de la tarea introducida.
Escenario alternativo	<ol style="list-style-type: none"> 9. El sistema valida las entradas necesarias. 10. El sistema detecta que las entradas son erróneas. 11. El sistema muestra un mensaje informativo al profesor y vuelve al paso 7.
Postcondiciones	La tarea se queda registrada en el sistema.
Requisito asociado	P02

ID	CU03P
Caso de uso	Visualizar tareas creadas
Actores	Profesor, Sistema
Descripción	Profesor consulta las tareas para realizar que se han generado con anterioridad.
Precondiciones	
Escenario principal	<ol style="list-style-type: none"> 1. El profesor se encuentra en la pantalla principal. 2. El profesor hace clic en la opción de menú "Visualizar/Eliminar tareas".

	3. El sistema genera el listado de tareas almacenadas para realizar por el alumno.
Escenario alternativo	2. El sistema muestra que la sección no está aún disponible.
Postcondiciones	
Requisito asociado	P03

ID	CU04P
Caso de uso	Eliminar tarea creada.
Actores	Profesor, Sistema
Descripción	Profesor elimina una tarea creada junto a sus soluciones.
Precondiciones	
Escenario principal	<ol style="list-style-type: none"> 1. El profesor se encuentra en la pantalla principal. 2. El profesor hace clic en la opción de menú "Visualizar/Eliminar tareas". 3. El sistema genera el listado de tareas almacenadas para realizar por el alumno. 4. El profesor selecciona una de las tareas para eliminar haciendo clic en ella. 5. El sistema muestra un mensaje informativo pidiendo la confirmación del profesor para eliminar el ejercicio. 6. El profesor confirma haciendo clic sobre la opción "Si" en el mensaje informativo. 7. El sistema muestra la confirmación de la tarea eliminada.
Escenario alternativo	<ol style="list-style-type: none"> 6. El profesor hace clic en la opción "No". 7. El sistema vuelve al paso 3 con el listado generado.
Postcondiciones	El sistema registra que la tarea ha sido eliminada.
Requisito asociado	P04

ID	CU05P
Caso de uso	Visualizar tareas solucionadas
Actores	Profesor, Sistema
Descripción	Profesor consulta las tareas que han sido solucionadas por alumnos.

Precondiciones	
Escenario principal	<ol style="list-style-type: none"> 1. El profesor se encuentra en la pantalla principal. 2. El profesor hace clic en la opción de menú "Visualizar tareas solucionadas". 3. El sistema genera el listado de tareas solucionadas por alumnos.
Escenario alternativo	2. El sistema muestra que la sección no está aún disponible.
Postcondiciones	
Requisito asociado	P05

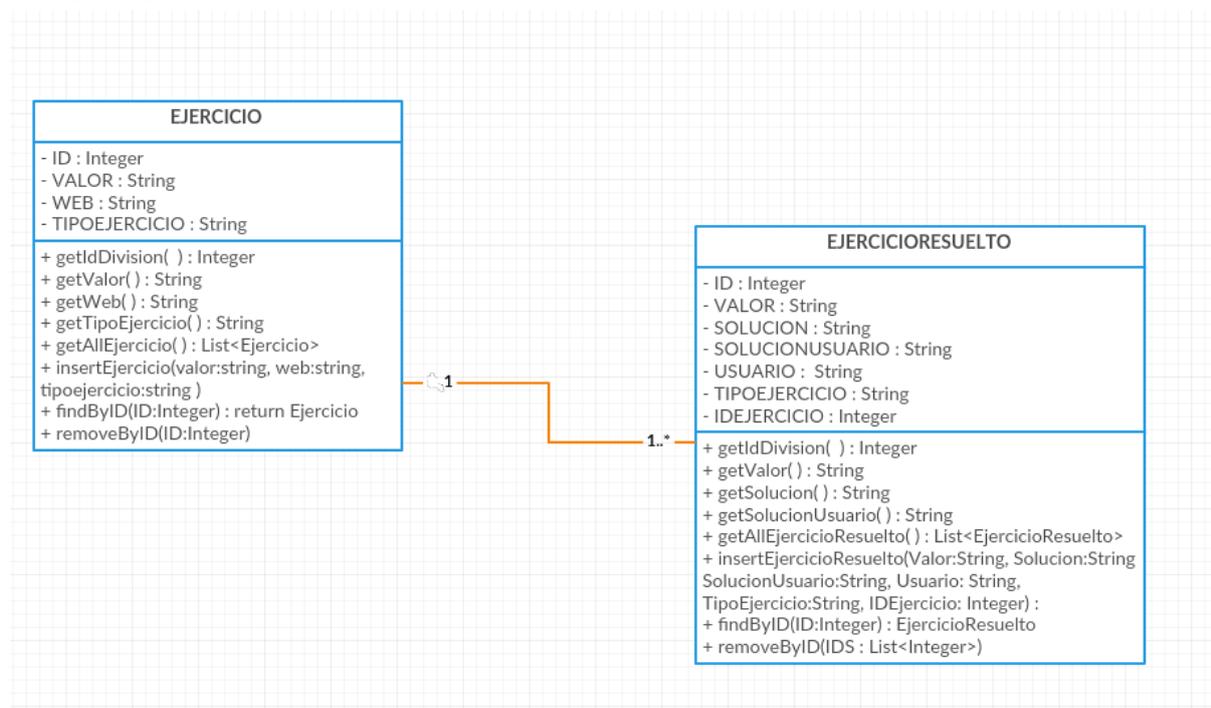
ID	CU06P
Caso de uso	Visualizar tareas solucionadas
Actores	Profesor, Sistema
Descripción	Profesor consulta las tareas que han sido solucionadas por alumnos.
Precondiciones	
Escenario principal	<ol style="list-style-type: none"> 1. El profesor se encuentra en la pantalla principal. 2. El profesor hace clic en la opción de menú "Visualizar tareas solucionadas". 3. El sistema genera el listado de tareas solucionadas por alumnos. 4. El profesor selecciona una de las tareas haciendo clic en ella. 5. El sistema genera la solución y la muestra al profesor.
Escenario alternativo	2. El sistema muestra que la sección no está aún disponible.
Postcondiciones	
Requisito asociado	P06

4.2. Modelo de clases

Una vez llegados a este punto y tras realizar un estudio sobre las necesidades del proyecto, se llegó a la conclusión de que lo principal y necesario siguiendo el objetivo propuesto sería almacenar tanto las tareas como las soluciones a estas junto a todos los detalles de estas.

No se ha hecho hincapié en el registro de los usuarios ya que esto está establecido como una mejora futura en la conexión con la herramienta “SIETTE”, así que esta fase ha sido simulada.

El siguiente diagrama nos muestra un esquema del modelo de clases que se ha establecido:



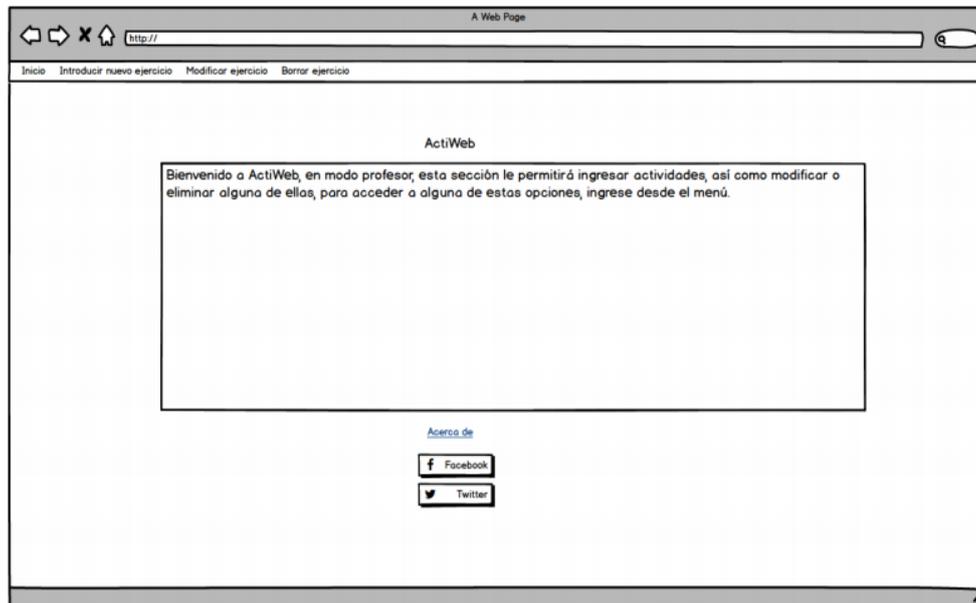
Se puede observar en este esquema las dos entidades diferenciadas: “EJERCICIO” y “EJERCICIORESUELTO”, las cuales están vinculadas debido a su relación, para existir soluciones de una tarea debe estar creada anteriormente la propia tarea, por ello la relación de uno a muchos.

Para la entidad “EJERCICIO” se han establecido los atributos ID, como identificador único, VALOR, para los parámetros que contendrá el mismo para su estructura posterior, WEB, su subdirección en la aplicación y TIPOEJERCICIO, para distinguir entre los diferentes tipos.

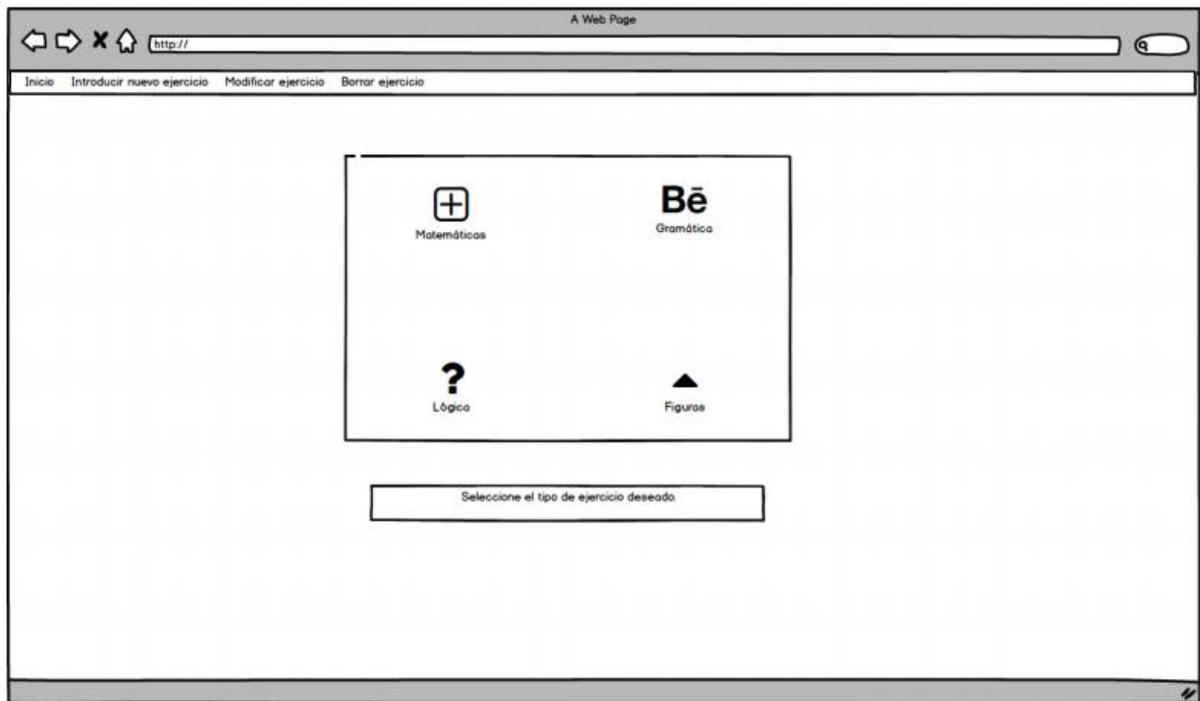
Para la entidad “EJERCICIORESUELTO” se han establecido los atributos (sin nombrar los comunes con “EJERCICIO”) SOLUCIÓN, como solución correcta del ejercicio, SOLUCIONUSUARIO, como la solución que ha procedido a rellenar el

usuario anteriormente, USUARIO, como el usuario que ha procedido a la resolución del ejercicio, IDEJERCICIO, para la relación que tiene con la entidad EJERCICIO.

Antes de dar paso a la implementación cabe mencionar que se ha tenido que hacer una serie de tareas que también forman parte de la implementación, la realización de “mockups”, maquetas para demostración, para una primera presentación de lo que sería la vista de nuestra aplicación y darse con ello el visto bueno a un primer modelo. A continuación se detallan algunos de los diseños propios que fueron aceptados:



Maquetado de la pantalla principal



Maquetado de las opciones disponibles



Maquetado de la vista en dispositivo móvil

IMPLEMENTACIÓN

Para realizar la implementación de la aplicación web de nuestro proyecto se ha decidido separar el desarrollo de este en diferentes iteraciones, estas iteraciones a su vez contendrán fases en las que se especifica cada una de las tareas a realizar.

Es importante resaltar que estamos ante una metodología incremental, lo cual por su propia naturaleza y por la interacción con el cliente, en este caso mediante reuniones con el tutor, se han ido añadiendo o eliminando elementos según las necesidades que iban surgiendo y que debía cubrir la propia aplicación.

Por lo tanto y después de lo mencionado anteriormente, para estructurar y recoger mejor todas las tareas que se han debido de realizar se muestra la siguiente tabla:

ITERACIÓN	Tarea
PRIMERA (ELEMENTOS COMUNES)	1. Creación de estructura global, logo, pié de página, cuerpo y diseño.
	2. Creación de menús profesor y alumno.
	3. Creación de mensajes informativos.
	4. Pruebas y corrección de errores.
SEGUNDA (BLOQUE PROFESOR)	1. Creación de la interfaz.
	2. Creación de la capa de datos.
	3. Creación de la capa controlador.
	4. Pruebas y corrección de errores.
TERCERA (BLOQUE ALUMNO)	1. Creación de la interfaz.
	2. Creación de la capa de datos.
	3. Creación de la capa controlador.
	4. Pruebas y corrección de errores.

A continuación, se van a detallar cada una de las iteraciones con ejemplos de cada desarrollo de las tareas y en el caso de la segunda y tercera iteración se especifican

diagramas de secuencia los cuales explican el comportamiento de las acciones que pueden realizar profesor y alumno.

5.1. Primera iteración : Elementos Comunes.

Para esta primera iteración se ha tenido que tener en cuenta con qué escenario nos vamos a encontrar y el tipo de usuario que va a usar la aplicación.

Debido a que es un ambiente escolar, no podemos establecer una interfaz complicada, con muchos elementos u opciones de menú infinitas, ya que el alumno puede perderse en ellas.

Además, la interfaz debería de considerar tener colores que resulten amigables al usuario junto a algunos elementos que lo apoyen, el color que se ha decidido a tener más en cuenta es un tono azul oscuro, ya que hay teorías que defienden que este resulta un color que cansa menos de cara a la visión (hay que tener en cuenta que un alumno puede estar un tiempo considerable resolviendo un problema).

Así continuación se mostrará cuáles son los elementos en común que han sido considerados válidos.

5.1.1. Estructura global, pie de página y cuerpo

Para la estructura global, dado que expresamente se hizo la petición de que esta aplicación web tuviera que ser responsive, es decir, poder ser visualizada por varios dispositivos como puede ser una tablet, se decidió tomar un diseño sencillo, sin demasiados elementos que puedan agolparse entre sí cuando se tratase de una visualización mediante dispositivo móvil.

Por todo ello, como se muestra en las maquetas realizadas, tenemos un diseño lineal consistente en un menú, un cuerpo de página y seguidamente un pie de página que nos informa de la posibilidad de seguir a esta aplicación mediante las redes sociales, redes que pueden ser usadas también para el apoyo en diversos ejercicios tanto por alumno como profesores, dado que estas redes sociales actualmente son muy relevantes a día de hoy.

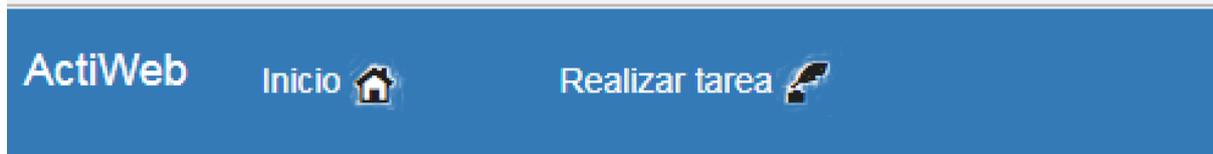
Una vez que se ha desarrollado esta estructura, que consistía básicamente en la aplicación de clases de Bootstrap sobre los elementos principales que formarán parte de toda nuestra aplicación, estos ya pueden ser importados al resto de vistas ahorrando en tiempo de desarrollo, ya que conformarán una plantilla.

5.1.2. Menús profesor y alumno

En estos menús se ha tenido en consideración también que deben ser responsive, además, deben posibilitar al usuario volver en cualquier momento al inicio de la aplicación y poder moverse por las diferentes opciones disponibles.

Como elemento adicional, se han aplicado iconos que refuercen aún más la descripción de una opción de menú, dotándolo además de un aspecto más amigable al usuario.

En la siguiente figura se observa el menú correspondiente al alumno:



Este consiste básicamente en el botón que vuelve a la página principal y el que permite realizar un ejercicio.

En la siguiente otra figura puede observarse el menú correspondiente para el profesor:

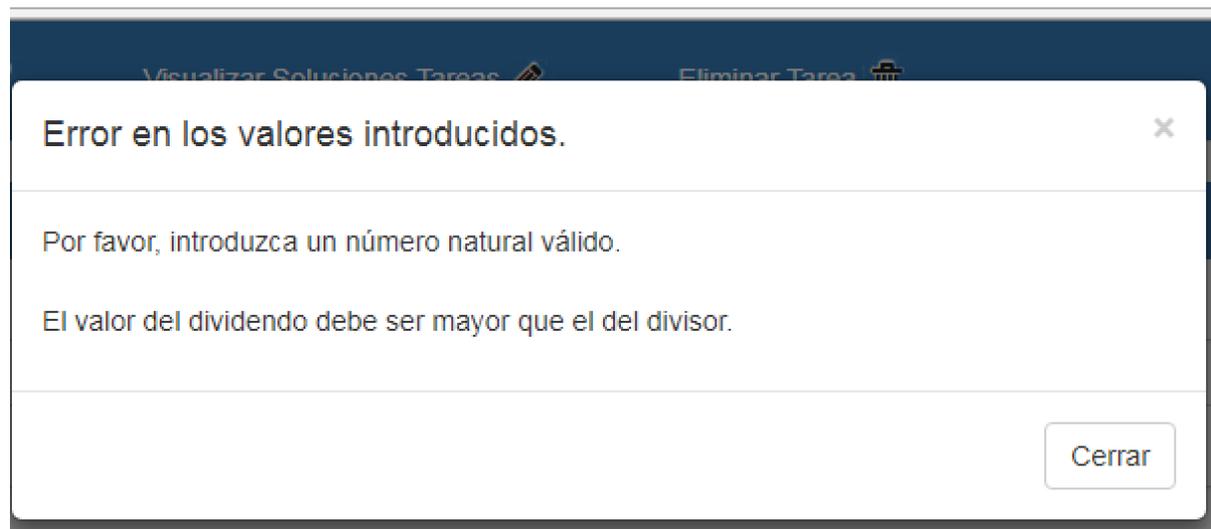


En este podemos observar, como era de esperar, cómo tenemos más opciones que corresponden al profesor.

5.1.3. Creación de mensajes informativos

Como en cualquier aplicación de hoy en día tenemos mensajes informativos, estos, valga la redundancia, informarán al usuario de mensajes importantes que deban ser de su conocimiento como, por ejemplo, un error en los datos introducidos o información acerca de cómo proceder a la resolución de alguna operación.

Por ello se desarrolló una plantilla con las clases de Bootstrap que facilitará esta información al usuario; un ejemplo de ella es la siguiente página:



Como se hizo con la estructura, este mensaje una vez creado, también servirá como plantilla en donde deba visualizarse cualquier mensaje informativo.

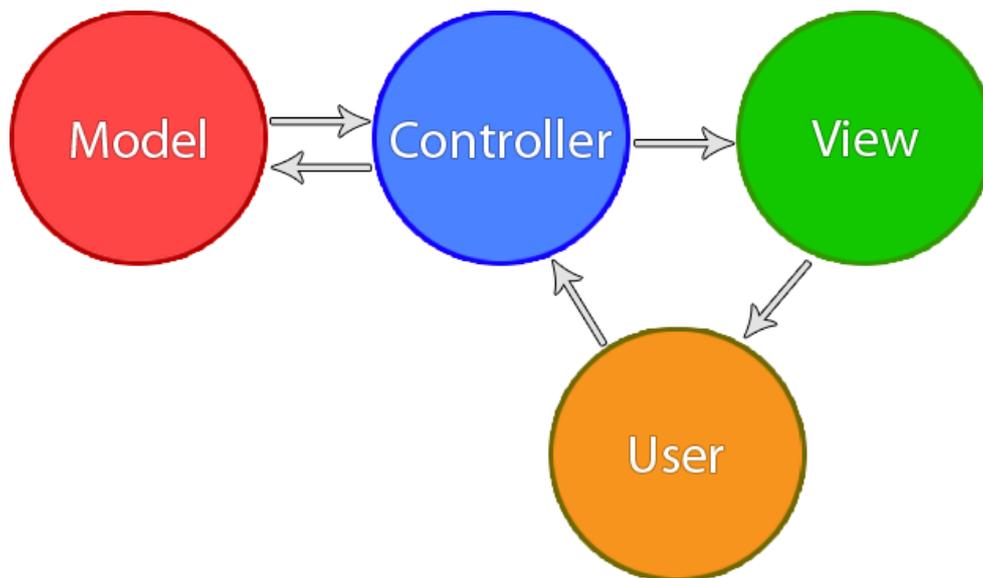
5.1.4. Pruebas y corrección de errores

En lo que atañe a las pruebas, al ser todos estos elementos partes comunes de lo que será el resto de la aplicación, se han desarrollado con bastante cautela y con aprobación del usuario final, por lo que la corrección de errores ha sido mínima, cuestiones de aspecto como puede ser algún icono o descripción de las opciones de menús.

5.2. Segunda iteración : Bloque profesor

En lo que concierne a esta segunda iteración, con respecto a la tercera tenemos algunas similitudes, entre ellas que vamos a desarrollar siguiendo un patrón, este patrón que se ha seguido se ha tomado como referencia del Modelo Vista Controlador (MVC), patrón que fue modificado debido a la inyección del plugin para el manejo de datos con nuestra base de datos H2.

El patrón Modelo Vista Controlador consiste en separar en diferentes módulos lo que sería la construcción de nuestra aplicación web, para entenderlo mejor nos ayudamos de la siguiente imagen:

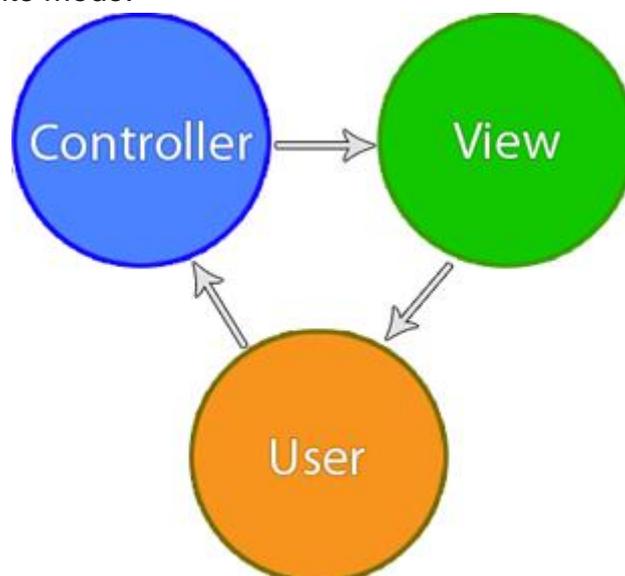


El modelo se hace cargo de la gestión de la información que tenemos almacenada.

El controlador se hace cargo tanto de la información enviada por el usuario mediante la vista, es decir, eventos como la pulsación de un botón con la respectiva información como la carga de datos a través del modelo para ser mostrada al usuario a través de la vista.

La vista se hace cargo de mostrar la información de manera “amigable” al usuario, información que puede provenir del modelo a través del controlador o de la propia vista, ofreciendo al usuario opciones sobre las cuales poder ejecutar determinadas acciones.

Como al principio de esta sección se ha mencionado, este patrón fue modificado para incluir en el controlador la parte que corresponde al modelo, con lo que quedaría del siguiente modo:



En el mismo controlador tendremos un bloque bien diferenciado que hará el papel de lo que era el modelo en el patrón MVC con las consultas necesarias a la base de datos y la conversión del resultado de estas para su tratamiento.

Es importante también destacar, que a diferencia de algunas otras aplicaciones, la verificación de datos introducidas por el usuario se efectúa en la vista mediante JavaScript o jQuery, ya que una petición al controlador es más costosa en rendimiento que una comprobación en la propia vista.

5.2.1 Bloque profesor

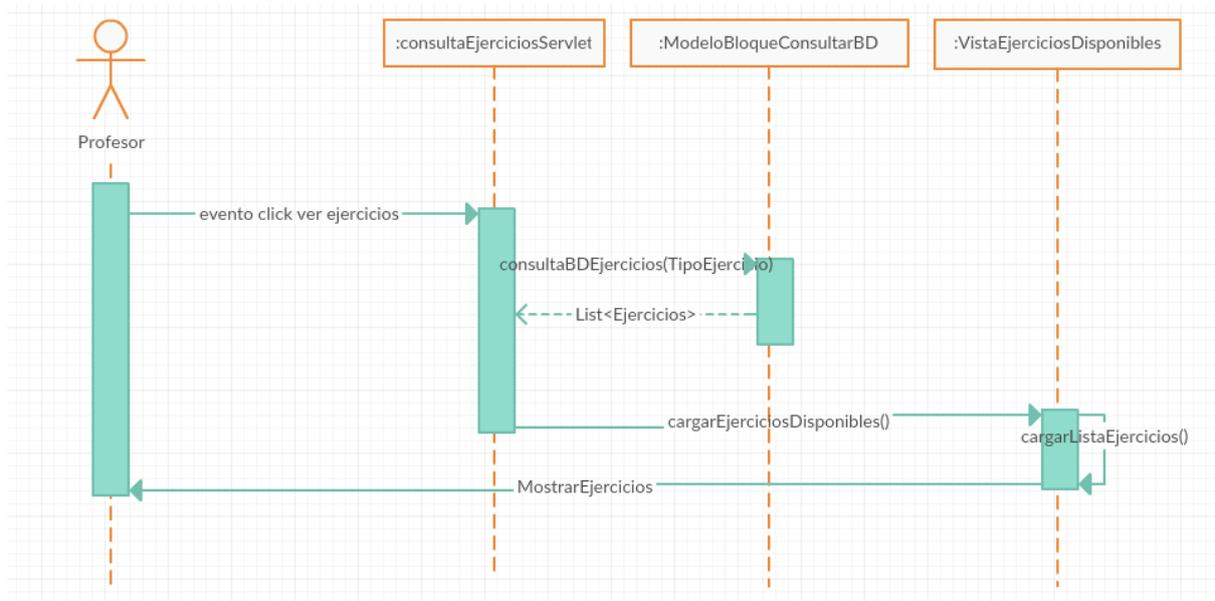
En este bloque se han integrado todas las acciones que pertenecen al usuario con perfil de profesor, una vez que se han implementado todos los elementos que tenemos en común para toda la aplicación anteriormente mencionados y el menú correspondiente a este perfil.

Las acciones correspondientes al usuario profesor, tal y como se mencionaban en los casos de uso, son:

- Visualizar ejercicios disponibles.
- Crear tarea.
- Visualizar tareas creadas.
- Eliminar tarea generada.
- Visualizar tareas-soluciones.
- Visualizar solución tarea.

Para entrar en mejor detalle conforme a estas acciones y la forma en la que hemos desarrollado la implementación de este bloque nos serviremos de los diagramas de secuencia que se detallan a continuación:

Diagrama de secuencia Visualizar ejercicios disponibles



En este diagrama se muestra la siguiente secuencia de acciones:

Un usuario con perfil profesor mediante el evento proveniente de otra vista (en el menú principal) ordena en la aplicación que se muestre los ejercicios de un determinado tipo.

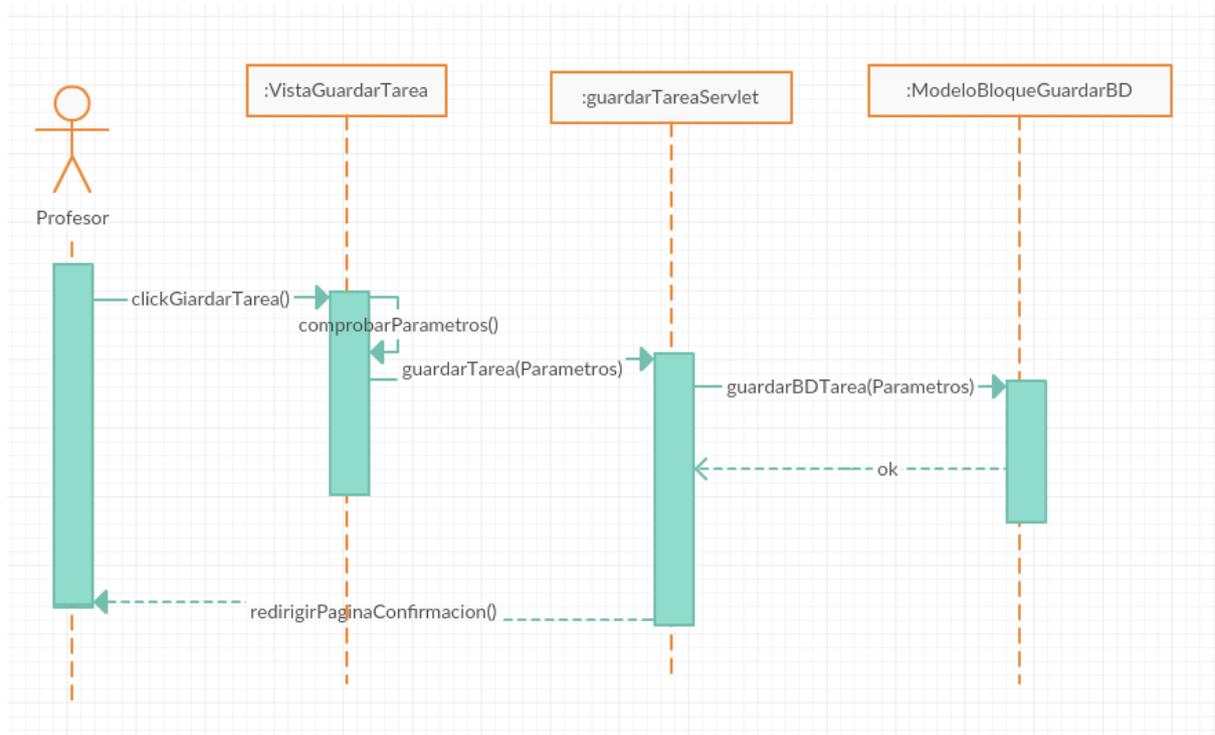
Esta acción es recibida por el controlador y acto seguido este ejecuta la orden de consulta a la capa de modelo para encontrar dicha información.

En cuanto es encontrada esta información por el modelo, vuelve al controlador con esta.

El controlador se encarga de enviar la información a la vista.

La vista se encarga de tratar la lista de ejercicios mediante JQuery para estructurarla para luego ser mostrada al usuario.

Diagrama de secuencia Crear tarea



En este diagrama se muestra la siguiente secuencia de acciones:

El actor profesor mediante la vista ordena guardar una tarea con los parámetros introducidos.

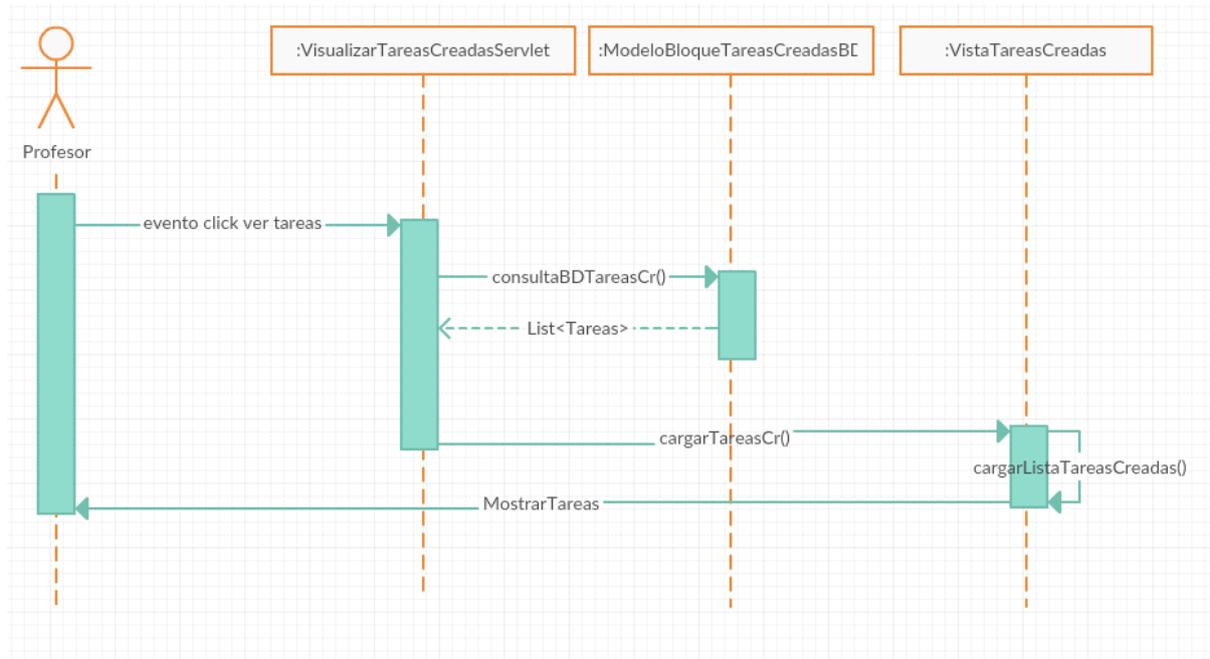
La vista mediante JQuery comprueba dichos parámetros y una vez verificados los envía al controlador.

El controlador envía la orden a la capa de modelo para que se guarde dicha tarea.

El modelo envía la confirmación de dicha tarea creada al controlador.

El controlador una vez recibe dicha verificación envía redireccionar al usuario a una página genérica de confirmación.

Diagrama de secuencia Visualizar tareas creadas



En este diagrama se muestra la siguiente secuencia de acciones:

Un usuario con perfil profesor mediante el evento proveniente de otra vista (en el menú principal) ordena en la aplicación que se muestre las tareas creadas.

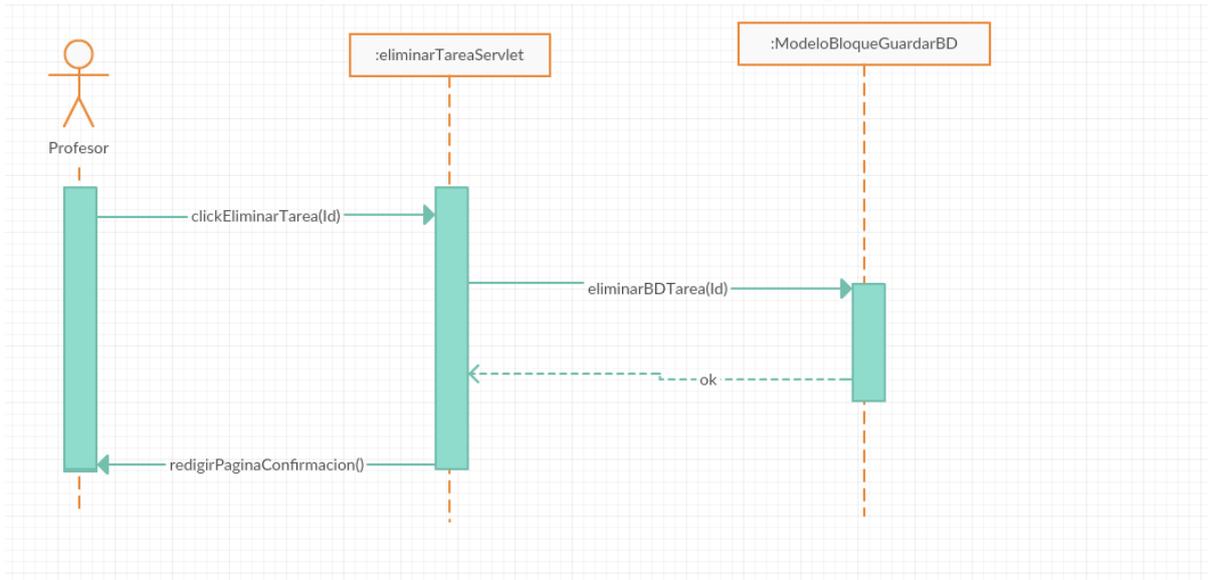
Esta acción es recibida por el controlador y acto seguido el controlador ejecuta la orden de consulta a la capa de modelo para encontrar dicha información.

En cuanto es encontrada esta información por el modelo, vuelve al controlador con esta.

El controlador se encarga de enviar la información a la vista.

La vista se encarga de tratar la lista de tareas creadas mediante JQuery para estructurarla para luego ser mostrada al usuario.

Diagrama de secuencia Eliminar tarea generada



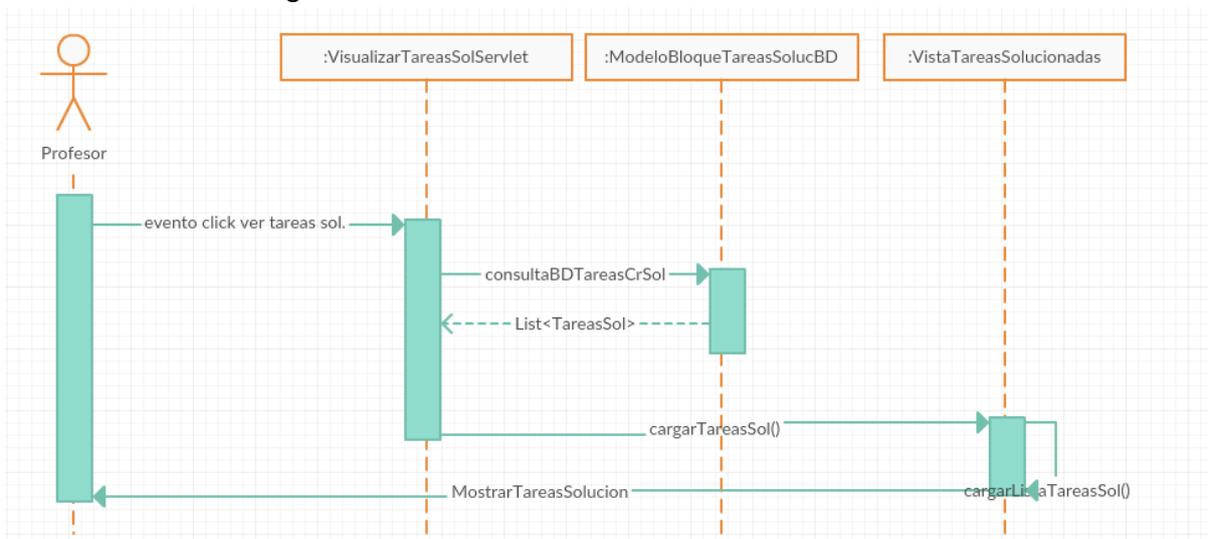
En este diagrama se muestra la siguiente secuencia de acciones:

El controlador envía la orden a la capa de modelo para que se elimine dicha tarea.

El modelo envía la confirmación de borrado de dicha tarea creada al controlador.

El controlador una vez recibe dicha verificación redirecciona al usuario a una página genérica de confirmación.

Diagrama de secuencia Visualizar tareas-soluciones



En este diagrama se muestra la siguiente secuencia de acciones:

Un usuario con perfil profesor mediante el evento proveniente de otra vista (en el menú principal) ordena en la aplicación que se muestre las tareas que fueron

solucionadas por los alumnos.

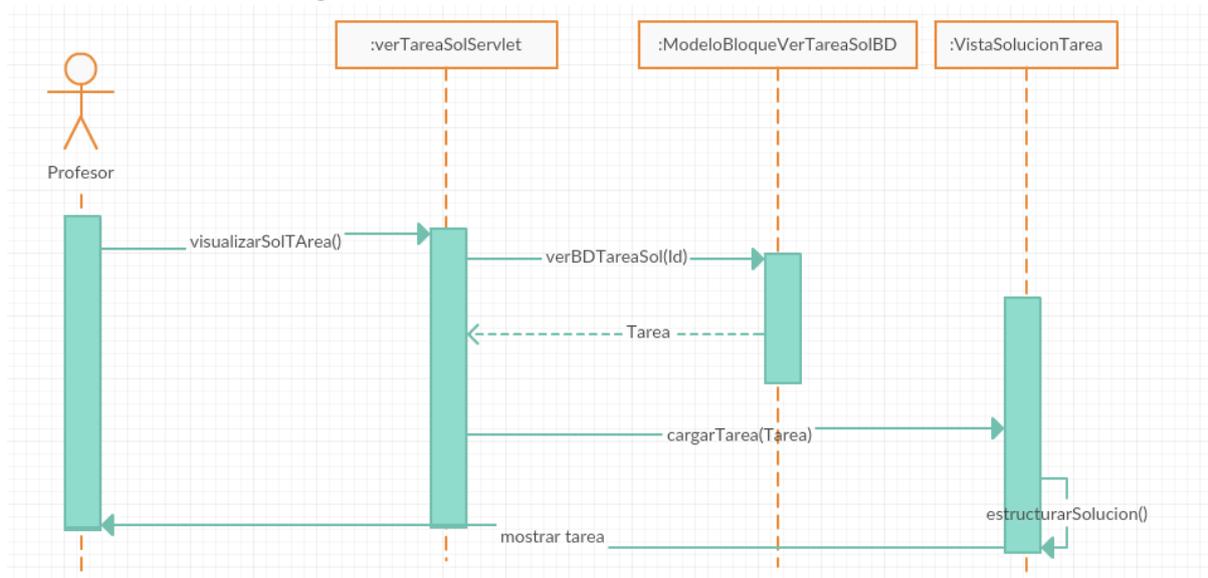
Esta acción es recibida por el controlador y acto seguido el controlador ejecuta la orden de consulta a la capa de modelo para encontrar dicha información.

En cuanto es encontrada esta información por el modelo, vuelve al controlador con esta.

El controlador se encarga de enviar la información a la vista.

La vista se encarga de tratar la lista de tareas solucionadas mediante JQuery para estructurarla para luego ser mostrada al usuario.

Diagrama de secuencia Visualizar solución tarea



En este diagrama se muestra la siguiente secuencia de acciones:

El actor profesor mediante el evento procedente de otra vista ordena consultar una tarea con solución de usuario.

El controlador envía la orden a la capa de modelo para que se consulte dicha tarea.

El modelo envía la información de dicha tarea creada al controlador.

El controlador una vez recibe dicha solución de la tarea la envía a la vista.

La vista se encarga de estructurar dicha información para luego mostrarla al usuario profesor.

Con todos estos diagramas se ha pretendido mostrar el funcionamiento correcto que debe tener nuestra aplicación para poder facilitar así el desarrollo a más bajo nivel. A continuación se mostrarán también los diagramas pertenecientes al usuario alumno, en los cuales se presenta alguno con cierta similitud a los del bloque profesor.

5.2.2. Bloque alumno

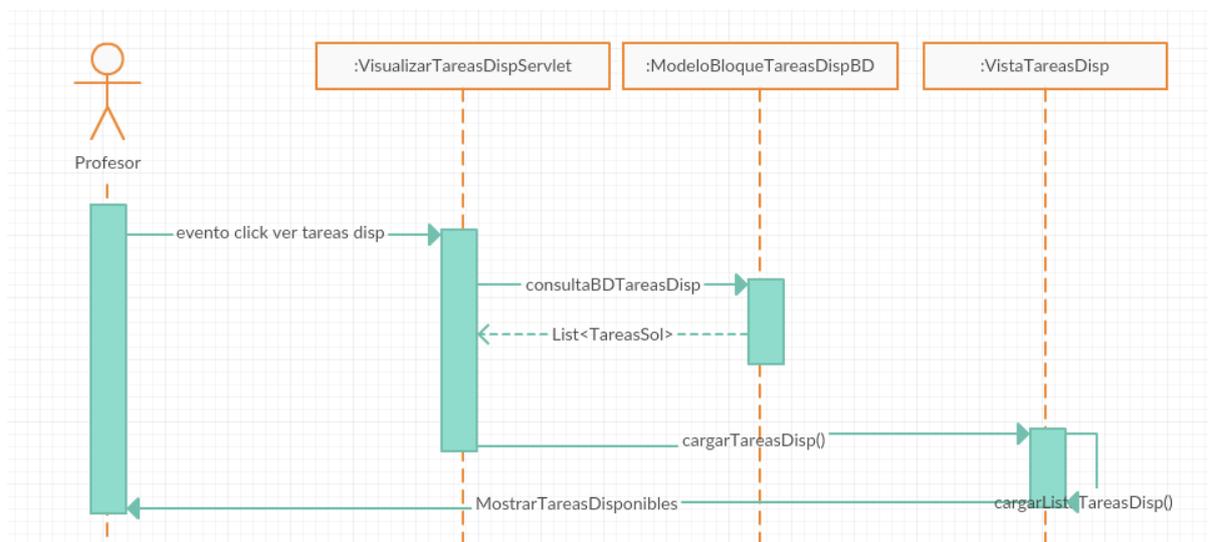
En este bloque se han recogido todas las acciones que pertenecen al usuario con perfil de alumno, una vez que se han implementado todos los elementos que tenemos en común para toda la aplicación anteriormente mencionados y el menú correspondiente a este perfil, tal y como se hizo en el bloque profesor.

Las acciones correspondientes al usuario alumno en este caso son:

- Visualizar tareas disponibles
- Realizar tarea

Nuevamente, como se ha realizado con el perfil profesor, se describen las acciones de cada escenario con los diagramas de secuencia:

Diagrama de secuencia Visualizar tareas disponibles



En este diagrama se muestra la siguiente secuencia de acciones:

Un usuario con perfil alumno, mediante el evento proveniente de otra vista (en el menú principal), ordena en la aplicación que se muestren las tareas que disponibles para los alumnos.

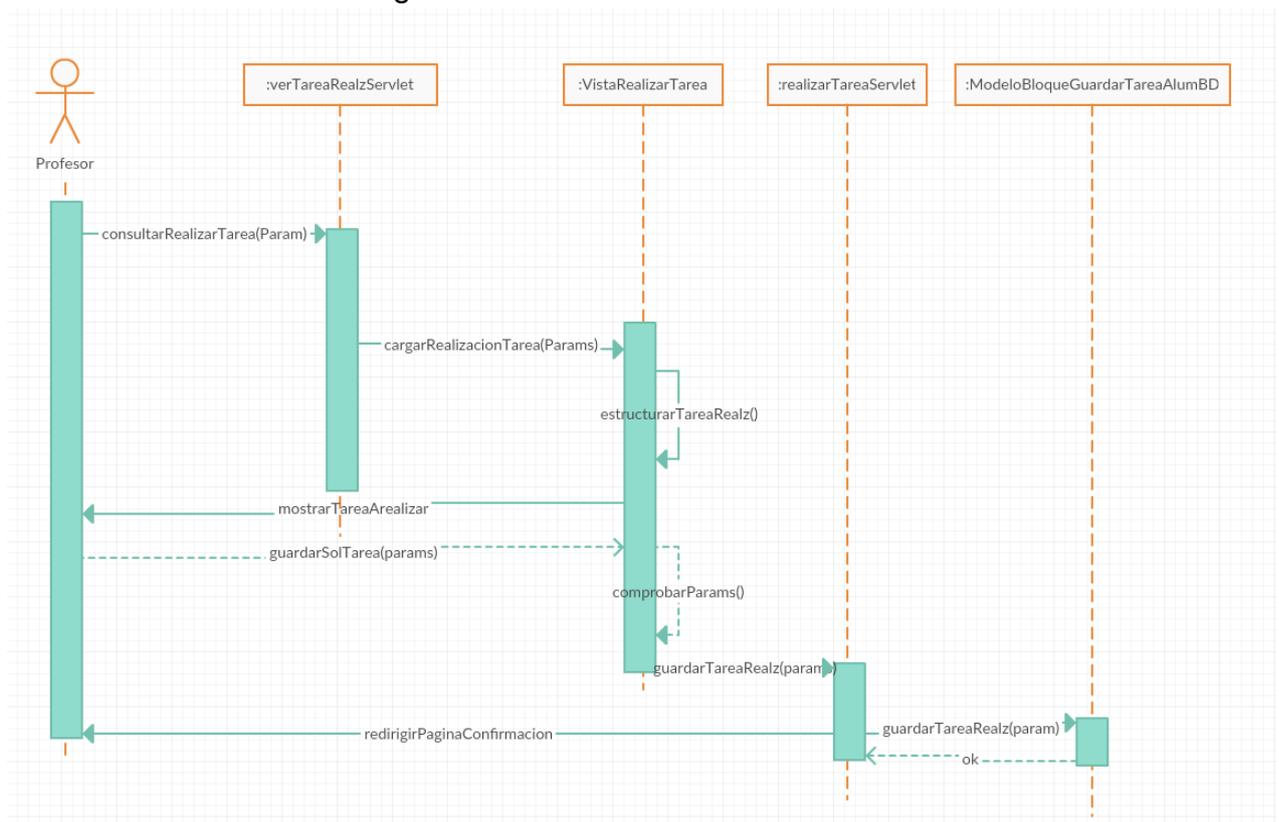
Esta acción es recibida por el controlador y acto seguido el controlador ejecuta la orden de consulta a la capa de modelo para encontrar dicha información.

En cuanto es encontrada esta información por el modelo, vuelve al controlador con ella.

El controlador se encarga de enviar la información a la vista.

La vista se encarga de tratar la lista de tareas disponibles para solucionar mediante JQuery para estructurarla para luego ser mostrada al usuario.

Diagrama de secuencia realizar tarea



En este diagrama se muestra la siguiente secuencia de acciones:

Un usuario con perfil alumno mediante el evento proveniente de otra vista (en el menú principal) ordena en la aplicación que se muestre un determinado ejercicio a realizar.

Esta acción es recibida por el controlador y acto seguido el controlador generando la tarea envía la información a la vista.

En cuanto es encontrada esta información por la vista, esta estructura la tarea a realizar para el usuario y se la muestra.

Una vez realizada la solución el usuario en la misma vista envía la solución. La vista se encarga de validar la información proveniente del usuario y después la envía al controlador encargado de guardar la tarea solucionada.

La tarea solucionada se envía por el controlador hacia el bloque del modelo encargado para guardar la misma solución, una vez guardado confirma al controlador la acción.

Una vez que el controlador recibe la confirmación redirige al alumno a una página de confirmación.

Con respecto a los demás casos, este último ha sido algo diferente, en él intervienen dos controladores, una vista, un modelo y el actor. Este caso es específico ya que uno de los controladores se debe encargar de generar toda la tarea a solucionar para el aprendizaje del alumno, esta tarea constará de diversos pasos que deberá solucionar el mismo alumno.

Por último, debe mencionarse que estos escenarios están pensados para los casos de éxito. Obviamente, puede suceder que algún caso no termine en éxito: por ejemplo, para el caso de introducir una nueva tarea por parte del profesor, puede que este no haya introducido bien los parámetros, en cuyo caso la vista avisaría al mismo usuario con un aviso al respecto o también no se haya guardado correctamente esta misma por un fallo en el sistema, lo cual reenviaría al usuario a una página de error advirtiéndolo de este suceso.

5.2.3. Pruebas y corrección de errores

En lo que atañe a las pruebas en estos dos últimos bloques, se han realizado bastantes trazas para averiguar qué valores podrían estar fallando, todas estas pruebas y su corrección se han realizado mediante los siguientes elementos:

Mediante el depurador propio de Netbeans: este ha sido muy útil sobre todo en la capa de controlador y en el modelo donde se recogían las diferentes excepciones para ser tratadas, una vez identificados estos problemas en la línea correspondiente fue fácil su corrección.

Mediante el depurador de Internet Explorer: por experiencia propia se ha podido comprobar que respecto a otros navegadores, este es el que mejor se adapta en cuanto a correcciones en la capa de la vista, ya que su depurador de código JQuery es excelente, pudiendo emular a la par a cómo lo hace el de Netbeans con las otras capas mencionadas.

CONCLUSIONES

En esta última sección se recogen algunas de las conclusiones obtenidas a nivel personal, cómo ha influido el proyecto para mejorar ciertos aspectos tanto académicos como profesionales, así como también mencionar qué futuras mejoras podría tener para que esta pueda ser una aplicación completa dedicada al sector de la educación.

6.1. Resultado final y conclusiones.

En la realización de este proyecto se perseguía que la aplicación tuviera una serie de especificaciones que se debían de respetar, por ello se explica cómo se han resuelto:

Fácil usabilidad: tanto la estructura propia de las páginas como los menús han sido adecuados para que en ningún momento pueda el usuario sentirse “perdido” entre las diferentes opciones disponibles.

Multiplataforma: la aplicación debía ser responsive y así se ha considerado, para que pueda ejecutarse desde cualquier dispositivo que sea ordenador, tablet o móvil con alta resolución.

Aprendizaje del alumno: esto podría considerarse el mayor objetivo de todos, el cual ha sido solventado con la carga siempre de todos los pasos que repercute solucionar un problema, así como la inclusión de mensajes informativos que notifican al alumno con pistas que le puedan servir de apoyo.

Conforme al último punto descrito cabe destacar que a día de hoy no se ha podido encontrar una herramienta en línea que pueda cubrir esta necesidad.

En el ámbito personal y profesional, se puntualiza sobre todo la mejora de la capacidad de análisis que se ha desarrollado, anteriormente, se había realizado un trabajo similar en la asignatura empleando este entorno. Sin embargo, el trabajo realizado en aquella ocasión fue en grupo por lo que algún aspecto pudo no ser llevado tan a la práctica como lo ha sido con este proyecto.

A nivel técnico, se ha mejorado en cuanto al conocimiento de la arquitectura MVC y al conocimiento de JQuery, además del nuevo conocimiento adquirido con Bootstrap, herramienta casi indispensable actualmente.

6.2. Futuras mejoras

Como futuras posibles mejoras podrían tomarse en cuenta las siguientes:

Futura integración con la herramienta SIETTE con la que se pensó realizar pero que no ha sido llevada a cabo porque el trabajo ya de por sí tenía entidad suficiente.

Inclusión de más tipos de ejercicios a realizar, como idea podría tomarse la de crear ejercicios con plantillas ya realizadas antes, subida de ficheros words, por ejemplo, con posterior subida de solución en el mismo formato.

Sistema de evaluación por parte del profesor, que supondría la incorporación de nueva entidad calificaciones.

BIBLIOGRAFÍA

- <https://es.wikipedia.org/wiki/Wikipedia:Portada>
- <http://getbootstrap.com/>
- <https://www.w3schools.com/jquery/>
- http://librosweb.es/libro/bootstrap_3/
- <https://www.vmware.com/es.html>
- <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>
- Apuntes de la asignatura “Tecnologías de Aplicaciones Web”

MANUAL DE USUARIO

Anexo : Manual de Usuario

Introducción

ActiLine es una aplicación dedicada al sector de la educación. Por ello nos brinda la posibilidad de acceso tanto a profesores como alumnos con las respectivas opciones típicas de cada una de las labores de estos dos perfiles.

Para mayor comodidad del usuario, el presente manual se va a dividir en dos secciones separadas, obviamente, por cada perfil.

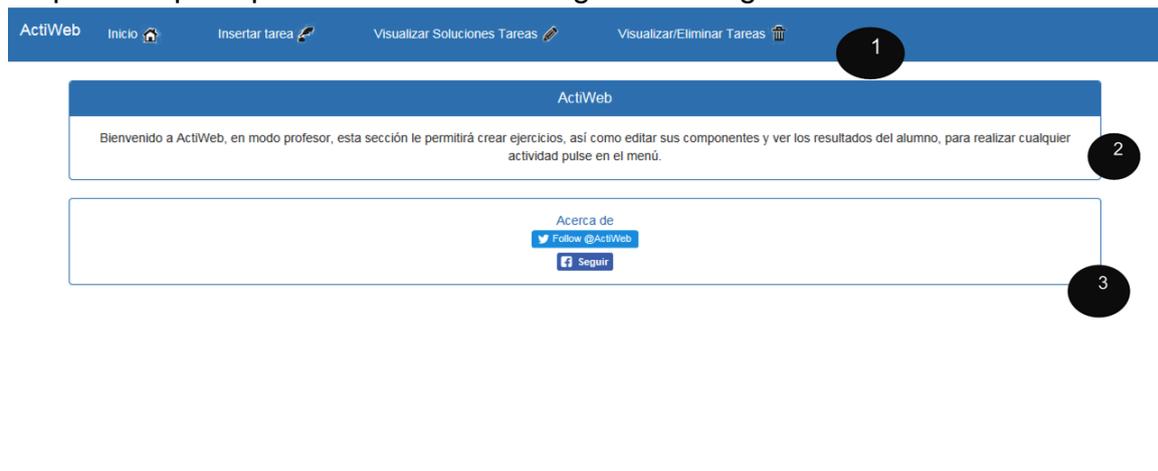
En cada perfil, se podrá observar cómo se divide de igual manera la aplicación en subapartados que serán :

- Pantalla principal.
- Acciones disponibles para el usuario.
- Acciones con detalle para el usuario.

Perfil Profesor

Pantalla principal

La pantalla principal se muestra en la siguiente imagen:



Igualmente se muestra en la siguiente imagen su versión móvil para completar su comprensión:



Cómo se puede observar, la pantalla principal se divide en tres secciones claramente diferenciadas:

El menú principal: desde este puede seleccionarse cualquier opción que se desee realizar; se incluye el botón inicio para volver a esta misma pantalla de bienvenida.

El cuerpo de la página: donde se presentará también en las diferentes páginas información al usuario o petición de información por parte de este.

Pie de página: siempre presente en la aplicación, informa al usuario la posibilidad de conocer datos sobre la aplicación y seguir a esta en las redes sociales.

Acciones disponibles para perfil Profesor

Como se detalla en el menú profesor, un usuario con perfil profesor puede realizar las siguientes acciones:

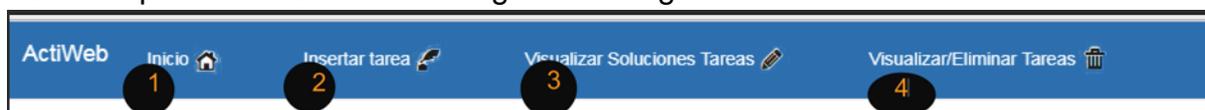
Volver a la página principal: esta opción permite al profesor volver a la pantalla principal de su perfil en la aplicación.

Insertar una tarea: esta opción permite al profesor agregar una tarea de una serie de ejercicios predefinidos en el sistema para la posterior realización de esta por parte de los alumnos.

Visualizar Solución tareas: esta opción permite al profesor visualizar las soluciones planteadas por sus alumnos a las tareas creadas con anterioridad.

Visualizar/Eliminar tareas: esta opción permite al profesor visualizar las tareas que ha generado con anterioridad y además le permite eliminar cualquiera de ellas previa confirmación.

Estas se pueden observar en la siguiente imagen:



Para su versión móvil, es necesario pulsar el botón señalado para ver las opciones de menú:

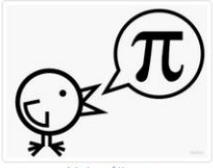


Acciones detalladas para perfil Profesor

Como se ha mencionado en el apartado menú de acciones disponibles para el profesor, en esta sección se explica con detalle cada una de las acciones:

Insertar tarea: para ello se debe acceder desde la opción de menú mencionada en el apartado anterior con lo que nos aparecerá la siguiente página:

Seleccione el tipo de ejercicio deseado.



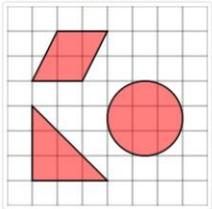
Matemáticas



Gramática



Lógica



Figuras

Acerca de
[Follow @ActiWeb](#)

Estos son los tipos de ejercicios disponibles desde los que se puede realizar una tarea para los alumnos.

ActiWeb Inicio Insertar tarea Visualizar Soluciones Tareas Visualizar/Eliminar Tareas

Introduzca dividendo y divisor.

Dividendo:

Divisor

Acerca de
[Follow @ActiWeb](#)
[Seguir](#)

Donde se le pide al usuario la información necesaria para completar la tarea para poder ser visible por los alumnos.

eb Inicio Insertar tarea Visualizar Soluciones Tareas Visualizar/Eliminar Tareas

Confirmación.



El ejercicio fue guardado correctamente, para realizar un nuevo ejercicio seleccione la opción en el menú superior.

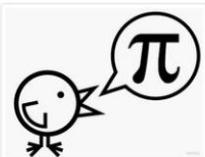
Acerca de
[Follow @ActiWeb](#)
[Seguir](#)

Una vez validados los datos introducidos y comprobado que estos son correctos, aparecerá la pantalla de verificación.

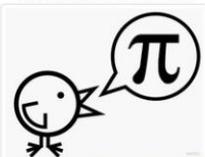
Visualizar Soluciones Tareas: nuevamente, una vez accedida a esta desde el menú principal tenemos la siguiente página, donde se visualizan los tipos de ejercicios que actualmente pueden ser visualizados con su solución:

ActiWeb Inicio Insertar tarea Visualizar Soluciones Tareas Visualizar/Eliminar Tareas

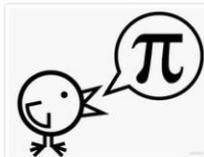
Seleccione el tipo de ejercicio deseado para visualizar soluciones.



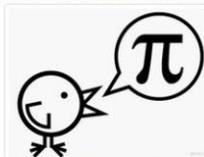
Visualizar División Naturales



Visualizar División con decimales en divisor



Visualizar División con decimales en dividendo



Visualizar División con decimales en dividendo y divisor

Acerca de
[Follow @ActiWeb](#)
[Seguir](#)

Una vez seleccionado uno de los tipos de tareas y el ejercicio del listado, puede observarse la solución introducida por el alumno:

Corrección de la división:

1	9	8	7	.	2	5	3
0							662,41
0			0	7			
0				1	2		
0					0	5	
0						2	

Solución: 662,41

Acerca de
[Follow @ActiWeb](#)
[Seguir](#)

Como se observa en la imagen superior, al pasar por encima el ratón aparece el valor correcto que debió ser introducido por el usuario.

Visualizar/Eliminar tareas: esta opción de menú nos lleva a la siguiente página donde se muestra un listado en el que se pueden visualizar todas las tareas creadas y con la posibilidad de borrar cualquiera de ellas:

Seleccione el ejercicio de matemáticas deseado.

- Division sin signo: 1987/3 (2 Soluciones)
- Division sin signo: 2343/3 (1 Soluciones)
- Division sin signo: 1212/12 (1 Soluciones)
- Division sin signo: 1980/2 (1 Soluciones)
- Division con decimal en numerador: 1987,25/3 (1 Soluciones)
- Division con decimal en numerador: 12,2/12 (1 Soluciones)
- Division con decimal en numerador: 12,2/12 (1 Soluciones)
- Division con decimal en denominador: 1987/3,14 (2 Soluciones)

Si el usuario selecciona una de las tareas y confirma su borrado, aparecerá una ventana de confirmación de igual forma que en insertar tarea.

Perfil Alumno

Pantalla principal

La pantalla principal se muestra en la siguiente imagen:



Igualmente se muestra en la siguiente imagen su versión móvil para completar su comprensión:



Como se puede observar, la pantalla principal se divide en tres secciones claramente diferenciadas:

El menú principal: desde este puede seleccionarse cualquier opción que se desee realizar, se incluye el botón inicio para volver a esta misma pantalla de bienvenida.

El cuerpo de la página: donde se presentará también en las diferentes páginas información al usuario o petición de información por parte de este.

Pie de página: siempre presente en la aplicación informa al usuario la posibilidad de conocer datos sobre la aplicación y seguir a esta en las redes sociales.

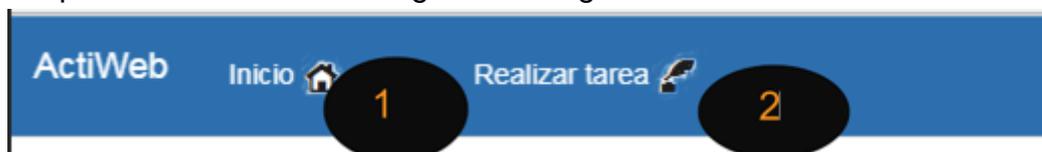
Acciones disponibles para el perfil Alumno

Como se detalla en el menú Alumno, un usuario con este perfil puede realizar las siguientes acciones:

Volver a la página principal: esta opción permite al alumno volver a la pantalla principal de su perfil en la aplicación.

Realizar tarea: esta opción permite al alumno realizar una tarea y enviar su solución para la corrección posterior del profesor, esta tarea fue creada previamente por el profesor.

Estas se pueden observar en la siguiente imagen:

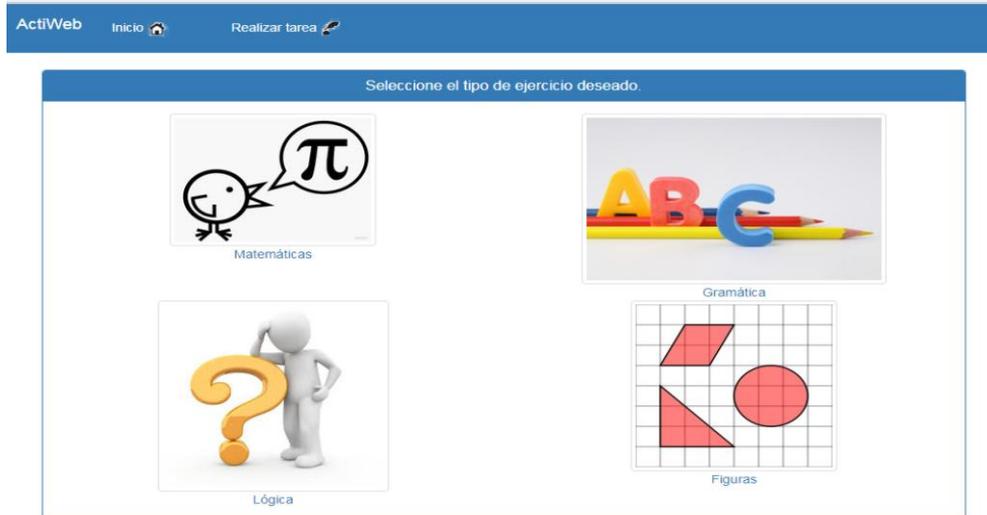


Para su versión móvil, es necesario pulsar el botón señalado para ver las opciones de menú:



Acciones detalladas para perfil Alumno

Como se ha mencionado en el apartado menú de acciones disponibles, en este apartado sólo tendríamos una para el alumno, la cual es la de realizar una tarea. Para ello accedemos primero desde el menú de alumno y encontramos la siguiente página:



Una vez seleccionada la sección y elegido el tipo de ejercicio específico, nos aparecerá un listado; al seleccionar uno de sus elementos se podrá rellenar la solución:

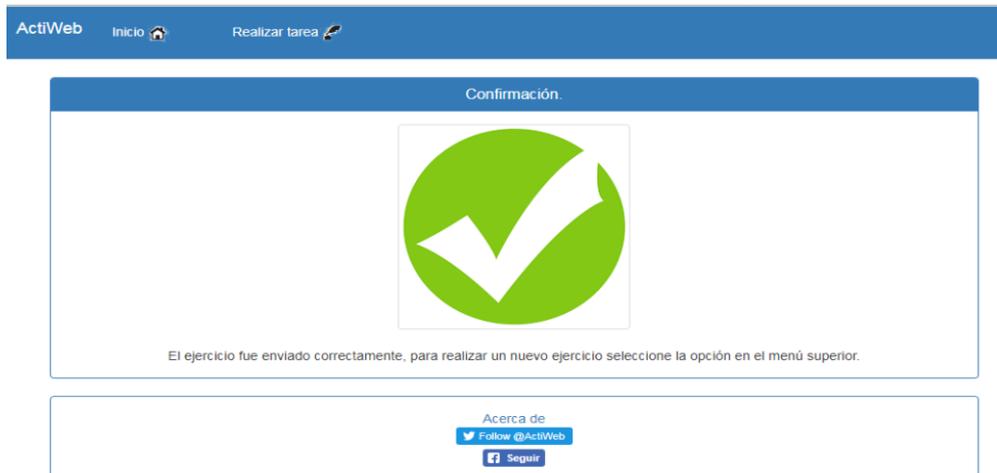
The screenshot shows the task resolution interface in ActiWeb. At the top, there is a navigation bar with 'ActiWeb', 'Inicio', and 'Realizar tarea'. Below this, a blue header reads 'Procede a resolver la división:'. The main area contains a division problem: $1987 \div 3 = 664$. The numbers are entered in input fields. Below the problem, there are three buttons: 'Click para corregir', 'Enviar resultado', and 'Acerca de'. The 'Acerca de' button has two sub-buttons: 'Follow @ActiWeb' and 'Seguir'.

En esta página de resolución de una tarea podemos observar que tenemos dos botones bastante importantes:

Clic para corregir: este permite corregir la solución que hemos dado tantas veces como queramos.

Enviar resultado: este permite enviar la solución que hemos dado al problema

Una vez enviada la solución, nos aparecerá la siguiente página de confirmación:



Por último, se muestra como a continuación se aprecia que, incluso en un smartphone puede realizarse un ejercicio:

