



Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Informática

Trabajo de Fin de Grado

Aplicación web responsiva para el
diseño de actividades educativas.

Responsive web app to design educational activities.

Krishna Hemnani Sadarangani

La Laguna, 7 de Junio de 2016

D. **Carina Soledad Gonzalez Gonzalez**, con N.I.F. 54.064.251-Z profesor Titular de Universidad adscrita al Departamento de Ingeniería Informática y de Sistema de la Universidad de La Laguna, como tutor.

D. **Yeray Del Cristo Barrios Fleitas**, con N.I.F. 54.106.627-R investigador adscrito al Departamento de Ingeniería Informática y de Sistema Universidad de La Laguna, como cotutor.

C E R T I F I C A (N)

Que la presente memoria titulada:

“Aplicación web responsiva para un diseño de actividades educativa.”

ha sido realizada bajo su dirección por D. **Krishna Hemnani Sadarangani**, con N.I.F. 45.940.836-S.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 7 de Junio de 2016.

Agradecimientos

Esta memoria representa la finalización de largo periodo de años, de esfuerzo y de formación en la carrera universitaria como Ingeniero Informático, en la que he recibido el apoyo y la confianza de mis familiares y amigos, quienes han acompañado en cada momento y a los cuales deseo agradecerle esta oportunidad.

Además quería aprovechar dar la gracias a todos los profesores que me han impartido clases o prácticas en cada curso, de la etapa universitaria y han aportado sus conocimientos para ilustrarme el camino para llegar a ser un ingeniero.

En especial quería agradecer a mi tutora y cotutor, que siempre han estado atentos y se han preocupado de que este proyecto salga adelante.

Finalmente y de todo de corazón, quiero agradecer a mis padres que siempre han dado el apoyo moral que necesitaba, estando en las buenas y en las malas. Gracias de verdad.

Resumen

El objetivo de este trabajo ha sido realizar una aplicación web adaptable y multidispositivo para la enseñanza de las matemáticas básicas. Principalmente, esta aplicación permite que educadores puedan diseñar sus propias actividades educativas, que más tarde podrán utilizarla con sus estudiantes. De esta forma, el profesorado puede crear secuencias de aprendizajes adaptadas a sus necesidades.

Palabras clave: Aplicación web, diseño adaptable, educación y TIC, herramientas de diseño para educadores, Diseñar, Niños, Divertidas.

Abstract

The target of this project is to make a responsive web app that you can run in any electronic device existing, for basic math education. Explicitly, this application allows the teachers design there own activities that can be done later, by their students. This way, the teachers create sequences of adapted education to their need.

Keywords: Web App, Responsive, Education, Teachers, Design, Kids, Math, Fun and Entertaining.

Índice General

Capítulo 1. Introducción	5
Proyecto DiverAP	5
1.1. Antecedentes.	5
1.2. Análisis.....	7
Capítulo 2. Planificación	8
2.1. Objetivos y Alcance.	8
2.1.1. Objetivos.....	8
2.1.2. Alcance	8
2.2. Cronograma	8
2.3. Costos.....	10
2.3. Lenguaje de programación y tecnología usada.....	12
Capítulo 3 Diseño.	13
3.1. Análisis.....	13
Diagramas de casos de uso.....	13
3.1.1. Editar Plataforma.....	14
3.1.2 Iniciar sesión	15
3.1.3 Realizar ejercicios.....	15
3.1.4 Visualizar Progreso	16
3.1.5. Preparar Clases.....	17
3.1.6. Realizar informes	18
3.2. Diseño de Bases de datos	19
3.3. Diseño de Interfaces.	21
Capítulo 4. Desarrollo	23
4.1. Selección de plantilla.....	23
4.2 Estructura de proyecto	25
4.3. Login	26

4.4.1. Home Profesor	27
4.4.2. Home Alumno	28
4.5. Editor de actividades.	28
4.5.1. Agregar Tema	29
4.5.2. Agregar Actividad.....	29
4.6. Ejecución de Actividades.	31
4.7. Interfaz multidispositivo	32
Capítulo 5. Conclusiones y líneas futuras	33
Bibliografía	36
Anexo A (Casos de uso extendido).	37
Anexo B (Diseño de bases de datos).	44
Anexo C (Diseño de Interfaces).	49

Índice de figuras

Figura 1.1. Divermates.

Figura 1.2. Ematic.

Figura 2.1 Diagrama de Gantt.

Figura 2.2. Diagrama de presupuesto.

Figura 2.3. Tecnologías usadas.

Figura 3.1. Diagrama de casos de uso general.

Figura 3.2. Diagrama de caso de uso, editar plataforma.

Figura 3.3. Diagrama de caso de uso, iniciar sesión.

Figura 3.4. Diagrama de caso de uso, realizar ejercicio.

Figura 3.5. Diagrama de caso de uso, progreso.

Figura 3.6. Diagrama de caso de uso, preparar clase.

Figura 3.7. Diagrama de caso de uso, realizar informes.

Figura 3.8. Tablas de MySQL.

Figura 3.9. Mockup de landing page.

Figura 3.10. Mockup de pantalla de la aplicación web.

Figura 3.11. Mockup de pantalla home de alumno.

Figura 3.12. Mockup de pantalla home de profesor.

Figura 4.1. Elección y evolución del logo definitivo.

Figura 4.2 Plantilla de DiverAP.

Figura 4.3. Estructura de proyecto.

Figura 4.4. Landing page DiverAP.

Figura 4.5. Pantalla de home DiverAP de alumno.

Figura 4.6. Pantalla de home DiverAP de profesor.

Figura 4.7. Pregunta de respuesta sencilla.

Figura 4.8. Pregunta Verdadero o Falso.

Figura 4.9. Pregunta multirespuesta.

Figura 4.10. Pregunta multirespuesta.

Figura 4.11. Código, nota de la respuestas de alumno

Figura 4.12. Código, nota de la actividad en general.

Figura 4.13. Ejercicio ejemplo.

Figura 4.14. Landing page responsivo.

Figura 4.15. Home alumno responsivo.

Figura 4.16. Home profesor responsivo.

Capítulo 1. Introducción

En este capítulo se explicará en que consiste el proyecto DiverAP, los antecedentes, el análisis de la competencia, la necesidad del desarrollo, los objetivos marcados y el alcance de este trabajo de fin de grado.

Proyecto DiverAP

El proyecto surge a raíz de un proyecto anterior llamado DiverMates[1], una aplicación desarrollada en Flash y Python que permitía a los alumnos llevar a cabo ejercicios de matemáticas en un entorno gamificado. La principal motivación de este trabajo es migrar la interfaz del sistema, creada en Flash, a tecnologías web actuales tales como HTML5, CSS3 y Javascript y añadir un módulo para el diseño de actividades destinada a los educadores. Además, otras motivaciones son el agregarle algunas mejoras, tales como el diseño adaptable, el despliegue para múltiples dispositivos y un motor de ejercicios más potente.

1.1. Antecedentes.

Como ya se ha mencionado anteriormente, este proyecto se basa en un proyecto anterior denominado DiverMates, diseñado para ofrecer una formación interactiva a alumnos con síndrome de Down. Se trata de una aplicación de escritorio que ofrece una formación basada en problemas y ejercicios de matemáticas, con un foco especial en la gamificación. El hecho de ser una aplicación significa que, si un usuario quiere utilizar la aplicación, antes debe instalarla en su ordenador personal. Esto además no se puede monitorizar en tiempo real ya que los datos manejados por los alumnos que están usando la aplicación en ese momento. Divermates incluye además el análisis del proceso de razonamiento del alumno en la resolución de los algoritmos. Este proceso se realiza aplicando técnicas de minería de datos que

permiten detectar patrones de interacción del alumno con su sistema, agruparlo y determinar el tipo de error, para luego brindar las pistas y ayudas que guíen al alumno en la resolución de los ejercicios de sumas y resta.

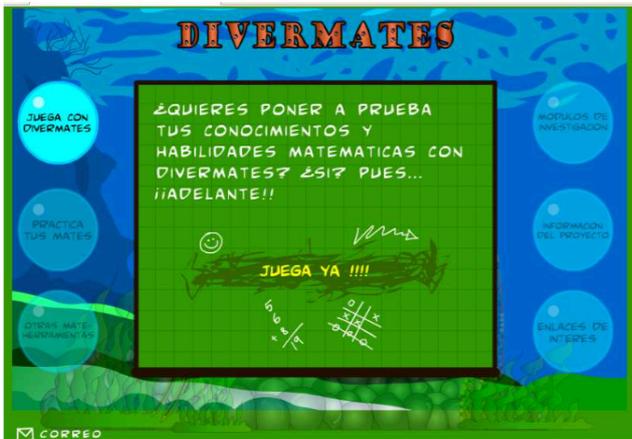


Figura 1.1 DiverMates

Es importante destacar que si bien en un principio el sistema se pensó para trabajar con alumnos con síndrome de Down, éste podrá ser utilizado por cualquier tipo de alumno.

Otro proyecto relacionado y tomado como punto de partida, es EMATIC (Enseñanza de Matemáticas por medio de las TIC) [2], que está dirigido especialmente a niños y niñas con dificultades de aprendizaje. Este sistema web permite a los usuarios aprender matemáticas al tiempo que se divierten con el desarrollo de actividades virtuales. El objetivo es el de formar en materia de matemáticas, pero también el de mejorar las capacidades cognitivas encargadas de la parte más lógica de nuestro cerebro. Parte de los elementos gráficos usados en este TFG han sido tomados de otro de los antecedentes de este proyecto.

EMATIC destaca por su adaptabilidad y despliegue en cualquier tipo de dispositivos como ordenadores, tabletas o móviles.



Figura 1.2 Ematic

En comparación con aplicaciones en el mercado cabe destacar Smartick[3], es una aplicación para sistemas operativos Android e IOS que educa a los niños a través de la realización de actividades de matemáticas. Al igual que Ematic, Smartick también puede ejecutarse en múltiples dispositivos.

Los alumnos que usan Smartick se enfrentan a diversos retos y con ello ganan puntos (ticks) según el resultado de los mismos. Este refuerzo positivo se convierte por tanto en parte de la mecánica de juego.

Algunas de las características sobre las que destaca esta aplicación son: sesiones cortas diarias, contenido adaptativo en tiempo real y feedback inmediato al alumno. Con sesiones diarias de 15 minutos, los ejercicios se van adaptando en tiempo real a la respuesta del alumno.

Tras cada sesión, el responsable del alumno recibe un correo electrónico con el resumen de la actividad. De forma que el educador pueda ajustar el plan docente aplicado al alumno.

1.2. Análisis

Los niños de hoy en día interactúan y acceden a dispositivos electrónicos a una edad cada vez más temprana. Para estos, usar la tecnología es un derecho de nacimiento y no conciben que una aplicación se vea en un dispositivo si y en otro no. Por otro lado, el despliegue de aplicaciones multidispositivo permite alcanzar un rango más grande de posibles usuarios, con lo que abarcamos también segmentos de usuarios con un nivel de recursos medio-bajo. Por suerte, las tecnologías actuales permiten que esto pueda suceder de una forma cada vez más fácil, rápida y barata.

El desarrollo de DiverAP permite aunar lo mejor de cada uno de estos proyectos y crear un sistema con un aspecto renovado y unas funcionalidades añadidas. También se hará esfuerzos en otorgar un papel más protagonista a los educadores, convirtiéndolos en generadores de conocimiento, y permitiéndoles diseñar y compartir actividades de cualquier tipo o materia.

Capítulo 2. Planificación

2.1. Objetivos y Alcance.

En este apartado se definen los objetivos y alcance del proyecto DiverAP.

2.1.1. Objetivos

Como objetivo principal de este proyecto consiste que la aplicación web sea responsiva a cualquier dispositivo.

Por otra parte, se han fijado los siguientes objetivos específicos:

- Diseñar e implementar la base de datos garantizando su optimización y consistencia, teniendo en cuenta que más adelante se pueda reutilizar.
- Permitir que se puedan diseñar diferentes tipos de ejercicios.
- Disponer de una interfaz usable.

2.1.2. Alcance

Se han desarrollado las características mínimas de una webapp, tales como: sistema de acceso o login, interfaces dinámicas basadas en los roles de educador y alumno y editor básico de ejercicios matemáticos.

2.2. Cronograma

Todo proyecto requiere de una planificación de su ejecución. En el siguiente gráfico se detallan los hitos a realizar y los tiempos estimados:

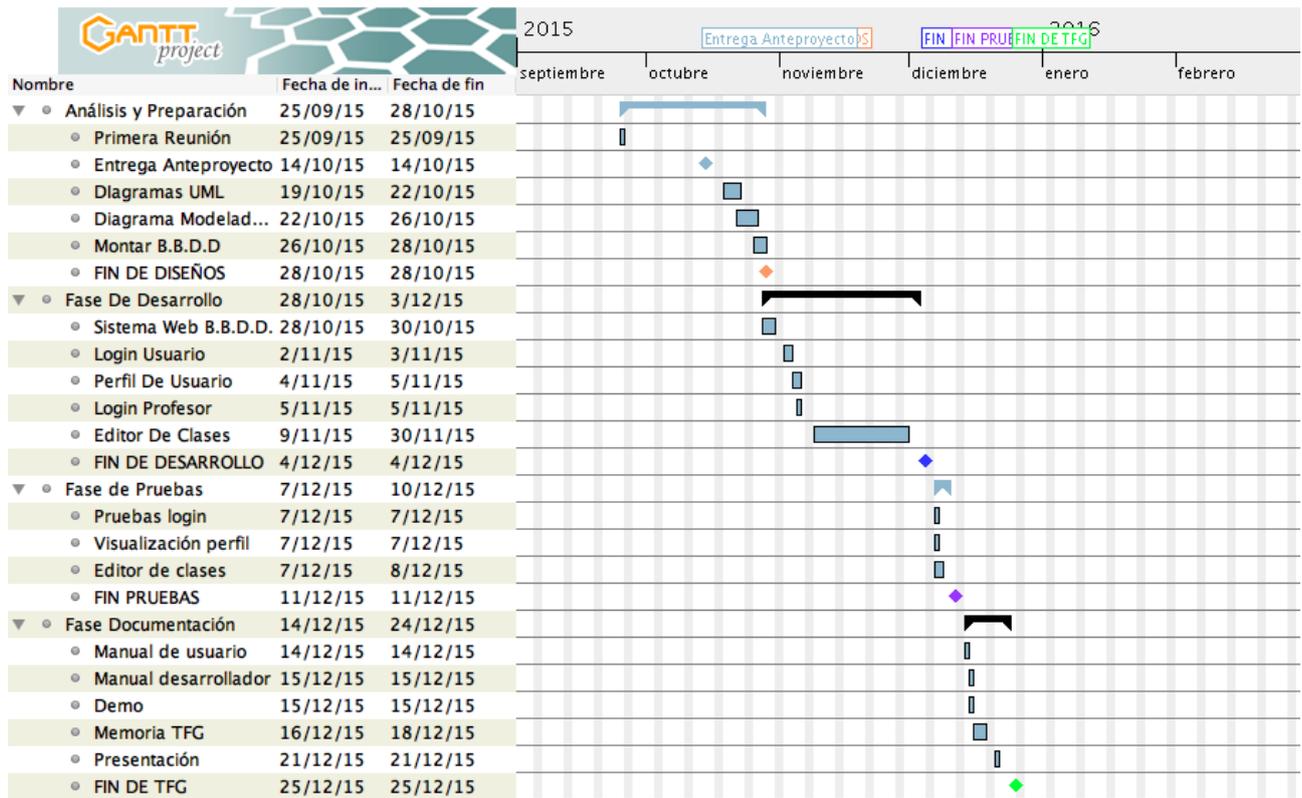


Figura 2.1. Diagrama de Gantt

A continuación se detallan cada una de las frases con su hitos correspondientes

Fase I: Preparación

Hito 1: Entrega de anteproyecto (Planificación). Se realizó el diagrama de Gantt y un plan de trabajo, para tener una idea de cuando y como se va realizar cada tarea.

Fase II: Diseño

Hito 2: Diseño

- Diagramas UML
- Modelado De Datos
- Diseño de Interfaz gráfica.

Fase III: Desarrollo y Pruebas

Hito 3: Desarrollo

- Login de usuario.
- Interfaz dinámica.
- Editor de clases personalizadas.

Hito 4: Test de pruebas

- Prueba uno
- Prueba dos

Fase IV: Documentación y Cierre.

Hito 5: Manual usuario

- Manual de usuario
- Guía de funcionamiento de la app.
- FAQ
- Perfil de profesor
- Perfil de usuario.

Hito 6: Demo. Video de una pequeña demostración de la aplicación.

Hito 7: Elaboración de Memoria Final. Finalizado la memoria del trabajo fin de grado y preparación de la presentación.

2.3. Costos

Tras analizar los costes derivados del desarrollo del software, se han presupuestado las fases de esta forma:

Fase I: Preparación	Carga de trabajo de 3%	114,00€
Fase II: Diseño	Carga de trabajo de 19,30%	1056,50€
Fase III: Desarrollo y Validación	Carga de trabajo de 67,70%	4157,80€
Fase IV: Documentación y Cierre.	Carga de trabajo de 10,00%	84,00€

Nos encontramos con un total de presupuesto de proyecto de 5172,30 €

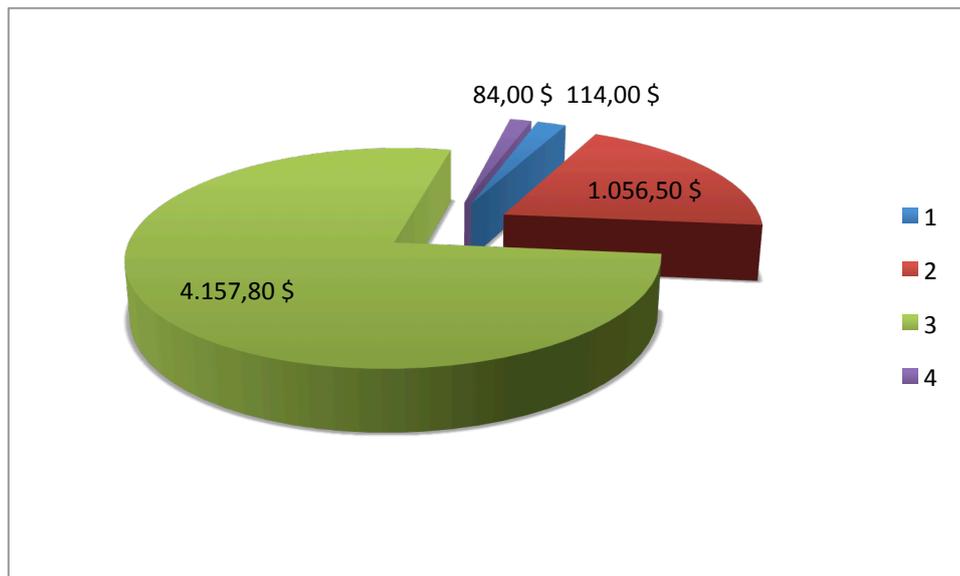


Figura 2.2 Diagrama de presupuesto

2.3. Lenguaje de programación y tecnología usada

Para la realización de este trabajo fin de grado, se han utilizado las siguientes tecnologías:

- Lenguajes de programación: HTML, CSS, Javascript, SQL y PHP(AJAX).
- Técnica desarrollo web en cliente: AJAX[7].
- Entorno de desarrollo: Brackets (1.6.0)[4].
- Servidor virtual: XAMPP(5.6.12-0)[5].
- Programa para la creación de logotipo y retoques del contenido web: Adobe Photoshop CS6 y Adobe Illustrator[6].
- Frameworks de desarrollo: Bootstrap (v3.1.1).

El uso de los lenguajes de programación web, HTML, CSS, Javascript, SQL y PHP, conforman una de las tecnologías más usadas en el desarrollo web en los últimos años. La combinación de estas tecnologías permite diseñar e implementar una página web de forma ágil, eficiente y con características añadidas con la adaptabilidad móvil y el esquema multidispositivo.



Figura 2.3. Tecnologías usada

Capítulo 3 Diseño.

En este capítulo se detallan los resultados del análisis realizado en la fase inicial del proyecto. Se desglosa en el análisis de casos de uso y casos de uso extendido, el esquema de la base de datos y el prototipado de las interfaces.

3.1. Análisis

El uso de la notación gráfica del lenguaje UML, permite definir, analizar y visualizar con detalle todas las funcionalidades de que debe disponer el software.

Diagramas de casos de uso

El primer caso de uso que nos encontramos es un caso de uso general para todo el sistema de DiverAP, en el cual existen tres tipos de actores principales: Profesor, Alumno y Desarrollador/Administrador.

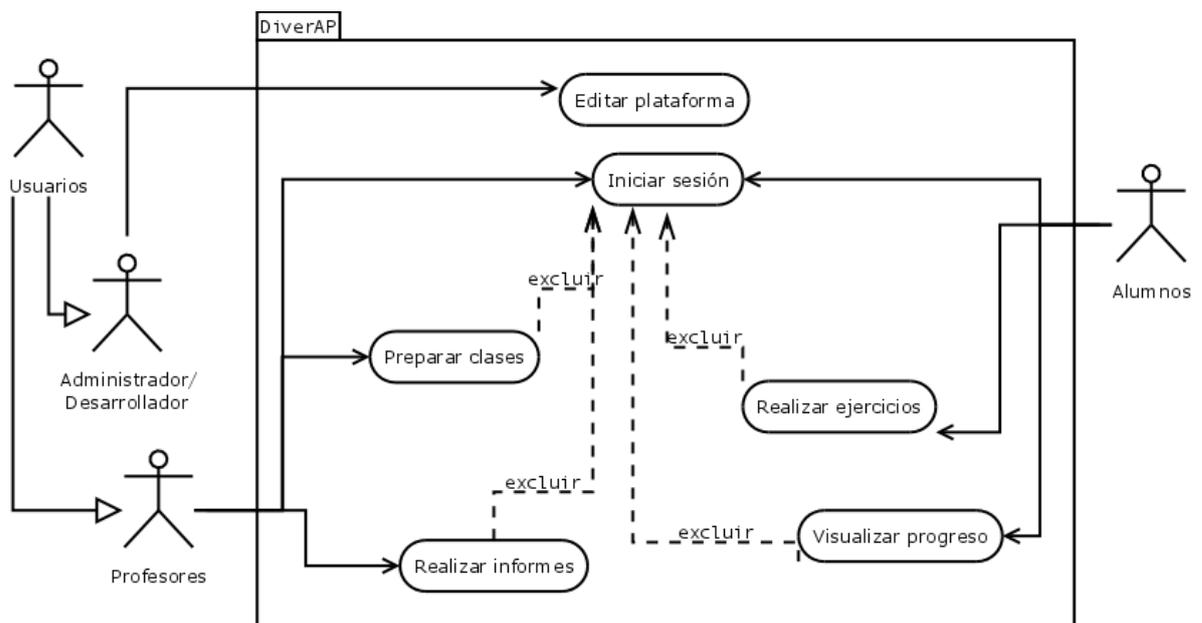


Figura 3.1 Diagrama de casos de uso general

Cada uno de los seis casos de uso se divide a su vez en múltiples sub-casos de usos. Algunos de estos solo suceden si el usuario ha accedido a la plataforma a través del inicio de sesión. Por esta razón este caso de uso es fundamental, ya que desde él se pueden llegar a otros diferentes casos.

3.1.1. Editar Plataforma.

El contexto de este caso de uso se desarrolla en el back-end de la aplicación web, con funcionalidades que permiten configurar y llevar a cabo tareas de mantenimiento con un rol de desarrollador y/o administrador.

Cómo tareas principales de este actor destacamos lo siguiente:

- Garantizar la protección de datos: Esta tarea es para comprobar si los educadores están incumpliendo la ley de protección de datos para los distintos ejercicios que desarrollan y sus respectivos contenidos, como imágenes, videos, etc, que será verificado por el administrador.
- Comprobación de credenciales: Ésta última tarea es esencial ya que en esta aplicación se almacenan información personal de los clientes, que no deber estar a disposición de cualquier usuario. Por esta razón, el administrador dispone acceso para verificar datos, especialmente relacionados al acceso de la plataforma (usuario y contraseña).

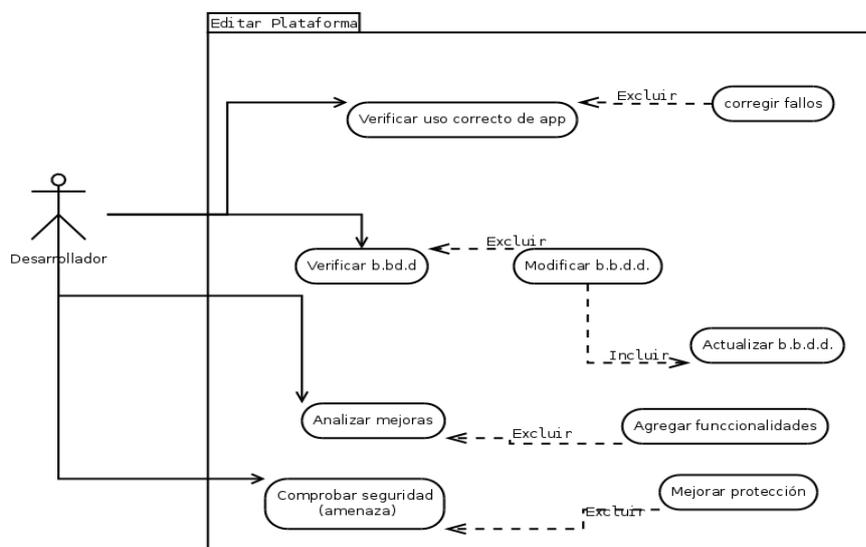


Figura 3.2. Diagrama de caso de uso, editar plataforma.

3.1.2 Iniciar sesión

El control del acceso por clave y contraseña, también conocido como login, permite conceder o denegar el acceso a los usuarios registrados en la plataforma. Los datos introducidos por los usuarios realizan una comprobación en el servidor de bases de datos (phpmyadmin). Si se recibe una respuesta, se les permitirá el acceso a la página principal, el home de cada perfil correspondiente. En caso contrario, salta un mensaje de error.

Si el usuario ha accedido recientemente a la plataforma, y aun se conservan las cookies de las credenciales, se redirecciona automáticamente a la página de inicio de la aplicación.

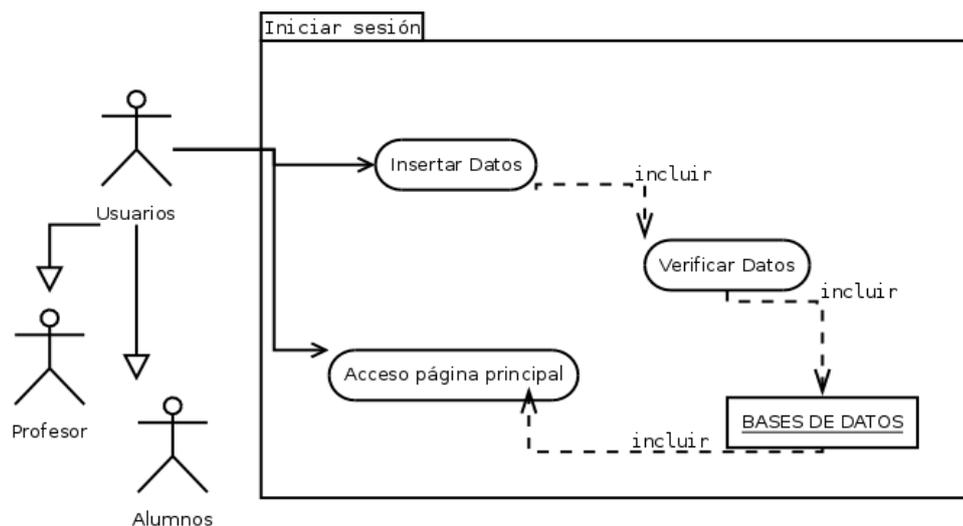


Figura 3.3. Diagrama de caso de uso, iniciar sesión

3.1.3 Realizar ejercicios.

El tipo de ejercicio condiciona la forma en la que se lleva a cabo. De esta forma, el usuario puede encontrarse con situaciones en las que pueda hacer pausas para descansar, y otras en las que se vea obligado a finalizar el ejercicio. La opción de abandonar el ejercicio siempre está disponible, pero como penalización, el estudiante recibirá un 0 como calificación. Al finalizar la actividad se muestra un resumen de los resultados obtenidos de la actividad.

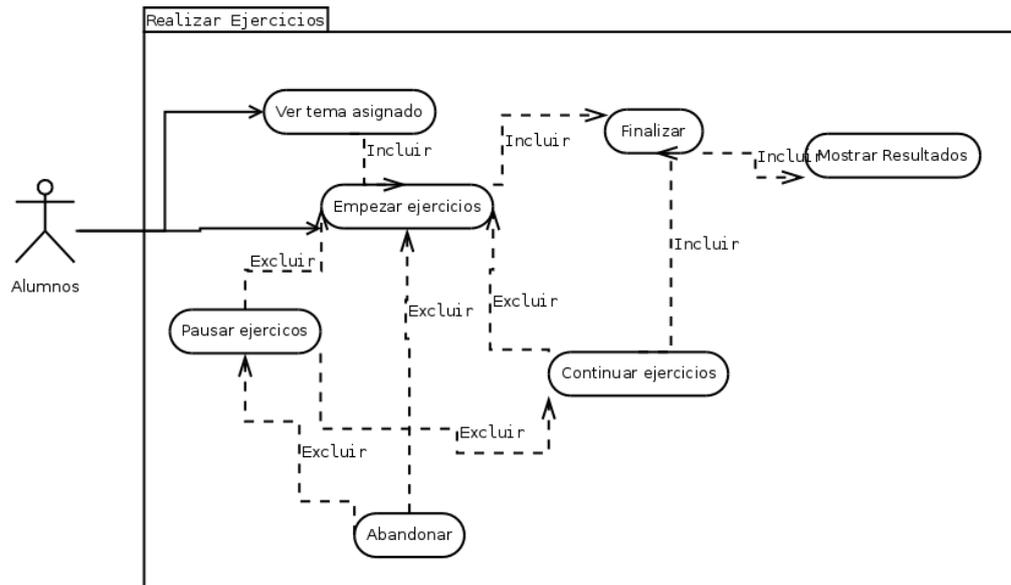


Figura 3.4. Diagrama de caso de uso, realizar ejercicio.

3.1.4 Visualizar Progreso

Los ejercicios finalizados por el estudiante, así como los que tiene pendiente de acabar, se pueden visualizar en nuestro panel de progreso. Dicho panel muestra además la progresión obtenida hasta el momento. Esta consecución de trabajo permite orientar al estudiante en su aprendizaje. Un usuario que ha desempeñado con éxito un grupo entero de ejercicios puede desbloquear nuevos grupos de ejercicios con distinto nivel de dificultad.

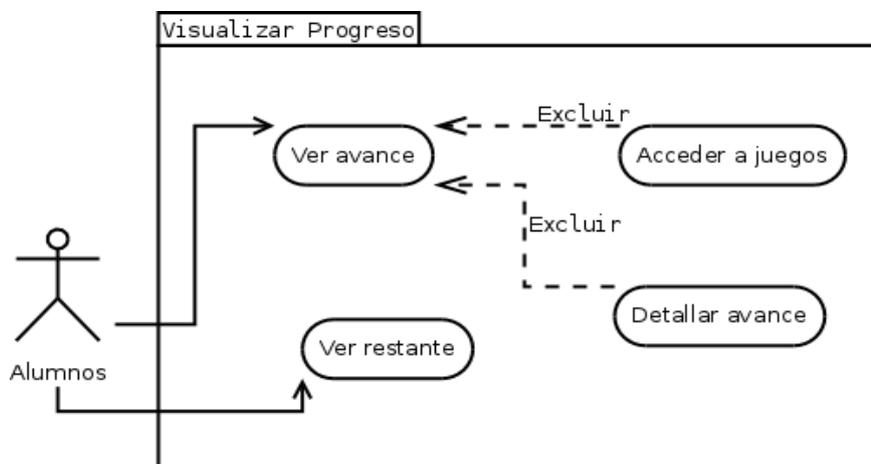


Figura 3.5. Diagrama de caso de uso, progreso.

3.1.5. Preparar Clases

El educador es quien lleva a cabo la elaboración de los ejercicios. En el proceso, deberá aportar una serie de datos que configuran el tipo de ejercicio y su contenido. Algunas de estas variables son:

- Tema o materia
- Tipo de ejercicio
- Formato
- Archivos adjuntos
- Tiempo de respuesta
- Fechas de apertura y cierre
- Visibilidad

Esto puede llevar tiempo, de forma que se le da la opción al educador de pausar el proceso y finalizarlo más tarde. Una vez que el ejercicio ya se ha dispuesto para los estudiantes y que alguno de ellos ya ha resuelto, el proceso de edición del ejercicio queda bloqueado hasta que se cierre el plazo de resolución.

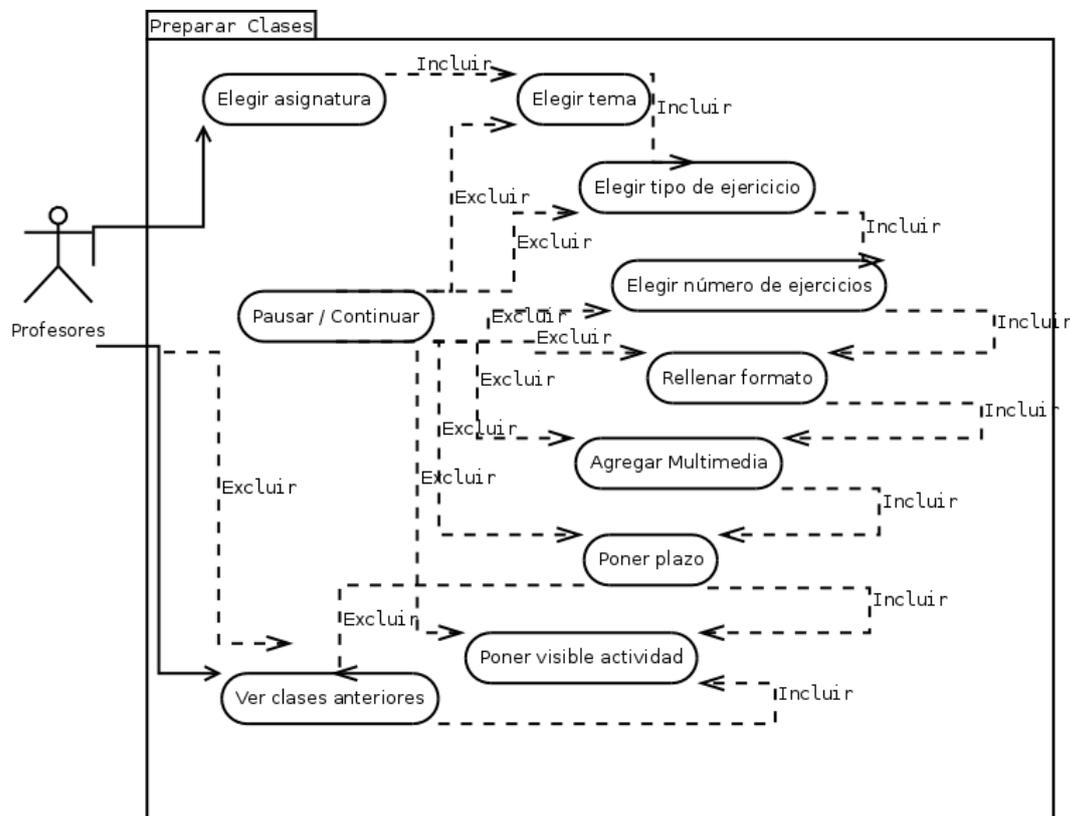


Figura 3.6. Diagrama de caso de uso, preparar clase.

3.1.6. Realizar informes

Los informe de estado y evolución son un parte fundamental de la labor docente en cualquier ámbito. Dichos informes pueden generarse en base a las asignaturas, temas, ejercicios, datos estadísticos, rendimiento o datos demográficos. Los informes pueden compartirse con otros usuarios en formato digital, dando además la posibilidad de aportar un feedback basado en consejos y/o comentarios.

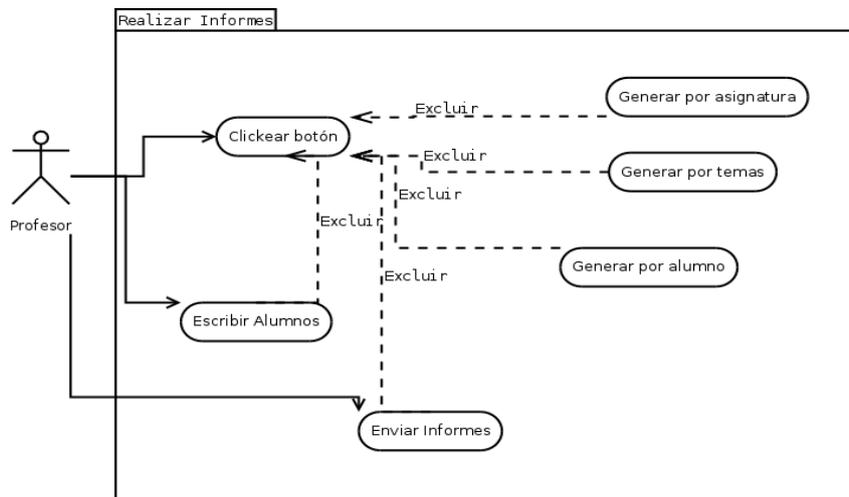


Figura 3.7. Diagrama de caso de uso, realizar informes.

Se describe en detalle a cada uno de los casos de usos, su casos de de uso extendido correspondiente. Para no extender el contenido de la memoria se incluye esto como apéndice.

Anexo A (Casos de uso extendido).

3.2. Diseño de Bases de datos

La base de datos que se ha diseñado, se apoyado siguiendo el esquema de bases de datos “entidad-relación”. Este esquema con su explicación y correspondiente modelo relacional, se ha añadido como Anexo B (Diseño de bases de datos)

La Figura 3.8. describe el esquema de tablas y relaciones existentes:

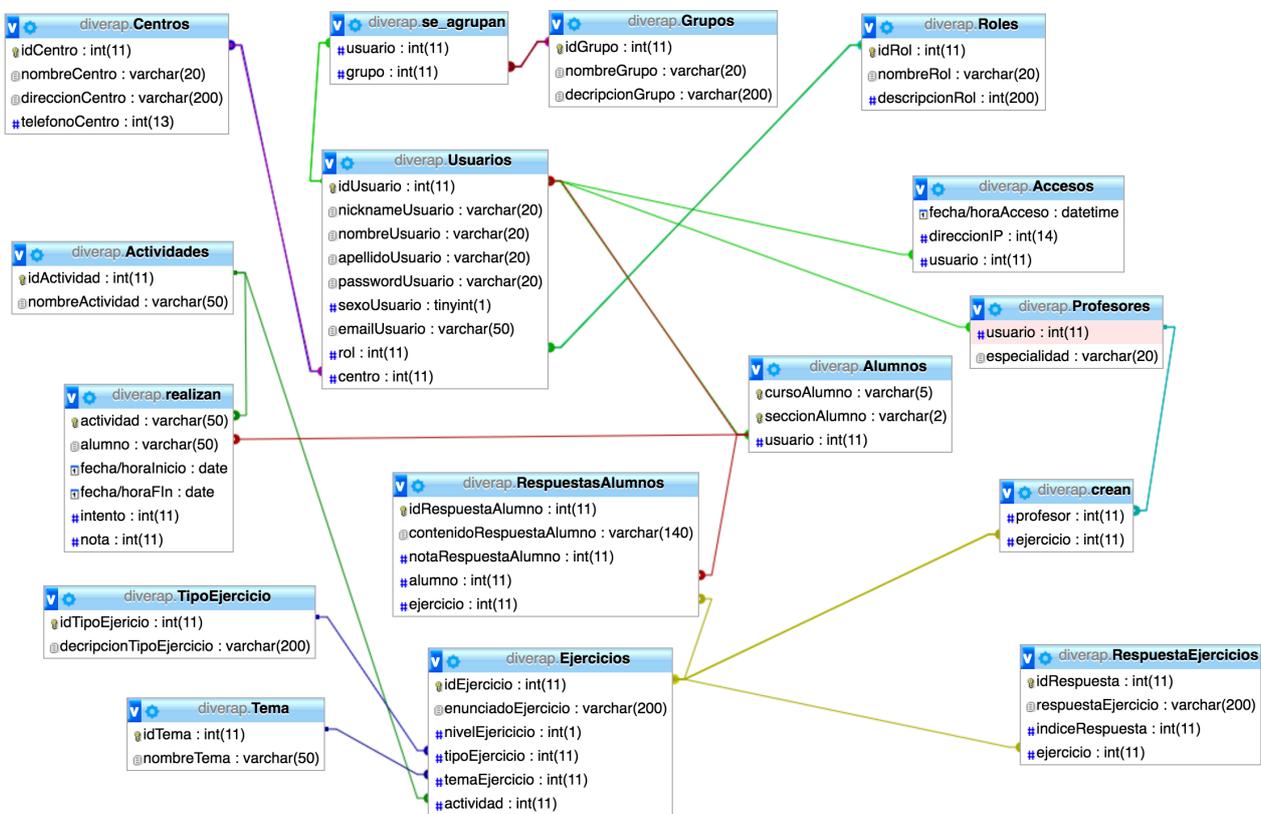


Figura 3.8. Tablas de MySQL

Como observamos tenemos dieciséis tablas, pero principalmente destaca entre todos, con alta concentraciones de relaciones, la tabla **Usuarios**. Esta tabla va almacenar todos los usuarios, indistintamente de su rol. Asimismo, para modular la información que almacenan las tablas, se aloja en partes, en las tablas **Centros** (lugar al que pertenece el usuario) y **Grupos** (el conjunto de personas con el que interactúa). Esta “agrupación” forma la tabla **se**

agrupan.

Los distintos tipos de usuarios distinguen por los papeles que pasan a tomar en la aplicación. Estos disponen de “identificadores de Rol”, creados en la tabla **Roles** (que podrá ser entre usuario de la plataforma o administrador) y permitirá cargar las interfaces de determinadas maneras.

En ningún momento debemos confundir esta relación a los sucesores de la tabla Usuario, **Alumnos** y **Profesores**, que se caracterizan por almacenar información adicional a parte de la tabla padre. (En el caso de Alumnos, el curso y sección que pertenece y por otra parte los Profesores, las especialidad que corresponde.

A parte de la tabla Usuarios con abundantes relaciones, la tabla Ejercicios funciona de la misma manera. Esta también distingue los ejercicios por “tipos de ejercicio” en la tabla **TipoEjercicio** (se contemplan tres tipos, verdadero o falso, de una respuesta y múltiple respuesta). La tabla **Tema** concreta el tema que corresponde el Ejercicio.

Los ejercicios son “creados” por Profesores. Por ello la tabla **crean** es relevante para almacenar quienes y cuantos creadores intervienen para hacer un ejercicio.

Cada ejercicio dispone de “respuestas” y es necesario guardar esta información en la tabla **RespuestasEjercicios**. Esta tabla va ser comparada con las “respuestas de los alumnos”. De esta manera es imprescindible la tabla **RespuestasAlumnos** que tendrá los resultados de los alumnos.

A todo esto se concreta que los ejercicios “pertenecen” a una **Actividad** específica. Estas actividades son **realizadas(tabla realizan)** por alumnos, que a su vez, se ha de tener cuenta, que estos realizan varias actividades.

Adicionalmente tenemos la tabla **Accesos** en la que se almacenan la direcciones IP y la fecha, hora de acceso. Esto es debido a los protocolos de seguridad que se desea añadir como funcionalidades en la aplicación.

Al tener verificadas las tablas se procedió a montar las bases de datos en el phpMyAdmin. Un servidor de bases de datos que nos ofrece el Xampp.

El uso fácil de este entorno web, su precio totalmente gratuito y su compatibilidad con el MAC, fueron las razones principales para escoger este programa.

3.3. Diseño de Interfaces.

Los mockups son una manera de diseñar las interfaces de la aplicación web en borrador, antes de comenzar a implementar el prototipo. Esto nos permite visualizar una base principal de la plataforma.

Destacamos aquí los principales mockups de las pantallas de DiverAP:

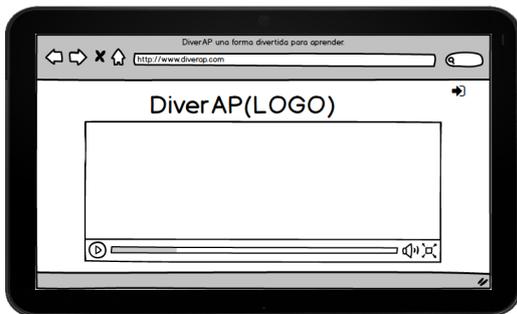


Figura 3.9. Mockup de la landing page.

Landing page. En esta pantalla dispondremos de un video introductorio explicando en lo que consiste la aplicación web. En la esquina tenemos botón para realizar el login a la plataforma.

Plantilla principal. Todas la pantallas una vez realizado el login, tendrán en principio, este formato. Se ubica arriba en la esquina-izquierda el logo de la aplicación y en el lado contrario, el botón para salir de la plataforma (cerrar sesión).

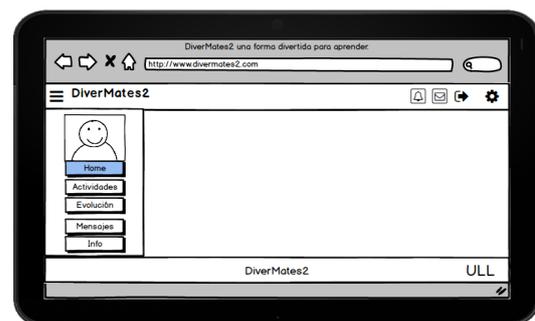


Figura 3.10 Mockup de la plantilla de la aplicación web

En el lateral izquierdo se sitúa el menú desplegable que podrá ocultarse al disminuir su resolución. Tenemos también un botón disponible en el caso de que quisiéramos una mayor visibilidad de la pantalla. Al final en el pie de la página, nos encontramos el footer de la aplicación.

Home. En la imagen en continuación se visualiza la página principal. Se caracteriza por tener un diseño similar a la **Figura 2.2.**, con la salvedad de que los accesos directos de la barra lateral cambian conforme el tipo de usuario que accede a la plataforma. Las funciones comunes a todos los roles son Inicio (la primera pantalla que se visualiza), Mensajes (se visualiza una especie buzón de mensajes) e Info (La información general del sistema. FAQ).

Para un estudiante, los accesos directos extras de que dispone, son actividades y evolución.

- Actividades, que le permitirá acceder a realizar los ejercicios que le corresponden
- Evolución, que corresponde ver el progreso y resultados de actividades ya realizadas por el estudiante.

El home de educador tendrá tres funcionalidades diferentes al de alumno, aunque una de ellas tenga el mismo nombre, su tarea es completamente diferente. La sección de actividades en el perfil profesor, al editor de texto que será la creación de actividades para alumnos. Otra de las características es informes, donde el educador accede a los diferentes tipos de resultados que el desea visualizar. Por último disponemos de grupos, donde el se tendrá un acceso directo a observar los grupos que pertenece o desea crear.



Figura 3.11. Home de estudiante.



Figura 3.12. home de educador.

Se puede encontrar un lista completa de todos los mockups en el Anexo C (Diseño de interfaces).

Capítulo 4. Desarrollo

4.1. Selección de plantilla

Antes de ponerse a desarrollar código desde cero, se trató buscar plantillas disponibles en la red. De las cientos y miles de modelos que estaban presentes, se limitó en escoger la apropiada y la que se ajustaba a la necesidad del la aplicación web. Entre todas, se decantó al final por AdminLTE.

AdminLTE es popular por su plantilla que ofrece tableros y unos paneles de control. A parte de ser responsive a HTML, la plantilla esta basada en el framework de Bootstrap3. También, usa todos los componentes de diseño de Bootstrap y remodela los típicos plugins que se usan para módulos de diseños que permite que sea más fácil y configurable. Estas eran unas de los y requerimientos que sea ajustaban totalmente con el proyecto.

Una vez elegida el diseño para nuestra plantilla, se procedió adaptar todas las funcionalidades, estilos y características al ajuste de la aplicación web.

En primer lugar se trabajó en la elección de un logo para la identidad visual de la plataforma. Dado que el proyecto se inspira en Divermates, se creyó conveniente utilizar también su símbolo, “La Lápiz”. El uso de personajes animados como técnica para acercarse más a los estudiantes es una buena estrategia que se ha decidido seguir.

Como método de inspiración, se investigaron múltiples web de diseño de logos online[8]. Finalmente se opto por la web <https://www.logaster.com.es/logo/>, una web de servicio comerciales gratuitos. El resultado del logo se muestra en la **Figura 4.1**. Posteriormente se rediseño el logotipo con la herramienta de Adobe Ilustratator con el fin de obtener una imagen vectorial, es decir, una imagen generada en tiempo de ejecución. Generar un imagen vectorial nos permite optimizar el tiempo de carga de la plataforma.

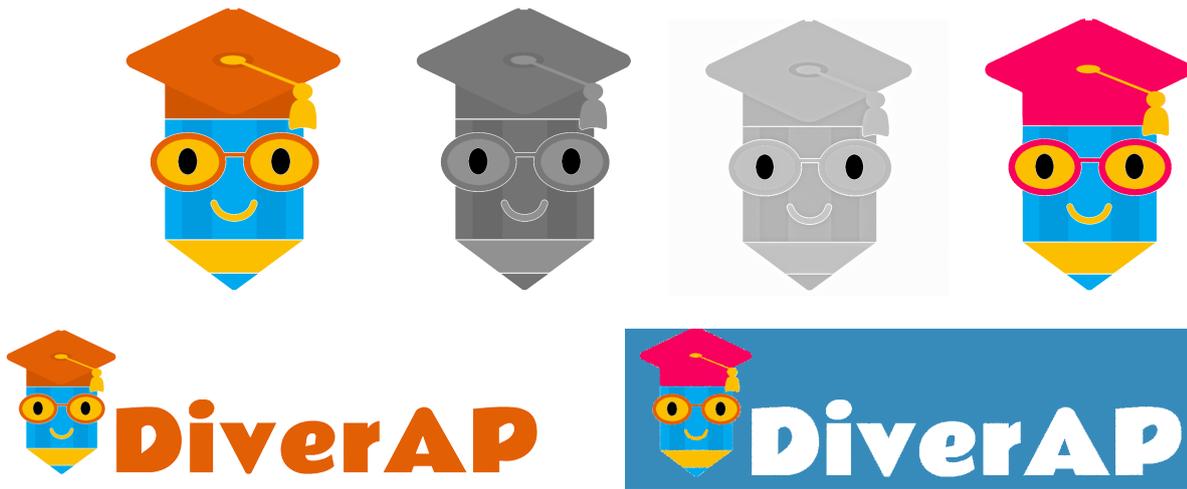


Figura 4.1 Elección y evolución del logo definitivo.

Al incorporar el logo se tuvo en cuenta los colores oscuros de la plantilla, por esta razón se escogió las letras en un coloro claro, en este caso el blanco. Luego a la plantilla se incorporó el nuevo logo y se procedió realizar un estudio de la estructura, tanto del proyecto, como el contenido del index.html.

En el fichero principal se implementa el contenido de la plantilla, que se caracteriza por su categorización de la estructura en un “wrapper” u otras palabra, envoltura. Dentro de esta envoltura se dispone de un “header main-header” que es la parte superior del portal. En él, se sitúa el logo, el botón desplegable del menú lateral y el botón para cerrar sesión. El menú desplegable se desarrolló en el “aside main-side-bar” que se modificó al ajuste del proyecto.

El contenido principal de la plataforma se define en “content-wrapper” que a su vez tiene su correspondiente secciones. Se eliminó todo el contenido de la plantilla AdminLTE para que más adelante se agregase y visualizará las funcionalidades de la aplicación.



Figura 4.2. Plantilla DiverAP.

Y por último, disponemos del footer que se reemplazó por el que se dispone en EMATIC. Tras los cambios, el resultado final se ve en la **Figura 4.2**.

Una vez lista la plantilla, se hizo una copia de la estructura html, y se copio al resto de fichero html. De esta forma, se garantiza una coherencia visual en todas las interfaces de la plataforma.

4.2 Estructura de proyecto

Una vez descargada la plantilla, realizadas las modificaciones y replicados los cambios al resto de interfaces, se integró el resultado en la estructura jerárquica del proyecto.

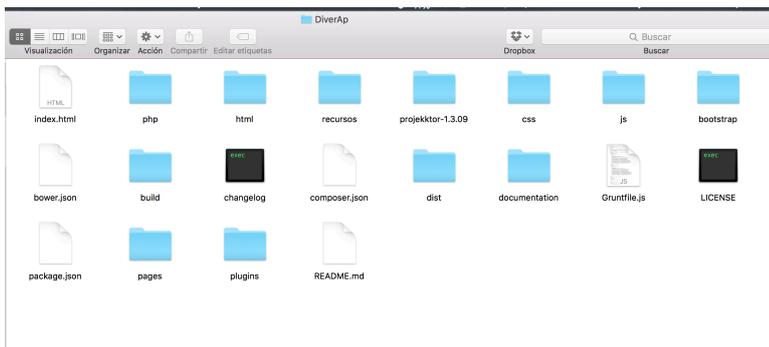


Figura 4.3 Estructura de datos.

Destacamos principalmente las siguientes carpetas:

- html (todos los htmls de la aplicación web).
- js (de los documentos javascript, los scripts de funcionalidad).
- css (todos los documentos de estilos css).
- php (todos los documentos que se generan conexión con las bases de datos).
- recursos (todos los contenidos multimedia, imágenes, audio, video, etc).
- index.html (Fichero inicial que indexa a la primera página de la app).
- config.xml (Archivo de configuración).

4.3. Login

La página principal o “landing page”, va tener una estructura bastante similar a la plantilla. El motivo de esto es por incorporar un video de presentación e informativo, que se reproduciría dependiendo de la elección del usuario, antes de iniciar sesión.

El sistema de login, se realiza poniendo una condición, sí se cumple (la consulta en la bases de datos en phpmyadmin, responde), accede. En el momento de insertar cualquier dato incorrecto, saltaría un error en la pantalla. De igual manera al introducir algún campo en vacío, salta el error.

Todo esto para evitar recargar la página, se ha realizado apoyando de Ajax. Al introducir los credenciales, los datos se envían con un post y si la consulta es válida, responde con un echo.

A continuación vemos en la **Figura 4.4** una captura de pantalla del landing page:

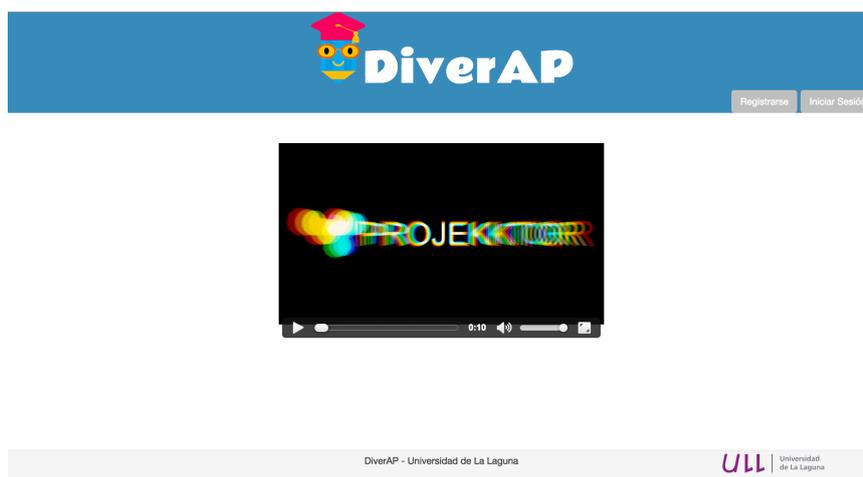


Figura 4.4. Landing page DiverAP.

4.4. Interfaz dinámica

Se debe tener en cuenta que el menú de la plantilla varía un poco, tras el acceso de los diferentes tipos de usuarios. Esto es para cargar determinado tipos de funcionalidades, dependiendo del tipo de usuario que esta accediendo. Los múltiples usuarios disfrutan de la misma aplicación pero sin tener que compartir las mismas interfaces. De esta manera el usuario tendrá la sensación que la aplicación ha sido diseñada única y exclusivamente para él y no para los usuarios con sus mismas características.

Aunque la estructura y contenido parezcan iguales, se incorporan algunas funcionalidades más en el perfil del profesor.

Estos cambios se visualizan en las siguientes dos figuras 4.5 y 4.6:



Figura 4.5. Home alumno DiverAP.

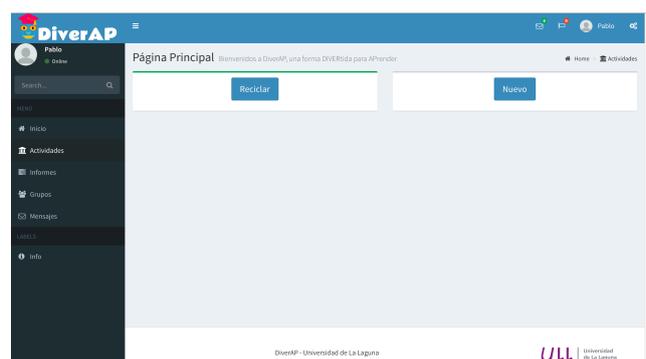


Figura 4.6. Home profesor DiverAP

Esto se realizó con una condición y consulta. Dicha condición ha de ser comprobada en el apartado anterior de la misma manera, a través de una consulta. Tras consultar la bases de datos e identificar de que tipo de rol corresponde al usuario, se carga su correspondiente perfil.

4.4.1. Home Profesor

El perfil de profesor destaca con una interfaz más sencilla y sin imágenes. Esto es para mostrar a los educadores las opciones de manera claras

y fácil de localizar. Se dispone de un menú lateral con las opciones de Actividades, Informes y Grupos.

- En actividades, el profesor podrá empezar crear o reciclar una actividad.
- Informes el profesor podrá generar y exportar los distintos tipos de información que desea obtener.
- Grupos. El profesor podrá acceder para visualizar los grupos que pertenece.

4.4.2. Home Alumno

El perfil de Alumno a parte de caracterizar con un menú lateral con las funcionalidades correspondientes, dos de ellas se visualizan en la pantalla principal.

- Actividades, será un acceso directo al conjunto de actividades que el alumno puede acceder para realizar.
- Evolución donde se visualizará el progreso y resultados.

Para que los alumnos puedan identificar y se sientan amigables con la interfaz se incorporan imágenes a estas opciones principales (que se han retomando del proyecto EMATIC).

4.5. Editor de actividades.

Esta funcionalidad, es la más importante de toda la aplicación, que pertenece al perfil del profesor.

Al comenzar a diseñar actividades nos encontramos primero con una lista de temas que existen actualmente. Todas las actividades pertenecientes a la plataforma van estar agrupadas por temas. Esta visualización se realiza con una consulta a la tabla **temas** de la base de datos, que carga toda lista que tenemos. Como ejemplo tenemos la **Figura 4.7**.

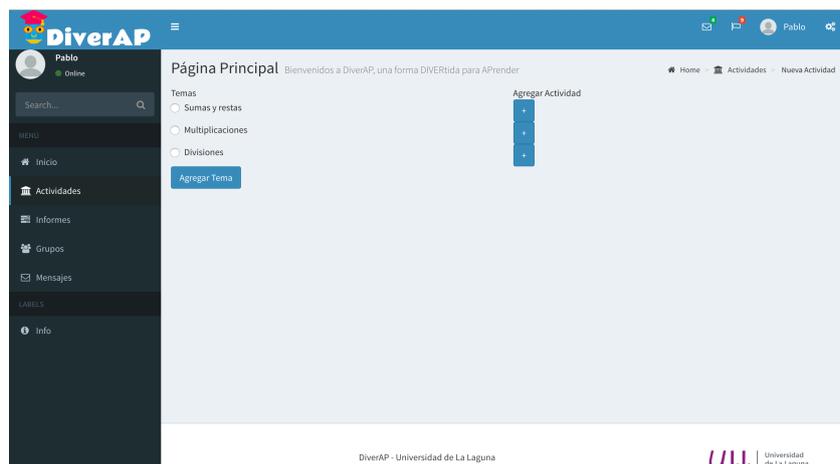


Figura 4.7. Home profesor DiverAP

En esta página visualizamos que podemos agregar tema o agregar actividad.

4.5.1. Agregar Tema

Al pinchar al botón de “Agregar Tema” se nos carga una página con su campo correspondiente para añadir un tema, que a su vez se agregará en la tabla Tema.

4.5.2. Agregar Actividad

Al agregar una actividad se nos carga la página previa antes de proceder rellenar el formulario de un ejercicio. Se tendría que rellenar los siguientes datos:

- Nombre de actividad
- Descripción actividad.
- Elegir tipo de ejercicio (combox).

Esto datos se almacenarán en la tabla Actividad.

4.5.3. Agregar Ejercicio

A continuación dependiendo del tipo de ejercicios que se haya escogido, se carga su correspondiente plantilla. Disponemos de tres tipos que se visualiza en las siguientes figuras.

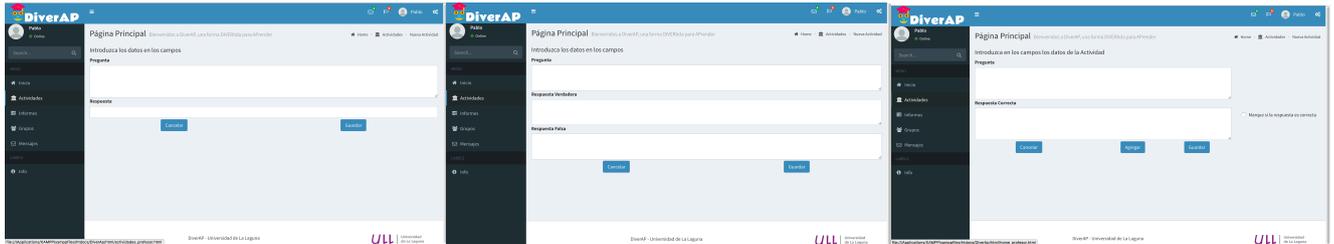


Figura 4.8.Respuesta simple Figura 4.9.Verdadero o Falso Figura 4.10. Múltiple respuesta

El enunciado se almacenará en la tabla Ejercicios y las respuestas en la tabla RespuestasEjercicios. El índice de la respuesta correcta se asigna automáticamente a uno cuando se diseñan ejercicios de tipo verdadero o falso y respuesta sencilla. En el caso de ejercicios múltiples, se divide entre los números de veces que se haya marcado respuesta correcta. Es importante tener en cuenta que a veces al realizar esta operación nos puede dar un número decimal aproximado. Para ello, es importante agregar el resto a resultado obtenido de los índices de estas respuestas. De esta manera el alumno tendrá la posibilidad de la máxima puntuación. El código implementado se explica esto en la en la **Figura 4.11**

```
22
23 indiceRespuesta = 1 / nRespuestasCorrecta;
24
25 if (indiceRespuesta == 1 )
26     indiceRespuesta= 1;
27
28 else if (indiceRespuesta !=1)
29 {
30     Resto= 1-indiceRespuestas;
31     indiceRespuesta+= Resto;
32
33 }
34
```

Figura 4.11. Código asignación de índice a respuestas correctas

Una vez hallada los índices de respuestas correctas, al comparar con las de alumno, se suma a las que haya respondido y con esto obtendremos el resultado final de la actividad.

```
while (indiceRespuestaAlumno | > 0 )  
    NotaActividad += indiceRespuestaAlumno;
```

Figura 4.12. Código, nota de actividad de alumno.

4.6. Ejecución de Actividades.

Las ejecución de las actividades es el resultado que se genera tras la creación de la actividad. Como visualizamos en la **Figura 4.13**, en esta sección simplemente cargamos lo datos de nuestros ejercicios a sus correspondientes actividades.

El estudiante al proceder realizar esta actividad, sus respuestas, se van almacenando en la tabla de RespuestasRespondidas, esto ocurre al clicar el botón “guardar”. De igual forma, se van comprobando la validación de estas respuestas.

Por último para mandar a evaluar la actividad entera, al realizar el último ejercicio se debe pinchar en el botón “finalizar actividad”. Se realiza el cálculo de la suma de la puntuación (cuyo cálculo fue realizado en el momento de responder la respuesta) y se genera y almacena el resultado general de la actividad (que fue calculado sumando el resultado de cada ejercicio).

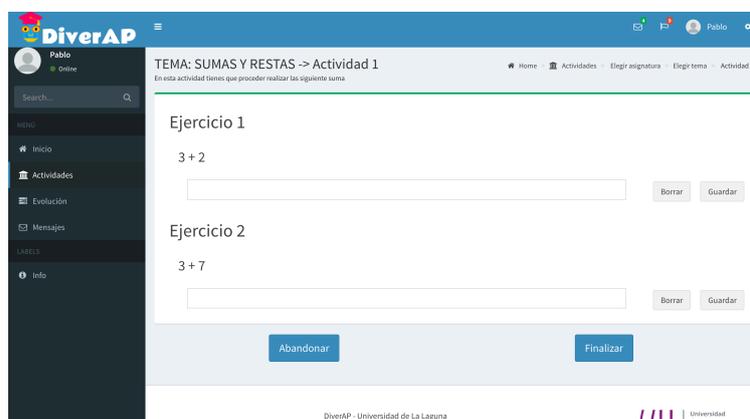


Figura 4.13. Ejercicio ejemplo

4.7. Interfaz multidispositivo

Para convertir esta plataforma en una aplicación web con diseño responsivo se basó en el framework de Bootstrap, cuya estructura permite este resultado final.

Esta tecnología se caracteriza por trabajar en un especie de sistema de cuadrilla, doce columnas cuyo ancho llega a 940 píxeles. En esta aplicación debajo de los 767 píxeles la plataforma, pasa tomar medidas de dispositivos que dispongan de estas resoluciones.

Visualizamos los cambios adaptativos en las siguientes imágenes:

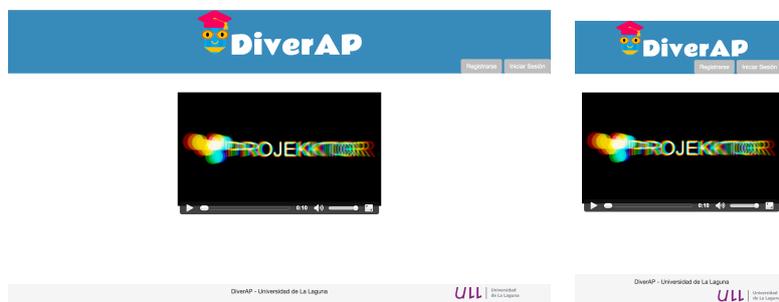


Figura 4.14. Landing page responsivo

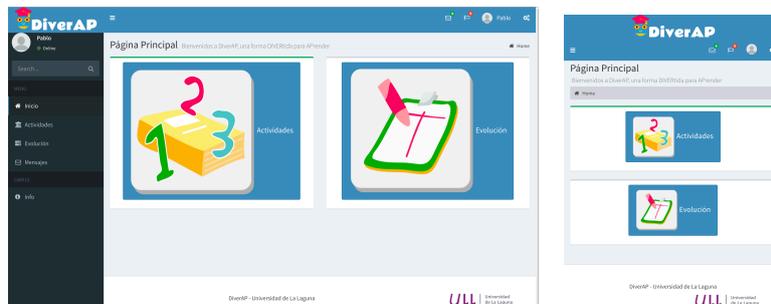


Figura 4.15. Home alumno responsivo

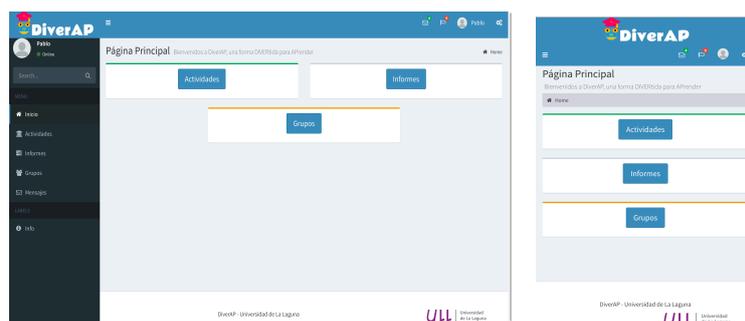


Figura 4.16. Home profesor responsivo

Capítulo 5. Conclusiones y líneas futuras

La realización de este trabajo final de grado se ha llevado a cabo tras las distintas fases del proyecto, logrando sus correspondiente hitos. Las fases de planificación y análisis han sido fundamental para este proyecto ya que era muy importante concretar una buena base para llevar acabo su adecuado diseño y desarrollo.

En las primeras dos fases se pudo aplicar los conocimientos adquiridos durante el desarrollo de la titulación del Grado de Ingeniería Informática del Itinerario 3 – Ingeniería del Software, en aspectos relacionados de gestión de proyecto, análisis de sistemas de software y modelado de datos. En las últimas fases de diseño y desarrollo, se adquirió nuevos conocimientos como el diseño de interfaces y desarrollo de aplicaciones web, tanto como el back-end, como el front-end. Esta última fase ha sido el más retador (a nivel personal) por el desconocimiento y funcionamiento de estas tecnologías. Como resultado final, se ha implementado DiverAP.

Con el fin de aportar en el mundo de la educación una importante herramienta de aprendizaje, se ha desarrollado esta plataforma web que soportará el diseño adaptativo. Gracias a su interfaz, los educadores tomarán el rol de diseñar y enseñar a partir de esta aplicación desde cualquier dispositivo electrónico, que después serán realizados por los estudiantes. Este proyecto ha llegado a su primera versión, de la cual se espera sacar provecho y mejorar en un futuro.

El sistema se enfocó principalmente en enseñanza de las matemáticas pero tiene la posibilidad y como futura propuesta, integrarle cualquier tipo de ejercicio, de cualquier materia. Más adelante se pretende integrarle un sistema inteligente. Esto consiste en que las actividades se adapten a las necesidades del alumno permitiendo que el aprendizaje sea personalizado y divertido. En otras palabras cuando al estudiante, el ejercicio le parezca fácil, automáticamente se le suba al nivel de dificultad o lo contrario. Además, se desea un sistema que guíe el alumno con pistas para resolver los problemas con ciertas dificultades. A la vez, incorporar un agente virtual inteligente que proporcione toda esta información y interactué con el alumno.

Todas estas características son futuras propuestas de módulos que se le desea añadir.

Contribuir una herramienta de esta potencial en el mundo de educación, favorecerá este sector bastante. Es muy importante que las forma de impartir clases o realizar actividades vaya modernizado, ya que en un futuro en el mundo laboral y ocio nos vemos interactuando continuamente con ello. Asimismo mejorar la impartición de la educación.

Summary and Conclusions

The fulfillment of this project has been taken place after various stages achieving to its corresponding landmarks. The phases of planning and analysis have been essential, due to its importance to set up a good base for its appropriate design and development.

In the first two stages, knowledge acquired (in subjects as project management, analysis of software requirements and database modeling), in the period of being a Software Engineer, were applied. The last two phases of design and development, knowledge was gained in designing interface and web application, such as back-end and frontend. These last phases have been the most challenging due to (personal) lack of awareness and not have been working on this technologies. As a final result, DiverAP as been introduced.

With the purpose to contribute in the world of education, this web app is an important tool of learning environment, which has been developed and designed with responsive design. Thanks to its interface the teachers can design, create and teach through out this app, in any electronic device, which later can be viewed and done by the students. This project has come to end of its first version, of which future lines are desired to develop.

The system was initially focused in teaching maths but it has the possibility to add any type of exercises, for any subject. Later it is expected to include a machine learning system. This consists of the system to adapt to the necessity of the student, allowing that they learn in a personalized and entertaining way. In other words, when the student feels the exercises are easy, automatically the level rises or viceversa. Besides, it is desired, the system to guide the student with hints to resolve the problems with certain difficulty. As well as, to include a virtual agent that provides all this information and interacts with student.

All these features are future proposal of modules and functions to work on and add to the platform.

Contributing a tool of this potential in the education system will benefit this sector a lot. It is very important to give classes or giving away homework modernizes, as in the future, in the professional world or spare time we are surrounded with technology, continuously.

Bibliografía

- [1] http://www.jmunozzy.org/files/9/Necesidades_Educativas_Especificas/Dis-calculia/documentos/1090-1095-disenoimplementacion.pdf
- [2] <http://riull.ull.es/xmlui/handle/915/281?show=full>
- [3] <http://www.xataka.com/aplicaciones/smartick-y-el-reto-de-que-los-ninos-amen-y-aprendan-las-matematicas>
- [4] <http://html5facil.com/tips/brackets-io-un-editor-de-texto-open-source-por-adobe/>
- [5] <https://es.wikipedia.org/wiki/XAMPP>
- [6] https://es.wikipedia.org/wiki/Adobe_Photoshop
- [7] https://librosweb.es/libro/ajax/capitulo_1.html
- [8] <http://www.designmantic.com/es/logo-design/variation?design=10893>

Anexo A (Casos de uso extendido).

RF- UC 1.1	Editar Plataforma – Verificar uso correcto de app.	
Actores	Desarrollador/Administrador	
Objetivos asociados	Mantenimiento de la aplicación web	
Descripción	Se realizarán comprobaciones que el uso es el correcto de la aplicaciones web.	
Precondición	Se avisa al cliente sobre el parón del sistema 2 semanas antes.	
Secuencia Normal	Paso	Acción
	1	Se ve la actividad del profesor y de los alumnos.
	2	Comprobación de todas las actividades se hayan realizado bien.
Postcondición	Se avisa al cliente cuando se hayan realizado los cambios.	
Excepciones	Paso	Acción
	1/2	Si se detecta algún fallo de nivel bajo, se procede a corregirlo y se realiza un informe de ello.
Frecuencia esperada	1 vez / en tres meses (trimestral)	

RF- UC 1.2	Editar Plataforma – Corregir fallos	
Actores	Desarrollador/Administrador	
Objetivos asociados	Mantenimiento de la aplicación web	
Descripción	Se procede corregir grandes fallos detectados.	
Precondición	Verificación uso correcto de app fallan cosas. Sistema no este en uso para realizar los cambios.	
Secuencia Normal	Paso	Acción
	1	Se lleva acabo el procedimiento dependiendo la característica de fallo.
	2	Al detectar que tipo, se realiza determinado pasos para que afecte lo menos posible a otras funcionalidades dela aplicación web.
Postcondición	Se avisa al cliente cuando se hayan realizado los cambios	
Excepciones	Paso	Acción
	2	Al no encontrar como solucionar se delega esto.
Frecuencia esperada	1 vez / en 1 año (anual)	

RF- UC 1.3	Editar Plataforma – Verificar B.B.D.D.	
Actores	Desarrollador/Administrador	
Objetivos asociados	Mantenimiento de la aplicación web	
Descripción	Se confirma que todo se ha almacenado bien en la bases de datos	
Precondición	Sistema no este funcionado actualmente.	
Secuencia Normal	Paso	Acción
	1	Se visualiza las tablas de bases de datos
	2	Se prueba algunas consultas
Postcondición	Se avisa al cliente cuando se hayan realizado los cambios.	
Excepciones	Paso	Acción
	1/2	Si se detecta algún fallo se procede corregirlo.
	3	Modificar y actualizar b.b.d.d. Repetir el procedimiento 1 y 2.
Frecuencia esperada	1 vez / en seis meses ()	

RF- UC 1.4	Editar Plataforma – Analizar mejoras	
Actores	Desarrollador/Administrador	
Objetivos asociados	Mantenimiento de la aplicación web	
Descripción	Se estudia posibles mejoras y funcionalidades a incorporar	
Precondición	Sistema no este funcionado actualmente.	
Secuencia Normal	Paso	Acción
	1	Se pregunta al cliente si desea mejorar algo
	2	Tras realizar un estudio, si se ve factible se procede incorporar el cambio.
Postcondición	Se avisa al cliente cuando se hayan realizado los cambios.	
Excepciones	Paso	Acción
	1	Si cliente no tiene queja no se incorpora nada nuevo.
Frecuencia	1 vez / en 1 año (anual)	

RF- UC 1.5	Editar Plataforma – Comprobar seguridad (amenaza)	
Actores	Desarrollador/Administrador	
Objetivos asociados	Mantenimiento de la aplicación web	
Descripción	Se comprueba la seguridad de web y red.	
Precondición	Sistema este funcionado actualmente.	
Secuencia Normal	Paso	Acción
	1	Se analiza posibles intentos de acceso al administrador del servidor web.
	2	Asimismo, posibles ataques de hackeo.
Postcondición	Se avisa al cliente cuando se hayan realizado los cambios.	
Excepciones	Paso	Acción
	1/2	Fortalecer seguridad, tomar medidas de criptográficas, etc.
Frecuencia esperada	1 vez / en seis meses (semestral)	

RF- UC 2.1	Iniciar sesión – Insertar Datos	
Actores	Alumnos y Profesores	
Objetivos asociados	Acceso página	
Descripción	Los usuarios introducirán los datos	
Precondición	Conexión a internet y navegador dirigido a la página principal.	
Secuencia Normal	Paso	Acción
	1	Los usuarios al acceder a la página escribirán su nombre de usuario en el primer campo.
	2	En el segundo campo la contraseña.
	3	Cliclean en el botón de "login" o enter.
Postcondición	Se espera a la verificación de datos.	
Excepciones	Paso	Acción
	1	No se cargue la página web.
Frecuencia esperada	80 veces / día	

RF- UC 2.2	Iniciar sesión – Verificar Datos	
Actores	Alumnos y Profesores	
Objetivos asociados	Acceso página	
Descripción	Los usuarios introducirán los datos y esperan a la comprobación	
Precondición	Usuario haya ingresado los campos y haya pinchando "enter".	
Secuencia Normal	Paso	Acción
	1	Se produce una conexión a las b.b.d.d
	2	Se comprueba los campos mediante una consulta SQL
	3	Al encontrar resultados, se devuelve los datos de verificación.
Postcondición	Acceso a la página principal	
Excepción 1	Paso	Acción
	1	Si no se produce conexión se reintenta durante un tiempo determinado.
Excepción 2	Paso	Acción
	1	Fallo en encontrar datos en la B.B.D.D
	2	De vuelve tipo de mensaje correspondiente al error.
Frecuencia esperada	40 veces / día	

RF- UC 2.3	Iniciar sesión – Acceder página principal	
Actores	Alumnos y Profesores	
Objetivos asociados	Acceso página	
Descripción	Los usuarios podrán acceder a la página directamente sin hacer login.	
Precondición	En lo cookies se hayan guardado los datos de login del usuario	
Secuencia Normal	Paso	Acción
	1	Se carga la página (home) sin pedir datos de login de usuario.
Postcondición	Acceso a la página principal	
Excepciones	Paso	Acción
	1	Se haya un pérdida de conexión, sesión caducada o los datos de login se hayan cambiado.
	2	Insertar datos.
Frecuencia esperada	40 veces / día	

RF- UC 3	Ver notificación – Ver número de notificaciones	
Actores	Alumnos	
Objetivos asociados	Avisar usuario de tareas que realizar.	
Descripción	Mantener al alumno avisado que tiene actividades o mensajes pendientes que hacer.	
Precondición	Ser usuario de la plataforma y haber iniciado sesión.	
Secuencia Normal	Paso	Acción
	1	Tras entrar en la página los usuarios reciben un aviso de los números de notificación que tiene pendiente.
	2	Al ver los números y la correspondencia a estos puede visualizar que tipo es.
	3	Si es de tipo ejercicio y si pincha esta notificación se va proceder realizar el ejercicio
Postcondición	Se espera a la verificación de datos.	
Excepción 1	Paso	Acción
	1/2	El usuario puede ver el número notificación, ver el tipo, pero no proceder a verlo.
Excepción 2	Paso	Acción
	3	Al proceder a realizar el ejercicios antes de comenzar puede dejarlo.
Frecuencia esperada	20 veces / día	

RF- UC 4.1	Realizar ejercicios – Ver tema asignado	
Actores	Alumnos	
Objetivos asociados	Alumno procede realizar sus actividades asignadas.	
Descripción	Al acceder a la plataforma, el alumno sin ver las notificación puede ir directamente ver por temas los ejercicios y realizarlos.	
Precondición	Ser usuario de la plataforma y haber iniciado sesión.	
Secuencia Normal	Paso	Acción
	1	Usuario accede directamente a la materia y tema que interesa.
	2	Visualiza el contenido y procede realizar los ejercicios.
Postcondición	Se espera a la verificación de datos.	
Excepciones	Paso	Acción
	1	Tras ver el tipo de ejercicio y contenido puede elegir a no hacerlo.
Frecuencia esperada	10 veces / día	

RF- UC 4.2	Realizar ejercicios – Empezar ejercicios	
Actores	Alumnos	
Objetivos asociados	Alumno procede realizar sus actividades asignadas.	
Descripción	Alumno ya empieza realizar los ejercicios independientemente como haya empezado.	
Precondición	Ser usuario de la plataforma, haber iniciado sesión, ver los ejercicios correspondiente a través de notificaciones o manualmente y darle a comenzar.	
Secuencia Normal	Paso	Acción
	1	Al comenzar el ejercicio puede tener varios opciones.
	2	Pausar/Continuar el ejercicio.
	3	Finalizar
Postcondición	Al finalizar se le muestra los resultados.	
Excepción 1	Paso	Acción
	2	Dependiendo del ejercicio se le permite diferente tipos de pausas. Asimismo puede continuar tras la pausa.
Excepción 2	Paso	Acción
	2	Abandonar el ejercicio.
Frecuencia esperada	20 veces / día	

RF- UC 5.1	Realizar informes – Clicar botón	
Actores	Profesores	
Objetivos asociados	Generación de informes (Seguimiento)	
Descripción	El profesor puede generar informes para realizar un seguimiento de sus alumnos. Esto permite visualizar el progreso de cada uno por separado.	
Precondición	Haber accedido como profesor en la plataforma	
Secuencia Normal	Paso	Acción
	1	Al no filtrar si pincha en generar informes se general de toda la plataforma.
	2	Se nos genera una .pdf
Postcondición	Se espera a la verificación de datos.	
Excepciones	Paso	Acción
	1	Se puede filtrar y generar informes, por asignatura, por temas o por alumnos.
Frecuencia esperada	1 vez / 2 semanas.	

RF- UC 5.2	Realizar informes – Escribir alumno	
Actores	Profesores	
Objetivos asociados	Generación de informes (Seguimiento)	
Descripción	El profesor puede generar un informe, puede escribirle al alumno intentando explicarle sus fallos o cualquier otro asunto.	
Precondición	Haber accedido como profesor en la plataforma y haber generado un informe.	
Secuencia Normal	Paso	Acción
	1	Profesor adjunta el .pdf y realiza comentarios al alumno.
Postcondición	Se espera a la verificación de datos.	
Excepciones	Paso	Acción
	1	Se le puede escribir sin adjuntar informe y tratar otro asunto.
Frecuencia esperada	1 vez / 2 semanas.	

RF- UC 5.3	Realizar informes – Enviar informes	
Actores	Profesores	
Objetivos asociados	Generación de informes (Seguimiento)	
Descripción	El profesor puede enviar informes a alumnos u otros profesores.	
Precondición	Haber accedido como profesor en la plataforma y haber generado informes.	
Secuencia Normal	Paso	Acción
	1	El envío de informe será exactamente como un correo o mensaje; destino y ficheros adjuntado.
Postcondición	Se espera a la verificación de datos.	
Excepciones	Paso	Acción
	1	En caso muy excepcional se lo puede enviar un informe a los padres.
Frecuencia esperada	1 vez / 2 semanas.	

RF- UC 6.1	Visualizar progreso – Ver avance	
Autores	Alumnos	
Objetivos asociados	Alumnos pueden ver su progreso.	
Descripción	Alumno visualiza su evolución respecto los ejercicios que realiza.	
Precondición	Haber accedido como alumno.	
Secuencia Normal	Paso	Acción
	1	Le aparecerá un barra de progreso respecto la evolución de curso en la página principal
	2	También se podrá clasificar esto, por asignaturas y temas.
	3	Si el alumno progresa bien puede llegar desbloquear juegos o recibir puntos de bonus.
Postcondición	Se espera a la verificación de datos.	
Excepciones	Paso	Acción
	1	Podemos filtrar y generar informes, por asignatura, por temas o por alumnos.
Frecuencia esperada	1 vez / 2 semanas.	

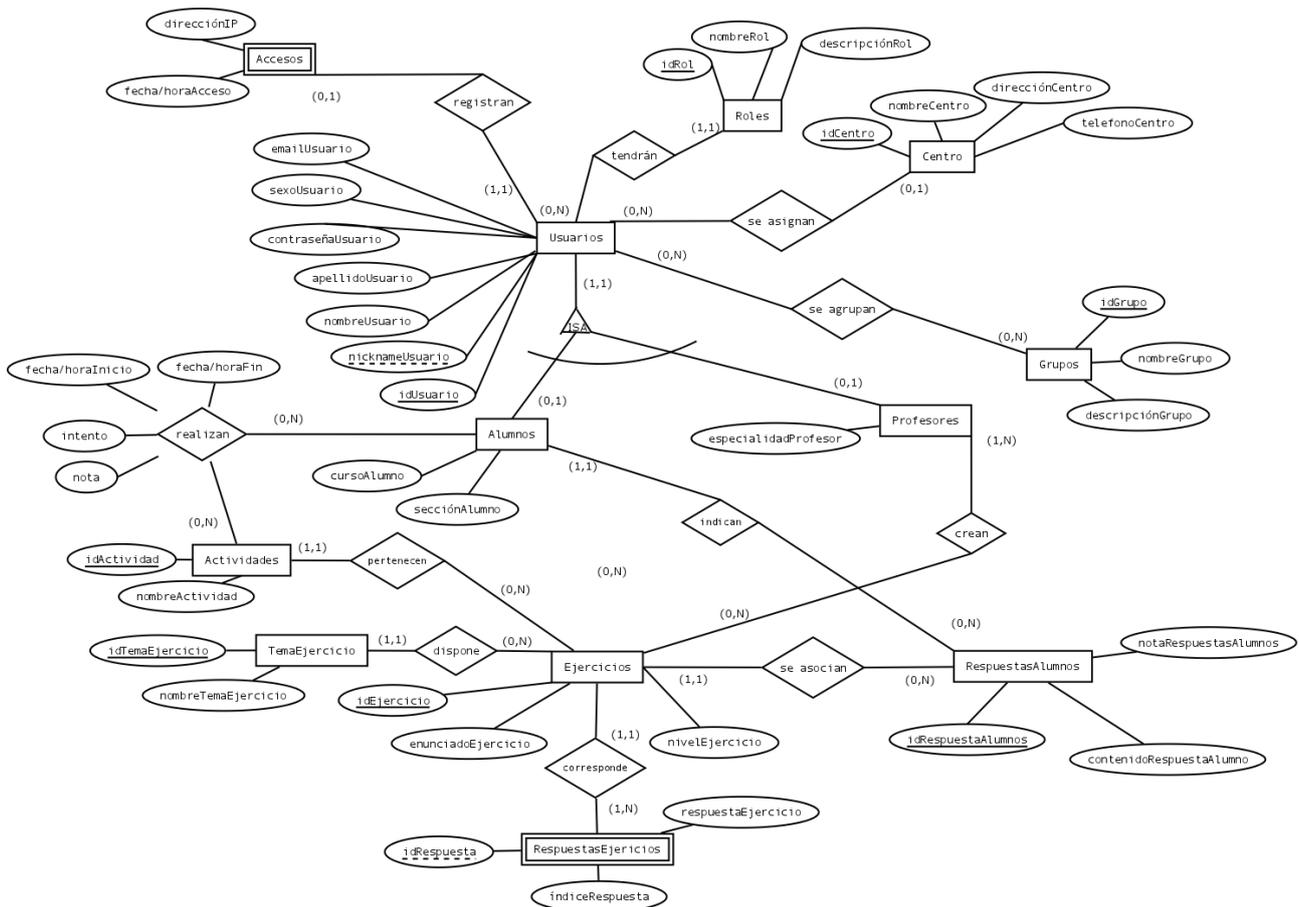
RF- UC 6.2	Realizar informes – Ver restante	
Actores	Alumnos	
Objetivos asociados	Generación de informes (Seguimiento)	
Descripción	El profesor puede generar informes para realizar un seguimiento de sus alumnos. Esto para realizar y visualizar el progreso de cada uno.	
Precondición	Haber accedido como alumno.	
Secuencia Normal	Paso	Acción
	1	El alumno podrá pinchar el porcentaje restante en general y mostrará los temas incompletos.
	2	Podrá visualizar lo mismo por asignatura.
Postcondición	Se espera a la verificación de datos.	
Excepciones	Paso	Acción
	1	En caso estar al día le saldrá 100%.
Frecuencia esperada	1 vez / 2 semanas.	

RF- UC 7.1	Preparar clases – Elegir asignatura y proceder hacer el ejercicio	
Actores	Profesores	
Objetivos asociados	Preparar clases virtuales.	
Descripción	El profesor prepara las clases o las actividades a realizar por los alumnos para que pueden realizarse virtualmente.	
Precondición	Haber accedido como profesor.	
Secuencia Normal	Paso	Acción
	1	Profesor elige primero de todo la asignatura.
	2	A continuación el tipo de ejercicio que desea hacer (Tipo-test, rellenar los blancos o responder a escrito)
	3	Elegir número de ejercicios
	4	Rellenar el formato que se abra para presentar los ejercicios.
	5	Agregar multimedia (imágenes, sonido o videos)
	6	Determinar plazo. (Fecha y tiempo para realizar el ejercicio).
	7	Poner visible el ejercicio (o programar fecha)
Postcondición	Se avisa al profesor con una notificación la disponibilidad de las actividades.	
Excepciones	Paso	Acción
	1/2/3/4/5 /6	En todos los etapas el profesor puede parar realizar una pausa y continuar realizando los ejercicios.
Frecuencia esperada	1 vez / 1 semanas.	

RF- UC 7.2	Preparar clases – Ver clases anteriores	
Actores	Profesores	
Objetivos asociados	Preparar clases virtuales.	
Descripción	El profesor prepara las clases o las actividades a realizar por los alumnos para que pueden realizarse virtualmente.	
Precondición	Haber accedido como profesor.	
Secuencia Normal	Paso	Acción
	1	Profesor retoma actividades realizadas anteriormente.
	2	Abre período de realización y cierre.
Postcondición	Se avisa al profesor con una notificación la disponibilidad de las actividades.	
Excepciones	Paso	Acción
	1	Si le desea agregar un cambio puede realizarlo.
Frecuencia esperada	1 vez / 1 semanas.	

Anexo B (Diseño de bases de datos).

Para desarrollar las bases de datos de esta aplicación, comenzamos diseñando un diagrama de e-r:



En él, nos encontramos con 10 entidades principales, 2 débiles y 3 relaciones NxM que forman tablas.

Se define la siguiente anotación para identificar las características de los atributos.

_____ = clave principal

===== = clave foránea

..... = clave débil

Entidad = Entidad débil

- **Entidades Principales:**

Usuarios: (*idUsuario*, *nicknameUsuario*, *nombreUsuario*, *apellidoUsuario*, *contraseñaUsuario*, *sexoUsuario*, *emailUsuario*)

Roles: (*idRol*, *nombreRol*, *descripciónRol*)

Centros: (*idCentro*, *nombreCentro*, *direcciónCentro*, *teléfonoCentro*)

Grupos: (*idGrupo*, *nombreGrupo*, *descripciónGrupo*)

Alumnos: (*cursoAlumno*, *secciónAlumno*)

Profesores: (*especialidad*)

Actividades (*idActividad*, *nombreActividad*)

Ejercicios: (*idEjercicio*, *enunciadoEjercicio*, *decripciónEjercicio*, *nivelEjercicio*).

TemaEjercicio: (*idTemaEjercicio*, *nombreTemaEjercicio*)

RespuestasRespondidas: (*idRespuestaRespondida*, *respuestaAlumno*)

- **Entidades débiles:**

Accesos: (*fecha/horaAcceso*, *direcciónIP*)

RespuestasEjercicios: (*idRespuesta*, *respuestaEjercicio*, *índiceRespuesta*)
RespuestasEjercicios tiene una clave débil, además necesita una clave foránea.
Esta entidad almacenará todas las respuestas de todos los ejercicios.

(El atributo “índiceRespuesta” es un porcentaje de acierto que se le asigna a la respuesta. Por ejemplo: Si tenemos una pregunta con una

respuesta, el índice tendrá valor de 100%. Si hubiera la posibilidad de dos respuestas pues se podría 50% así sucesivamente. 0% significa que la respuesta incorrecta.)

- **Relaciones :**

Usuario-tendrán-Roles, todos los usuarios tienen un solo rol correspondiente; los diferentes Roles pueden asignar a varios usuarios, por lo que tenemos una relación (1:N).

Usuarios-pertenecen-Centro, todo los usuarios pertenecen a un Centro designado. En cada Centro puede haber 0 o varios usuarios, por lo que tenemos (1:N).

Usuarios-realizan-Accesos: Esta es una relación: (1:1) ya que un usuario solamente podrá acceder a una determinada hora, con una dirección IP. El acceso corresponderá a exclusivamente a un usuario. La entidad Accesos depende de Usuarios. Sin usuario no podremos tener Accesos, por lo que la clave de Usuarios viaja a Accesos.

Por último de la entidad Usuarios, se hereda Alumnos y Profesores. Dos entidades que tienen algunos atributos más, para caracterizarlo correctamente.

Alumnos-indican-RespuestasAlumnos: De los Alumnos se desea almacenar todas las Respuestas correspondiente a ellos. Pero esta solo se almacenará si un alumno responde. Por este motivo tenemos una relación(1:N) ya que las RespuestasAlumno depende de Alumnos.

Ejercicios-tienen-TemaEjercicio: Relación(1:N) Cada ejercicio será de un tipo determinado. Como "Tema" es un dato complementario, podemos incorporar su clave en la tabla de Ejercicios.

Ejercicios-corresponde-RespuestasEjercicio: Sin Ejercicios no habrán respuestas y cada respuesta tiene que ser identificada con su único Ejercicio correspondiente. Esta dependencia convierte la entidad RespuestaEjercicio, en una entidad débil, por lo tanto nos encontramos con una relación (1:N) donde la clave de Ejercicio viaja RespuestaEjercicio.

Ejercicios-asocia-RespuestasAlumnos: A todo ejercicios se le asocia al conjunto de respuestas que responde los alumnos. Al contrario sin estos Ejercicios no dispondremos de Ejercicio. Por lo que tenemos una relación (1:N).

Ejercicios-pertenece-Actividades: Todos los Ejercicios corresponden a una Actividad y la Actividades pueden tener ninguno o varios ejercicios(1:N). Los ejercicios dependen de las Actividades.

- **Relaciones NxM:**

Estas relaciones son las que generan tablas.

se_agrupan: (idUsuario, idGrupo)

Esta relación formada con la clave idUsuario, que pueden estar en varios grupos y la clave de Grupos, que pueden tener varios Usuarios (En cada grupo habrá un mínimo usuario y máximo de M).

crean : (idProfesor, idEjercicio)

Esta relación se anota, ya que los Profesores pueden crear varios ejercicios y estos Ejercicios pueden ser creado por varios profesores.

realizan: (idAlumno, idActividades)

Los alumnos puedan realizar varios actividades y a su vez estas actividades pueden realizarse por diferentes alumnos, por lo que se genera tabla(N:M).

Modelo Relacional

El modelo relacional es el paso previo antes de proceder implementar nuestras tablas de bases de datos en el servidor de bases de datos. Por lo cual, este paso es vital que se elaboré correctamente, para más adelante no verse afectado de un mal diseño de bases de datos.

A continuación tenemos el modelo relacional:

Usuarios: (*idUsuario*, *nicknameUsuario*, *nombreUsuario*, *apellidoUsuario*, *contraseñaUsuario*, *sexoUsuario*, *emailUsuaario*) *idRol*, *idCentro*

Accesos: (*fecha/horaAcceso*, *direcciónIP*) *IdUsuario*

Roles: (*idRol*, *nombreRol*, *descripciónRol*)

Centros: (*idCentro*, *nombreCentro*, *direcciónCentro*, *teléfonoCentro*)

Grupos: (*idGrupo*, *nombreGrupo*, *descripciónGrupo*)

Alumnos: (*cursoAlumno*, *secciónAlumno*) *IdUsuario*

Profesores: (*especialidad*) *IdUsuario*

se agrupan: *idUsuario*, *idGrupo*

crean: *idProfesor*, *idEjercicio*

Actividades: (*idActividad*, *nombreActividad*)

realizan: *idActividad*, *IdUsuario*, *fecha/horaInicio*, *fecha/horaFin*, *nota*, *intento*.

Ejercicios:(*idEjercicio*, *enunciadoEjercicio*, *decripciónEjercicio*, *nivelEjercicio*)
idTemaEjercicio.

TemaEjercicio: (*idTemaEjercicio*, *nombreTemaEjercicio*)

RespuestasEjercicios: (*idRespuesta*, *respuestaEjercicio*, *índiceRespuesta*)
idEjercicio

RespuestasAlumnos:(*idRespuestaRespondida*, *contenidoRespuestaAlumno*, *notaRespuestaAlumno*) *IdUsuario*, *idEjercicio*,

Anexo C (Diseño de Interfaces).

