



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA (SG)

Grado en Ingeniería Informática de Servicios y Aplicaciones

**Aplicación para la introducción y el
estudio de los números enteros en la
Educación Primaria**

**Alumno: Sandra García Sanz
Tutor: María Luisa Martín Pérez**

Aplicación para la introducción y el estudio de los números enteros en la Educación Primaria

Sandra García Sanz

ÍNDICE PRINCIPAL

Bloque I: Memoria

Bloque II: Manual técnico

Bloque III: Manual de usuario

Bloque IV: Bibliografía y anexos

ÍNDICE

Tabla de contenido

0

1.1	Identificación del trabajo fin de grado	13
1.2	Motivación y propósito	13
1.3	Objetivos principales del proyecto.....	14
1.4	Organización del documento	14
1.5	Contenidos del CD	15
2.	ESTADO DEL ARTE	16
3.	DESCRIPCIÓN GENERAL DEL TRABAJO	18
3.1	Descripción del sistema.....	18
3.2	Funcionalidad	18
3.3	Tecnologías utilizadas.....	19
3.4	Cuestiones metodológicas	21
3.4.1	Metodología, proceso de desarrollo y ciclo de vida	21
3.5	Software utilizado	24
3.5.1	Software utilizado durante la documentación.....	24
3.5.2	Software utilizado durante el desarrollo.....	26
3.6	Planificación y presupuesto	27
3.6.1	Planificación	27
3.6.2	Presupuesto.....	45
3.7	Consideraciones	49
3.7.1	Consideraciones sobre la fase de análisis	49
3.7.2	Consideraciones sobre la fase de diseño	50
3.7.3	Consideraciones sobre la fase de implementación.....	50
3.7.4	Consideraciones sobre la fase de pruebas.....	50
3.8	Conclusiones y posibles ampliaciones.....	51
3.8.1	Conclusiones.....	51
3.8.2	Posibles ampliaciones	51
4.	ANÁLISIS	55
4.1	Introducción	55
4.2	Alcance del análisis.....	55
4.3	Requisitos específicos	55
4.3.1	Objetivos principales del sistema.....	56

4.3.2	Requisitos de información.....	59
4.3.3	Restricciones generales.....	61
4.3.4	Requisitos funcionales.....	62
4.3.5	Requisitos no funcionales	89
4.3.6	Resumen.....	91
5.	DISEÑO	92
5.1	Arquitectura lógica.....	92
5.2	Arquitectura física	95
5.3	Diagrama de clases.....	96
5.4	Diagrama de secuencia	98
5.5	Modelo lógico de datos.....	109
5.5.1	Diccionario de datos.....	109
5.6	Diseño de interfaz	112
6.	PRUEBAS.....	121
6.1	Introducción	121
6.2	Pruebas de caja negra	123
6.3	Pruebas de caja blanca.....	128
7.	MANUAL DE USUARIO.....	133
7.1	Introducción	133
7.2	Instalación de Software.....	133
7.2.1	Instalación del servidor	133
7.2.2	Instalación de la aplicación	144
7.3	Manual de usuario	144
7.4	Requisitos del sistema.....	167
8.	BIBLIOGRAFÍA	171
9.	WEBGRAFÍA	171
10.	ANEXOS	172
10.1	Índice de tablas	172
10.2	Glosario	175

BLOQUE I
MEMORIA

1. INTRODUCCIÓN

1.1 Identificación del trabajo fin de grado

Título:

Aplicación para la introducción y el estudio de los números enteros en la Educación Primaria

Autor:

Sandra García Sanz

Tutor:

María Luisa Martín Pérez

Área:

Matemática Aplicada

1.2 Motivación y propósito

No cabe duda de que estamos viviendo una de las etapas más apasionantes de la historia de la humanidad desde la llegada de la era de la información. Si echamos un vistazo atrás (y no hace falta irse muy lejos), nos podemos dar cuenta que estamos siendo testigos de cómo la tecnología transforma el mundo que nos rodea, de cómo se vive y cómo se trabaja.

Esta situación, además de dar oportunidades supone un gran reto para las empresas e instituciones ya que deben diseñar y aplicar la tecnología para potenciar las habilidades humanas, es decir, deben dar la posibilidad a las personas de hacer y conseguir más en sus vidas personales y profesionales.

Si todo esto, lo enfocamos en el mundo de la educación, también se puede observar cómo desde la enseñanza en guarderías, colegios, institutos hasta la universidad se va instaurando una nueva forma de enseñanza a través de la tecnología. Ésta, constantemente en actualización, da apoyo a los docentes en las tareas de comunicación de contenidos y desarrollo de prácticas, de forma que ayuda al aumento de la calidad de la enseñanza.

Por todo esto, en este Trabajo Fin de Grado se desarrollará y analizará una aplicación web donde los alumnos de Tercer Ciclo de Educación Primaria puedan aprender de una forma sencilla y atractiva los números enteros utilizando el medio tecnológico y, por otra parte, se facilita al profesorado una herramienta fácil de utilizar a la hora de actualizar temarios, ejercicios, test e incluso tener una comunicación con los alumnos.

La finalidad de esta aplicación web es acercar a los niños una parte de las Matemáticas y que a través de una interfaz tecnológica sea más ameno su estudio.

1.3 Objetivos principales del proyecto

El principal objetivo de este TFG es crear una aplicación web que permita adentrarse al usuario (en este caso, alumnos del Tercer Ciclo de Primaria) en el aprendizaje de los números enteros a través de unas nociones básicas de teoría, ejemplos didácticos, ejercicios y juegos.

La aplicación será capaz de realizar las siguientes operaciones:

- ✓ A nivel de alumno:
 - Consultar los diferentes temarios.
 - Realizar ejercicios.
 - Realizar test.
 - Comunicarse con un profesor
- ✓ A nivel de profesor:
 - Crear, modificar y eliminar temarios.
 - Crear, modificar y eliminar ejercicios.
 - Crear test.
 - Comunicarse con un alumno.

1.4 Organización del documento

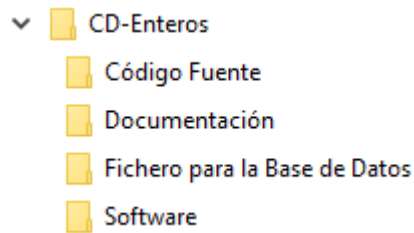
A modo de guía para el lector, a continuación, se describen las diferentes secciones que podemos encontrar a lo largo de la documentación del trabajo.

Se compone de cuatro partes bien diferenciadas:

- ✓ **Bloque I: “Memoria”.**
En esta parte se especifica una breve introducción, la planificación del desarrollo de la aplicación, junto con su metodología, herramientas utilizadas, conclusiones y posibles ampliaciones.
- ✓ **Bloque II: “Manual técnico”.**
Esta parte engloba los aspectos técnicos del análisis y diseño de la aplicación, con sus respectivos diagramas.
También contiene las pruebas que se han llevado a cabo a lo largo del desarrollo. Esta sección está destinada para analistas y programadores.
- ✓ **Bloque III: “Manual de usuario”.**
En esta parte se especifica cómo usar la aplicación web, además de cómo instalar los programas necesarios para poner en marcha dicha aplicación.
- ✓ **Bloque IV: “Bibliografía, Webgrafía y Anexos”.**
Este bloque contiene la bibliografía y webgrafía utilizada, el glosario de términos y el diccionario de tablas e ilustraciones utilizadas a lo largo de esta memoria.

1.5 Contenidos del CD

El CD adjunto a esta memoria dispone de cuatro carpetas con la siguiente estructura:



- La carpeta **Documentación** contiene el documento (en formato .PDF) correspondiente a la documentación del TFG. El documento abarca las siguientes partes:
 - Memoria
 - Manual Técnico.
 - Manual de Usuario.
 - Bibliografía, Webgrafía y Anexos.
- En la carpeta **Código Fuente** se encuentran todos los ficheros que componen la aplicación. Todos los ficheros .js y .html del front, los ficheros .php del backend, etc.
- En la carpeta **Software** se encuentran todos los programas utilizados en el desarrollo del proyecto, que a su vez son necesarios para la correcta ejecución del mismo.
- En la carpeta **Fichero para la base de datos** se encuentra el directorio necesario para importar los datos a la base de datos.

2. ESTADO DEL ARTE

Previo al análisis, diseño e implementación de la aplicación web “Aprendiendo los números enteros”, es conveniente realizar un estudio sobre las diferentes herramientas dirigidas a fomentar el aprendizaje y estudio sobre los números enteros dentro del ámbito escolar que nos podemos encontrar por la red.

“Aprendiendo los números enteros” está dirigida a crear un modelo de estudio online para los niños de Tercer Ciclo (más específicamente de 6º curso) de Primaria para que contribuya al aprendizaje de esta parte de las matemáticas de una forma amena y amigable. Por otra parte, también se quiere facilitar al profesorado la exposición de temarios y que esta aplicación sea una forma sencilla de tener un seguimiento de los alumnos.

Por todo esto, para el estudio de plataformas competidoras del entorno se tendrán en cuenta también aquellas aplicaciones que ofrezcan a alumnos y profesores una funcionalidad similar.

Después de una búsqueda exhaustiva por Internet, he comprobado que es bastante difícil para un niño encontrar aplicaciones similares a la nuestra. No obstante, he encontrado algunas que me han parecido muy interesantes para que el alumno practique con juegos a la vez que afianza los conceptos.

- ✓ De Anaya:
http://www.joaquincarrion.com/Recursosdidacticos/SEXTO/datos/03_Mates/datos/05_rdi/ud05/2/02.htm
http://www.joaquincarrion.com/Recursosdidacticos/SEXTO/datos/03_Mates/datos/05_rdi/ud05/1/01.htm
- ✓ GenMagic.org
<http://www.genmagic.net/repositorio/albums/userpics/granpe1c.swf>
- ✓ Junta de Andalucía
http://www.juntadeandalucia.es/averroes/centros-tic/41009470/helvia/aula/archivos/repositorio/0/193/html/recursos/la/U03/pages/recursos/143304_P37/es_carcasa.htm

Por otra parte, se han encontrado algunas webs que albergan contenidos educativos, la más interesante sería la siguiente:

- Web del Ministerio de Educación, Cultura y Deporte, “educaLAB” (<http://ntic.educacion.es/w3/eos/MaterialesEducativos/primaria/matematicas/conmates/unid-3/numeros-enteros1.htm>).
 Dentro de materiales educativos para los niños de primaria podemos encontrar una pequeña aplicación en la que aparte de otros temas, se explican los números enteros:

- La información que alberga es sencilla y muy fácil de seguir para un niño del Tercer Ciclo de Primaria, tiene un desglose breve de conceptos acompañados de teoría acompañada de un par de actividades para que los alumnos practiquen.
- Batería de ejercicios en la que los alumnos pueden comprobar la solución (en algunos facilitan alguna pista si así lo necesitan).
- Un apartado de definiciones importantes que se han visto a lo largo de la unidad.
- Y, por último, una breve evaluación de todo lo aprendido. Le da una idea de cómo hacer un control en clase al profesor.

Y, por último, lo que sí he encontrado en la red son numerosos blogs destinados al aprendizaje de diferentes áreas de las Matemáticas dirigidas a niños.

El blog que me ha parecido más completo ha sido el siguiente:

- ✓ Blog “Jugando y Aprendiendo” (Blog de Luisa María Arias Prada. Recursos para 6º de Primaria: <https://luisamariaarias.wordpress.com/matematicas/tema-3-numeros-enteros/>)

En este blog encontramos numerosos temas para alumnos de 6º de Primaria, centrándonos en el área de Matemáticas, alberga el tema de los números enteros, sus contenidos son a base de videos y links a aplicaciones que contienen juegos sobre los números enteros.

Me parece un blog muy interesante para un niño que necesite refuerzo en esta área, además tiene de una forma centralizada numerosos ejercicios con los que practicar.

De toda la información existente consultada, la aplicación que se presenta en este TFG resulta más completa y fácil de seguir en comparación con la web del Ministerio o del blog. Es decir, contiene lo básico y necesario que un niño necesita para afianzar los conceptos de los números enteros. Así mismo, se han incluido numerosos ejercicios con los que el alumno puede practicar.

3. DESCRIPCIÓN GENERAL DEL TRABAJO

3.1 Descripción del sistema

El sistema que se desarrolla en esta memoria deberá ser capaz de dar acceso a profesores de tal forma que accedan a un menú en el que puedan gestionar el mantenimiento de los alumnos, los temarios (con sus ejercicios y test) y en el que además puedan comunicarse con los alumnos.

Por otra parte, una vez que el profesor dé de alta a alumnos, el sistema debe ser capaz de diferenciar en un primer plano entre alumnos y profesores, y mostrar al alumno todos aquellos contenidos que el profesor ha ido añadiendo a la aplicación. Y en segundo plano, discriminar entre alumnos registrados y alumnos no registrados ya que no tendrán los mismos privilegios.

Dado el ámbito de usuarios con el que se tratará, la aplicación web se ha diseñado con un interfaz amigable y visualmente atractivo para que, ya en sí mismo, esto sea un aliciente para seguir navegando por la misma.

3.2 Funcionalidad

La aplicación web tiene las siguientes funcionalidades:

- ✓ Para el usuario “Profesor”:
 - **Mantenimiento de alumnos:** la aplicación va a permitir dar de alta alumnos, modificar sus datos, activarlos en la aplicación, así como eliminarlo si se quisiera.
 - **Mantenimiento de temario y ejercicios:** el profesor va a poder crear temarios y ejercicios, modificarlos y eliminarlos. Se da la posibilidad que, una vez creado un tema o ejercicio, sea visible para el alumno o no.
 - **Generación de test:** la aplicación le mostrará al profesor una serie de campos para preparar un control fácilmente en base a los ejercicios que previamente ha creado.
 - **Comunicación con el alumno:** mediante un cuadro de diálogo se permite al profesor escribir a un alumno para que cuando posteriormente se logue, vea dicho mensaje.
 - **Subida de archivos:** la aplicación tiene un menú de subida de archivos para que el profesor pueda subir ficheros tipo .pdf, .jpg, etc.
 - **Consulta de archivos:** podrá consultar los archivos que suben los alumnos (por ejemplo, un test resuelto).
 - **Añadir calificaciones:** mediante un cuadro de texto el profesor puede ir añadiendo las calificaciones oportunas.

- ✓ Para el usuario “Alumno registrado”:
 - **Consultar temario y ejercicios:** el alumno podrá consultar el temario y ejercicios que previamente el profesor ha subido a la aplicación y ha dejado visible.
 - **Realizar ejercicios:** el alumno podrá practicar lo estudiado en los temas gracias a unos ejercicios online con comprobación de resultados.
 - **Visualizar y descargar test:** el alumno podrá visualizar los test disponibles, así como descargárselos para realizarlos.
 - **Subida de archivos:** la aplicación tiene un menú de subida de archivos para que el alumno pueda subir ficheros tipo .pdf, .jpg, etc, por ejemplo, para subir un test resuelto.
 - **Consultar calificaciones:** la aplicación va a facilitar al alumno un listado de las calificaciones disponibles.
 - **Descargar ficheros:** el alumno podrá descargarse ficheros, por ejemplo, un test o más información adicional sobre un tema.
 - **Comunicación con el profesor:** mediante un cuadro de diálogo se permite al alumno escribir al profesor para que cuando posteriormente se logue, vea dicho mensaje.

- ✓ Para el usuario “Alumno no registrado”:
 - **Consultar temario y ejercicios:** para un alumno no registrado la aplicación estará disponible para consultar temarios y ejercicios.
 - **Realizar ejercicios:** el alumno no registrado podrá practicar con los ejercicios online.

3.3 Tecnologías utilizadas

Para el desarrollo de la aplicación se han utilizado los siguientes lenguajes de programación:

✓ **AngularJS:**



Utilizado para el frontEnd de la aplicación.

Es un framework de JavaScript de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

✓ **PHP:**

Utilizado para el backEnd de la aplicación.

Es un lenguaje de programación de código del lado del servidor diseñado para el desarrollo web de contenido dinámico.

✓ **JavaScript:**

Utilizado para el frontEnd de la aplicación.

Es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

✓ **HTML:**

Utilizado para el frontEnd de la aplicación.

Lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros.

✓ **CSS:**

Son estilos de la aplicación web.

Es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de las páginas web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etc.

✓ **Bootstrap:**

Son estilos de la aplicación web.

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.

✓ **Jquery:**

Utilizado para el frontEnd de la aplicación.

Es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

✓ **MySQL:**

Es un sistema de gestión de bases de datos relacional desarrollado por Oracle.

Está considerada como la base de datos open source más popular del mundo (sobre todo para entornos de desarrollo web).

3.4 Cuestiones metodológicas

3.4.1 Metodología, proceso de desarrollo y ciclo de vida

Para el desarrollo de la aplicación web “Aprendiendo los números enteros”, se ha optado por abordar el trabajo siguiendo un *modelo iterativo e incremental*.

Este proceso está enfocado hacia la construcción, el desarrollo y el mantenimiento del software. Combina buenas prácticas con una descripción consistente y bien documentada.

La idea fundamental de este modelo, se basa en dos principios:

- **Desarrollo iterativo:** Se basa en un desarrollo según "mini proyectos cortos" o "iteraciones", las cuales tienen una duración fija, típicamente cuatro semanas. Como resultado de cada iteración se obtiene un sistema que puede ser probado, integrado y ejecutado. Cada iteración tiene sus propias actividades: análisis, diseño implementación, pruebas y documentación.
- **Incremental:** Ampliación y refinamiento del sistema a través de múltiples iteraciones. El sistema converge hacia los requisitos y el diseño más adecuados. El sistema crece incrementalmente con el tiempo: desarrollo iterativo e incremental.

La elección de esta metodología está apoyada en los beneficios del desarrollo iterativo. Algunos de estos beneficios son los siguientes:

- Mitigación tan pronto sea posible de riesgos altos (la inestabilidad de los requisitos y el diseño disminuyen con el tiempo).
- Progreso visible en las primeras etapas.
- Temprana retroalimentación, compromiso de los usuarios y adaptación que nos lleva a refinar el sistema.
- Recogida de nuevos requisitos al final de cada iteración.
- Gestión de la complejidad más sencilla (pasos más cortos y más sencillos).
- Conocimiento adquirido, aprovechable iteración tras iteración.

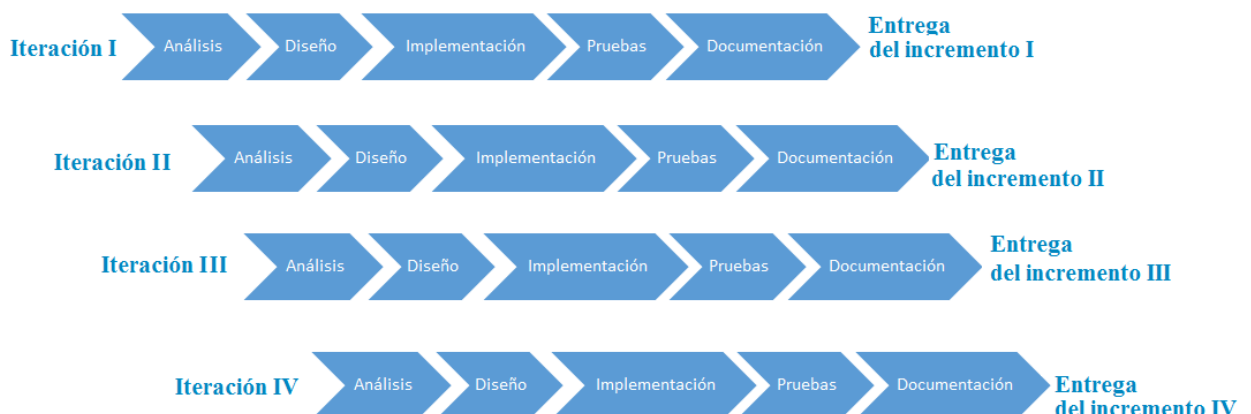
Cada mes de desarrollo será una iteración en la que se irán añadiendo funcionalidades con el propósito de mejorar el resultado final. Al final de cada iteración podemos detectar errores, riesgos y recoger nuevos requisitos.

El conjunto de iteraciones que podemos conectar en este tipo de modelo de desarrollo permite obtener un conjunto de versiones producto de las mejoras que se establecen en cada una de esas iteraciones. Este tipo de modelo de desarrollo permite volver a una fase anterior siempre en cuando el proyecto no esté en la fase de entrega al cliente.

El modelo de desarrollo iterativo-incremental combina elementos del modelo en cascada con la filosofía iterativa de construcción de prototipos. Al final de cada iteración se entrega u obtiene una versión mejorada.

En cada iteración se seguirán diferentes etapas:

- Análisis
- Diseño
- Implementación
- Pruebas
- Documentación



Ventajas

- Con un paradigma incremental se reduce el tiempo de desarrollo inicial, ya que se implementa la funcionalidad parcial.
- También provee un impacto ventajoso frente al cliente, que es la entrega temprana de partes operativas del Software.
- El modelo proporciona todas las ventajas del modelo en cascada realimentado, reduciendo sus desventajas sólo al ámbito de cada incremento.
- Permite entregar al cliente un producto más rápido en comparación del modelo de cascada.
- Resulta más sencillo acomodar cambios al acotar el tamaño de los incrementos.
- Por su versatilidad requiere de una planeación cuidadosa tanto a nivel administrativo como técnico.
- Se pueden incluir nuevos requisitos en cada iteración.

Desventajas

- El modelo Incremental no es recomendable para casos de sistemas de tiempo real, de alto nivel de seguridad, de procesamiento distribuido, y/o de alto índice de riesgos.
- Requiere de mucha planeación, tanto administrativa como técnica.
- Requiere de metas claras para conocer el estado del proyecto.

3.5 Software utilizado

En este apartado se detalla cada uno de los programas que se han ido utilizando a lo largo del desarrollo del trabajo.

3.5.1 Software utilizado durante la documentación



Windows 10 64 bits

Utilizado para la documentación Windows 10 es la versión más reciente de Microsoft Windows, línea de sistemas operativos producida por Microsoft Corporation. Esta versión introdujo una arquitectura de aplicaciones «universales», puede ejecutarse en toda la familia de productos de Microsoft con un código casi idéntico: Pc, Smartphone, Tablet, Xbox, etc.



Microsoft Office 2010

Microsoft Office 2010 es una versión de la suite ofimática Microsoft Office de Microsoft y sucesora de Microsoft Office 2007. Office 2010 incluye compatibilidad extendida para diversos formatos de archivos, actualizaciones de la interfaz de usuario, y una experiencia de usuario refinada.

Compatible con Windows XP, Windows Vista, Windows 7 y Windows 10. Por primera vez y con la introducción de Office 2010, la suite está disponible en una compilación para arquitecturas de 64 bits, aunque sólo para los sistemas operativos de núcleo NT 6.x, como Windows Vista, Windows 7 y Windows Server 2008/2008 R2.



Adobe Reader 11

Adobe Acrobat es una familia de programas informáticos desarrollados por Adobe Systems diseñados para visualizar, crear y modificar archivos con el formato Portable Document Format, más conocido como PDF.

Acrobat y Reader son de uso muy popular como forma de presentar información con un formato fijo similar al de una publicación. El uso del formato PDF es muy común para mostrar texto con un diseño visual ordenado.



StarUML genera todo tipo de diagramas compatibles con la plataforma de programas Microsoft Office.

StarUML se maneja con facilidad. En un vistazo a la interfaz se ven las funciones principales del programa. Otra característica importante del programa es que su código es compatible con C++ y Java.

Se puede comenzar a dibujar los gráficos manualmente o seleccionar las plantillas que contiene el archivo de instalación para modificarlas. Esta última opción es muy recomendable para quien no ha trabajado con archivos UML / MDA.



OpenProj es una utilidad para gestionar tareas, recursos y hacer un seguimiento visual de todo el proyecto.

Proporciona todo lo que puedas esperar de una herramienta de gestión de proyectos, desde varios tipos de gráficas hasta el coste final de desarrollo de un proyecto.

Entrando en detalles, OpenProj permite asignar recursos a tareas, crear dependencias entre tareas, ver los recursos disponibles, filtrar las tareas por estado y ordenarlas según prioridad, coste y otros factores.



DIA

Dia es un editor de diagramas con las herramientas necesarias para crearlos o modificarlos sin apenas conocimientos.

Incluye herramientas de dibujo para introducir distintos elementos geométricos a nuestras composiciones, pudiendo editar sus propiedades y con un espacio cuadrulado para organizar nuestros diagramas y sistema de capas.

Además, permite abrir y exportar los dibujos realizados a los formatos más conocidos, además de tener su propio formato para editar el documento posteriormente.

Se puede utilizar para dibujar diferentes tipos de diagramas como pueden ser: diagramas entidad-relación, diagramas UML, diagramas de flujo, diagramas de redes, etc.

3.5.2 Software utilizado durante el desarrollo



Google Chrome

Google Chrome es un navegador web práctico y ágil desarrollado por Google. Es gratuito y multidispositivo.

Su secreto es potencia y versatilidad así como un enorme catálogo de extensiones que eleva sus posibilidades hasta el infinito. Además es vistoso y fácil de usar.



Aplicación web que se utiliza para alojar proyectos utilizando el sistema de control de versiones Git. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.



Windows 10 64 bits

Sistema operativo sobre el que se ha llevado el desarrollo de la aplicación. Windows 10 es la versión más reciente de Microsoft Windows, línea de sistemas operativos producida por Microsoft Corporation. Esta versión introdujo una arquitectura de aplicaciones «universales», puede ejecutarse en toda la familia de productos de Microsoft con un código casi idéntico: Pc, Smartphone, Tablet, Xbox, etcétera.



Sublime Text 3

Herramienta utilizada para realizar el desarrollo del código fuente de la aplicación web.

Es un editor de texto y editor de código fuente.



Xampp

Software libre que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y el intérprete para lenguaje de script: PHP.

3.6 Planificación y presupuesto

3.6.1 Planificación

Una de las actividades cruciales del proceso de gestión es la planificación, la cual se basa en una buena estimación del esfuerzo requerido para realizar el proyecto, duración cronológica y coste.

3.6.1.1 Estimación del esfuerzo

La sociedad moderna cada vez es más dependiente del software. Al ser el SW un bien en el que se invierten grandes cantidades de dinero, se hace muy importante poder controlarlo. Sin embargo, a pesar de su importancia, es un bien intangible que es difícil medir o valorar.

Las organizaciones necesitan controlar esta inversión. En ese sentido, el medir el software no es un tema académico, sino un tema de valor, de inversión y de negocio.

De hecho, es conocido que, grandes proyectos han fracasado al no estar a tiempo o dentro de presupuesto por una mala estimación de esfuerzo o duración, o de las capacidades requeridas de los ingenieros y de la empresa.

El uso de métricas, y en este caso de una métrica de tamaño basada en la funcionalidad, nos puede ayudar a tener mejor control y una mejor evaluación de la inversión en proyectos de tecnología basados en SW.

Se van a realizar varias estimaciones mediante los métodos más utilizados, puntos de función, casos de uso y COCOMO.

3.6.1.2 Estimación por puntos de función

Es un método utilizado en Ingeniería del Software para medir el tamaño de un proyecto. Tiene por objetivo medir la funcionalidad entregada al cliente de manera independiente de la tecnología usada para la construcción y explotación del mismo y ser útil en cualquiera de las fases de la vida del software, desde el diseño inicial hasta la explotación y mantenimiento.

Tradicionalmente se ha medido el tamaño del software mediante distintas métricas: número de líneas de código, número de programas fuente, etc. Estas técnicas no resultan del todo eficientes en buenas prácticas profesionales.

Con el apoyo en esta técnica podemos conocer las dimensiones de la aplicación y estimar así el esfuerzo para su realización.

Los puntos de función se determinan indirectamente mediante la estimación de número de entradas, número de salidas, archivos de datos, consultas e interfaces, así como catorce valores de ajuste de complejidad que veremos posteriormente a la hora de estimar el esfuerzo de nuestro proyecto.

Lo primero que debemos hacer para realizar la siguiente estimación es determinar los puntos de función (PF) del proyecto, para ello habrá que seguir los siguientes pasos:

Primer Paso:

- ✓ Identificar el número de entradas de usuario
- ✓ Identificar el número de salidas de usuario
- ✓ Identificar el número de consultas de usuario
- ✓ Identificar el número de ficheros externos
- ✓ Identificar el número de ficheros internos

Segundo Paso:

- ✓ Determinar el grado de complejidad para cada función.

Tercer Paso:

- ✓ Calcular puntos de función no ajustados (PFNA).
- ✓ Calcular el factor de ajuste (en base a las 14 características generales FA).
- ✓ Calculo de los puntos de función ajustados (PFA).

Una vez realizados los pasos descritos anteriormente obtenemos como resultado final el tamaño del sistema en Puntos de Función Ajustados.

Comenzamos a aplicarlo a la aplicación web “Aprendiendo los números enteros”:

- ❖ **Número de entradas de usuario:** se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación.
 - Temarios: *complejidad simple*
 - Ejercicios: *complejidad simple*
 - Usuarios: *complejidad simple*
 - Test: *complejidad media*
 - Ficheros: *complejidad simple*
 - Total: 5 (4 entradas de usuario de complejidad simple y 1 media)*
- ❖ **Número de salidas de usuario:** se cuenta cada salida que proporciona al usuario información orientada a la aplicación, informes, pantallas, mensajes de error...
 - Menú principal: *complejidad simple*
 - Página de teoría: *complejidad simple*
 - Página de ejercicios: *complejidad media*
 - Menú alumno: *complejidad media*
 - Menú profesor: *complejidad media*
 - Página gestión alumnos: *complejidad media*
 - Página de contacto: *complejidad simple*
 - Página gestión temario: *complejidad alta*

- Página gestión ejercicios: *complejidad alta*
 - Página de carga de ficheros: *complejidad simple*
 - Página de generación de test: *complejidad alta*
 - Página de calificaciones: *complejidad simple*
 - Mensajes de error: *complejidad simple*
- Total: 13 (6 salidas de usuario de complejidad simple, 4 medias y 3 altas)*

❖ **Número de consultas de usuario:** Una petición es una entrada interactiva que genera alguna respuesta del software inmediata en forma de salida interactiva.

- Consulta de tema: *complejidad simple*
 - Consulta de ejercicio: *complejidad media*
 - Consulta de test: *complejidad simple*
 - Consulta de calificaciones: *complejidad simple*
 - Consulta alumnos: *complejidad media*
 - Consulta de ficheros: *complejidad simple*
- Total: 6 (4 consultas de usuario de complejidad simple y 2 medias)*

❖ **Número de ficheros lógicos internos:** Se cuenta cada archivo maestro lógico, esto es, un grupo lógico de datos que puede ser una parte de una gran base de datos o un archivo independiente.

- Ficheros de teoría: *complejidad media*
 - BBDD: *complejidad alta*
 - Ficheros de test: *complejidad media*
- Total: 3 (1 fichero lógico interno de complejidad alta y 2 medias)*

❖ **Número de ficheros externos:** Se cuentan todas las interfaces legibles por la máquina, que se utilizan para transmitir información a otro sistema.

- Manual de usuario: *complejidad simple*
- Total: 1 (1 fichero externo de complejidad simple)*

Una vez identificadas y clasificadas las funciones procedemos a calcular sus puntos de función sin ajustar PFNA:

Tipo de función	Complejidad	Total x complejidad	Total por tipo	Suma
Ficheros internos	Simple	0x7	0	35
	Media	2x10	20	
	Alta	1x15	15	
Ficheros externos	Simple	1x5	5	5
	Media	0x7	0	
	Alta	0x10	0	
Entradas de usuario	Simple	4x3	12	16
	Media	1x4	4	
	Alta	0x6	0	
Salidas de usuario	Simple	6x4	24	65
	Media	4x5	20	
	Alta	3x7	21	
Consultas de usuario	Simple	4x3	12	20
	Media	2x4	8	
	Alta	0x6	0	
TOTAL PUNTOS DE FUNCIÓN:				141

Tabla 1: Identificación de puntos de función no ajustados.

Una vez obtenidos los **PFNA**, deberán ser ajustados mediante un factor de ajuste (FA). El cálculo del factor de ajuste está basado en 14 características generales de los sistemas que miden la funcionalidad general y complejidad/influencia de la aplicación.

A cada característica de la siguiente tabla se le atribuye un peso de 0 a 5, e indica el grado de complejidad/influencia que tiene cada característica.

Grado	Descripción Complejidad	Grado	Descripción Influencia
0	No está presente o su complejidad no es tenida en cuenta	0	No está presente o no influye
1	Complejidad mínima	1	Influencia mínima
2	Complejidad moderada	2	Influencia moderada
3	Complejidad promedio	3	Influencia promedio
4	Complejidad significativa	4	Influencia significativa
5	Complejidad fuerte	5	Influencia fuerte

Tabla 2: Valores de evaluación de complejidad.

Calculamos el grado de complejidad de cada característica para el cálculo del factor de ajuste (FA):

Factor de Ajuste	Complejidad
1. Comunicación de datos	4
2. Funciones distribuidas	0
3. Rendimiento	4
4. Gran carga de trabajo	2
5. Frecuencia de transacciones	3
6. Entrada on-line de datos	4
7. Requisito de manejo del usuario final	1
8. Actualizaciones online	2
9. Procesos complejos	2
10. Utilización de otros sistemas	0
11. Facilidad de mantenimiento	3
12. Facilidad de operación	3
13. Instalación en múltiples lugares	1
14. Facilidad de cambio	4
Total	33

Tabla 3: Factores de complejidad.

Cálculo del factor de ajuste (FA) a partir de la suma de los 14 factores de complejidad (FC).

$$FA = (0,01 \times \Sigma FC) + 0,65 = (0,01 \times 33) + 0,65 = 0,98$$

* Siendo ΣFC la suma de los factores de complejidad.

Cálculo de puntos de función (PF)

$$PF = PFNA \times FA = 141 \times 0,98 = 138,18$$

Obtención del número de líneas de código (LDC)

Por último se muestra una tabla de equivalencias entre los puntos de función y los lenguajes de programación que se han utilizado para la obtención de las líneas de código estimadas.

Lenguaje	LDC/PF
PHP (similar a Java)	53
JAVASCRIPT	47
HTML	34

(Fuente QSM)

Tabla 4: Equivalencia entre líneas de código y puntos de función.

Debido a que el 15% de la aplicación se desarrollará en lenguaje PHP y sabemos que para cada punto de función equivale a 53 líneas de código (LDC):

$$\text{Líneas de Código PHP} = (138,18 \times 0,15) \times 53 = 1098,50$$

El 50% de la aplicación se desarrollará en JAVASCRIPT que tienen una equivalencia de 47 líneas de código, por tanto:

$$\text{Líneas de Código JavaScript} = (138,18 \times 0,50) \times 47 = 3247,20$$

Y el 35% restante será código HTML

$$\text{Líneas de Código HTML} = (138,18 \times 0,35) \times 34 = 1644,34$$

$$\text{Total de líneas de código} = 1098,50 + 3247,20 + 1644,34 = 5990 \approx 6 \text{ KLDC}$$

3.6.1.3 Estimación mediante COCOMO

COCOMO es el acrónimo de Modelo Constructivo de Costos (CONstructive COSt MOdel) es un modelo matemático utilizado para la estimación de costos software.

Se trata de un tipo de estimación para medir el esfuerzo y tiempo que supondrá realizar el nuevo sistema. Para poder llevarse a cabo, se tiene que haber estimado el número de líneas de código de nuestro sistema.

A continuación se detalla el procedimiento a seguir, para después realizar todos los cálculos. Existen tres modos diferentes de COCOMO en función de las características del sistema que se desarrollará. En función del modo elegido, los valores de los parámetros utilizados para los cálculos varían.

- **Orgánico:** proyectos relativamente sencillos, con poca innovación técnica, poca presión de tiempo, menores de 50 KDLC líneas de código, en los cuales se tiene experiencia de proyectos similares y se encuentran en entornos estables.
- **Empotrado:** proyectos bastantes complejos, en los que apenas se tiene experiencia y se engloban en un entorno de gran innovación técnica.
- **Semi-acoplado:** proyectos intermedios en complejidad y tamaño (menores de 300 KDLC), donde la experiencia en este tipo de proyectos es variable, y las restricciones intermedias.

Proyecto de Software	PERSONAS –MES	TIEMPO DE DESARROLLO
Orgánico	PM= 3,2 KLDC ^{1,05}	TD= 2,5 PM ^{0,38}
Empotrado	PM= 3,0 KLDC ^{1,12}	TD= 2,5 PM ^{0,35}
Semiacoplado	PM= 2,8 KLDC ^{1,20}	TD= 2,5 PM ^{0,32}

Tabla 5: Constantes de COCOMO.

El sistema a desarrollar de este proyecto se clasifica en un sistema Orgánico, puesto que por la aproximación que se ha hecho anterior mente de KLDC cumple que tenemos menos de 50 KLDC, además ya se tiene experiencia en proyectos similares y se va a encontrar en un entorno estable, por tanto:

$$\text{Esfuerzo nominal} = 3,2 \times \text{KLDC}^{1,05} = 3,2 \times 6^{1,05} = \mathbf{20,99 \approx 21 \text{ personas/mes}}$$

Si queremos pasar a COCOMO Intermedio se debe aplicar un factor de esfuerzo. Este factor son 15 atributos del proyecto. Cada uno de ellos se mide en diferentes grados para los cuales toma un valor distinto.

El valor de los factores aparece reflejado en la siguiente tabla:

FACTORES	VALORES DE LOS FACTORES					
	Muy Bajo	Bajo	Medio	Alto	Muy Alto	Extra
Fiabilidad requerida	0,75	0,88	1,00	1,15	1,14	
Tamaño de la base de datos		0,94	1,00	1,08	1,16	
Complejidad del software	0,70	0,85	1,00	1,11		1,65
Restricciones de tiempo de ejecución			1,00	1,06		1,66
Restricciones de memoria			1,00	1,15		1,56
Volatilidad del hardware		0,87	1,00	1,07		
Restricciones de tiempo de respuesta		0,87	1,00	0,86		
Calidad de los analistas	1,46	1,19	1,00	0,91		
Experiencia con el tipo de aplicación	1,29	1,13	1,00	0,90		
Experiencia con el hardware	1,21	1,10	1,00	0,95		
Experiencia con el lenguaje de programación	1,14	1,07	1,00	0,86		
Calidad de los programadores	1,42	1,17	1,00	0,91		
Técnicas modernas de programación	1,24	1,10	1,00	0,91		
Empleo de herramientas	1,24	1,10	1,00			
Restricciones a la duración del proyecto	1,23	1,08	1,00			

Tabla 6: Factores de ajuste de COCOMO.

Se calcula el esfuerzo con los factores seleccionados de la tabla anterior:

NOTA: Se han tenido en cuenta los factores de fiabilidad requerida, complejidad del software, calidad de los analistas, experiencia con el tipo de aplicación, experiencia con el lenguaje de programación y calidad de los programadores.

$$\text{Esfuerzo} = 21 \times 1,15 \text{ (fiabilidad)} \times 0,85 \text{ (complejidad SW)} \times 0,91 \text{ (calidad analistas)} \times 0,90 \text{ (experiencia con la aplicación)} \times 0,86 \text{ (experiencia con el lenguaje)} \times 0,91 \text{ (calidad programadores)} = \mathbf{13,16 \text{ personas/mes}}$$

En base al cálculo anterior, el tiempo de desarrollo será:

$$\text{Tiempo de Desarrollo: } 2,5 \text{ PM}^{0,38} = 2,5 \times (13,16)^{0,32} = \mathbf{5,70 \text{ meses} \approx 6 \text{ meses}}$$

Y por último el número medio de personas:

$$\text{Nº Medio de personas} = 13,16/5,70 = \mathbf{2,31 \text{ personas/mes} \approx 2 \text{ personas/mes}}$$

En resumen, se necesitarán 2 personas al mes durante 6 meses para realizar el trabajo.

3.6.1.4 Estimación mediante casos de uso

Para la estimación del tamaño de un sistema, una de las técnicas más difundidas, estables y fiables es el "**Método de Puntos de Casos de Uso**".

Este método está basado en los casos de uso. Se deben revisar los aspectos clave de los requerimientos para calcular un recuento de puntos de casos de uso sin ajustar, estudiar los factores técnicos y el entorno para crear los factores de ajuste y ajustar los factores para obtener los puntos de casos de uso ajustados (UCP) que después se transforma en estimación de esfuerzo (horas-persona).

A continuación se detalla este proceso.

En primer lugar se calculan los puntos de casos de uso sin ajustar, para ello clasificamos:

- Cada interacción entre actor y caso de uso según su complejidad y se le asigna un peso (número de actores x complejidad).

TIPO DE INTERACCIÓN	PESO
Simple (a través de un API)	1
Medio (a través de un protocolo)	2
Complejo (a través de una interfaz gráfica)	3

Tabla 7: Relación interacción-peso COCOMO.

Actores:

- ✓ Alumno no registrado
- ✓ Alumno registrado
- ✓ Profesor

$$\text{UAW} = \text{n}^\circ \text{ actores (3)} * \text{ complejidad (3)} = 9$$

- Clasificar la complejidad de cada caso de uso según el número de transacciones (escenarios).

TIPO DE INTERACCIÓN	NÚMERO DE TRANSACCIONES	PESO
Simple (a través de un API)	3 o menos	5
Medio (a través de un protocolo)	De 4 a 7	10
Complejo (a través de una interfaz gráfica)	7 o más	15

Tabla 8: Relación interacción-transacciones-peso COCOMO.

Se estiman los siguientes casos de uso:

- ✓ Gestión de alumnos (complejo)
- ✓ Gestión de temarios (complejo)
- ✓ Gestión de ejercicios (complejo)
- ✓ Generación de test (medio)
- ✓ Comunicación con alumno (medio)
- ✓ Subir archivos (simple)
- ✓ Consultar archivos (simple)
- ✓ Consultar temarios (simple)
- ✓ Consultar ejercicios (simple)
- ✓ Realizar ejercicios (complejo)
- ✓ Visualizar y descargar test (medio)
- ✓ Consultar calificaciones (simple)
- ✓ Comunicación con el profesor (medio)

$$\text{UUCW} = \text{n}^\circ \text{ UC} * \text{ complejidad} = (5 * 5) + (4 * 10) + (4 * 15)$$

$$\text{UUCW} = 25 + 40 + 60$$

$$\text{UUCW} = 125$$

$$\text{UUCP} = \text{Peso de los actores sin ajustar} + \text{Peso de los UC sin ajustar}$$

$$\text{UUCP} = 9 + 125 = 134$$

En segundo lugar, se calcula el factor de complejidad técnica (TCF). Cada uno de estos factores tiene un peso definido con los cuales se obtendrá puntos ponderados por cada uno de ellos, según la valoración que se le asigne. Para una mejor comprensión cada uno de estos puntos se debe evaluar según la siguiente escala:

DESCRIPCIÓN	VALOR
Irrelevante	De 0 a 2.
Medio	De 3 a 4.
Esencial	5

Tabla 9: Valoraciones posibles en cálculo del TCF COCOMO.

A continuación, se muestra una tabla con los ítems:

FACTOR	DESCRIPCIÓN	PESO	VALOR	COMENTARIOS
T1	Sistema distribuido	2	0	Sistema centralizado
T2	Objetivos de rendimiento	1	1	Velocidad limitada por entradas del usuario
T3	Eficiencia respecto al usuario final	1	4	Escasas restricciones de eficiencia
T4	Procesamiento complejo	1	1	No hay cálculos complejos
T5	Código reutilizable	1	3	Podría reutilizarse
T6	Instalación sencilla	0.5	2	Escasos requerimientos
T7	Fácil utilización	0.5	5	Muy fácil
T8	Portabilidad	2	3	No sería muy difícil portarlo a otros sistemas
T9	Fácil de cambiar	1	3	Coste medio mantenimiento
T10	Uso concurrente	1	2	Medio
T11	Características de seguridad	1	3	Normal
T12	Accesible por terceros	1	4	Los usuarios web tiene acceso directo si no son usuarios registrados
T13	Se requiere formación especial	1	1	Facilidad de uso, no requiere excesiva formación especial

Tabla 10: Factores de complejidad técnica COCOMO.

$$\begin{aligned} \text{TFactor} &= \sum (\text{valor} * \text{peso}) = \\ &(2*0) + (1*1) + (1*4) + (1*1) + (1*3) + (0.5*2) + (0.5*5) + (2*3) + (1*3) + (1*2) + (1*3) + (1*4) + (1*1) \\ &= 0 + 1 + 4 + 1 + 3 + 1 + 2.5 + 6 + 3 + 2 + 1 + 1 + 1 = 26.5 \end{aligned}$$

$$\text{TCF} = 0.6 + (0.01 * \text{TFactor}) = 0.6 + (0.01 * 26.5) = 0.865$$

En tercer lugar, se calcula el factor de entorno (EF), a través de la fórmula que se presenta a continuación:

$$\text{EFactor} = \text{Sum}(\text{Valor} * \text{Peso})$$

El valor se calcula de la misma forma que en el paso anterior.

FACTOR	DESCRIPCIÓN	PESO	VALOR
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	4
E2	Experiencia en la aplicación.	0.5	4
E3	Experiencia en orientación a objetos.	1	4
E4	Capacidad del analista líder.	0.5	5
E5	Motivación.	1	5
E6	Estabilidad de los requerimientos	2	5
E7	Personal a tiempo parcial	-1	0
E8	Dificultad del lenguaje de programación	-1	0

Tabla 11: Factores de entorno COCOMO

$$E_{\text{Factor}} = \sum (\text{valor} * \text{peso}) = (1.5 * 4) + (0.5 * 4) + (1 * 4) + (0.5 * 5) + (1 * 5) + (2 * 5) + (-1 * 0) + (-1 * 0) = 6 + 2 + 4 + 2.5 + 5 + 10 + 0 + (-0) = 29.5$$

$$EF = 1.4 + (-0.03 * E_{\text{Factor}}) = 1.4 + (-0.03 * 29.5) = 0.515$$

En cuarto lugar, se calculan los puntos de caso de uso ajustado mediante la siguiente expresión:

$$UCP = UUCP * TCF * EF = 134 * 0.865 * 0.515 = 59.69$$

Por último, se calcula la estimación del esfuerzo de programación, mediante la siguiente expresión:

$$\text{Esfuerzo} = UCP * \text{Factor Productividad} = 59.69 * 20 = 1193,80 \text{ horas/persona}$$

En la que el factor de productividad según Karner toma un valor aproximado a 20 horas/persona por UCP, obteniéndose al contar los factores de entorno, a través de la fórmula siguiente:

$$\text{Valor} = (R_1 - R_6 \text{ con } n_i < 3) + (R_7 - R_8 \text{ con } n_j > 3)$$

Como este método es una estimación al alza, consideraremos que las horas por persona resultantes no se corresponden solo a la parte de la programación, sino que se corresponde al proyecto en su totalidad, por lo que la estimación por partes del proyecto sería:

Para las estimaciones anteriores estamos considerando un único trabajador.

Y considerando que se trabajan 8 horas diarias durante 26 días al mes:

$$1193,80 \text{ horas} / 8 \text{ horas/día} = 149,23 \text{ días}$$

$$149,23 \text{ días} / 26 \text{ días} = 5,74 \text{ meses} \approx 5,7 \text{ meses}$$

Si hacemos una distribución genérica del esfuerzo por fases:

ACTIVIDAD	PORCENTAJE
Análisis (10%)	14,9 días
Planificación (5%)	7,46 días
Diseño (20%)	29,84 días
Programación (50%)	74,61 días
Pruebas (15%)	22,38 días

Tabla 12: Estimación por fases.

Para la planificación temporal, nos quedaremos con esta estimación ya que creo que se ajusta más al trabajo fin de grado que se va a llevar a cabo.

3.6.1.5 Planificación temporal

Descripción de la planificación temporal

En este apartado se detalla la planificación temporal del proyecto a realizar, presentando los diagramas de tareas de Gantt, con sus respectivas duraciones en días.

Desde el punto de vista de la planificación temporal, el trabajo a realizar dentro del proyecto estará dividido en una serie de unidades que podemos medir de forma cuantitativa. Estas unidades son las tareas, cada una de ellas tendrá una determinada duración, que será un tiempo estimado.

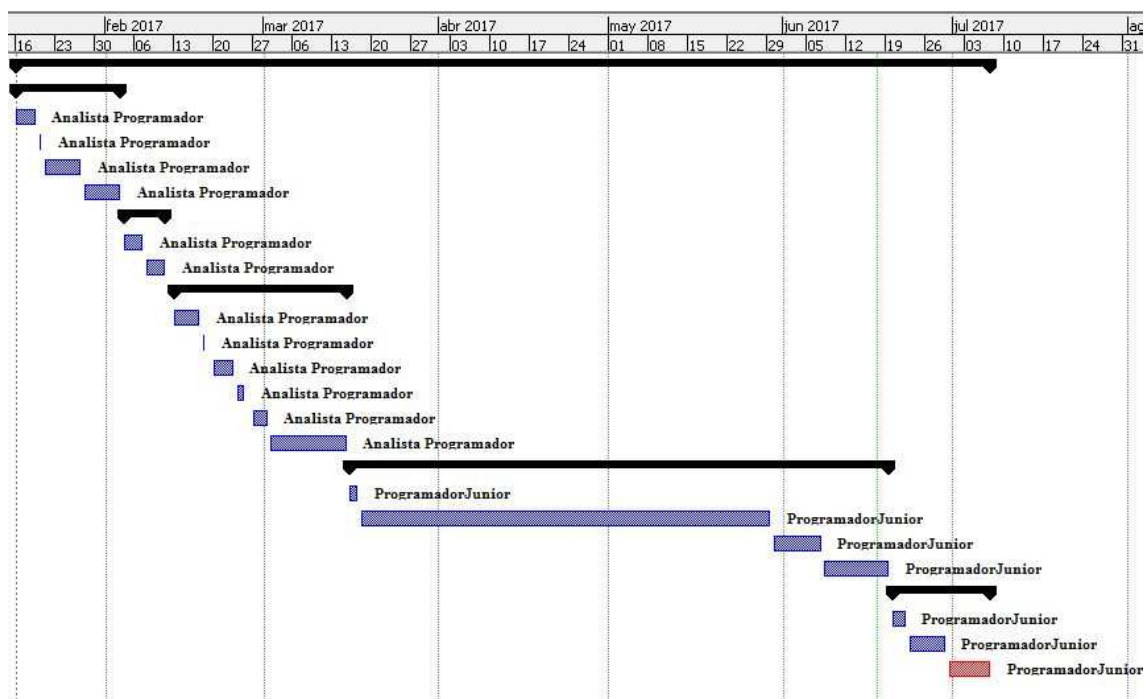
Estas tareas se relacionan entre sí mediante órdenes de dependencia o de simultaneidad. Las órdenes son los vínculos entre las tareas. De esta forma podremos definir, por ejemplo, que una tarea no pueda ser iniciada hasta haber acabado otro, o bien, que dos tareas puedan ser iniciadas a la vez.

Cronograma

En la imagen siguiente podemos observar que el proyecto se inicia el 16 de Enero de 2017y finaliza al 7 de Julio de 2017, lo que indica un total de 149 días laborables, teniendo en cuanto jornadas de 8 horas de Lunes a Sábado.

Nombre	Duración	Inicio	Terminado	Nombres del Recurso
☐ TFG	149 days	16/01/17 8:00	7/07/17 17:00	Ordenador Personal;Recuursos software;Otros...
☐ Análisis	17 days	16/01/17 8:00	3/02/17 17:00	Analista Programador
Estudio del problema	4 days	16/01/17 8:00	19/01/17 17:00	
Estudio de aplicaciones similares	1 day	20/01/17 8:00	20/01/17 17:00	
Estudio de objetivos	6 days	21/01/17 8:00	27/01/17 17:00	
Estudios de requisitos	6 days	28/01/17 8:00	3/02/17 17:00	
☐ Planificación	7 days	4/02/17 8:00	11/02/17 17:00	Analista Programador
Estimación presupuestaria	3 days	4/02/17 8:00	7/02/17 17:00	
Estimación presupuestaria	4 days	8/02/17 8:00	11/02/17 17:00	
☐ Diseño	27 days	13/02/17 8:00	15/03/17 17:00	Analista Programador
Modelado estático del sistema	5 days	13/02/17 8:00	17/02/17 17:00	
Revisión del modelo estático	1 day	18/02/17 8:00	18/02/17 17:00	
Modelado de comportamiento del sistema	4 days	20/02/17 8:00	23/02/17 17:00	
Revisión del modelo de comportamiento	2 days	24/02/17 8:00	25/02/17 17:00	
Modelado de base de datos	3 days	27/02/17 8:00	1/03/17 17:00	
Diseño de manual técnico	12 days	2/03/17 8:00	15/03/17 17:00	
☐ Implementación	82 days	16/03/17 8:00	19/06/17 17:00	ProgramadorJunior
Preparación de recursos materiales	2 days	16/03/17 8:00	17/03/17 17:00	
Implementación del sistema	62 days	18/03/17 8:00	29/05/17 17:00	
Revisión básica del sistema	8 days	30/05/17 7:00	7/06/17 17:00	
Diseño del manual de usuario	10 days	8/06/17 7:00	19/06/17 17:00	
☐ Pruebas	16 days	20/06/17 7:00	7/07/17 17:00	ProgramadorJunior
Pruebas parciales del sistema	3 days	20/06/17 7:00	22/06/17 17:00	
Prueba global del sistema	6 days	23/06/17 7:00	29/06/17 17:00	
Depuración de errores	7 days	30/06/17 7:00	7/07/17 17:00	

Diagrama de Gantt



A continuación, se explica brevemente como están constituidas cada una de las fases.

Fase de Análisis:

☐ Análisis	17 days	16/01/17 8:00	3/02/17 17:00
Estudio del problema	4 days	16/01/17 8:00	19/01/17 17:00
Estudio de aplicaciones similares	1 day	20/01/17 8:00	20/01/17 17:00
Estudio de objetivos	6 days	21/01/17 8:00	27/01/17 17:00
Estudios de requisitos	6 days	28/01/17 8:00	3/02/17 17:00

Esta es la fase inicial, por lo que se realizará un estudio detallado de lo que nos solicita el cliente, para ello es necesario tener una toma de contacto con el cliente, en la cual se especificaran las necesidades y requisitos que quiere que cumpla su aplicación.

Fase de Planificación:

☐ Planificación	7 days	4/02/17 8:00	11/02/17 17:00
Estimación presupuestaria	3 days	4/02/17 8:00	7/02/17 17:00
Estimación presupuestaria	4 days	8/02/17 8:00	11/02/17 17:00

En esta segunda fase se va a realizar un estudio de la planificación para tener un conocimiento del presupuesto estimado necesario para desarrollar el trabajo.

Fase de Diseño:

En la tercera fase, se diseñará la estructura de la aplicación y de todos los componentes que la engloban.

☐ Diseño	27 days	13/02/17 8:00	15/03/17 17:00
Modelado estático del sistema	5 days	13/02/17 8:00	17/02/17 17:00
Revisión del modelo estático	1 day	18/02/17 8:00	18/02/17 17:00
Modelado de comportamiento del sistema	4 days	20/02/17 8:00	23/02/17 17:00
Revisión del modelo de comportamiento	2 days	24/02/17 8:00	25/02/17 17:00
Modelado de base de datos	3 days	27/02/17 8:00	1/03/17 17:00
Diseño de manual técnico	12 days	2/03/17 8:00	15/03/17 17:00

Fase de Implementación:

☐ Implementación	82 days	16/03/17 8:00	19/06/17 17:00
Preparación de recursos materiales	2 days	16/03/17 8:00	17/03/17 17:00
Implementación del sistema	62 days	18/03/17 8:00	29/05/17 17:00
Revisión básica del sistema	8 days	30/05/17 7:00	7/06/17 17:00
Diseño del manual de usuario	10 days	8/06/17 7:00	19/06/17 17:00

En la cuarta fase, se llevará a cabo la implementación.

Fase de Pruebas:

<input type="checkbox"/> Pruebas	16 days	20/06/17 7:00	7/07/17 17:00
Pruebas parciales del sistema	3 days	20/06/17 7:00	22/06/17 17:00
Prueba global del sistema	6 days	23/06/17 7:00	29/06/17 17:00
Depuración de errores	7 days	30/06/17 7:00	7/07/17 17:00

La quinta y última fase se basa en la realización de pruebas y depuración de errores de la aplicación obtenida.

A continuación podemos observar cada una de las tareas de las que consta la aplicación y los recursos (materiales y humanos) necesarios para desarrollar las tareas definidas.

Nombre	Trabajo	Duración	Inicio	Terminado
<input type="checkbox"/> TFG	2.384 horas	149 days	16/01/17 8:00	7/07/17 17:00
<i>Recursos software</i>	<i>596 horas</i>	<i>74,5 days</i>	<i>16/01/17 8:00</i>	<i>28/04/17 13:00</i>
<i>Otros recursos</i>	<i>596 horas</i>	<i>74,5 days</i>	<i>16/01/17 8:00</i>	<i>28/04/17 13:00</i>
<input type="checkbox"/> Análisis	136 horas	17 days	16/01/17 8:00	3/02/17 17:00
Estudio del problema	32 horas	4 days	16/01/17 8:00	19/01/17 17:00
<i>Analista Programador</i>	<i>32 horas</i>	<i>4 days</i>	<i>16/01/17 8:00</i>	<i>19/01/17 17:00</i>
Estudio de aplicaciones si	8 horas	1 day	20/01/17 8:00	20/01/17 17:00
<i>Analista Programador</i>	<i>8 horas</i>	<i>1 day</i>	<i>20/01/17 8:00</i>	<i>20/01/17 17:00</i>
Estudio de objetivos	48 horas	6 days	21/01/17 8:00	27/01/17 17:00
<i>Analista Programador</i>	<i>48 horas</i>	<i>6 days</i>	<i>21/01/17 8:00</i>	<i>27/01/17 17:00</i>
Estudios de requisitos	48 horas	6 days	28/01/17 8:00	3/02/17 17:00
<i>Analista Programador</i>	<i>48 horas</i>	<i>6 days</i>	<i>28/01/17 8:00</i>	<i>3/02/17 17:00</i>
<input type="checkbox"/> Planificación	56 horas	7 days	4/02/17 8:00	11/02/17 17:00
Estimación presupuestari	24 horas	3 days	4/02/17 8:00	7/02/17 17:00
<i>Analista Programador</i>	<i>24 horas</i>	<i>3 days</i>	<i>4/02/17 8:00</i>	<i>7/02/17 17:00</i>
Estimación presupuestari	32 horas	4 days	8/02/17 8:00	11/02/17 17:00
<i>Analista Programador</i>	<i>32 horas</i>	<i>4 days</i>	<i>8/02/17 8:00</i>	<i>11/02/17 17:00</i>
<input type="checkbox"/> Diseño	216 horas	27 days	13/02/17 8:00	15/03/17 17:00
Modelado estático del sis	40 horas	5 days	13/02/17 8:00	17/02/17 17:00
<i>Analista Programador</i>	<i>40 horas</i>	<i>5 days</i>	<i>13/02/17 8:00</i>	<i>17/02/17 17:00</i>
Revisión del modelo está	8 horas	1 day	18/02/17 8:00	18/02/17 17:00
<i>Analista Programador</i>	<i>8 horas</i>	<i>1 day</i>	<i>18/02/17 8:00</i>	<i>18/02/17 17:00</i>
Modelado de comportami	32 horas	4 days	20/02/17 8:00	23/02/17 17:00
<i>Analista Programador</i>	<i>32 horas</i>	<i>4 days</i>	<i>20/02/17 8:00</i>	<i>23/02/17 17:00</i>
Revisión del modelo de c	16 horas	2 days	24/02/17 8:00	25/02/17 17:00
<i>Analista Programador</i>	<i>16 horas</i>	<i>2 days</i>	<i>24/02/17 8:00</i>	<i>25/02/17 17:00</i>
Modelado de base de dat	24 horas	3 days	27/02/17 8:00	1/03/17 17:00
<i>Analista Programador</i>	<i>24 horas</i>	<i>3 days</i>	<i>27/02/17 8:00</i>	<i>1/03/17 17:00</i>
Diseño de manual técnico	96 horas	12 days	2/03/17 8:00	15/03/17 17:00
<i>Analista Programador</i>	<i>96 horas</i>	<i>12 days</i>	<i>2/03/17 8:00</i>	<i>15/03/17 17:00</i>

☐ Implementación	656 horas	82 days	16/03/17 8:00	19/06/17 17:00
Preparación de recursos	16 horas	2 days	16/03/17 8:00	17/03/17 17:00
<i>Programador Junior</i>	<i>16 horas</i>	<i>2 days</i>	<i>16/03/17 8:00</i>	<i>17/03/17 17:00</i>
Implementación del sistema	496 horas	62 days	18/03/17 8:00	29/05/17 17:00
<i>Programador Junior</i>	<i>496 horas</i>	<i>62 days</i>	<i>18/03/17 8:00</i>	<i>29/05/17 17:00</i>
Revisión básica del sistema	64 horas	8 days	30/05/17 7:00	7/06/17 17:00
<i>Programador Junior</i>	<i>64 horas</i>	<i>8 days</i>	<i>30/05/17 7:00</i>	<i>7/06/17 17:00</i>
Diseño del manual de usuario	80 horas	10 days	8/06/17 7:00	19/06/17 17:00
<i>Programador Junior</i>	<i>80 horas</i>	<i>10 days</i>	<i>8/06/17 7:00</i>	<i>19/06/17 17:00</i>
☐ Pruebas	128 horas	16 days	20/06/17 7:00	7/07/17 17:00
Pruebas parciales del sistema	24 horas	3 days	20/06/17 7:00	22/06/17 17:00
<i>Programador Junior</i>	<i>24 horas</i>	<i>3 days</i>	<i>20/06/17 7:00</i>	<i>22/06/17 17:00</i>
Prueba global del sistema	48 horas	6 days	23/06/17 7:00	29/06/17 17:00
<i>Programador Junior</i>	<i>48 horas</i>	<i>6 days</i>	<i>23/06/17 7:00</i>	<i>29/06/17 17:00</i>
Depuración de errores	56 horas	7 days	30/06/17 7:00	7/07/17 17:00
<i>Programador Junior</i>	<i>56 horas</i>	<i>7 days</i>	<i>30/06/17 7:00</i>	<i>7/07/17 17:00</i>
<i>Ordenador Personal</i>	<i>1</i>	<i>1 day</i>	<i>3/08/17 8:00</i>	<i>3/08/17 17:00</i>

3.6.1.6 Presupuesto y costes por fases según OpenProject

En esta sección se especificará el coste total del proyecto obtenido a través de OpenProject.

En la imagen vemos un resumen de ello:

Información de Proyecto

General Estadísticas Notas

Nombre: TFG

Fecha Inicio: 16/01/17 8:00 Terminado: 7/07/17 17:00

Baseline Inicio: Baseline Termino:

Inicio Actual: Término actual:

Duración: 149 days Baseline Duracion: 0 days

Duración Actual: 0 days Duración Remanente: 149 days

Trabajo: 2.384 horas Baseline Trabajo: 0 horas

Trabajo Actual: 0 horas Trabajo Remanente: 2.384 horas

Costo: 14205,51 € Baseline Costo: 0,00 €

Costo Actual: 0,00 € Costo Remanente: 14205,51 €

Desarrollo de costes por fases:

Fase de Análisis:



Fase de Planificación:



Fase de Diseño:Fase de Implementación:Fase de Pruebas:**Desarrollo de costes en Recursos Humanos**

El proyecto está desarrollado por dos tipos de trabajadores (analista y programador), Cada uno de ellos desarrollará su labor, el analista durante el 35% del tiempo total y el 65% restante lo hará el programador.

Analista Programador Costo 5997,60 € Presu... 0,00 €

Programador Junior Costo 7706,72 € Presu... 0,00 €

Desarrollo de costes en Recursos Materiales:

Recursos software
Costo 16,32 €
Presu... 0,00 €

Otros recursos
Costo 378,00 €
Presu... 0,00 €

Ordenador Personal
Costo 106,87 €
Presu... 0,00 €

3.6.2 Presupuesto

El desarrollo del trabajo fin de grado ha sido realizado por Sandra García Sanz actuando de forma simultánea como analista-programador y programador.

El sueldo de cada puesto de trabajo es el siguiente.

PUESTO	SALARIO ANUAL	SALARIO MENSUAL
Analista programador	36.000 €/año	3000 €/mes
Programador	25.000 €/año	2083 €/mes

Tabla 13: Salarios trabajadores.

Se ha considerado que el trabajador va a tener 12 pagas, una por mes, y cada mes va a tener 26 días laborables y 8 horas por día.

Como el proyecto se realiza durante 5,7 meses, 3,25 meses aproximados trabajando como programador, y 2,45 como analista, el total de recursos humanos es:

$$3000 \text{ €} \times 2 \text{ meses (AP)} = 6.000 \text{ €}$$

$$2083 \text{ €} \times 3,7 \text{ meses (P)} = 7707,10 \text{ €}$$

$$\text{Total} = \underline{\underline{13707,10 \text{ €}}}$$

Recursos hardware

Se ha considerado que el ordenador personal está valorado en 900€ con un ciclo de vida de cuatro años. Cada cuatro años se cambiará de ordenador, por tanto al año, se devaluará 225€, y como el proyecto durará 5,7 meses, entonces el coste de uso del ordenador será de:

$$225 \text{ €/año ordenador} \quad \frac{\quad}{\quad} \quad 1 \text{ año}$$

$$X \quad \frac{\quad}{\quad} \quad 5,7/12 \text{ año}$$

$$X = (5,7/12) * 225 = 106,87 \text{ € valdrá el ordenador durante el TFG.}$$

Cómo el proyecto será desarrollado por un trabajador, solo se necesitará un ordenador, por tanto, los gastos de productos hardware son **un total de 106,87 €**.

Los requisitos asociados a los equipos incorporados para el desarrollo del proyecto serán:

Sistema:

- Windows 10
- Intel (R) Core(TM) i7 CPU920 @ 2.67GHz 2.66GHz.
- 6.00 GB de RAM.
- Sistema operativo de 64 bits.

El valor de estos requisitos viene incluido en el precio del equipo.

Recursos Software

Anteriormente se ha realizado una descripción de cada una de las aplicaciones que se utilizarán en el desarrollo del trabajo. Se va a especificar el coste que supone el uso de cada una de ellas.

Para abaratar el coste total del proyecto, se utilizará en la medida de lo posible software libre.

A continuación se muestra una tabla, con la relación software/coste:

APLICACIÓN	DESCRIPCIÓN	COSTE
Adobe Reader 11	Programa de manipulación de archivos PDF.	0.00 €
Open Project 1.4	Herramienta de planificación de proyectos.	0.00 €
StarUML 5.02	Herramienta para el diseño de diagramas de la aplicación,	0.00 €
Google Chrome	Navegador web para buscar información y cargar la aplicación desde nuestro sistema.	0.00 €
PDF creator 1.0.2	Convertor de documentos a archivos PDF.	0.00 €
Dia 0.97.1	Herramienta para la construcción de diferentes diagramas.	0.00 €
Windows 10	Sistema Operativo	0.00 €
Microsoft office 2010	Herramienta ofimática para la realización de la documentación.	40 €

Tabla 14: Recursos Software / coste.

Cómo podemos observar en la tabla se muestra el coste total anual de las aplicaciones empleadas.

El total de la suma del coste de aplicaciones es 40 €/año.

Como el proyecto durará 149 días, el coste de las aplicaciones utilizadas deberá ser proporcional al tiempo de la duración del proyecto.

$$\begin{array}{r} 40 \text{ €/año ordenador} \quad \underline{\hspace{2cm}} \quad 365 \text{ días} \\ X \quad \quad \quad \underline{\hspace{2cm}} \quad 149 \text{ días} \end{array}$$

$$X = (149 \cdot 40) / 365 = 16,32 \text{ € valdrá el software durante el proyecto.}$$

Otros Recursos

En este apartado se incluirán los gastos que no tienen nada que ver con el desarrollo de la aplicación web, pero a su vez, son necesarios para el desarrollo de la misma.

Podemos destacar los siguientes gastos:

- Material de oficina (folios, bolígrafos, cartuchos de impresora, cd's ...): 50€.
- Impresión y reprografía (impresión y encuadernación de la documentación): 100€.
- Conexión a internet: 228 € (40€/mes * 5,7 meses).

Por tanto el coste total de otros recursos asciende a: 378 €.

Presupuesto Total

En la siguiente tabla se encuentra un desglose de los costes necesarios para llevar a cabo el proyecto. Se va a incluir el esfuerzo y todos los recursos utilizados:

RECURSOS	COSTE
Humanos	13707,10 €
Hardware	106,87 €
Software	16,32 €
Otros	378 €

Tabla 15: Presupuesto total estimado.

TOTAL: 14208,29 €

3.6.2.1 Duración real del proyecto y comparativa estimación/realidad

El trabajo fin de grado ha sido realizado entre el 1 de Febrero de 2017 y el 9 de Julio de 2017. Durante ese tiempo se ha trabajado de lunes a sábado una media de 8 horas diarias.

Por tanto se han empleado 136 días y un total de 1088 horas.

Si comparamos con la estimación anterior en la que obtuvimos un total de 149,23 días laborales y 1193,80 horas, podemos ver que conseguimos una rebaja temporal de aproximadamente el 10%.

Con esta rebaja temporal disminuyó también el presupuesto utilizado, quedando así:

El sueldo de cada puesto de trabajo es el siguiente.

PUESTO	SALARIO ANUAL	SALARIO MENSUAL
Analista programador	36.000 €/año	3000 €/mes
Programador	25.000 €/año	2083 €/mes

Tabla 16: Salarios trabajadores.

Se ha considerado que el trabajador va a tener 12 pagas, una por mes, y cada mes va a tener 26 días laborables y 8 horas por día.

Como el proyecto se realiza durante 5,23 meses, 1.5 meses aproximados trabajando como programador, y 1.5 como analista, el total de recursos humanos es:

$$\begin{aligned}
 &3000 \text{ €} \times 1.83 \text{ meses (Analista)} = 5490 \text{ €} \\
 &2083 \text{ €} \times 3.39 \text{ meses (Programador)} = 7061,37 \text{ €}
 \end{aligned}$$

$$\text{Total} = \underline{\underline{12551,37 \text{ €}}}$$

El coste de hardware y software se reducirá, ya que se estimó para 5,7 meses y tendremos que recalcularlo para 5,23 meses.

$$\text{P. Hardware para 5,7 meses} = 106,87\text{€} \rightarrow \text{P. Hardware para 5,23 meses} = 98,05\text{€}$$

$$\text{P. Software para 5,7 meses} = 16,32\text{€} \rightarrow \text{P. Software para 5,23 meses} = 14,97\text{€}$$

El resto de materiales quedará igual, a excepción de la conexión ADSL que quedará en 209,20€. Por tanto, en recursos materiales se reduce a **359,20€**.

Por tanto, el coste total del proyecto sería:

RECURSOS	COSTE
Humanos	12551,37 €
Hardware	98,05 €
Software	14,97 €
Otros	359,20 €

Tabla 17: Presupuesto total real.

TOTAL: 13023,59 EUROS

Hemos conseguido bajar los costes del proyecto un 8%.

La estimación por puntos de función nos calculó un total de 6KLDC, por lo que fue bastante acertada.

En concreto esta es la comparación de la estimación y la realidad:

LENGUAJE	LDC Estimadas	LDC reales
PHP	1098,50	750
JAVASCRIPT	3247,20	3000
HTML	1644,34	2000
TOTAL	5990	5750

Tabla 18: Comparación LDC estimadas/reales.

3.7 Consideraciones

En este punto se va a tener en cuenta los aspectos más relevantes a la hora de comprender y desarrollar la aplicación web “Aprendiendo los números enteros”.

3.7.1 Consideraciones sobre la fase de análisis

➤ Sobre casos de uso:

Para representar las diferentes situaciones con las que se pueden encontrar los usuarios de la aplicación se utilizará la técnica de casos de uso, ya que es una técnica simple y muy representativa.

➤ Sobre los diagramas de secuencia:

Los diagramas de secuencia nos ayudaran a la hora de modelar las iteraciones entre actores y el sistema.

Lo normal es un diagrama de secuencia por cada caso de uso.

3.7.2 Consideraciones sobre la fase de diseño

A la hora de diseñar la aplicación web se determinó realizar el frontEnd con el framework AngularJS que basado en el lenguaje JavaScript y el backEnd con PHP, y para el tratamiento de los datos, se usó un sistema gestor de bases de datos relacional MySQL.

➤ Sobre el diseño de la interfaz:

A la hora de decidir sobre el diseño de la interfaz, puesto que iba a ser una aplicación destinada a la enseñanza de niños de 6º de Primaria, se tomó la decisión que ésta fuera visualmente amigable, interactiva, que llamara la atención y que sobre todo fuera de fácil manejo tanto para el alumno como para el profesor. De ésta manera, el profesor podría ser el “administrador” de la aplicación sin necesidad de tener conocimientos avanzados en materias informáticas.

3.7.3 Consideraciones sobre la fase de implementación

En esta fase, es donde se determinará el trato de los datos introducidos por los usuarios de la aplicación. Como estos datos deben perdurar en el tiempo, se realizará como se ha dicho anteriormente, a través de una base de datos en MySQL.

3.7.4 Consideraciones sobre la fase de pruebas

Para el funcionamiento correcto de la aplicación web deberá estar alojada en un servidor externo, para poder realizar las pruebas correspondientes al funcionamiento de la misma, la aplicación estará configurada para trabajar en un servidor local. Se llevará a cabo mediante el uso de la herramienta XAMPP.

3.8 Conclusiones y posibles ampliaciones

3.8.1 Conclusiones

Aunque desde bien pequeña quise ser “informática”, siempre me ha llamado la atención el ámbito de la enseñanza. Así que cuando llegó el momento de elegir una idea para realizar el trabajo fin de grado, al leer la propuesta de mi tutora, lo tuve claro, me pareció una idea estupenda y que puede que, cuando crezca un poco mis sobrinos puedan utilizarla.

Después de estos meses de desarrollo y de volver a las “aulas” puedo decir que me siento satisfecha de haber puesto en práctica lo aprendido en las últimas asignaturas cursadas en el Curso de Adaptación y aplicar todos aquellos conocimientos que he ido aprendiendo a lo largo de estos años que llevo trabajando. Además, también he aprendido cosas nuevas, como a programar con el framework AngularJS.

En esta carrera y profesión, ¡siempre se está aprendiendo!

3.8.2 Posibles ampliaciones

Esta aplicación web se podría ampliar con varias ideas que, con más tiempo podrían haberse realizado:

- Comunicación con los padres vía correo electrónico.
- Ampliación de la aplicación para otras áreas de la enseñanza de las Matemáticas.
- Comunicación automática por correo electrónico de las calificaciones.
- Dar la posibilidad al profesor de añadir ciertas observaciones sobre un alumno a la hora de añadir calificaciones.
- Ampliar el envío de ejercicio entre alumno y profesor, para que un alumno con tan solo un click pueda enviar los ejercicios directamente al profesor sin necesidad de tener que descargárselo, realizarlo y volvérselo a enviar al profesor.

BLOQUE II:
MANUAL TÉCNICO

4. ANÁLISIS

4.1 Introducción

Al realizar el Análisis del Sistema se pretende obtener una colección completa y detallada de los requisitos del sistema, tomando como punto de partida los requisitos identificados en el estudio que se ha hecho anteriormente (punto 3 del manual de Memoria). El documento generado en esta fase será la base para el desarrollo de la fase de diseño e implementación del sistema.

Con este trabajo se ha querido desarrollar un sistema que permita tanto a alumnos como profesores tener un acceso rápido a los conceptos de los números enteros

El sistema almacenará y manipulará datos sobre temarios y ejercicios, y en el que existirán varios tipos de usuarios, por una parte, los usuarios registrados, que serán los profesores que harán de “administradores” y los alumnos registrados y por otra, cualquier tipo de usuario no registrado.

4.2 Alcance del análisis

Esta fase tiene como objetivo obtener una especificación detallada del sistema que se va a diseñar. Mediante su producto, el Documento de Análisis del Sistema, se pretende captar cuales son las necesidades que tiene el cliente.

En primer lugar, se definirá cual es el alcance del sistema que se desea desarrollar, así como el entorno tecnológico asociado al proyecto. Además, se identificarán los diferentes participantes que aparecen a lo largo de la vida del proyecto, así como los usuarios finales.

A continuación, se definirán cuáles son los requisitos software que debe cumplir el sistema a desarrollar, tomando como punto de partida los casos de uso y los requisitos de usuario.

4.3 Requisitos específicos

A continuación, se detallan las restricciones que deberá cumplir el sistema a diseñar:

- En un primer lugar la aplicación funcionará en cualquier S.O. solo será necesario un navegador para poder utilizarla.
- La interfaz, de la aplicación Web, con la que interaccionará los usuarios finales, debe proporcionar todas las funcionalidades descritas en los requisitos software del sistema, de una forma amigable e intuitiva, evitando posibles ambigüedades que puedan ocasionar confusión al usuario.
- El idioma usado en el desarrollo del sistema será el castellano.

4.3.1 Objetivos principales del sistema

OBJ-01	Gestión de usuarios
Versión	1.0
Autor	Sandra García Sanz
Fuentes	Autor del TFG.
Descripción	El sistema deberá gestionar los registros de usuarios, tanto alumnos como profesores. Debe ser capaz de gestionar tanto nuevos registros como las conexiones al sistema de los usuarios ya registrados.
Subobjetivos	OBJ-01.01 Registro de usuario. OBJ-01.02 Login de usuario. OBJ-01.03 Desconexión de usuario.
Importancia	Muy importante
Urgencia	Alta
Estado	Validado
Estabilidad	Alta
Comentarios	

Tabla 19: Objetivo 1 - Gestión de usuarios.

OBJ-02	Gestión de teoría
Versión	1.0
Autor	Sandra García Sanz
Fuentes	Autor del TFG.
Descripción	El sistema deberá gestionar el temario de la asignatura. Tanto la inserción, modificación y borrado del mismo, como su visualización por partes de los alumnos.
Subobjetivos	OBJ-02.01 Edición de teoría. OBJ-02.02 Consulta de teoría.
Importancia	Importante
Urgencia	Alta
Estado	Validado
Estabilidad	Alta
Comentarios	

Tabla 20: Objetivo 2 - Gestión de teoría

OBJ-03	Gestión de ejercicios
Versión	1.0
Autor	Sandra García Sanz
Fuentes	Autor del TFG.
Descripción	El sistema deberá gestionar los ejercicios de la asignatura. Tanto la inserción, modificación y borrado de los mismos, como su visualización por partes de los alumnos.
Subobjetivos	OBJ-03.01 Edición de ejercicios. OBJ-03.02 Consulta de ejercicios. OBJ-03.03 Corrección de ejercicios.
Importancia	Importante
Urgencia	Alta
Estado	Validado
Estabilidad	Alta
Comentarios	

Tabla 21: Objetivo 3 - Gestión de ejercicios.

OBJ-04	Gestión de test
Versión	1.0
Autor	Sandra García Sanz
Fuentes	Autor del TFG.
Descripción	El sistema deberá gestionar los test de la asignatura. Tanto la inserción de los mismos, como su visualización por partes de los alumnos.
Subobjetivos	OBJ-04.01 Inserción de test. OBJ-04.02 Consulta de test.
Importancia	Media
Urgencia	Media
Estado	Validado
Estabilidad	Alta
Comentarios	

Tabla 22: Objetivo 4 - Gestión de test.

OBJ-05	Gestión de ficheros
Versión	1.0
Autor	Sandra García Sanz
Fuentes	Autor del TFG.
Descripción	El sistema deberá gestionar la carga y descarga de ficheros por parte de alumnos y profesores, así como la visualización.
Subobjetivos	OBJ-05.01 Carga de ficheros OBJ-05.02 Descarga de ficheros OBJ-05.03 Visualización de ficheros
Importancia	Media
Urgencia	Media
Estado	Validado
Estabilidad	Alta
Comentarios	

Tabla 23: Objetivo 5 - Gestión de ficheros.

OBJ-06	Gestión de calificaciones
Versión	1.0
Autor	Sandra García Sanz
Fuentes	Autor del TFG.
Descripción	El sistema deberá gestionar inserción y consulta de calificaciones de los alumnos.
Subobjetivos	OBJ-06.01 Inserción de calificación OBJ-06.02 Consulta de calificación
Importancia	Importante
Urgencia	Alta
Estado	Validado
Estabilidad	Alta
Comentarios	

Tabla 24: Objetivo 6 - Gestión de calificaciones.

OBJ-07	Gestión de alumnos
Versión	1.0
Autor	Sandra García Sanz
Fuentes	Autor del TFG.
Descripción	El sistema proporcionará la interfaz necesaria para gestionar los datos personales y de contacto de los alumnos de la asignatura.
Subobjetivos	
Importancia	Muy importante
Urgencia	Alta
Estado	Validado
Estabilidad	Alta
Comentarios	

Tabla 25: Objetivo 7 - Gestión de alumnos.

OBJ-08	Contacto entre alumnos y profesores
Versión	1.0
Autor	Sandra García Sanz
Fuentes	Autor del TFG.
Descripción	El sistema permitirá el contacto entre alumno y profesor.
Subobjetivos	N/A
Importancia	Alta
Urgencia	Alta
Estado	Validado
Estabilidad	Alta
Comentarios	

Tabla 26: Objetivo 8 - Contacto entre alumnos y profesores.

4.3.2 Requisitos de información

Identificador	RI-01
Definición	Información de usuarios
Autores	Sandra García Sanz
Objetivos asociados	OBJ-01- Gestión de usuarios - OBJ-01.01 Registro de usuarios OBJ-05 - Gestión de datos de alumnos OBJ-06 - Gestión de calificaciones OBJ-07 - Gestión de ficheros OBJ-08 - Contacto
Descripción	El sistema deberá almacenar la información correspondiente a los usuarios de la aplicación.
Datos específicos	<ul style="list-style-type: none"> ■ Nombre ■ Apellidos ■ DNI ■ Rol ■ Correo electrónico ■ Contraseña ■ Estado ■ Notas ■ Fecha de creación
Importancia	Alta
Prioridad	Muy alta

Tabla 27: Requisitos de información de usuarios.

Identificador	RI-02
Definición	Información de teoría
Autores	Sandra García Sanz
Objetivos asociados	OBJ-02- Gestión de teoría - OBJ-02.01 Edición de teoría - OBJ-02.02 Consulta de teoría
Descripción	El sistema deberá almacenar la información correspondiente a los temarios creados por el profesor.
Datos específicos	<ul style="list-style-type: none"> ■ Nombre ■ Número de tema ■ Descripción ■ Texto ■ Visibilidad
Importancia	Alta
Prioridad	Muy alta

Tabla 28: Requisitos de información de teoría.

Identificador	RI-03
Definición	Información de ejercicios
Autores	Sandra García Sanz
Objetivos asociados	OBJ-03 - Gestión de ejercicios - OBJ-03.01 Edición de ejercicios - OBJ-03.02 Consulta de ejercicios
Descripción	El sistema deberá almacenar la información correspondiente a los ejercicios creados por el profesor.
Datos específicos	<ul style="list-style-type: none"> ■ Id ■ Nombre ■ Texto ■ Soluciones ■ Fecha de creación
Importancia	Alta
Prioridad	Muy alta

Tabla 29: Requisitos de información de ejercicios.

Identificador	RI-04
Definición	Información de test
Autores	Sandra García Sanz
Objetivos asociados	OBJ-04 - Gestión de test - OBJ-04.01 Inserción de test
Descripción	El sistema deberá almacenar la información correspondiente a los test creados por el profesor.
Datos específicos	<ul style="list-style-type: none"> ■ Nombre ■ Fecha ■ Descripción ■ Texto
Importancia	Alta
Prioridad	Muy alta

Tabla 30: Requisitos de información de test.

Identificador	RI-05
Definición	Información de ficheros
Autores	Sandra García Sanz
Objetivos asociados	OBJ-05 – Gestión de ficheros
Descripción	El sistema deberá almacenar los ficheros subidos por profesor y alumnos.
Datos específicos	<ul style="list-style-type: none"> ■ Nombre ■ Tipo ■ Propietario
Importancia	Alta
Prioridad	Muy alta

Tabla 31: Requisitos de información de ficheros.

4.3.3 Restricciones generales

Identificador	CRQ-01
Definición	Unicidad de usuario
Autores	Sandra García Sanz
Objetivos asociados	OBJ-01- Gestión de usuarios
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: Solo se podrá crear un usuario por correo electrónico. En caso de ya estar registrado dicho correo electrónico se informará al usuario.
Requisitos de información asociados	RI-01– Información de usuarios

Tabla 32: Restricción: unicidad de usuario.

Identificador	CRQ-02
Definición	Alumno validado por profesor
Autores	Sandra García Sanz
Objetivos asociados	OBJ-01- Gestión de usuarios
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: Solo se podrá crear un usuario alumno si previamente el profesor ha dado de alta sus datos, incluyendo el correo electrónico. En caso de no estar registrado dicho correo electrónico se informará al usuario alumno en el momento del registro.
Requisitos de información asociados	RI-01– Información de usuarios

Tabla 33: Restricción: validar alumno.

4.3.4 Requisitos funcionales

4.3.4.1 Definición de los actores

A continuación, se presentan los actores que pueden interactuar con la aplicación web.

ACT-01	Alumno no registrado
Descripción	Es un usuario que podrá acceder a la aplicación y consultar los temas que haya disponibles así como los ejercicios asociados a ella.

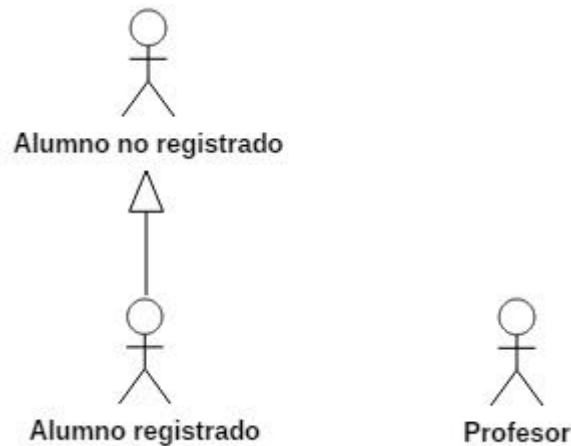
Tabla 34: Actor alumno no registrado.

ACT-02	Alumno registrado
Descripción	Es un usuario “alumno no registrado” que, a través de un formulario y completando los datos obligatorios quedará registrado en la aplicación y podrá acceder a un mayor contenido (test, comunicarse con el profesor, subir y descargar archivos...)

Tabla 35: Actor alumno registrado.

ACT-03	Profesor
Descripción	Es un usuario que tendrá el rol de “administrador”, podrá dar gestión a alumnos, temarios y ejercicios; además, podrá comunicarse con el alumno, subir y descargar archivos, incluir calificaciones...

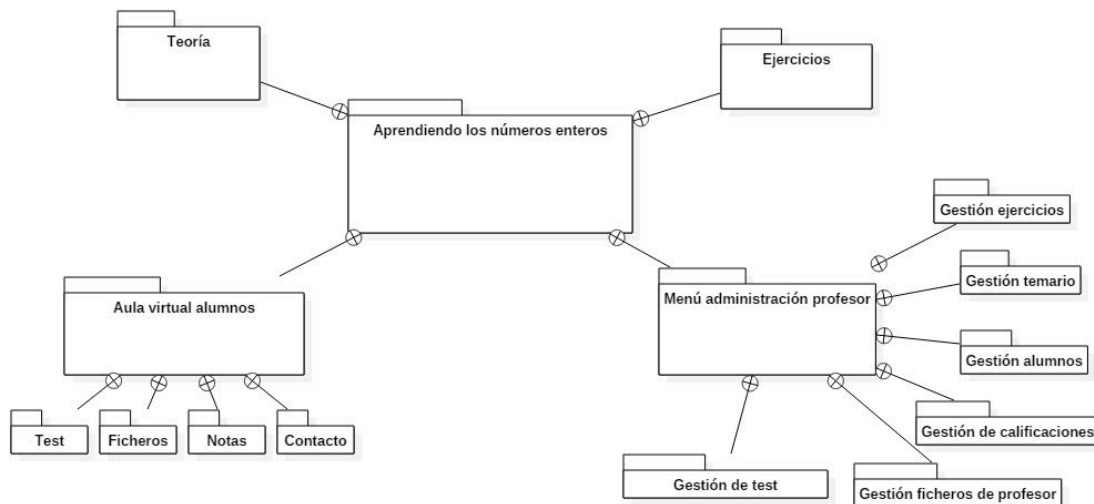
Tabla 36: Actor profesor.



4.3.4.2 Diagrama de subsistemas

Un diagrama de subsistemas muestra las partes funcionales de las que se compone el sistema.

En la siguiente imagen podemos observar que nuestra aplicación está compuesta por cinco subsistemas principales.



4.3.4.3 Diagrama de casos de uso

Un diagrama de casos de uso es una especie de diagrama de comportamiento del sistema al afrontar una tarea o requisito.

A continuación, se muestra los diagramas de casos de uso asociados a cada actor de la aplicación.

Diagrama de casos de uso del actor Alumno no registrado

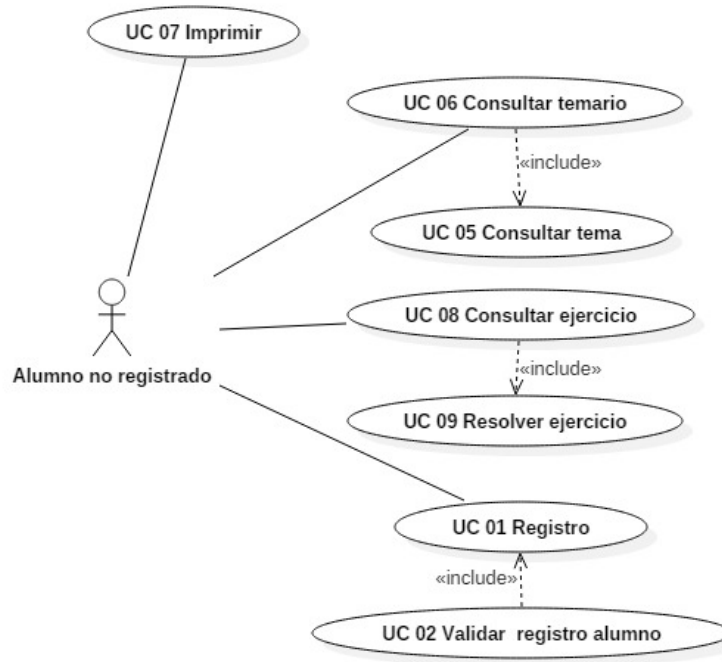


Diagrama de casos de uso del actor Alumno registrado

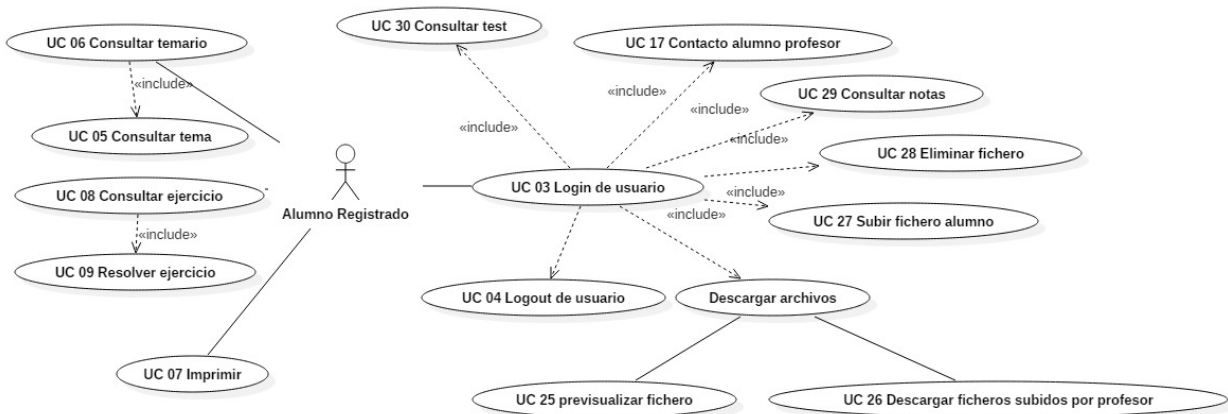
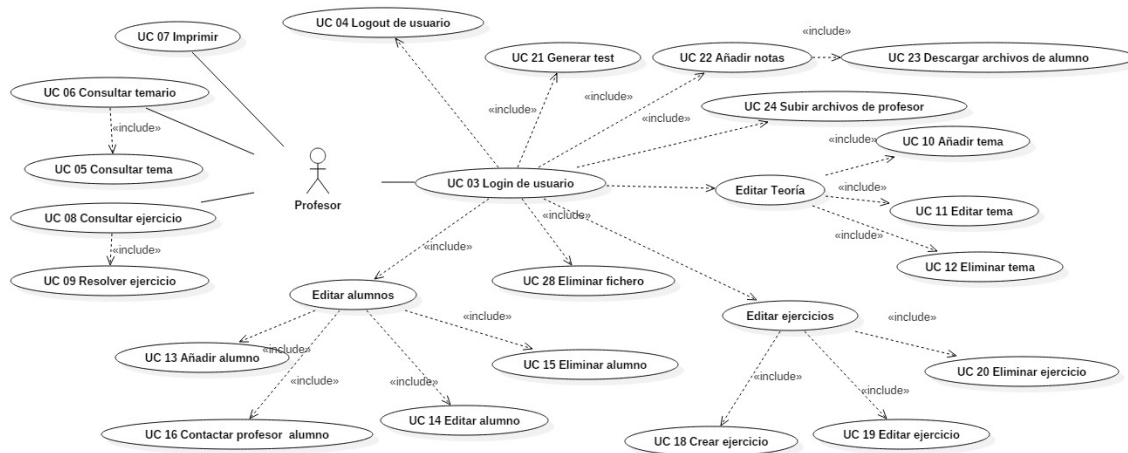


Diagrama de casos de uso del actor Profesor



4.3.4.4 Definición de casos de uso

UC01	Registro														
Actores	Profesor, alumno no registrado														
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor o un alumno no registrado deseen registrarse en la aplicación.														
Precondiciones	El profesor no debe estar dado de alta. El alumno no registrado no debe estar dado de alta. El alumno no registrado debe estar validado por el profesor.														
Secuencia normal	<table border="0"> <tr> <td>Paso 1</td> <td>El usuario pulsa en el menú principal sobre el botón aula virtual.</td> </tr> <tr> <td>Paso 2</td> <td>El sistema muestra la página de inicio de sesión.</td> </tr> <tr> <td>Paso 3</td> <td>El usuario pulsa sobre el botón “Crear cuenta”.</td> </tr> <tr> <td>Paso 4</td> <td>El sistema muestra el formulario de registro.</td> </tr> <tr> <td>Paso 5</td> <td>El usuario completa los campos del formulario y pulsa sobre el botón “Crear Cuenta”.</td> </tr> <tr> <td>Paso 6</td> <td>El sistema valida que todos los campos son correctos y crea el usuario.</td> </tr> <tr> <td>Paso 7</td> <td>El sistema devuelve un mensaje de éxito/error en la operación.</td> </tr> </table>	Paso 1	El usuario pulsa en el menú principal sobre el botón aula virtual.	Paso 2	El sistema muestra la página de inicio de sesión.	Paso 3	El usuario pulsa sobre el botón “Crear cuenta”.	Paso 4	El sistema muestra el formulario de registro.	Paso 5	El usuario completa los campos del formulario y pulsa sobre el botón “Crear Cuenta”.	Paso 6	El sistema valida que todos los campos son correctos y crea el usuario.	Paso 7	El sistema devuelve un mensaje de éxito/error en la operación.
Paso 1	El usuario pulsa en el menú principal sobre el botón aula virtual.														
Paso 2	El sistema muestra la página de inicio de sesión.														
Paso 3	El usuario pulsa sobre el botón “Crear cuenta”.														
Paso 4	El sistema muestra el formulario de registro.														
Paso 5	El usuario completa los campos del formulario y pulsa sobre el botón “Crear Cuenta”.														
Paso 6	El sistema valida que todos los campos son correctos y crea el usuario.														
Paso 7	El sistema devuelve un mensaje de éxito/error en la operación.														
Postcondiciones	El profesor o el alumno queda almacenado en la base de datos.														
Excepciones	<ul style="list-style-type: none"> ➤ Error en la conexión a base de datos ➤ Si el usuario no ha introducido todos los datos necesarios para el registro el sistema muestra un mensaje de alerta. ➤ Si el correo electrónico ya existe en la base de datos el sistema muestra un mensaje de alerta. 														
Rendimiento	3 segundos														
Frecuencia	Media														
Importancia	Vital														

Tabla 37: Caso de Uso 1 – Registro.

UC02	Validar registro alumno																
Actores	Alumno no registrado																
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un alumno no registrado desee registrarse en la aplicación.																
Precondiciones	El alumno debe haber sido dado de alta por el profesor en el sistema. El alumno no debe estar registrado.																
Secuencia normal	<table border="1"> <tbody> <tr> <td>Paso 1</td> <td>El usuario pulsa en el menú principal sobre el botón aula virtual.</td> </tr> <tr> <td>Paso 2</td> <td>El sistema muestra la página de inicio de sesión.</td> </tr> <tr> <td>Paso 3</td> <td>El usuario pulsa sobre el botón “Crear cuenta”.</td> </tr> <tr> <td>Paso 4</td> <td>El sistema muestra el formulario de registro.</td> </tr> <tr> <td>Paso 5</td> <td>El usuario completa los campos del formulario y pulsa sobre el botón “Crear Cuenta”.</td> </tr> <tr> <td>Paso 6</td> <td>El sistema valida que todos los campos son correctos.</td> </tr> <tr> <td>Paso 7</td> <td>El sistema comprueba que el alumno ha sido dado de alta previamente por el profesor.</td> </tr> <tr> <td>Paso 8</td> <td>El sistema almacena el usuario en la base de datos y devuelve un mensaje de éxito/error en la operación.</td> </tr> </tbody> </table>	Paso 1	El usuario pulsa en el menú principal sobre el botón aula virtual.	Paso 2	El sistema muestra la página de inicio de sesión.	Paso 3	El usuario pulsa sobre el botón “Crear cuenta”.	Paso 4	El sistema muestra el formulario de registro.	Paso 5	El usuario completa los campos del formulario y pulsa sobre el botón “Crear Cuenta”.	Paso 6	El sistema valida que todos los campos son correctos.	Paso 7	El sistema comprueba que el alumno ha sido dado de alta previamente por el profesor.	Paso 8	El sistema almacena el usuario en la base de datos y devuelve un mensaje de éxito/error en la operación.
Paso 1	El usuario pulsa en el menú principal sobre el botón aula virtual.																
Paso 2	El sistema muestra la página de inicio de sesión.																
Paso 3	El usuario pulsa sobre el botón “Crear cuenta”.																
Paso 4	El sistema muestra el formulario de registro.																
Paso 5	El usuario completa los campos del formulario y pulsa sobre el botón “Crear Cuenta”.																
Paso 6	El sistema valida que todos los campos son correctos.																
Paso 7	El sistema comprueba que el alumno ha sido dado de alta previamente por el profesor.																
Paso 8	El sistema almacena el usuario en la base de datos y devuelve un mensaje de éxito/error en la operación.																
Postcondiciones	El alumno queda guardado en la base de datos.																
Excepciones	<ul style="list-style-type: none"> ➤ Error en la conexión a base de datos ➤ Si el usuario no ha introducido todos los datos necesarios para el registro el sistema muestra un mensaje de alerta. ➤ Si el alumno no está dado de alta por el profesor el sistema muestra un mensaje de alerta. ➤ Si el correo electrónico ya existe en la base de datos el sistema muestra un mensaje de alerta. 																
Rendimiento	3 segundos																
Frecuencia	Media																
Importancia	Vital																

Tabla 38: Caso de Uso 2 – Validar registro alumno

UC03		Login de usuario	
Actores	Alumno registrado, profesor		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario desee iniciar sesión en la aplicación.		
Precondiciones	El usuario debe haberse registrado en el sistema.		
Secuencia normal	Paso 1	El usuario pulsa en el menú principal sobre el botón aula virtual.	
	Paso 2	El sistema muestra la página de inicio de sesión con el formulario correspondiente.	
	Paso 3	El usuario introduce sus datos de acceso y pulsa el botón “Acceder”	
	Paso 4	El sistema comprueba los datos introducidos.	
	Paso 5	El sistema crear una sesión de usuario.	
	Paso 6	El sistema redirige a la zona privada de alumno o profesor en función del tipo de usuario.	
Postcondiciones	Se ha creado una sesión para dicho usuario.		
Excepciones	<ul style="list-style-type: none"> ➤ Error en la conexión a base de datos ➤ Si el usuario no ha introducido todos los datos necesarios para el registro el sistema muestra un mensaje de alerta. ➤ Si el usuario no está registrado en la aplicación o los datos introducidos no son correctos el sistema muestra un mensaje de alerta. 		
Rendimiento	3 segundos		
Frecuencia	Media		
Importancia	Vital		

Tabla 39: Caso de Uso 3 – Login de usuario.

UC04	Logout de usuario	
Actores	Alumno registrado, profesor	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario quiera cerrar sesión en la aplicación.	
Precondiciones	El usuario debe haber iniciado sesión en el sistema.	
Secuencia normal	Paso 1	El usuario pulsa en uno de los diferentes botones de “Cerrar sesión” que hay en la zona privada de la aplicación.
	Paso 2	El sistema elimina la sesión del usuario.
	Paso 3	El sistema redirige al usuario a la página inicial.
Postcondiciones	La sesión es eliminada	
Excepciones	➤ Error en la conexión a base de datos	
Rendimiento	3 segundos	
Frecuencia	Media	
Importancia	Vital	

Tabla 40: Caso de Uso 4 – Logout de usuario.

UC05	Consultar tema	
Actores	Alumno registrado, alumno no registrados	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario quiera consultar los temarios de la asignatura.	
Precondiciones	El usuario ha accedido a la zona de consulta de temario	
Secuencia normal	Paso 1	El usuario pulsa en el botón “Consultar” sobre alguno de los temas de la asignatura.
	Paso 2	El sistema muestra los datos del tema seleccionado
Postcondiciones	N/A	
Excepciones	➤ Error en la conexión a base de datos	
Rendimiento	3 segundos	
Frecuencia	Media	
Importancia	Vital	

Tabla 41: Caso de Uso 5 – Consultar tema.

JC06		Consultar temario
Actores	Alumno registrado, alumno no registrados	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario quiera consultar un tema de la asignatura.	
Precondiciones	N/A	
Secuencia normal	Paso 1	El usuario pulsa en el botón “Teoría” el menú principal de la aplicación.
	Paso 2	El sistema muestra el listado de temas de la asignatura
Postcondiciones	N/A	
Excepciones	➤ Error en la conexión a base de datos	
Rendimiento	3 segundos	
Frecuencia	Media	
Importancia	Vital	

Tabla 42: Caso de Uso 6 – Consultar temario.

JC07		Imprimir
Actores	Alumno registrado, alumno no registrados, profesor	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario desee imprimir uno de los apartados habilitados para ello.	
Precondiciones	N/A	
Secuencia normal	Paso 1	El usuario pulsa el botón imprimir.
	Paso 2	El sistema muestra la visualización del documento a imprimir y la posibilidad de imprimir el documento o cancelar la impresión.
Postcondiciones	N/A	
Excepciones	➤ Error en la conexión a base de datos	
Rendimiento	3 segundos	
Frecuencia	Media	
Importancia	Vital	

Tabla 43: Caso de Uso 7 – Imprimir.

UC08	Consultar ejercicio	
Actores	Alumno registrado, alumno no registrado	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario desee consultar un ejercicio.	
Precondiciones	N/A	
Secuencia normal	Paso 1	El usuario pulsa el botón “Ejercicios” en el menú principal de la aplicación.
	Paso 2	El sistema muestra el listado de ejercicios disponibles divididos en dos secciones. “Ejemplos prácticos” y “Ejercicios por temas”
	Paso 3	El usuario pulsa sobre el ejercicio/tema deseado.
	Paso 4	El sistema muestra el listado de ejercicios disponibles para el tema seleccionado o el ejemplo seleccionado.
	Paso 5	El sistema muestra el ejercicio seleccionado.
Postcondiciones	N/A	
Excepciones	➤ Error en la conexión a base de datos	
Rendimiento	3 segundos	
Frecuencia	Media	
Importancia	Vital	

Tabla 44: Caso de Uso 8 – Consultar ejercicio.

UC09	Resolver ejercicio	
Actores	Alumno registrado, alumno no registrado	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario desee comprobar las soluciones de un ejercicio.	
Precondiciones	El usuario ha consultado un ejercicio	
Secuencia normal	Paso 1	El usuario introduce las soluciones al ejercicio en los campos habilitados.
	Paso 2	El sistema comprueba si las soluciones introducidas por el usuario son iguales a las almacenadas por el profesor.
	Paso 3	El sistema muestra un mensaje con los aciertos y/o fallos obtenidos.
Postcondiciones	N/A	
Excepciones	➤ Error en la conexión a base de datos	
Rendimiento	3 segundos	
Frecuencia	Media	
Importancia	Vital	

Tabla 45: Caso de Uso 9 – Resolver ejercicio.

UC10	Crear tema
Actores	Profesor
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee crear un nuevo tema.
Precondiciones	El profesor ha iniciado sesión
Secuencia normal	Paso 1 El profesor pulsa sobre el botón “Temario” del menú de administración de profesor.
	Paso 2 El sistema muestra un listado con el temario actual.
	Paso 3 El usuario pulsa sobre el botón “añadir tema”
	Paso 4 El sistema da la posibilidad de editar los campos del nuevo tema y además muestra el editor de texto.
	Paso 5 El usuario completa los campos requeridos y pulsa el botón “Guardar cambios”
	Paso 6 El sistema almacena el nuevo tema.
Postcondiciones	El tema queda guardado en el sistema.
Excepciones	➤ Error en la conexión a base de datos
Rendimiento	3 segundos
Frecuencia	Media
Importancia	Vital

Tabla 46: Caso de Uso 10 – Crear tema.

UC11	Editar tema	
Actores	Profesor	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee editar un tema.	
Precondiciones	El profesor ha iniciado sesión y el tema ya existe.	
Secuencia normal	Paso 1	El profesor pulsa sobre el botón “Temario” del menú de administración de profesor.
	Paso 2	El sistema muestra un listado con el temario actual.
	Paso 3	El usuario pulsa sobre el botón de edición de un tema
	Paso 4	El sistema da la posibilidad de editar los campos del tema incluyendo el editor de texto.
	Paso 5	El usuario completa los campos requeridos y pulsa el botón “Guardar cambios”
	Paso 6	El sistema almacena los cambios.
Postcondiciones	El tema queda guardado en el sistema.	
Excepciones	➤ Error en la conexión a base de datos	
Rendimiento	3 segundos	
Frecuencia	Media	
Importancia	Vital	

Tabla 47: Caso de Uso 11 – Editar tema.

UC12		Eliminar tema	
Actores	Profesor		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee eliminar un tema.		
Precondiciones	El profesor ha iniciado sesión y el tema ya existe.		
Secuencia normal	Paso 1	El profesor pulsa sobre el botón “Temario” del menú de administración de profesor.	
	Paso 2	El sistema muestra un listado con el temario actual.	
	Paso 3	El usuario pulsa sobre el botón de borrado de un tema	
	Paso 4	El sistema sombrea el tema a borrar, permitiendo al usuario deshacer la eliminación	
	Paso 5	El usuario pulsa botón “Guardar cambios”	
	Paso 6	El sistema elimina el tema.	
Postcondiciones	El tema queda eliminado.		
Excepciones	➤ Error en la conexión a base de datos		
Rendimiento	3 segundos		
Frecuencia	Media		
Importancia	Vital		

Tabla 48: Caso de Uso 12 – Eliminar tema.

UC13	Crear alumno
Actores	Profesor
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee insertar un alumno de la asignatura en la aplicación.
Precondiciones	El profesor debe haber iniciado sesión.
Secuencia normal	<p>Paso 1 El usuario pulsa en el menú de administración de profesor sobre el botón “Alumnos”.</p> <p>Paso 2 El sistema muestra el listado de alumnos ya introducidos.</p> <p>Paso 3 El profesor pulsa sobre el botón “Añadir alumno”.</p> <p>Paso 4 El sistema muestra el formulario de inserción de alumnos.</p> <p>Paso 5 El usuario completa los campos del formulario y pulsa sobre el botón “Guardar Cambios”.</p> <p>Paso 6 El sistema valida que todos los campos son correctos y crea el alumno.</p> <p>Paso 7 El sistema devuelve un mensaje de éxito/error en la operación.</p>
Postcondiciones	El alumno queda grabado en el sistema.
Excepciones	<ul style="list-style-type: none"> ➤ Error en la conexión a base de datos ➤ Si hay cualquier error en la base de datos, el alumno no queda almacenado y se muestra un mensaje de error al profesor.
Rendimiento	3 segundos
Frecuencia	Media
Importancia	Vital

Tabla 49: Caso de Uso 13 – Crear alumno.

UC14	Editar alumno
Actores	Profesor
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee editar los datos personales de un alumno de la asignatura en la aplicación.
Precondiciones	El profesor debe haber iniciado sesión.
Secuencia normal	<p>Paso 1 El usuario pulsa en el menú de administración de profesor sobre el botón “Alumnos”.</p> <p>Paso 2 El sistema muestra el listado de alumnos ya introducidos.</p> <p>Paso 3 El profesor pulsa sobre el botón de edición en cualquiera de los alumnos disponibles</p> <p>Paso 4 El sistema muestra el formulario de edición de los datos del alumno.</p> <p>Paso 5 El usuario completa los campos del formulario y pulsa sobre el botón “Guardar Cambios”.</p> <p>Paso 6 El sistema valida que todos los campos son correctos y crea el alumno.</p> <p>Paso 7 El sistema devuelve un mensaje de éxito/error en la operación.</p>
Postcondiciones	Los cambios quedan guardados en la aplicación.
Excepciones	<ul style="list-style-type: none"> ➤ Error en la conexión a base de datos ➤ Si hay cualquier error en la base de datos, los cambios no quedan guardados y se muestra un mensaje de error al profesor.
Rendimiento	3 segundos
Frecuencia	Media
Importancia	Vital

Tabla 50: Caso de Uso 14 – Editar alumno.

UC15	Eliminar alumno	
Actores	Profesor	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee eliminar uno o varios alumnos de la asignatura.	
Precondiciones	El profesor debe haber iniciado sesión.	
Secuencia normal	Paso 1	El usuario pulsa en el menú de administración de profesor sobre el botón “Alumnos”.
	Paso 2	El sistema muestra el listado de alumnos ya introducidos.
	Paso 3	El profesor pulsa sobre el botón de borrado en cualquiera de los alumnos disponibles
	Paso 4	El sistema sombrea el alumno a borrar, permitiendo al usuario deshacer la eliminación
	Paso 5	El usuario pulsa botón “Guardar cambios”
	Paso 6	El sistema elimina el usuario.
	Paso 7	El sistema devuelve un mensaje de éxito/error en la operación.
Postcondiciones	Los cambios quedan guardados en la aplicación. Alumno eliminado de la BBDD.	
Excepciones	<ul style="list-style-type: none"> ➤ Error en la conexión a base de datos ➤ Si hay cualquier error en la base de datos, los cambios no quedan guardados y se muestra un mensaje de error al profesor. 	
Rendimiento	3 segundos	
Frecuencia	Media	
Importancia	Vital	

Tabla 51: Caso de Uso 15 – Eliminar alumno.

UC16		Contacto profesor - alumno	
Actores	Profesor, alumno registrado		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee contactar con un alumno.		
Precondiciones	El profesor debe haber iniciado sesión.		
Secuencia normal	Paso 1	El usuario pulsa en el menú de administración de profesor sobre el botón “Alumnos”.	
	Paso 2	El sistema muestra el listado de alumnos ya introducidos.	
	Paso 3	El profesor pulsa sobre el nombre del alumno con el que desea contactar.	
	Paso 4	El sistema muestra el historial de mensajes entre el alumno y el profesor	
	Paso 5	El usuario escribe el mensaje deseado y pulsa el botón “enviar mensaje”	
	Paso 6	El sistema almacena el mensaje para que el alumno lo pueda leer.	
Postcondiciones	Los mensajes quedan guardados en la aplicación.		
Excepciones	➤ Error en la conexión a base de datos		
Rendimiento	3 segundos		
Frecuencia	Media		
Importancia	Vital		

Tabla 52: Caso de Uso 16 – Contacto profesor - alumno.

UC17		Contacto alumno - profesor	
Actores	Profesor, alumno registrado		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un alumno desee contactar con un profesor.		
Precondiciones	El alumno debe haber iniciado sesión.		
Secuencia normal	Paso 1	El usuario pulsa en el menú de administración de alumno sobre el botón “Contactar con profesor”.	
	Paso 2	El sistema muestra el historial de mensajes entre el alumno y el profesor	
	Paso 3	El usuario escribe el mensaje deseado y pulsa el botón “enviar mensaje”	
	Paso 4	El sistema almacena el mensaje para que el profesor lo pueda leer.	
Postcondiciones	Los mensajes quedan guardados en la aplicación.		
Excepciones	➤ Error en la conexión a base de datos		
Rendimiento	3 segundos		
Frecuencia	Media		
Importancia	Vital		

Tabla 53: Caso de Uso 17 – Contacto alumno - profesor.

UC18	Crear ejercicio
Actores	Profesor
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee crear un nuevo tema.
Precondiciones	El profesor debe haber iniciado sesión.
Secuencia normal	Paso 1 El profesor pulsa sobre el botón “Ejercicios” del menú de administración de profesor.
	Paso 2 El sistema muestra las posibilidades de edición de ejercicios.
	Paso 3 El usuario pulsa sobre el botón “añadir ejercicio”
	Paso 4 El sistema da la posibilidad de asignar el nuevo ejercicio a un tema o crear un nuevo tema.
	Paso 5 El usuario escoge un tema para el ejercicio
	Paso 6 El sistema muestra el formulario con todos los campos necesarios para un ejercicio.
	Paso 7 El usuario completa los campos y pulsa sobre el botón “Guardar”
	Paso 8 El sistema comprueba los campos y almacena el ejercicio en la aplicación
Postcondiciones	El ejercicio queda almacenado en BBDD.
Excepciones	<ul style="list-style-type: none"> ➤ Error en la conexión a base de datos ➤ Si hay cualquier error en la base de datos, los cambios no quedan guardados y se muestra un mensaje de error al profesor.
Rendimiento	3 segundos
Frecuencia	Media
Importancia	Vital

Tabla 54: Caso de Uso 18 – Crear ejercicio.

UC19		Editar ejercicio	
Actores	Profesor		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee modificar un tema.		
Precondiciones	El profesor debe haber iniciado sesión y el tema tiene que existir.		
Secuencia normal	Paso 1	El profesor pulsa sobre el botón “Ejercicios” del menú de administración de profesor.	
	Paso 2	El sistema muestra las posibilidades de edición de ejercicios.	
	Paso 3	El usuario pulsa sobre el botón “editar ejercicio”	
	Paso 4	El sistema muestra el listado de ejercicios almacenados.	
	Paso 5	El usuario pulsa sobre el botón de edición del ejercicio seleccionado.	
	Paso 6	El sistema muestra el formulario con todos los campos editables de un ejercicio.	
	Paso 7	El usuario completa los campos y pulsa sobre el botón “Guardar cambios”	
	Paso 8	El sistema comprueba los campos y almacena el ejercicio en la aplicación	
Postcondiciones	Los cambios se guardan correctamente.		
Excepciones	<ul style="list-style-type: none"> ➤ Error en la conexión a base de datos ➤ Si hay cualquier error en la base de datos, los cambios no quedan guardados y se muestra un mensaje de error al profesor. 		
Rendimiento	3 segundos		
Frecuencia	Media		
Importancia	Vital		

Tabla 55: Caso de Uso 19 – Editar ejercicio.

UC20		Eliminar ejercicio	
Actores	Profesor		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee eliminar un tema.		
Precondiciones	El profesor debe haber iniciado sesión y el tema tiene que existir.		
Secuencia normal	Paso 1	El profesor pulsa sobre el botón “Ejercicios” del menú de administración de profesor.	
	Paso 2	El sistema muestra las posibilidades de edición de ejercicios.	
	Paso 3	El usuario pulsa sobre el botón “borrar ejercicio”	
	Paso 4	El sistema muestra el listado de ejercicios almacenados.	
	Paso 5	El usuario pulsa sobre el botón de borrado del ejercicio seleccionado.	
	Paso 6	El sistema sombrea el ejercicio a borrar, permitiendo al usuario deshacer la eliminación	
	Paso 7	El usuario pulsa sobre el botón “Guardar cambios”	
	Paso 8	El sistema elimina el ejercicio de la aplicación	
Postcondiciones	El tema queda eliminado el sistema.		
Excepciones	<ul style="list-style-type: none"> ➤ Error en la conexión a base de datos ➤ Si hay cualquier error en la base de datos, los cambios no quedan guardados y se muestra un mensaje de error al profesor. 		
Rendimiento	3 segundos		
Frecuencia	Media		
Importancia	Vital		

Tabla 56: Caso de Uso 20 – Eliminar ejercicio.

UC21	Generar test
Actores	Profesor
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee generar un test.
Precondiciones	El profesor debe haber iniciado sesión y debe existir al menos un ejercicio creado previamente.
Secuencia normal	Paso 1 El profesor pulsa sobre el botón “Generar test” del menú de administración de profesor.
	Paso 2 El sistema muestra unos campos para que el profesor pueda presentar el test. Aparece un nombre de test, una fecha y una breve descripción para que el profesor pueda dar indicaciones de cómo realizar el test.
	Paso 3 El profesor elige los ejercicios que quiere añadir al test.
	Paso 4 El profesor puede previsualizar el test para comprobar cómo quedará una vez lo genere o un alumno lo imprima.
	Paso 5 El profesor pulsa sobre el botón “Guardar”.
	Paso 6 El sistema almacena el test y estaría disponible para que un alumno lo consultara.
Postcondiciones	El test se ha generado correctamente.
Excepciones	<ul style="list-style-type: none"> ➤ Error en la conexión a base de datos ➤ Si hay cualquier error en la base de datos, los cambios no quedan guardados y se muestra un mensaje de error al profesor.
Rendimiento	3 segundos
Frecuencia	Baja
Importancia	Importante

Tabla 57: Caso de Uso 21 – Generar test.

UC22	Añadir notas
Actores	Profesor
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee eliminar notas.
Precondiciones	El profesor debe haber iniciado sesión y el alumno al que se le va añadir una nota debe existir en la base de datos.
Secuencia normal	Paso 1 El profesor pulsa sobre el botón “Calificaciones” del menú de administración de profesor.
	Paso 2 El sistema muestra al profesor el listado de alumnos a los que puede añadir una nota.
	Paso 3 El profesor pulsa sobre el botón “Añadir nota” y añade la nota que corresponda.
	Paso 4 El profesor pulsa sobre el botón “Guardar”
	Paso 5 El sistema guarda la nota correspondiente en la aplicación
Postcondiciones	La nota queda almacenada correctamente.
Excepciones	<ul style="list-style-type: none"> ➤ Error en la conexión a base de datos ➤ Si hay cualquier error en la base de datos, los cambios no quedan guardados y se muestra un mensaje de error al profesor.
Rendimiento	3 segundos
Frecuencia	Media
Importancia	Vital

Tabla 58: Caso de Uso 22 – Añadir notas.

UC23		Descargar archivos de alumno	
Actores	Profesor		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee descargarse algún archivo de alumno.		
Precondiciones	El profesor debe haber iniciado sesión.		
Secuencia normal	Paso 1	El profesor pulsa sobre el botón “Calificaciones” del menú de administración de profesor.	
	Paso 2	El sistema muestra al profesor un listado de los alumnos.	
	Paso 3	El profesor pulsar sobre “Consultar archivo” de un determinado alumno.	
	Paso 4	El sistema muestra un listado de aquellos archivos que el alumno haya subido a la aplicación web.	
	Paso 5	El profesor puede descargarse el archivo o previsualizarlo.	
Postcondiciones			
Excepciones	➤ Error en la conexión a base de datos		
Rendimiento	3 segundos		
Frecuencia	Media		
Importancia	Alta		

Tabla 59: Caso de Uso 23 – Descargar archivos de alumno.

UC24	Subir archivo de profesor	
Actores	Profesor	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor desee subir un archivo.	
Precondiciones	El profesor debe haber iniciado sesión.	
Secuencia normal	Paso 1	El profesor pulsa sobre el botón “Envía ficheros” del menú de administración de profesor.
	Paso 2	El sistema muestra un pequeño menú donde le da la posibilidad al profesor de elegir un archivo local y posteriormente subirlo a la aplicación.
	Paso 3	El profesor pulsar el botón “Subir”.
	Paso 4	El sistema guarda el fichero.
Postcondiciones	El fichero queda almacenado.	
Excepciones	<ul style="list-style-type: none"> ➤ Error en la conexión a base de datos ➤ Si hay cualquier error en la base de datos, los cambios no quedan guardados y se muestra un mensaje de error al profesor. 	
Rendimiento	3 segundos	
Frecuencia	Media	
Importancia	Alta	

Tabla 60: Caso de Uso 24 – Subir archivo de profesor.

UC25		Previsualizar ficheros	
Actores	Alumno registrado		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un alumno registrado desee previsualizar un fichero.		
Precondiciones	El alumno registrado debe haber iniciado sesión.		
Secuencia normal	Paso 1	El alumno pulsa el botón “Descarga ficheros” del menú de alumno.	
	Paso 2	El sistema muestra al alumno un listado de los posibles archivos que puede visualizar	
	Paso 3	El alumno pulsa el botón “Previsualizar”.	
	Paso 4	El sistema muestra en la misma pantalla el fichero.	
Postcondiciones	El fichero queda previsualizado.		
Excepciones			
Rendimiento	3 segundos		
Frecuencia	Media		
Importancia	Alta		

Tabla 61: Caso de Uso 25 – Previsualizar ficheros.

UC26		Descargar ficheros subidos por profesor	
Actores	Alumno registrado		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un alumno registrado desea descargar un fichero subido por el profesor.		
Precondiciones	El alumno registrado debe haber iniciado sesión.		
Secuencia normal	Paso 1	El alumno pulsa el botón “Descarga ficheros” del menú de alumno.	
	Paso 2	El sistema muestra al alumno un listado de los posibles archivos que puede descargarse.	
	Paso 3	El alumno pulsa el botón “Descargar”.	
	Paso 4	El sistema descarga el fichero.	
Postcondiciones	El fichero queda descargado y a disponibilidad del alumno.		
Excepciones			
Rendimiento	3 segundos		
Frecuencia	Alta		
Importancia	Alta		

Tabla 62: Caso de Uso 26 – Descargar ficheros subidos por profesor.

UC27		Subir fichero de alumno	
Actores	Alumno registrado		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un alumno desee subir un fichero.		
Precondiciones	El alumno debe haber iniciado sesión.		
Secuencia normal	Paso 1	El alumno pulsa sobre el botón “Envía ficheros” del menú de alumno registrado.	
	Paso 2	El sistema muestra un pequeño menú donde le da la posibilidad al alumno de elegir un archivo local y posteriormente subirlo a la aplicación.	
	Paso 3	El alumno pulsa el botón “Subir”.	
	Paso 4	El sistema guarda el fichero.	
Postcondiciones	El fichero queda almacenado.		
Excepciones	<ul style="list-style-type: none"> ➤ Error en la conexión a base de datos ➤ Si hay cualquier error en la base de datos, los cambios no quedan guardados y se muestra un mensaje de error al profesor. 		
Rendimiento	3 segundos		
Frecuencia	Media		
Importancia	Alta		

Tabla 63: Caso de Uso 27 – Subir fichero de alumno.

UC28		Eliminar fichero	
Actores	Profesor, alumno registrado		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un profesor o un alumno registrado desee eliminar un fichero.		
Precondiciones	El profesor o el alumno registrado deben haber iniciado sesión y el fichero debe estar subido previamente.		
Secuencia normal	Paso 1	El profesor o el alumno registrado pulsa sobre el botón “Envía ficheros” del menú de administración de profesor o del menú de alumno.	
	Paso 2	El sistema muestra un pequeño menú donde el usuario podrá elegir la opción de eliminar fichero.	
	Paso 3	El usuario pulsa sobre el botón “Eliminar fichero”	
	Paso 4	El sistema elimina el fichero de la aplicación	
Postcondiciones	El fichero queda eliminado el sistema.		
Excepciones	➤ Error en la conexión a base de datos		
Rendimiento	3 segundos		
Frecuencia	Baja		
Importancia	Media		

Tabla 64: Caso de Uso 28 – Eliminar fichero.

UC29		Consultar notas	
Actores	Alumno registrado		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un alumno registrado desee consultar las notas que ha obtenido.		
Precondiciones	El alumno debe haber iniciado sesión.		
Secuencia normal	Paso 1	El alumno registrado pulsa sobre el botón “Consulta tus notas” del menú de alumno.	
	Paso 2	El sistema muestra la nota obtenida de la última evaluación en color verde, y en color azul, muestra el resto de notas que ha ido sacado a lo largo del curso.	
Postcondiciones			
Excepciones	➤ Error en la conexión a base de datos		
Rendimiento	3 segundos		
Frecuencia	Alta		
Importancia	Vital		

Tabla 65: Caso de Uso 29 – Consultar notas.

UC30		Consultar test	
Actores	Alumno registrado		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un alumno registrado desee consultar el test actual.		
Precondiciones	El alumno registrado debe haber iniciado sesión.		
Secuencia normal	Paso 1	El alumno pulsa sobre el botón “Test” del menú de alumno registrado.	
	Paso 2	El sistema muestra, si existe, el test en pantalla dando la posibilidad de poderlo imprimir.	
Postcondiciones			
Excepciones	➤ Error en la conexión a base de datos		
Rendimiento	3 segundos		
Frecuencia	Media		
Importancia	Alta		

Tabla 66: Caso de Uso 30 – Consultar test.

4.3.5 Requisitos no funcionales

Los requisitos no funcionales describen propiedades o cualidades que el sistema debe tener.

Identificador	RQNF-01
Definición	Disponibilidad
Descripción	La aplicación debe estar disponible las 24 horas del día, los 7 días de la semana

Tabla 67: Requisito no funcional 1 – Disponibilidad.

Identificador	RQNF-02
Definición	Mantenibilidad
Descripción	La aplicación debe ser capaz de recuperarse fácilmente de cualquier error que pudiera suceder.

Tabla 68: Requisito no funcional 2 – Mantenibilidad.

Identificador	RQNF-03
Definición	Interoperabilidad
Descripción	La aplicación debe ser compatible con la mayoría de los navegadores web: Internet Explorer, Mozilla Firefox, Google Chrome.

Tabla 69: Requisito no funcional 3 – Interoperabilidad.

Identificador	RQNF-04
Definición	Interfaz
Descripción	La aplicación debe estar diseñada de tal forma que para un usuario con básicos conocimientos de informática, sea fácil de comprender y que le resulte atractiva e intuitiva, ya que va dirigida a niños del Tercer Ciclo de Primaria.

Tabla 70: Requisito no funcional 4 – Interfaz.

Identificador	RQNF-05
Definición	Usabilidad
Descripción	El usuario deberá poder utilizar la aplicación sin problemas después de haber leído el manual de usuario. El sistema debe restringir que el usuario solo realice las operaciones permitidas para su rol.

Tabla 71: Requisito no funcional 5 – Usabilidad.

Identificador	RQNF-06
Definición	Escalabilidad
Descripción	El sistema debe ser escalable, es decir, se debe poder añadir nuevas funcionalidades con facilidad y el funcionamiento actual no debe verse alterado.

Tabla 72: Requisito no funcional 6 – Escalabilidad.

Identificador	RQNF-07
Definición	Seguridad
Descripción	Los usuarios deberán identificarse en el sistema a través de su usuario y su contraseña.

Tabla 73: Requisito no funcional 7 – Seguridad.

Identificador	RQNF-08
Definición	Accesibilidad
Descripción	A la aplicación debe poder accederse a través de un navegador.

Tabla 74: Requisito no funcional 8 – Accesibilidad.

4.3.6 Resumen

TIPO	ID	DESCRIPCIÓN
OBJETIVOS	OBJ-01	Gestión de usuarios
	OBJ-02	Gestión de teoría
	OBJ-03	Gestión de ejercicios
	OBJ-04	Gestión de test
	OBJ-05	Gestión de ficheros
	OBJ-06	Gestión de calificaciones
	OBJ-07	Gestión de alumnos
	OBJ-08	Contacto entre alumnos y profesores
ACTORES	ACT-01	Alumno no registrado
	ACT-02	Alumno registrado
	ACT-03	Profesor
REQUISITOS DE INFORMACIÓN	RI-01	Información de usuarios
	RI-02	Información de teoría
	RI-03	Información de ejercicios
	RI-04	Información de test
	RI-05	Información de ficheros
REQUISITOS FUNCIONALES	UC-01	Registro
	UC-02	Validar registro alumno
	UC-03	Login de usuario
	UC-04	Logout de usuario
	UC-05	Consultar tema
	UC-06	Consultar temario
	UC-07	Imprimir
	UC-08	Consultar ejercicio
	UC-09	Resolver ejercicio
	UC-10	Crear tema
	UC-11	Editar tema
	UC-12	Eliminar tema
	UC-13	Crear alumno
	UC-14	Editar alumno
	UC-15	Eliminar alumno
	UC-16	Contacto profesor – alumno
	UC-17	Contacto alumno – profesor
	UC-18	Crear ejercicio
	UC-19	Editar ejercicio
	UC-20	Eliminar ejercicio
	UC-21	Generar test
	UC-22	Añadir notas
	UC-23	Descargar archivos de alumno
	UC-24	Subir archivo de profesor
	UC-25	Previsualizar ficheros
	UC-26	Descargar ficheros subidos por profesor
	UC-27	Subir fichero de alumno
	UC-28	Eliminar fichero

	UC-29	Consultar notas
	UC-30	Consultar notas
REQUISITOS NO FUNCIONALES	RQNF-01	Disponibilidad
	RQNF-02	Mantenibilidad
	RQNF-03	Interoperabilidad
	RQNF-04	Interfaz
	RQNF-05	Usabilidad
	RQNF-06	Escalabilidad
	RQNF-07	Seguridad
	RQNF-08	Accesibilidad

5. DISEÑO

En este punto, se va a especificar cómo se va a organizar, desarrollar y comportar el sistema.

5.1 Arquitectura lógica

La arquitectura lógica describe cuáles son los componentes lógicos del sistema, cómo interactúan entre ellos y por otra parte, guía el diseño del sistema.

La aplicación web “Aprendiendo los números enteros” se va a desarrollar bajo una arquitectura de dos capas. Estas dos capas serán por un lado, el cliente y por otro, el servidor.

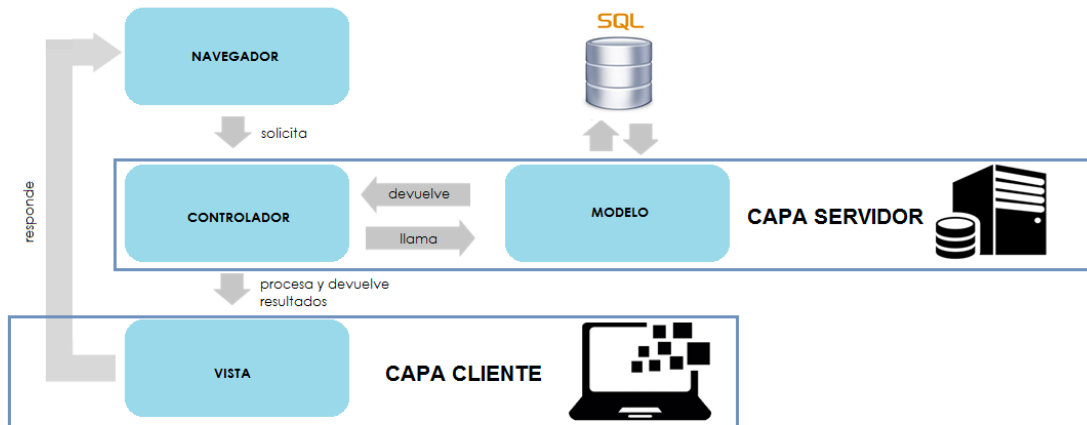
En el cliente, nos vamos a encontrar la lógica de presentación que será la encargada de generar la interfaz de usuario en función de las acciones llevadas a cabo y, la lógica de aplicación o de negocio que será la encargada de modelar los procesos de negocio y es donde se realiza todo el procesamiento necesario para atender las peticiones de usuario.

En el servidor, nos encontramos con la lógica de administración de datos, ésta se encarga de hacer persistente toda la información.

Teniendo presente la arquitectura de dos capas, el diseño de la aplicación se basa en el patrón MVC (Modelo – Vista – Controlador), es decir, en el cliente tendremos la vista y en el servidor el modelo y el controlador.

Se hace una breve explicación sobre dicho patrón:

Es un modelo de abstracción de desarrollo de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el sistema de gestión de base de datos y la lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

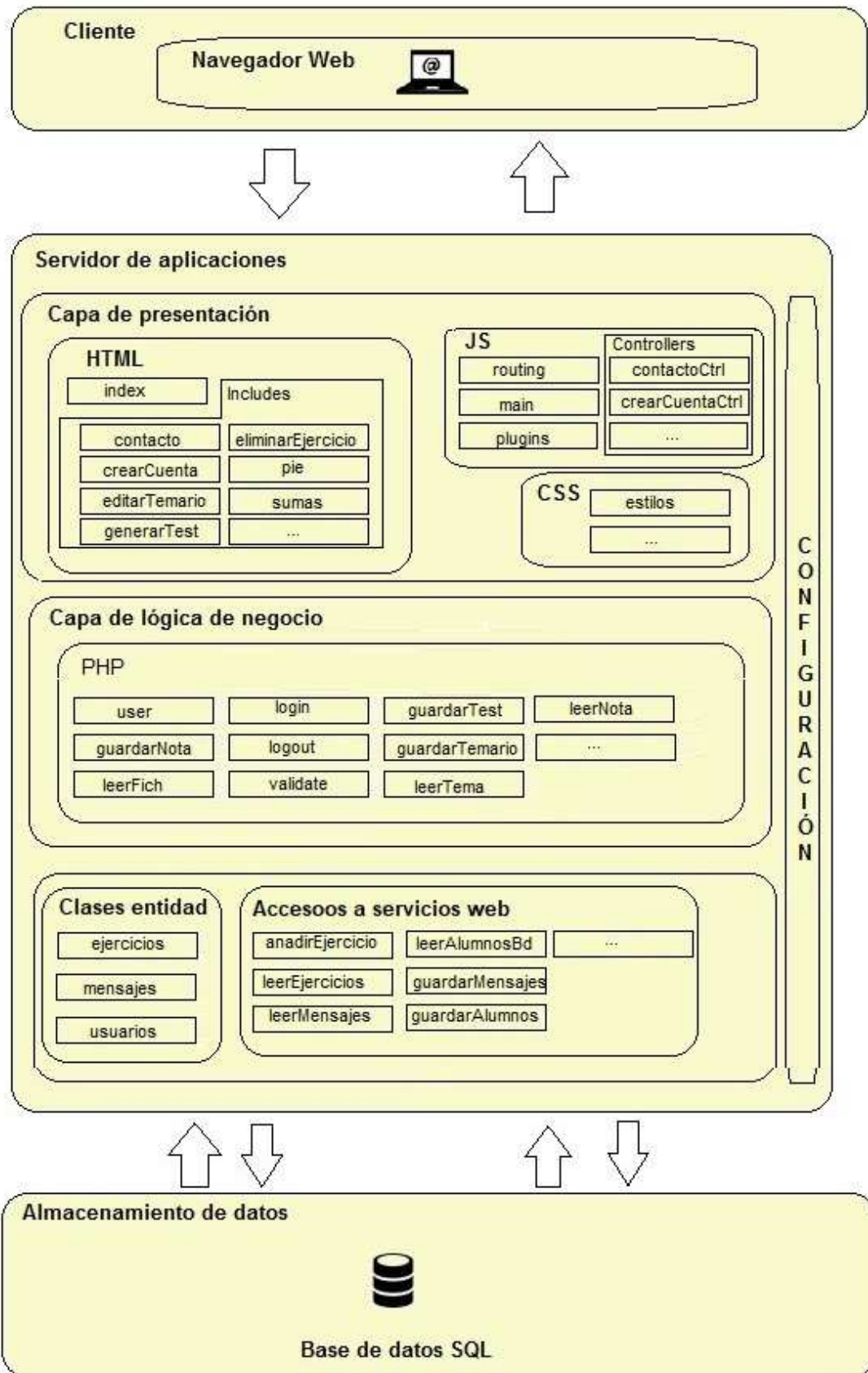


A continuación se describe cada componente lógico de la aplicación web desarrollada:

- Lógica de presentación. Las principales tareas son:
 - Obtener información del usuario.
 - Enviar información del usuario a la lógica de aplicación o de negocio para su procesamiento.
 - Recibir los resultados del procesamiento de la lógica de aplicación.
 - Presentar los resultados al usuario.

- Lógica de aplicación o de negocio.
 - Recibir la entrada del nivel de presentación.
 - Interactuar con la lógica de administración de datos.
 - Enviar el resultado del procesamiento al nivel de presentación.

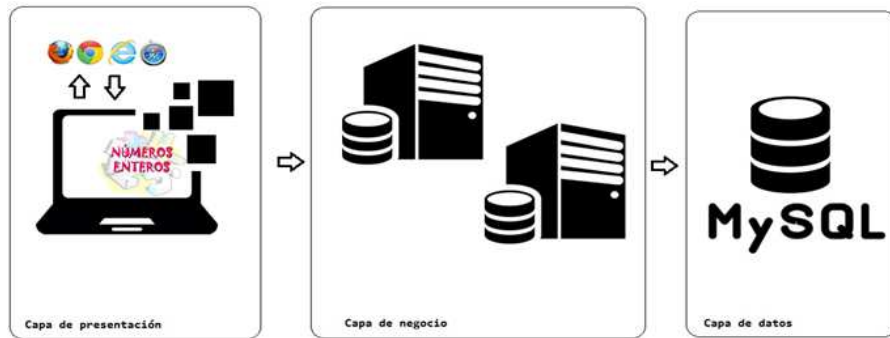
- Lógica de administración de datos.
 - Almacena, recupera, mantiene y asegura la integridad de los datos.



5.2 Arquitectura física

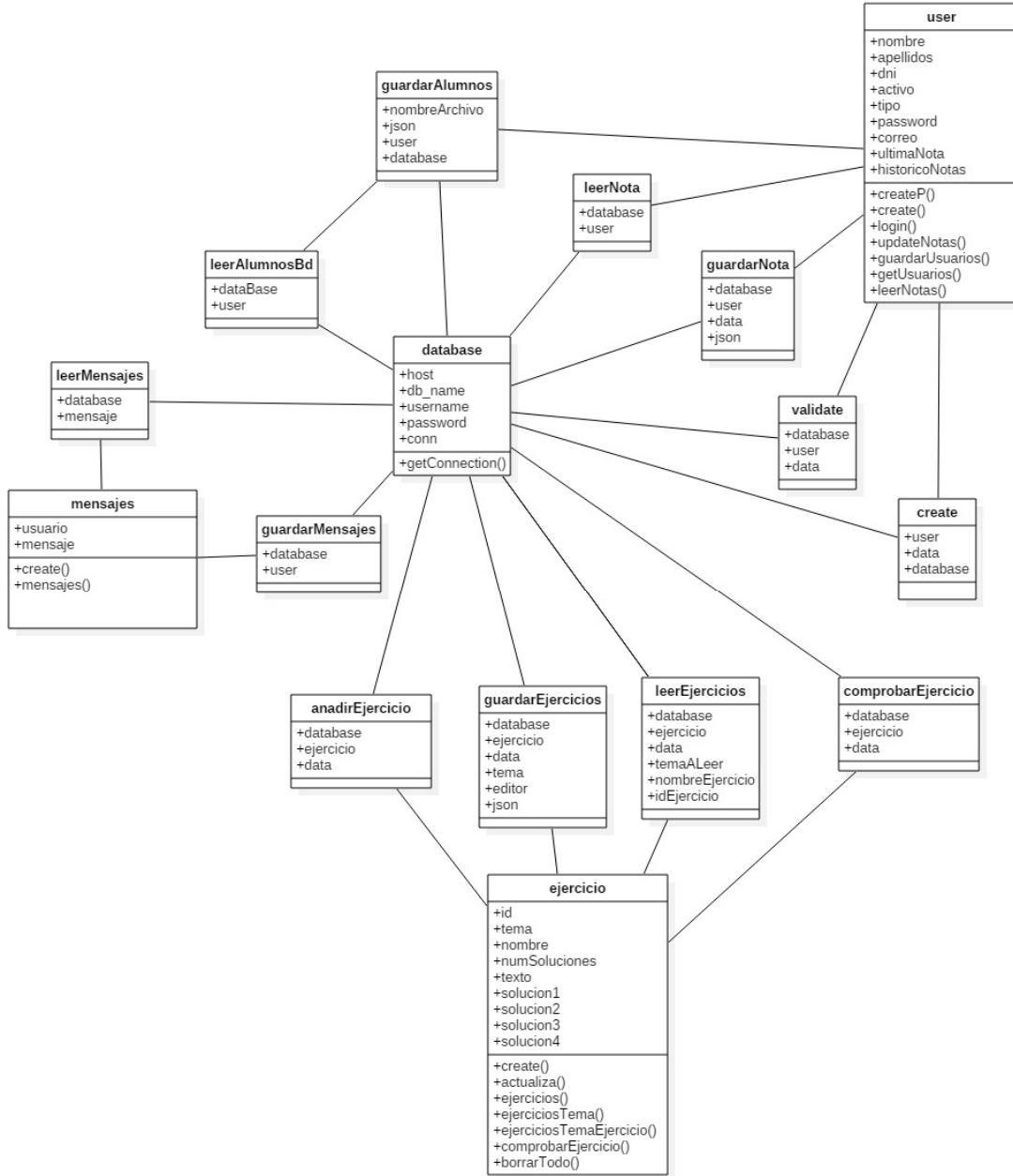
La arquitectura física describe los componentes físicos y su relación entre ellos.

Los usuarios de la aplicación acceden a la herramienta a través de un navegador web, esto va a permitir que nuestra aplicación sea independiente de la plataforma siempre y cuando se utilice un navegador.

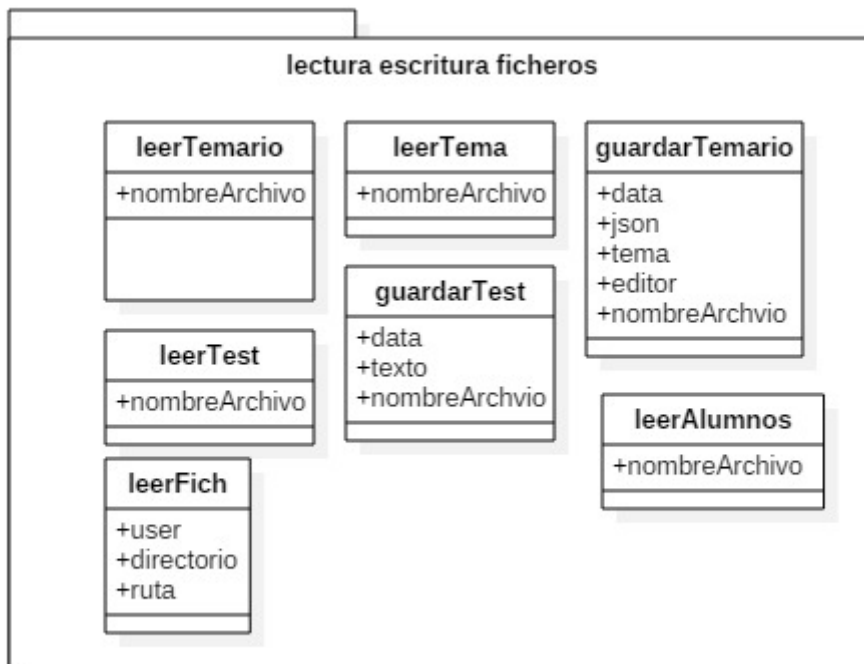


5.3 Diagrama de clases

En los diagramas de clases se describen todas las clases implementadas para el desarrollo de la aplicación.



Por otra parte, tenemos las clases que se utilizan para la lectura y escritura de ficheros:



Y por último, las clases que tenemos para las sesiones:

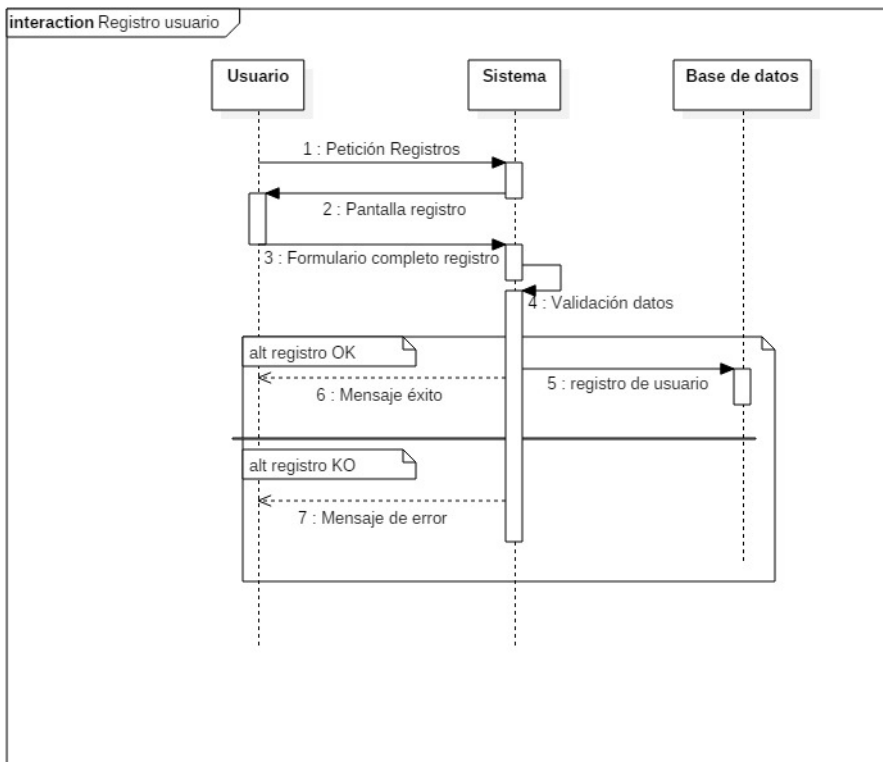


5.4 Diagrama de secuencia

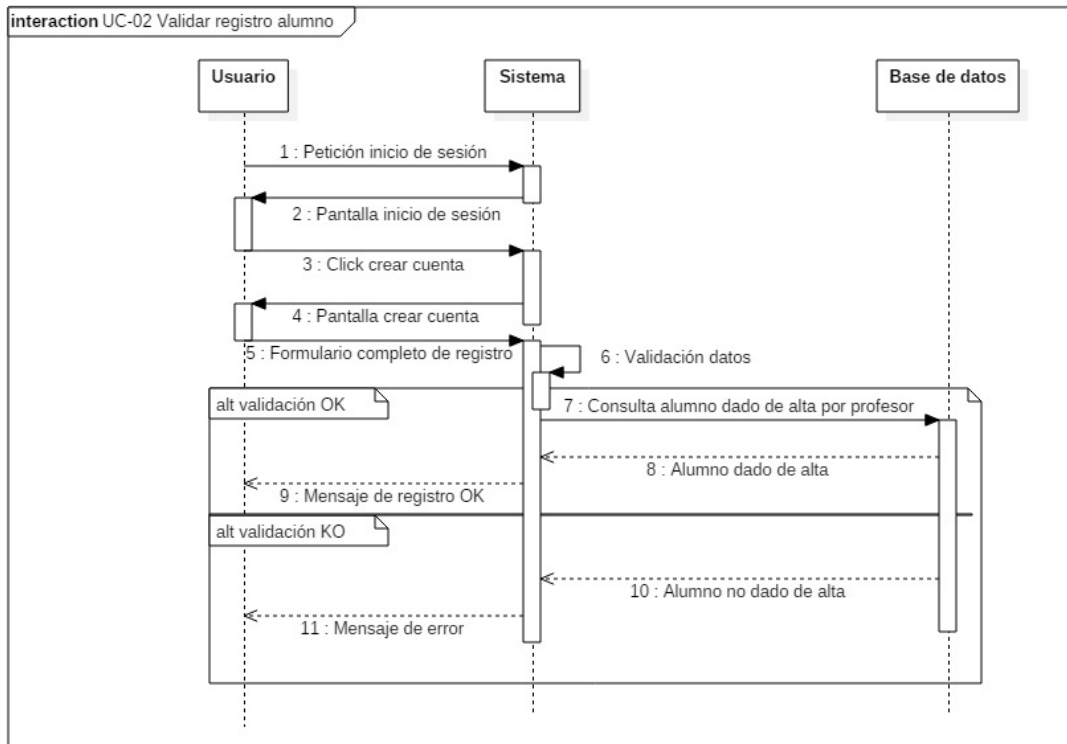
Los diagramas de secuencia van a mostrar la interacción de un conjunto de objetos de nuestra aplicación. Se modela un diagrama de secuencia por cada caso de uso.

De los 30 casos de uso que se analizaron en puntos anteriores, se ha decidido realizar los más importantes y no repetir algunos, como por ejemplo de tipo CRUD (creación, leer, modificación y eliminación).

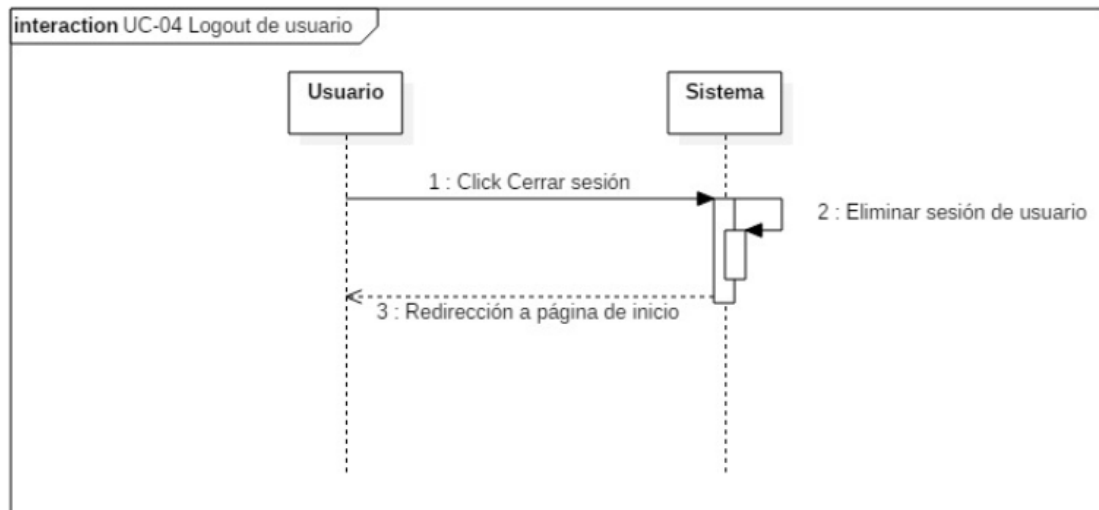
UC-01 Registro



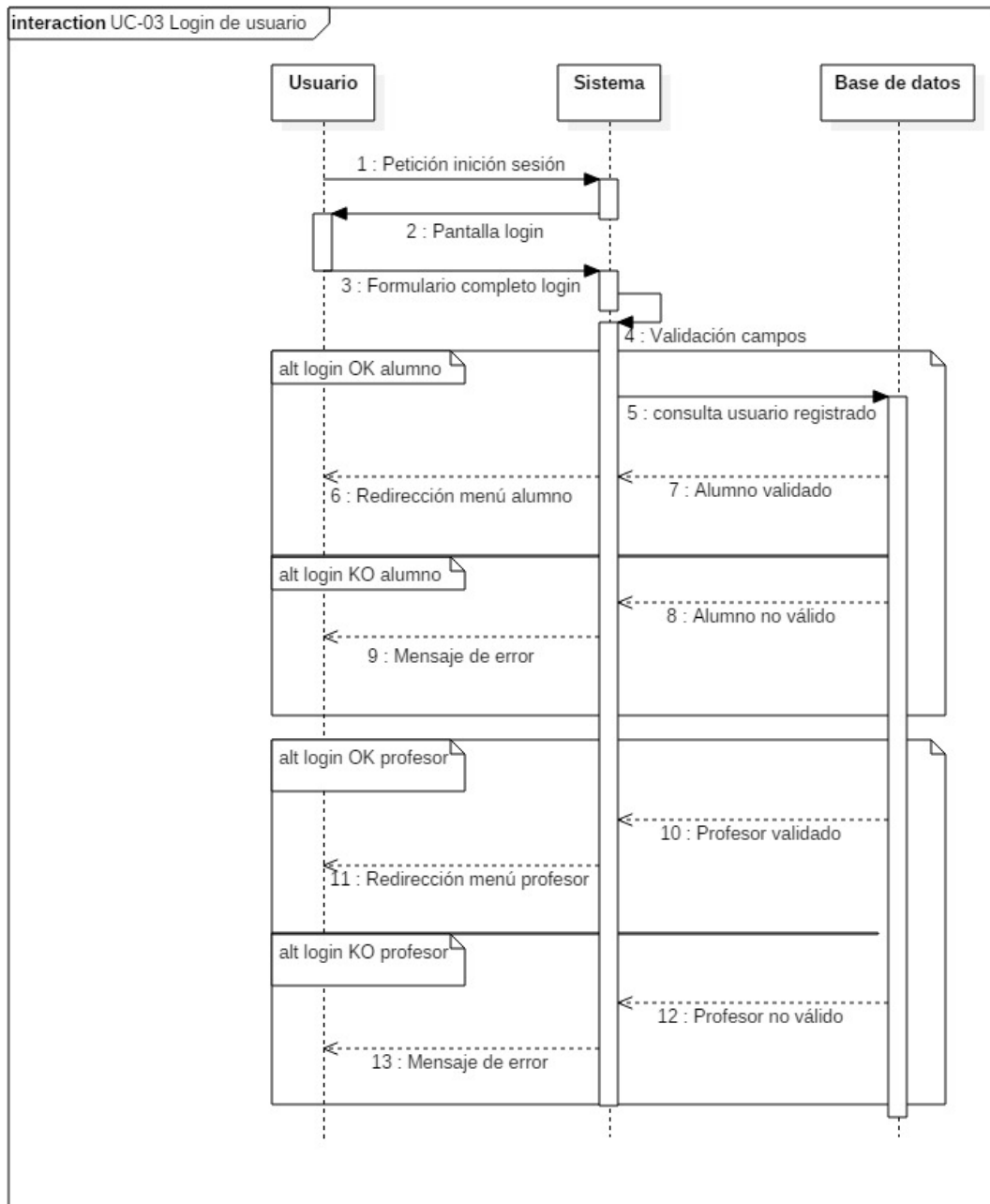
UC-02 Validar registro alumno

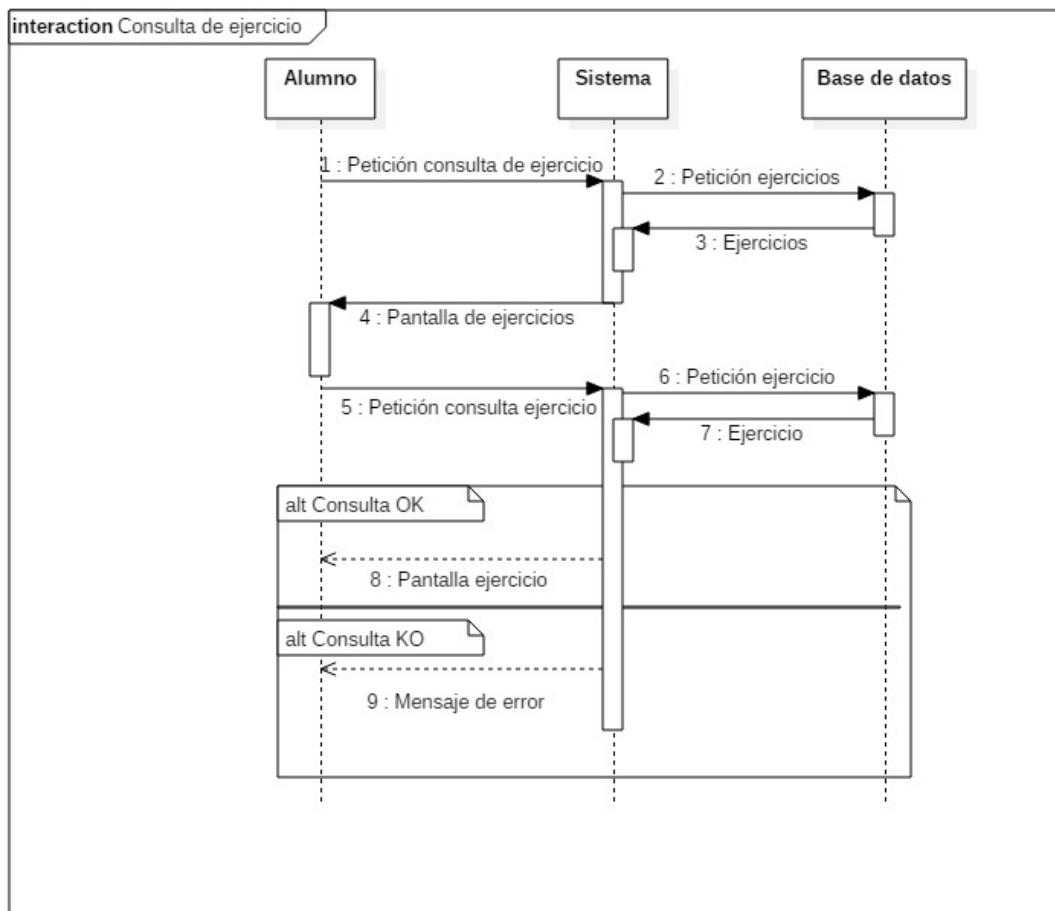


UC-04 Logout de usuario

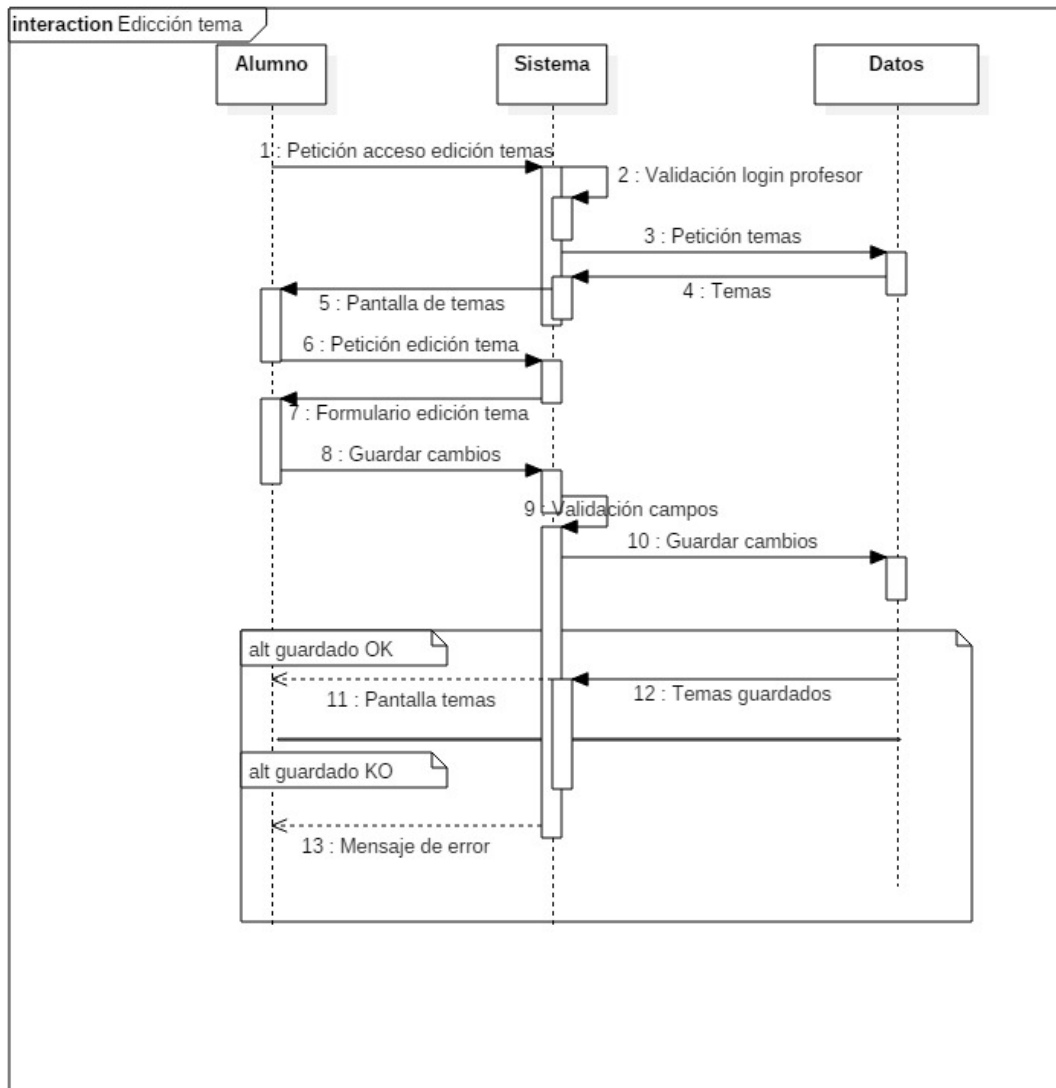


UC-03 Login de usuario

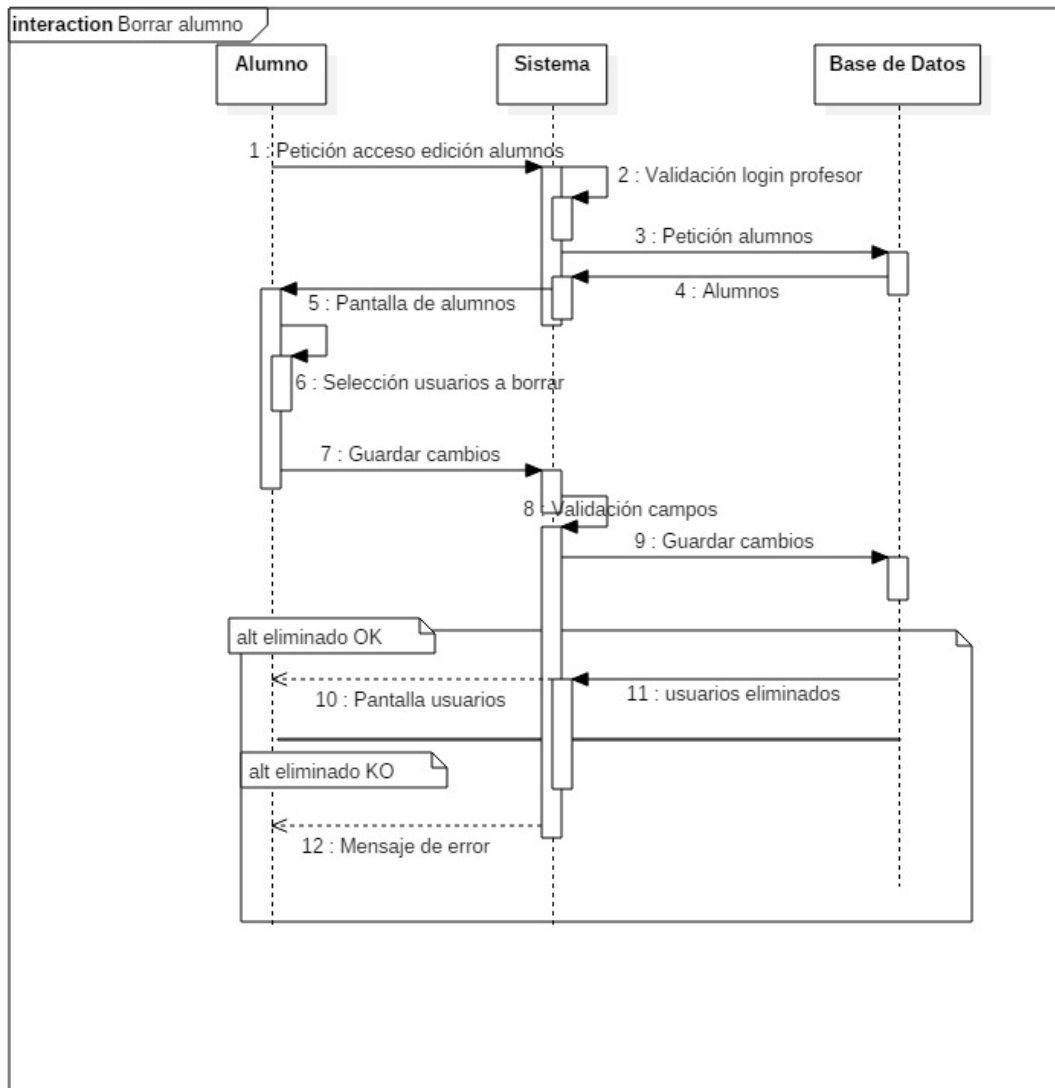


UC-08 Consultar ejercicio

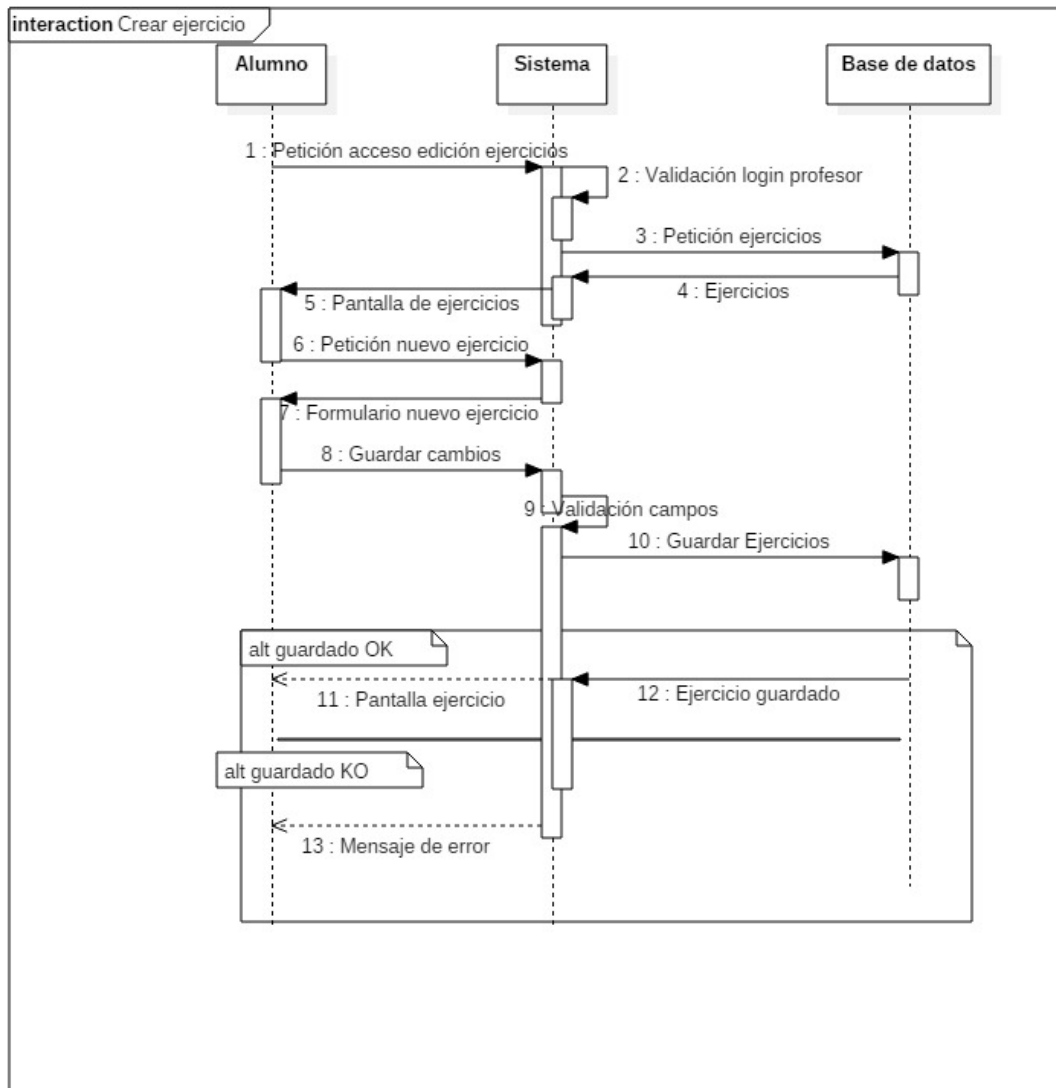
UC-11 Editar tema



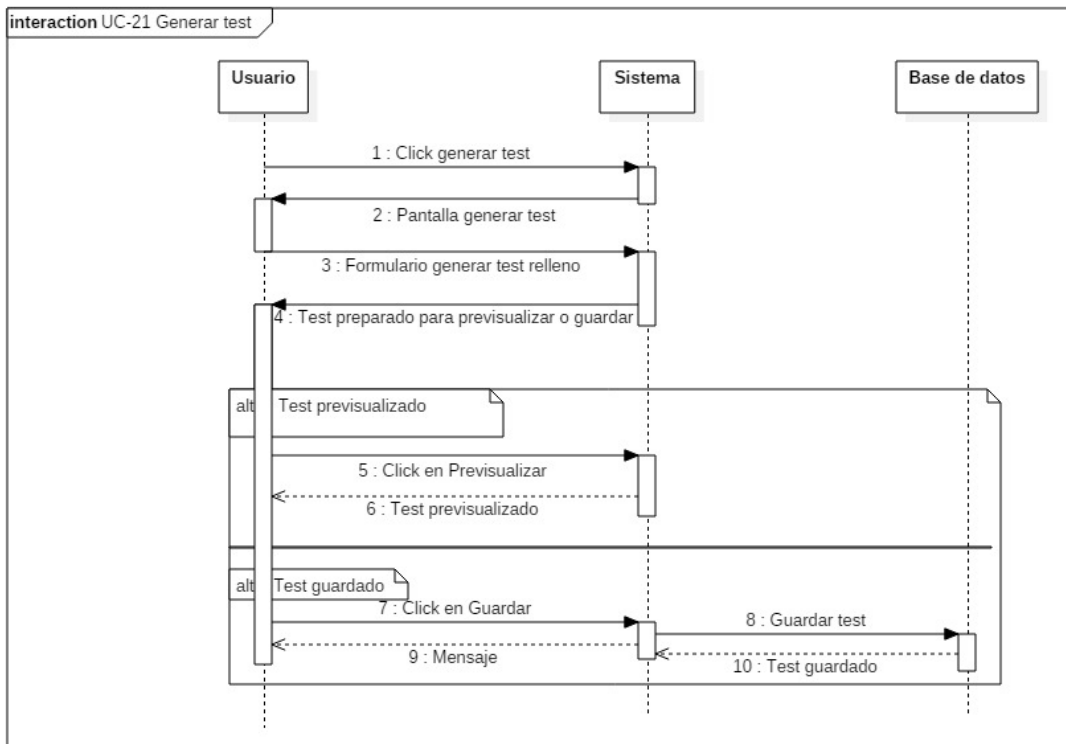
UC-15 Eliminar alumno



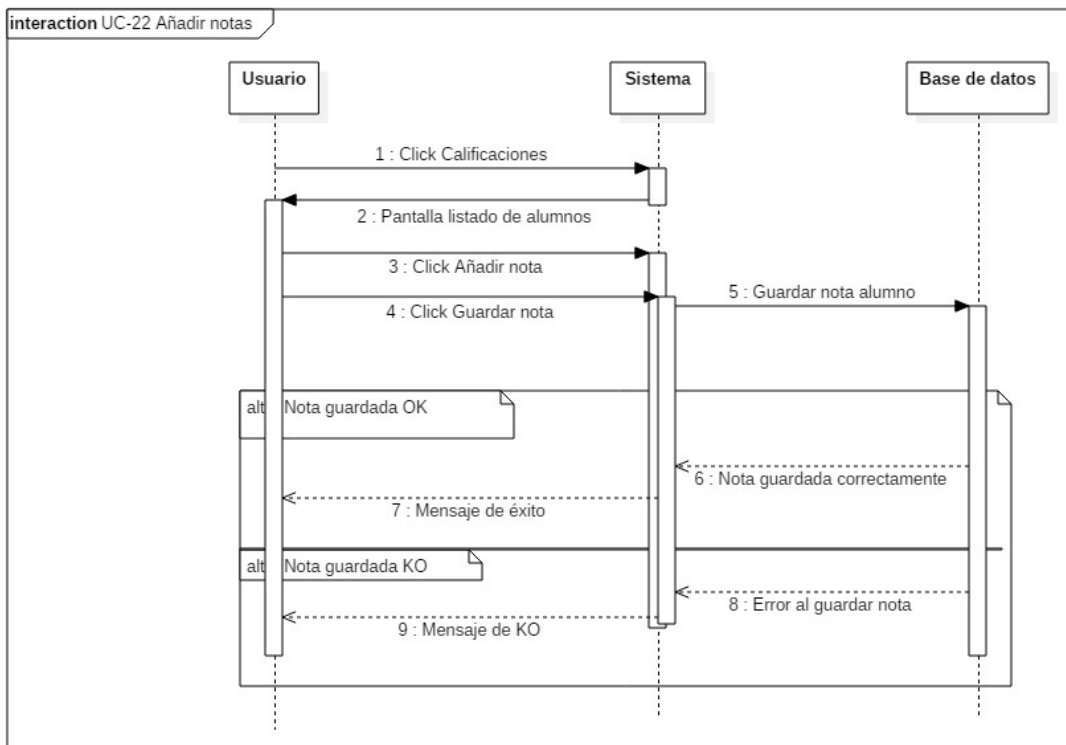
UC-18 Crear ejercicio

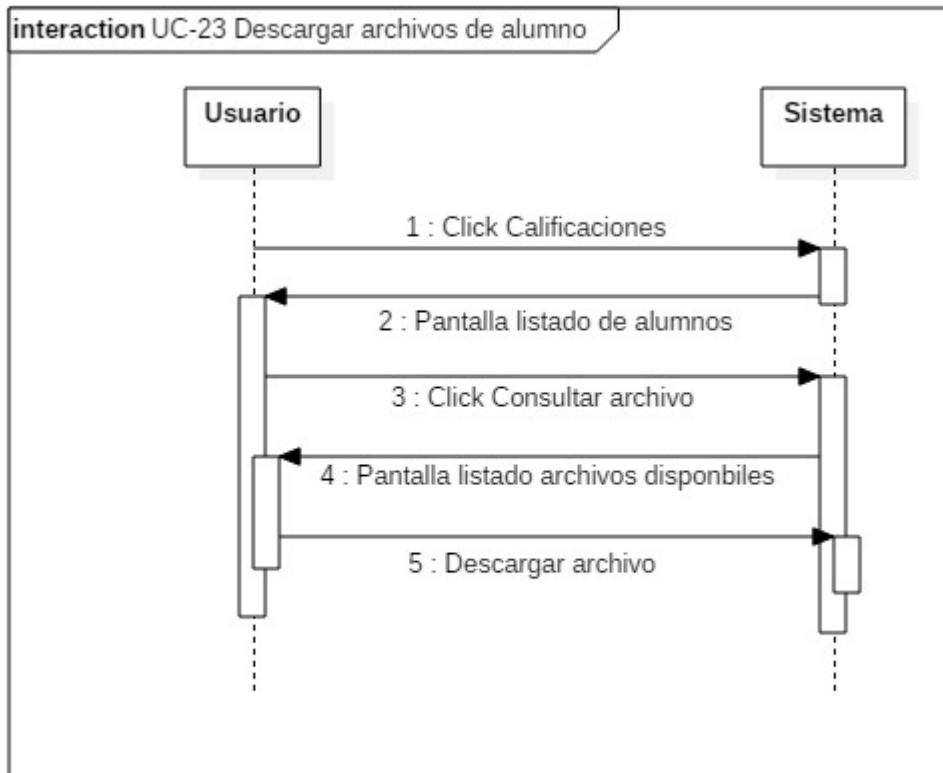
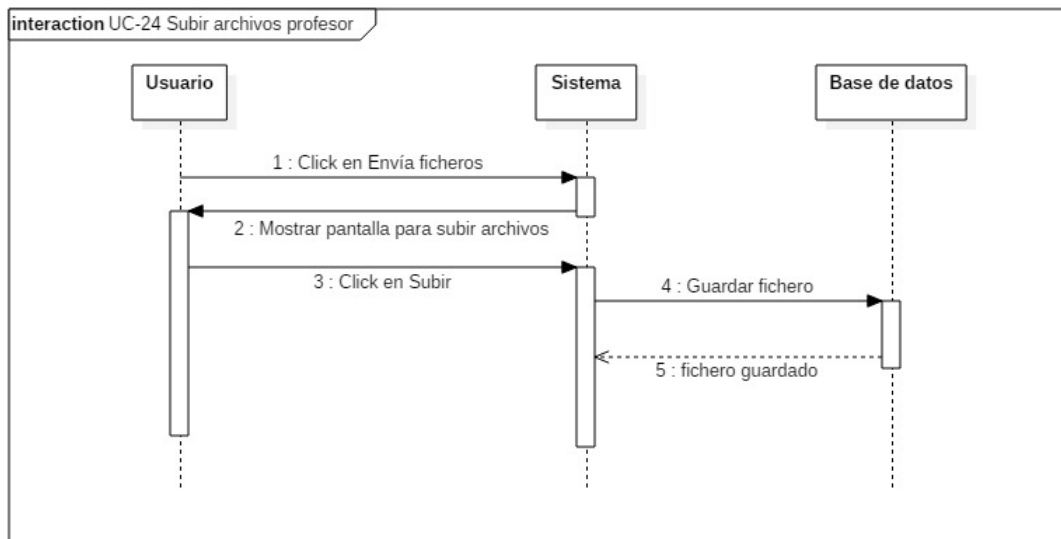


UC-21 Generar test

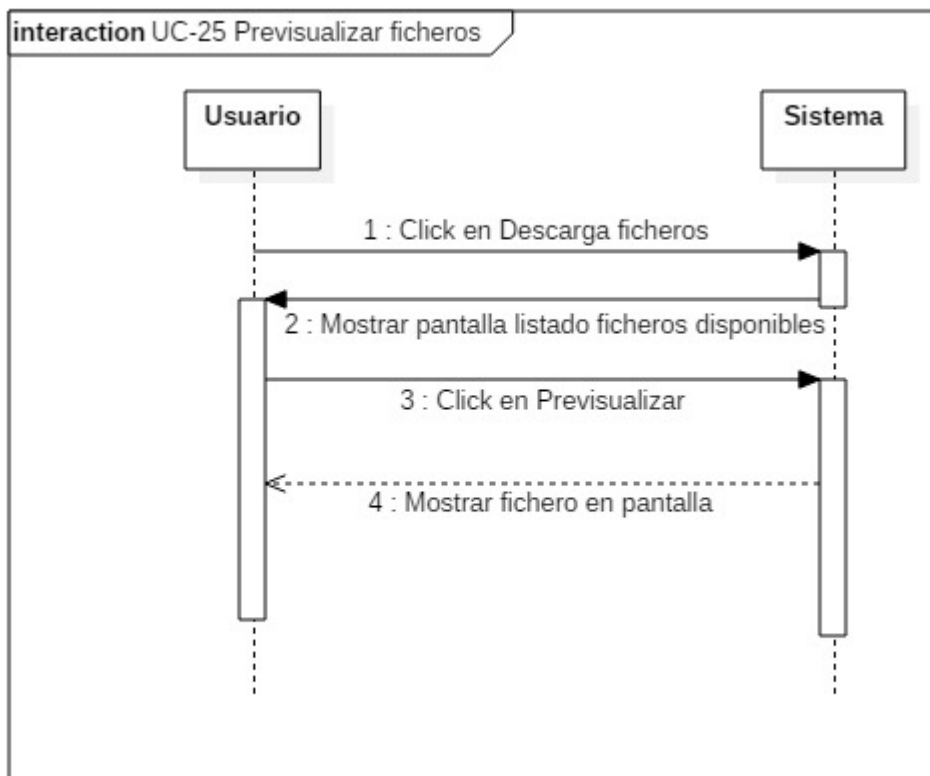


UC-22 Añadir notas

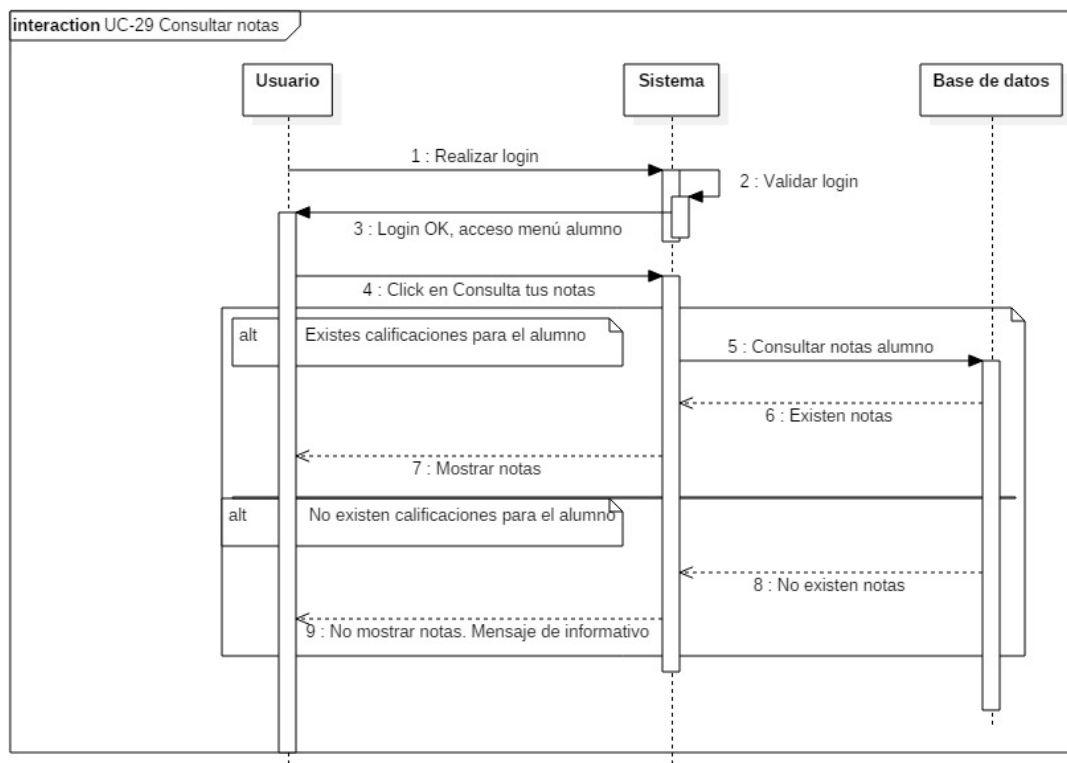


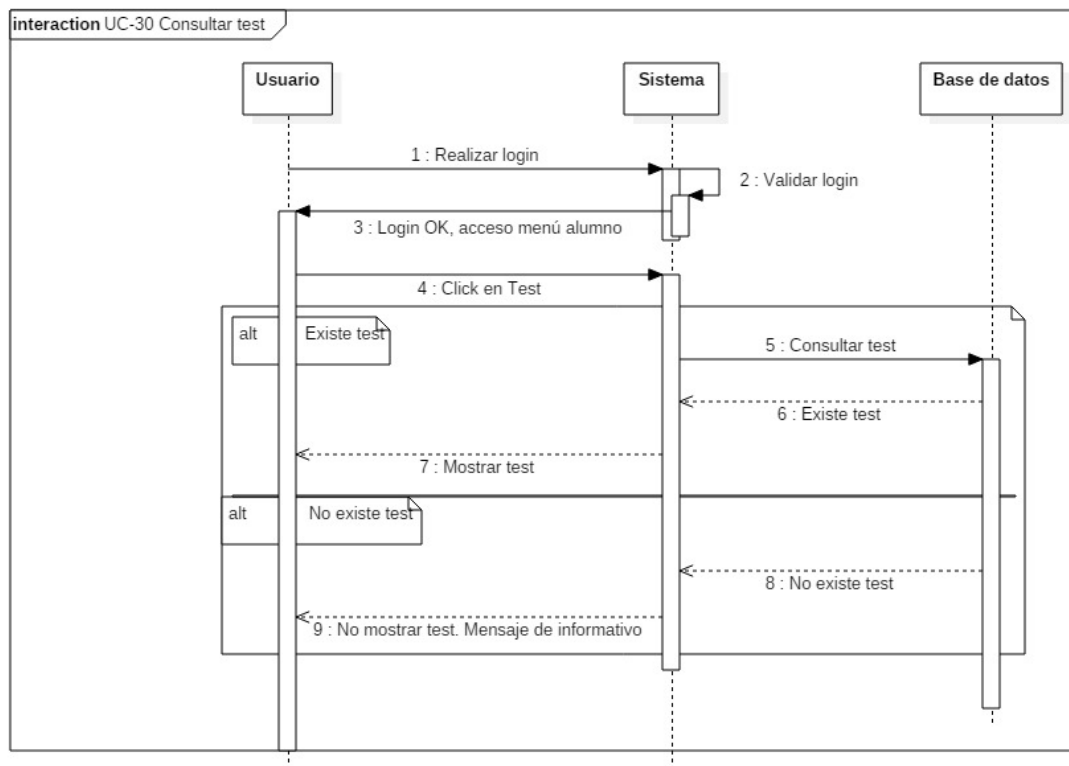
UC-23 Descargar archivos de alumno*UC-24 Subir archivo de profesor*

UC-25 Previsualizar ficheros



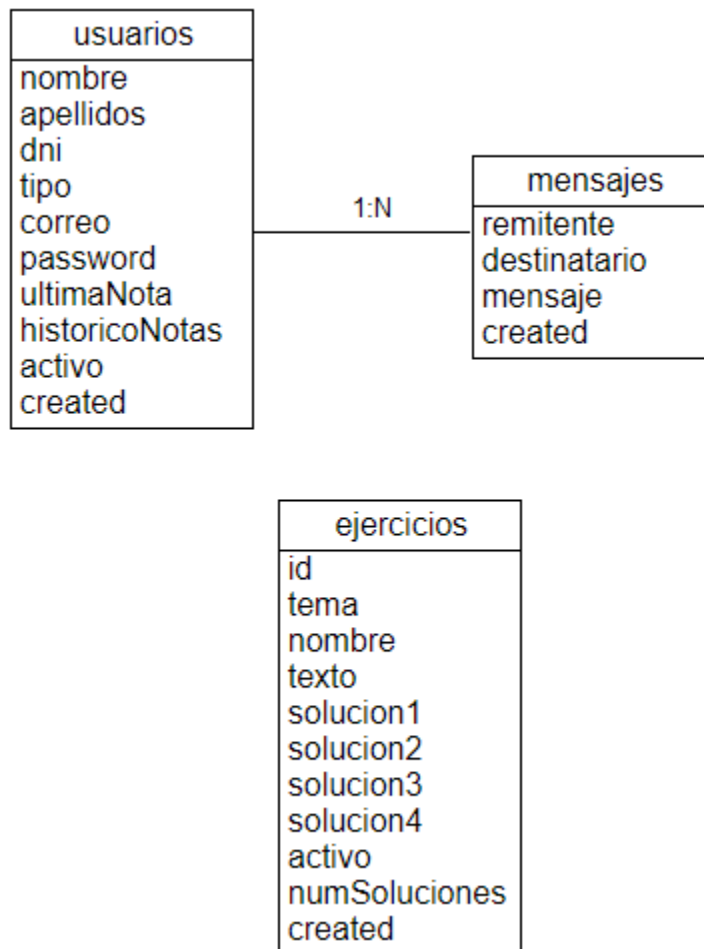
UC-29 Consultar notas



UC-30 Consultar test

5.5 Modelo lógico de datos

La aplicación presenta un modelo simple de tablas:



5.5.1 Diccionario de datos

En este apartado se representan las características lógicas de los datos que se van a utilizar, se incluye el nombre de la entidad, atributos, tipo de cada atributo, su longitud así como una breve descripción.

Entidad	Atributo	Tipo	Longitud	Descripción	Información
usuarios	nombre	varchar	512	Nombre de usuario	Not Null
	apellidos	varchar	255	Apellidos del usuario	Not Null
	dni	varchar	9	Dni del usuario	
	tipo	varchar	20	Tipo de usuario (alumno/profesor)	Not Null
	correo	varchar	512	Correo electrónico del usuario	Clave primaria
	password	varchar	512	Contraseña del usuario	Cifrado en MD5
	ultimaNota	text		Última nota del usuario	
	historicoNotas	text		Histórico del usuario	
	activo	text		Activo =1 Inactivo = 0	
created	timestamp		Fecha de alta en el sistema	Not Null	

Tabla 75: Diccionario de datos: entidad usuarios.

Entidad	Atributo	Tipo	Longitud	Descripción	Información
mensajes	remite	varchar	255	Remite del mensaje	Not Null
	destinatario	text		Destinatario del mensaje	Not Null
	mensaje	varchar	512	Mensaje	Not Null
	created	timestamp		Fecha de alta en el sistema	Not Null

Tabla 76: Diccionario de datos: entidad mensajes.

Entidad	Atributo	Tipo	Longitud	Descripción	Información
ejercicios	id	int	11	Id del ejercicio	Autoincrementable Not Null
	tema	text		Tema del ejercicio	Not Null
	nombre	text		Nombre del ejercicio	
	texto	text		Enunciado del ejercicio	
	solucion1	text		Solución del ejercicio	Not Null
	solucion2	text		Solución del ejercicio	
	solucion3	text		Solución del ejercicio	
	solucion4	text		Solución del ejercicio	
	activo	text		Estado del ejercicio	Not Null
	numSoluciones	text		Número de soluciones que tiene el ejercicio	Not Null
	created	text		Fecha de alta en el sistema	Not Null

Tabla 77: Diccionario de datos: entidad ejercicios.

5.6 Diseño de interfaz

En este apartado se realiza una aproximación de la interfaz de usuario que va a tener la aplicación.

No se especifican todas las interfaces de aplicación, ya que muchas son similares.

A continuación, se detallan aquellas interfaces de usuario más relevantes y representativas. Se ha querido que esta aplicación tenga un interfaz sencillo, fácil y sobre todo atractivo ya que la aplicación va destinada a niños.

Todas las diferentes interfaces de la aplicación van a tener la misma estructura común, tendrán una cabecera (que servirá siempre para volver a la página principal), el contenido y un pie de página.

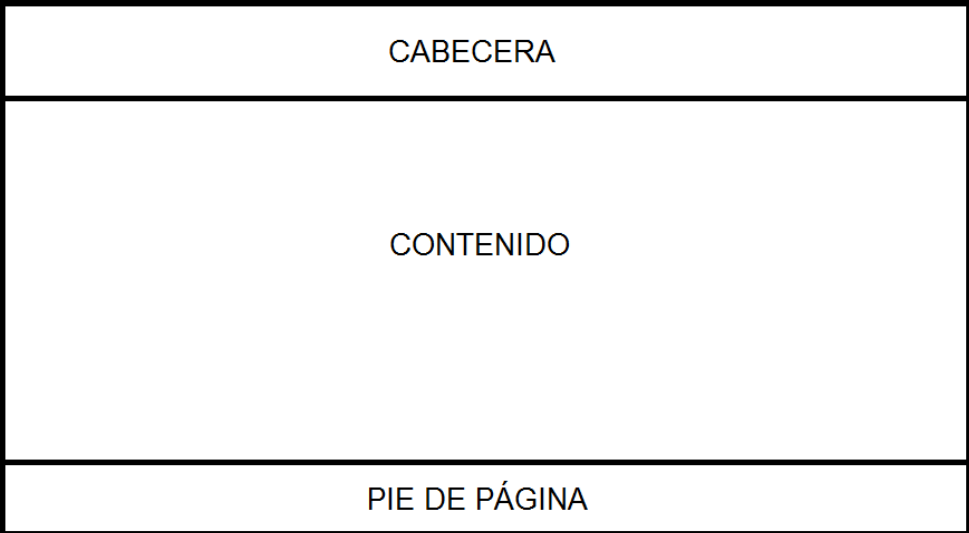
DI-01	Estructura común
Definición	Estructura común a todas las interfaces de usuario de la aplicación “Aprendiendo los números enteros”.
Activación	Esta interfaz se activa al entrar en la aplicación a través de un navegador web.
Boceto	

Tabla 78: Diseño de interfaz 1 – Estructura común.

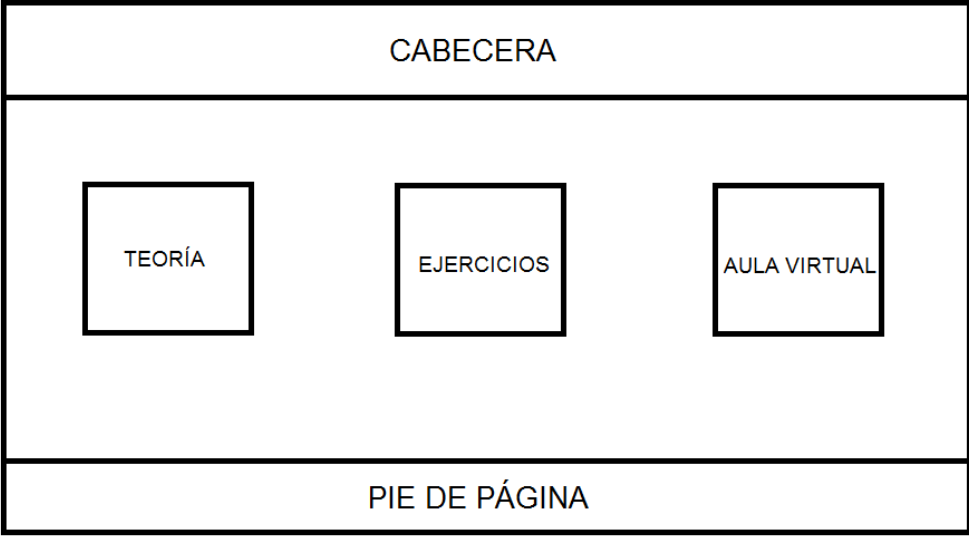
DI-02	Página principal
Definición	Esta es la página es la que los usuarios visualizarán nada más acceder a la aplicación.
Activación	Esta interfaz se activa al entrar en la aplicación a través de un navegador web.
Boceto	

Tabla 79: Diseño de interfaz 2 – Página principal.

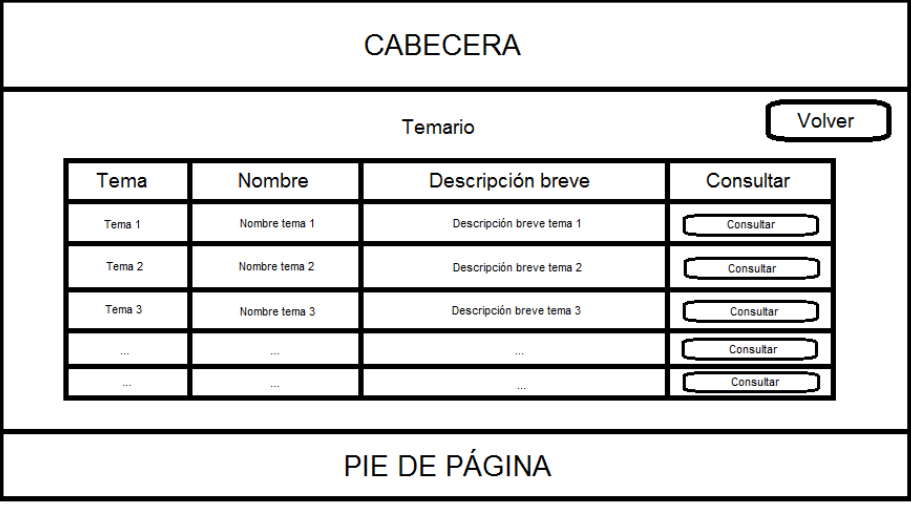
DI-03	Teoría
Definición	Esta página mostrará un listado con los diferentes temas disponibles.
Activación	Esta interfaz se activa cuando el usuario hace click en Teoría.
Boceto	

Tabla 80: Diseño de interfaz 3 – Teoría.

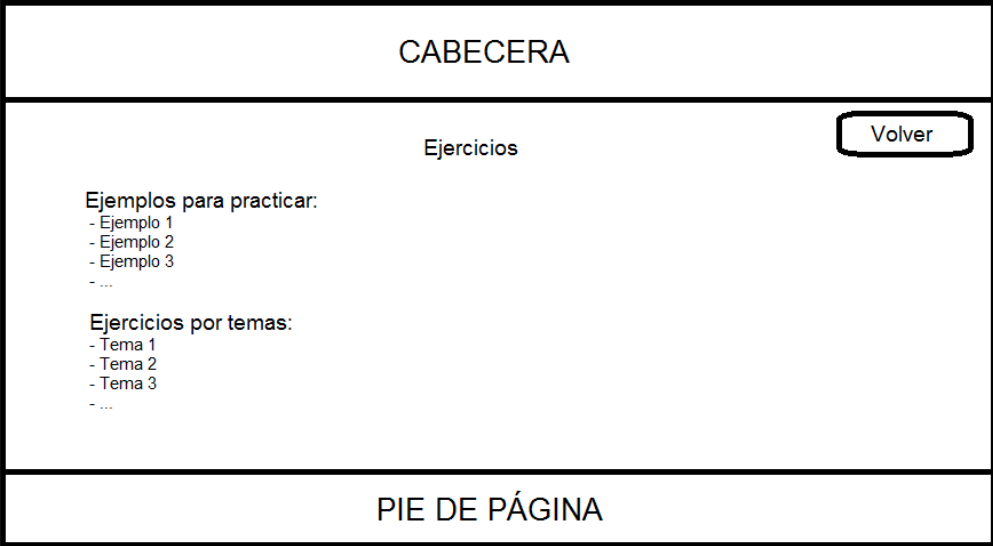
DI-04	Ejercicios
Definición	Esta página mostrará un listado con los diferentes ejercicios, por una parte estarán los ejemplos para practicar y por otra, los ejercicios propios de cada tema.
Activación	Esta interfaz se activa cuando el usuario hace click en Ejercicios.
Boceto	

Tabla 81: Diseño de interfaz 4 – Ejercicios.

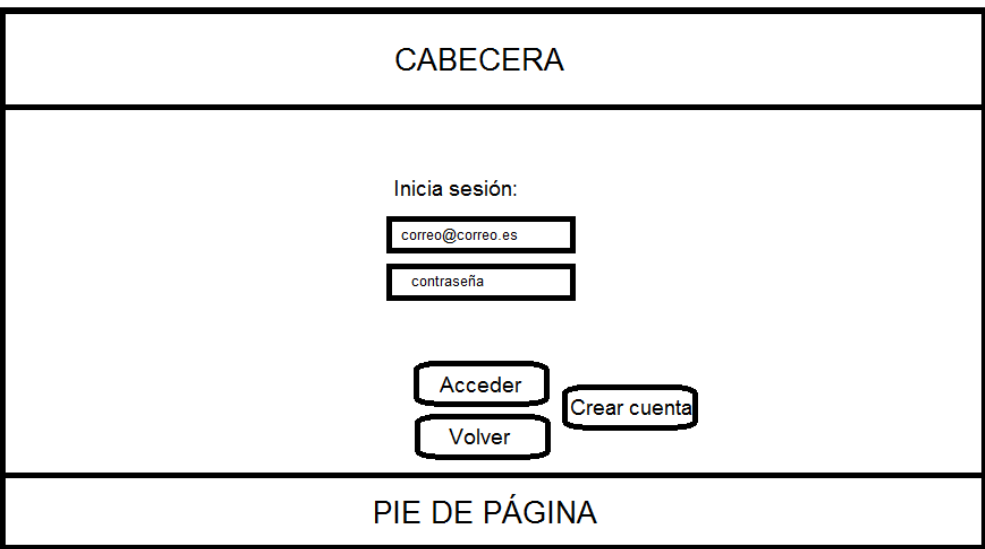
DI-05	Aula virtual
Definición	Esta página mostrará un formulario para que, un usuario registrado se pueda logar o si no está registrado se da la posibilidad que se registre.
Activación	Esta interfaz se activa cuando el usuario hace click en Aula Virtual.
Boceto	

Tabla 82: Diseño de interfaz 5 – Aula virtual.

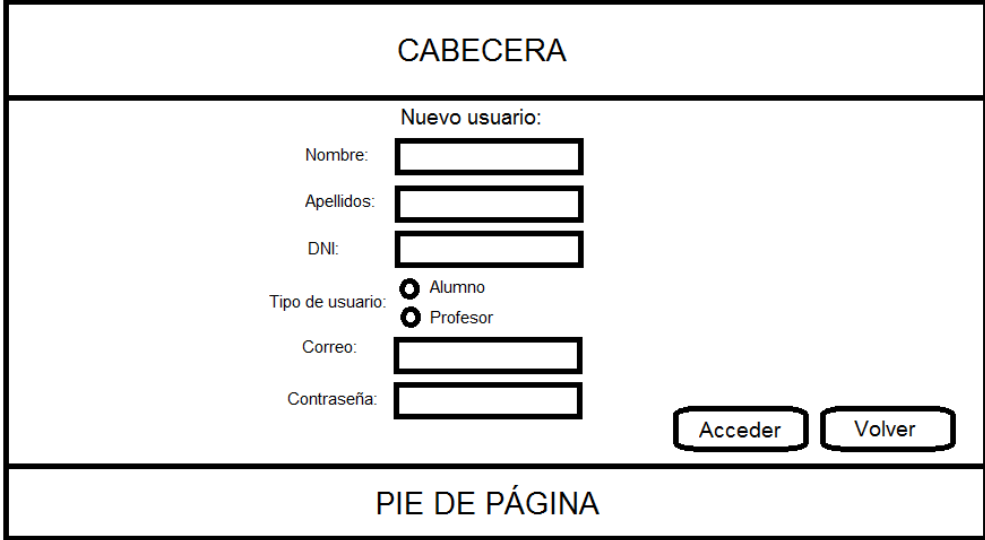
DI-06	Crear cuenta
Definición	Esta página mostrará un formulario para que un usuario se registre si así lo desea.
Activación	Esta interfaz se activa cuando el usuario hace click en Crear cuenta una vez haya entrado en el menú de Aula
Boceto	

Tabla 83: Diseño de interfaz 6 – Crear cuenta.

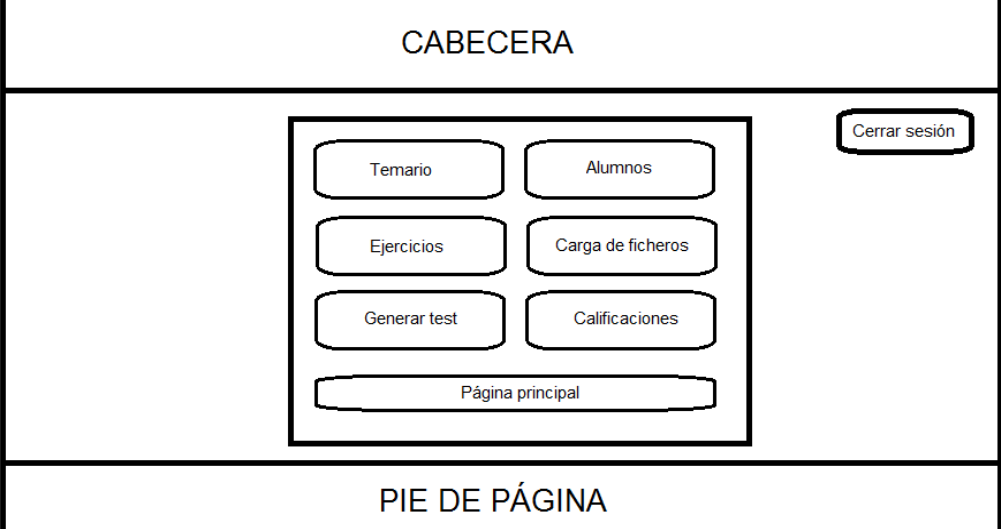
DI-07	Menú profesor
Definición	En esta página se mostrará el menú de administración del profesor con las diferentes opciones que puede realizar.
Activación	Esta interfaz se activa cuando el usuario se loga como rol de profesor.
Boceto	

Tabla 84: Diseño de interfaz 7 – Menú profesor.

DI-08	Menú alumno
Definición	En esta página se mostrará el menú disponible para un alumno registrado.
Activación	Esta interfaz se activa cuando el usuario se loga como rol de alumno.
Boceto	

Tabla 85: Diseño de interfaz 8 – Menú alumno.

DI-09	Menú profesor – edición temario
Definición	En esta página se mostrará un listado de los temas dados de alta así como la posibilidad añadir, modificar y eliminar un tema.
Activación	Esta interfaz se activa cuando el profesor hace click sobre Temario dentro del menú de administración.
Boceto	

Tabla 86: Diseño de interfaz 9 – Menú profesor – edición temario.

DI-10	Menú profesor – edición temario – añadir tema
Definición	En esta página se mostrará la posibilidad de dar de alta un tema.
Activación	Esta interfaz se activa cuando el profesor hace click Añadir tema.
Boceto	

Tabla 87: Diseño de interfaz 10 – Menú profesor – edición temario – añadir tema.

DI-11	Menú profesor – edición ejercicios
Definición	En esta página se mostrará un menú para poder añadir, modificar y eliminar ejercicios.
Activación	Esta interfaz se activa cuando el profesor hace click Ejercicios.
Boceto	

Tabla 88: Diseño de interfaz 11 – Menú profesor – edición ejercicios.

DI-12	Menú profesor – edición ejercicios – añadir ejercicio
Definición	En esta página se mostrará un formulario sencillo para que el profesor pueda un ejercicio fácilmente.
Activación	Esta interfaz se activa cuando el profesor hace click Añadir ejercicio.
Boceto	

Tabla 89: Diseño de interfaz 12 – Menú profesor – edición ejercicios – añadir ejercicio.

DI-13	Menú profesor – cargar ficheros
Definición	En esta página se mostrará una serie de funcionalidades para poder subir archivos a la aplicación, también se da la posibilidad de eliminar archivos que previamente se hayan subido.
Activación	Esta interfaz se activa cuando el profesor hace click en Carga de ficheros.
Boceto	

Tabla 90: Diseño de interfaz 13 – Menú profesor – cargar ficheros.

DI-14	Menú profesor – generar test
Definición	En esta página se mostrará un formulario para que el profesor pueda generar un test a partir de los distintos ejercicios de cada tema. Se da la posibilidad de previsualizarlo antes de guardarlo.
Activación	Esta interfaz se activa cuando el profesor hace click en Generar Test.
Boceto	<p style="text-align: center;">CABECERA</p> <p style="text-align: right;">Volver Cerrar sesión</p> <p>Fecha</p> <p>Nombre de la prueba</p> <p>Comentarios</p> <p>Lista de ejercicios posibles para añadir al test</p> <p>Previsualizar Guardar</p> <p style="text-align: center;">PIE DE PÁGINA</p>

Tabla 91: Diseño de interfaz 14 – Menú profesor – generar test.

DI-15	Menú profesor – calificaciones																				
Definición	En esta página se mostrará un listado de los alumnos dados de alta. Se da la posibilidad al profesor de consultar los archivos que han ido subido los alumnos a lo largo del curso. El profesor puede añadir notas a cada alumno. Se muestra un histórico de notas y la media de cada alumno.																				
Activación	Esta interfaz se activa cuando el profesor hace click en Calificaciones.																				
Boceto	<p style="text-align: center;">CABECERA</p> <p style="text-align: right;">Volver Cerrar sesión</p> <table border="1"> <thead> <tr> <th>Alumnos</th> <th>Consulta ficheros</th> <th>Calificación</th> <th>Histórico de notas</th> <th>Media</th> </tr> </thead> <tbody> <tr> <td>Alumno 1</td> <td>Consultar</td> <td>5</td> <td>5-5-8</td> <td>6.00</td> </tr> <tr> <td>Alumno 2</td> <td>Consultar</td> <td>7</td> <td>7</td> <td>7</td> </tr> <tr> <td>Alumno 3</td> <td>Consultar</td> <td>8.6</td> <td>8.6</td> <td>8.6</td> </tr> </tbody> </table> <p>Anterior Siguiente Guardar notas</p> <p style="text-align: center;">PIE DE PÁGINA</p>	Alumnos	Consulta ficheros	Calificación	Histórico de notas	Media	Alumno 1	Consultar	5	5-5-8	6.00	Alumno 2	Consultar	7	7	7	Alumno 3	Consultar	8.6	8.6	8.6
Alumnos	Consulta ficheros	Calificación	Histórico de notas	Media																	
Alumno 1	Consultar	5	5-5-8	6.00																	
Alumno 2	Consultar	7	7	7																	
Alumno 3	Consultar	8.6	8.6	8.6																	

Tabla 92: Diseño de interfaz 15 – Menú profesor – calificaciones.

DI-16	Menú profesor – alumnos
Definición	En esta página se mostrará un listado de los alumnos dados de alta. Se da la posibilidad al profesor de modificarlos o eliminarlos, así como activar a un alumno en la aplicación o no. También se puede añadir un nuevo alumno.
Activación	Esta interfaz se activa cuando el profesor hace click en Alumno.
Boceto	

Tabla 93: Diseño de interfaz 16 – Menú profesor – alumnos.

DI-17	Contactar con el profesor
Definición	En esta página se da la posibilidad de contactar con el profesor.
Activación	Esta interfaz se activa cuando el alumno hace click en Contactar con el profesor.
Boceto	

Tabla 94: Diseño de interfaz 17 – Contactar con el profesor.

6. PRUEBAS

6.1 Introducción

Las pruebas software son los procesos que permiten verificar y revelar la calidad de un producto software.

Son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de una aplicación informática. Básicamente, es una fase en el desarrollo de software consistente en probar las aplicaciones construidas.

Las pruebas de software se integran dentro de las diferentes fases del ciclo del software dentro de la Ingeniería de Software. Así se ejecuta un programa y mediante técnicas experimentales se intenta descubrir qué errores tiene.

El objetivo de las pruebas es la detección de defectos en el software, descubrir un error es el éxito de una prueba. El testing puede probar la presencia de errores, pero no la ausencia de ellos.

Hay muchos planteamientos a la hora de abordar el proceso de pruebas de software, pero verificar productos complejos de forma efectiva requiere de un proceso de investigación más que seguir un procedimiento al pie de la letra. Una definición de "testing" es: proceso de evaluación de un producto desde un punto de vista crítico, donde el "tester" es la persona que realiza las pruebas, y somete al producto a una serie de acciones inquisitivas, respondiendo el producto con un comportamiento como reacción.

➤ **Recomendaciones para que las pruebas sean satisfactorias:**

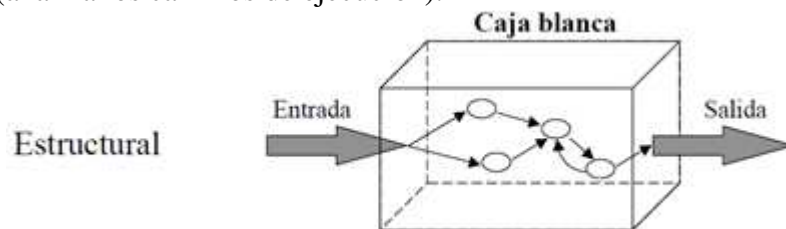
- Cada caso de prueba debe definir el resultado de salida esperado que se comparará con el realmente obtenido.
- El programador debe evitar probar sus propios programas, ya que desea (consciente o inconscientemente) demostrar que funcionan sin problemas.
 - Además, es normal que las situaciones que olvidó considerar al codificar queden de nuevo olvidados al crear los casos de prueba.
- Se debe inspeccionar a conciencia el resultado de cada prueba, así, se podrá descubrir posibles defectos de la codificación.
- Al generar casos de prueba, se deben incluir tanto datos de entrada válidos y esperados como no válidos e inesperados.
- Las pruebas deben centrarse en dos objetivos:
 - Probar si el software no hace lo que debe hacer.
 - Probar si el software hace lo que debe hacer, es decir, si provoca efectos secundarios adversos.

- No deben hacerse planes de prueba suponiendo que, prácticamente, no hay defectos en los programas y, por lo tanto, dedicando pocos recursos a las pruebas: **siempre hay defectos y errores.**
- La experiencia parece indicar que donde hay un defecto hay otros, es decir, la probabilidad de descubrir nuevos defectos en una parte del software es proporcional al número de defectos ya descubierto.
- Las pruebas son una tarea tanto o más creativa que el desarrollo de software.

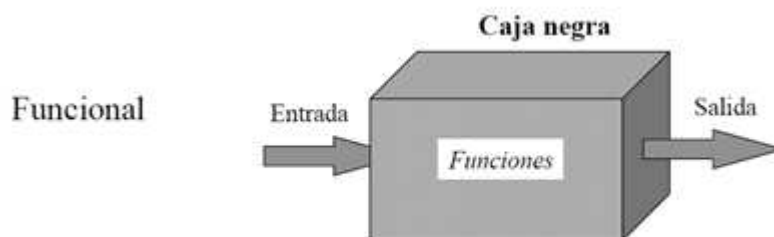
Es interesante planificar y diseñar las pruebas para poder detectar el máximo número y variedad de defectos con el mínimo consumo de tiempo y esfuerzo.

En este TFG se han llevado a cabo dos tipos de diseño de pruebas:

1.- El enfoque estructural o de **caja blanca**. Se centra en la estructura interna del programa (analiza los caminos de ejecución).



2.- El enfoque funcional o de **caja negra**. Se centra en las funciones, entradas y salidas.



6.2 Pruebas de caja negra

Las pruebas de caja negra se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ello se denominan pruebas funcionales, y el “tester” se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro.

Las pruebas de caja negra están especialmente indicadas en aquellos módulos que van a ser interfaz con el usuario.

Las pruebas de caja negra se apoyan en la especificación de requisitos del módulo. De hecho, se habla de "cobertura de especificación" para dar una medida del número de requisitos que se han probado. Es fácil obtener coberturas del 100% en módulos internos, aunque puede ser más laborioso en módulos con interfaz al exterior. En cualquier caso, es muy recomendable conseguir una alta cobertura en esta línea.

Caso de prueba	PRCN-01 Crear usuario
Propósito	Crear un nuevo usuario para que puede acceder al contenido de la aplicación.
Prerrequisitos	El usuario que se va a registrar no debe estar almacenado en la BD.
Datos de entrada	En el formulario de inicio de sesión se le pedirán los datos de “usuario” y “contraseña”.
Resultado esperado	El usuario (profesor o alumno) queda registrado y almacenado en la base de datos.
Resultado obtenido	Correcto

Tabla 95: Caso de prueba 1 - Crear usuario.

Caso de prueba	PRCN-02 Iniciar sesión en la aplicación
Propósito	El usuario introduce las credenciales de su cuenta (correo y contraseña) para acceder al sistema como un usuario registrado (profesor o alumno registrado).
Prerrequisitos	El usuario debe estar registrado en la BD.
Datos de entrada	En el formulario de inicio de sesión se le pedirán los datos de “correo” y “contraseña”.
Resultado esperado	El usuario (profesor o alumno) inicia sesión en la aplicación, teniendo acceso a sus funcionalidades.
Resultado obtenido	Correcto

Tabla 96: Caso de prueba 2 - Iniciar sesión en la aplicación.

Caso de prueba	PRCN-03 Cerrar sesión en la aplicación
Propósito	El usuario cierra sesión en la aplicación y el sistema eliminará su sesión. Se redirige a la página principal.
Prerrequisitos	El usuario debe haber iniciado sesión con sus credenciales.
Datos de entrada	N/A
Resultado esperado	El sistema elimina la sesión de usuario y redirige al usuario a la pantalla de inicio.
Resultado obtenido	Correcto

Tabla 97: Caso de prueba 3 - Cerrar sesión en la aplicación.

Caso de prueba	PRCN-04 Añadir tema
Propósito	El usuario con rol profesor accede al menú de administración y da de alta un tema.
Prerrequisitos	El profesor debe haber iniciado sesión en la aplicación.
Datos de entrada	Se deben rellenar los siguientes datos para dar de alta un tema: nombre, tema, descripción y visibilidad.
Resultado esperado	El sistema almacena el nuevo tema.
Resultado obtenido	Correcto

Tabla 98: Caso de prueba 4 - Añadir tema.

Caso de prueba	PRCN-05 Modificar tema
Propósito	El usuario con rol profesor accede al menú de administración y modifica información de un tema ya existente.
Prerrequisitos	El profesor debe haber iniciado sesión en la aplicación y el tema a modificar debe existir.
Datos de entrada	En la modificación de datos de un tema se puede modificar cualquier información.
Resultado esperado	El sistema almacena las modificaciones realizadas por el usuario.
Resultado obtenido	Correcto

Tabla 99: Caso de prueba 5 - Modificar tema.

Caso de prueba	PRCN-06 Eliminar tema
Propósito	El usuario con rol profesor accede al menú de administración y elimina un tema ya existente.
Prerrequisitos	El profesor debe haber iniciado sesión en la aplicación y el tema a eliminar debe existir.
Datos de entrada	N/A
Resultado esperado	El sistema elimina el tema.
Resultado obtenido	Correcto

Tabla 100: Caso de prueba 6 - Eliminar tema.

Caso de prueba	PRCN-07 Añadir ejercicio
Propósito	El usuario con rol profesor accede al menú de administración y añade un nuevo ejercicio.
Prerrequisitos	El profesor debe haber iniciado sesión en la aplicación.
Datos de entrada	Para dar de alta un nuevo ejercicio hay que rellenar los siguientes datos: tema asociado al ejercicio, título del ejercicio, enunciado del ejercicio y solución.
Resultado esperado	El sistema almacena el nuevo ejercicio.
Resultado obtenido	Correcto

Tabla 101: Caso de prueba 7 - Añadir ejercicio.

Caso de prueba	PRCN-08 Eliminar ejercicio
Propósito	El usuario con rol profesor accede al menú de administración y elimina un ejercicio previamente creado.
Prerrequisitos	El profesor debe haber iniciado sesión en la aplicación y el ejercicio a eliminar debe existir.
Datos de entrada	N/A
Resultado esperado	El sistema elimina el ejercicio.
Resultado obtenido	Correcto

Tabla 102: Caso de prueba 8 - Eliminar ejercicio.

Caso de prueba	PRCN-09 Modificar ejercicio
Propósito	El usuario con rol profesor accede al menú de administración y modifica un ejercicio previamente creado.
Prerrequisitos	El profesor debe haber iniciado sesión en la aplicación y el ejercicio a modificar debe existir.
Datos de entrada	En la modificación de datos de un ejercicio se puede modificar cualquier información.
Resultado esperado	El sistema almacena las modificaciones del ejercicio que ha realizado el usuario.
Resultado obtenido	Correcto

Tabla 103: Caso de prueba 9 - Modificar ejercicio.

Caso de prueba	PRCN-10 Añadir alumno
Propósito	El usuario con rol profesor accede al menú de administración y añade un nuevo alumno.
Prerrequisitos	El profesor debe haber iniciado sesión en la aplicación.
Datos de entrada	Para dar de alta un nuevo alumno hay que rellenar los siguientes datos: nombre, apellidos, contacto, teléfono y visibilidad.
Resultado esperado	El sistema almacena el nuevo alumno.
Resultado obtenido	Correcto

Tabla 104: Caso de prueba 10 - Añadir alumno.

Caso de prueba	PRCN-11 Eliminar alumno
Propósito	El usuario con rol profesor accede al menú de administración y elimina un alumno previamente creado.
Prerrequisitos	El profesor debe haber iniciado sesión en la aplicación y el alumno a eliminar debe existir.
Datos de entrada	N/A
Resultado esperado	El sistema elimina el alumno y todos los datos asociados a él.
Resultado obtenido	Correcto

Tabla 105: Caso de prueba 11 - Eliminar alumno.

Caso de prueba	PRCN-12 Modificar alumno
Propósito	El usuario con rol profesor accede al menú de administración y modifica un alumno previamente creado.
Prerrequisitos	El profesor debe haber iniciado sesión en la aplicación y el alumno a modificar debe existir.
Datos de entrada	En la modificación de datos de un alumno se puede modificar cualquier información.
Resultado esperado	El sistema almacena las modificaciones del alumno que ha realizado el profesor.
Resultado obtenido	Correcto

Tabla 106: Caso de prueba 12 - Modificar alumno.

Caso de prueba	PRCN-13 Cargar/Enviar archivo
Propósito	El usuario registrado carga un archivo para que sea visible bien por el profesor o por el alumno registrado.
Prerrequisitos	El usuario debe haber iniciado sesión con sus credenciales.
Datos de entrada	Archivo que se desea subir a la aplicación.
Resultado esperado	El sistema almacena el archivo y a partir de este momento está visible para el profesor o el alumno registrado.
Resultado obtenido	Correcto

Tabla 107: Caso de prueba 13 - Cargar/Enviar archivo.

Caso de prueba	PRCN-14 Generar test
Propósito	El profesor generar un test a partir de los ejercicios disponibles.
Prerrequisitos	El profesor debe haber iniciado sesión con sus credenciales.
Datos de entrada	Para generar el test se rellenan los siguientes datos: fecha, nombre de la prueba, descripción de la prueba y ejercicios que se quiere incluir.
Resultado esperado	El sistema almacena el test y está disponible para los alumnos.
Resultado obtenido	Correcto

Tabla 108: Caso de prueba 14 - Generar test.

Caso de prueba	PRCN-15 Añadir calificaciones
Propósito	El profesor añade una nueva nota a un alumno.
Prerrequisitos	El profesor debe haber iniciado sesión con sus credenciales.
Datos de entrada	Nota que quiere añadir.
Resultado esperado	El sistema almacena la nueva nota y calcula la nueva media del alumno.
Resultado obtenido	Correcto

Tabla 109: Caso de prueba 15 - Añadir calificaciones.

Caso de prueba	PRCN-16 Consulta de notas
Propósito	El alumno consulta notas asociadas a las entregas que ha ido realizando.
Prerrequisitos	El alumno debe haber iniciado sesión con sus credenciales.
Datos de entrada	N/A
Resultado esperado	El alumno visualiza las notas, en un color está la nota actual y otro el resto de notas de entregas pasadas.
Resultado obtenido	Correcto

Tabla 110: Caso de prueba 16 - Consulta de notas.

6.3 Pruebas de caja blanca

En programación, se denomina pruebas de caja blanca a un tipo de pruebas de software que se realiza sobre las funciones internas de un módulo. Así como las pruebas de caja negra ejercitan los requisitos funcionales desde el exterior del módulo, las de caja blanca están dirigidas a las funciones internas.

Entre las técnicas usadas se encuentran:

- La cobertura de caminos (son pruebas que hacen que se recorran todos los posibles caminos de ejecución).
- Pruebas sobre las expresiones lógico-aritméticas.
- Pruebas de camino de datos (definición-uso de variables).
- Comprobación de bucles (se verifican los bucles para 0,1 e interacciones, y luego para las interacciones máximas, máximas menos uno y más uno).

En este TFG se ha llevado a cabo la técnica de cobertura de caminos, la cual consiste en escribir casos de prueba suficientes para que se ejecuten todos los caminos de un programa. Entendiendo camino como una secuencia de sentencias encadenadas desde la entrada del programa hasta su salida.

La aplicación de este criterio de cobertura asegura que los casos de prueba diseñados permiten que todas las sentencias del programa sean ejecutadas al menos una vez y que las condiciones sean probadas tanto para su valor verdadero como falso.

Ya que las pruebas de caja blanca se han realizado durante el desarrollo de la aplicación, se ha asegurado que todas ellas resultaban positivas antes de continuar con la siguiente. En el caso de las más complejas se ha intentado poner a prueba el máximo número de variaciones posibles para asegurar que ninguna combinación u orden concreto de órdenes a la aplicación tenga la posibilidad de provocar un error.

BLOQUE III
MANUAL
DE
USUARIO

7. MANUAL DE USUARIO

7.1 Introducción

En este manual se explica detalladamente tanto la instalación del software necesario para la ejecución de la aplicación, como su uso.

Este apartado estará dividido en las siguientes partes:

- Instalación del software.
- Manual de usuario.

“Aprendiendo los números enteros” está diseñada para que su uso sea sencillo a la vez que potente a la hora de gestionar contenidos, y con esta guía quedará completamente claro su uso.

7.2 Instalación de Software

7.2.1 Instalación del servidor

7.2.1.1 *Instalación del servidor XAMPP*

Para que la aplicación funcione correctamente desde un ordenador sin conexión a Internet y sin servicio de alojamiento web, se debe preparar el equipo de la mejor forma posible. El ordenador donde se instalará la aplicación actuará como servidor local, por lo que se debe instalar, los siguientes programas:

- Sistema gestor de base de datos PHPMyAdmin.
- MySQL
- Servidor Apache

La aplicación XAMPP contiene estos tres programas que necesitamos, por lo que, a continuación se procede a la instalación conjunta del paquete.

7.2.1.1.1 Descarga de la aplicación

XAMPP es un programa gratuito al que se puede acceder mediante el siguiente enlace:

<https://goo.gl/gqY6Fi>

Dentro de esta página podremos descargarnos todas las versiones actuales del programa. En este caso se ha utilizado la versión XAMPP 7.1.6 (que incluye PHP 7.1.6), que se puede descargar en el siguiente enlace:

<https://www.apachefriends.org/xampp-files/7.1.6/xampp-win32-7.1.6-0-VC14-installer.exe>

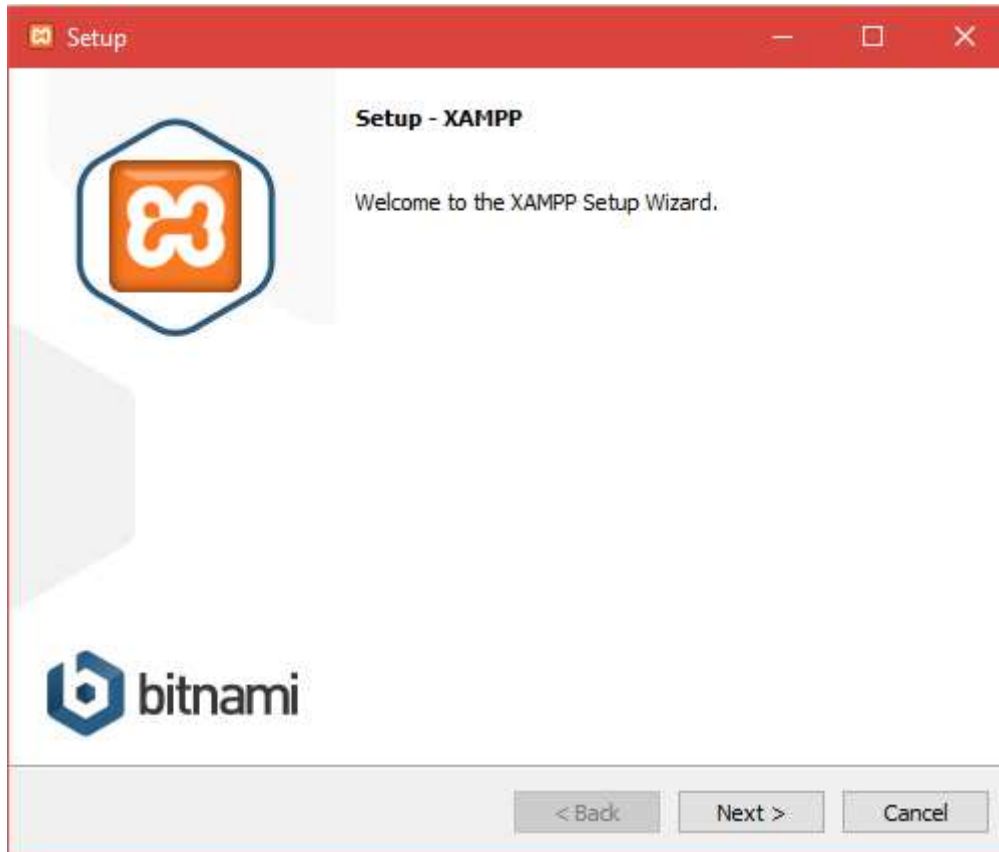
Al hacer click en la url anterior, se iniciará la descarga de XAMPP pasados unos segundos (almacenaremos la descarga en local):



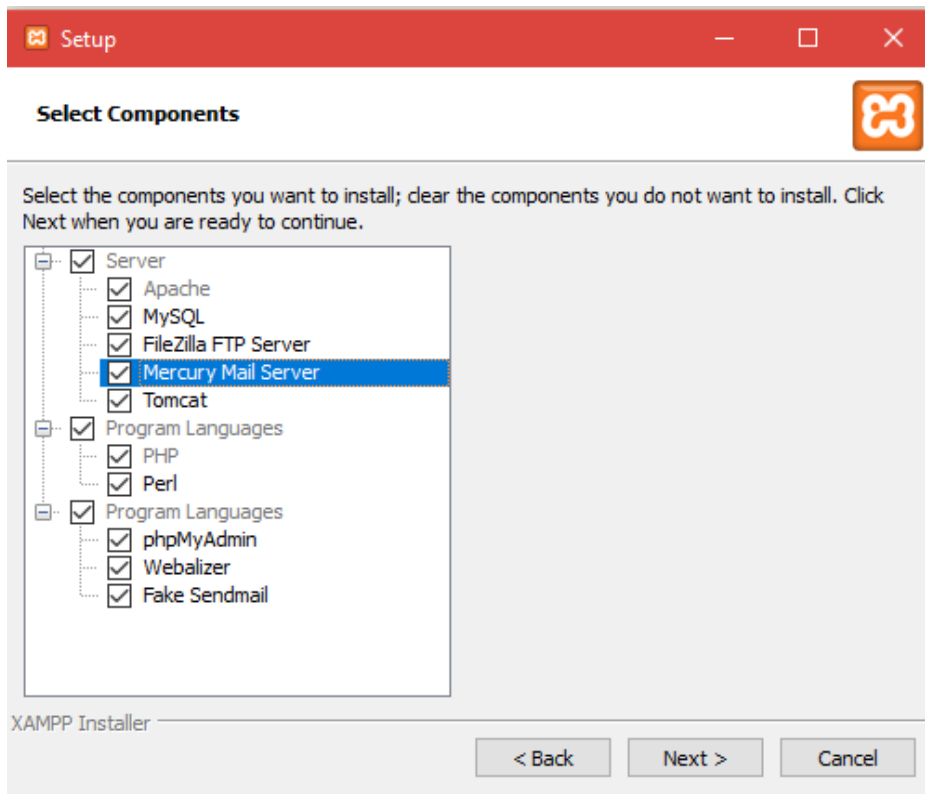
Una vez descargado el archivo pasamos al siguiente paso.

7.2.1.1.2 Instalación de la aplicación

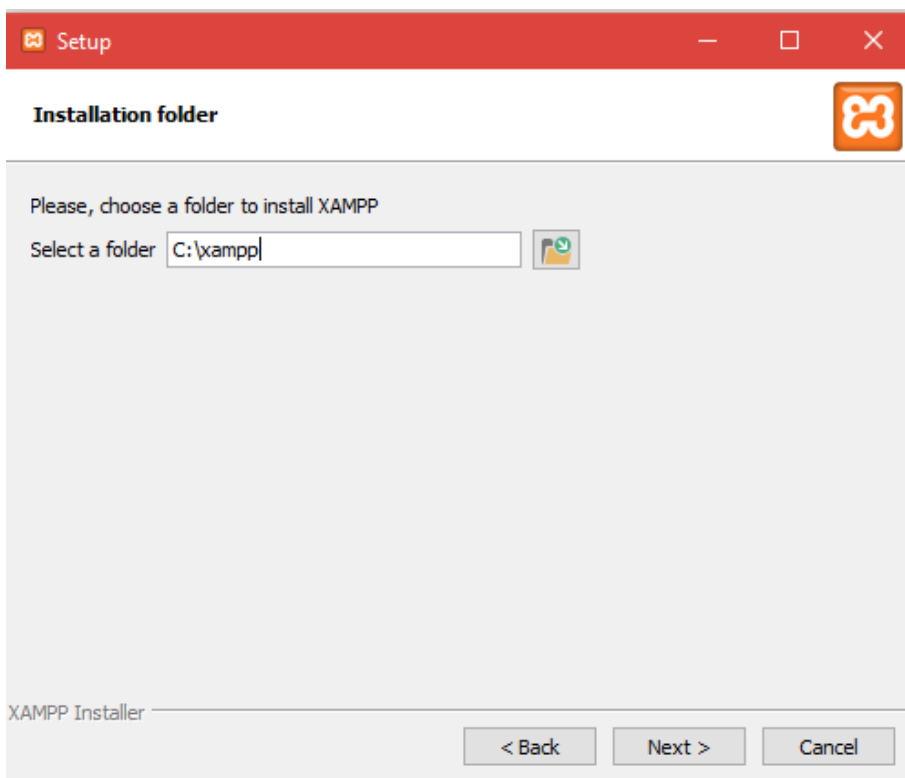
Se ejecuta el instalador que hemos descargado y nos aparecerá la siguiente ventana de bienvenida:



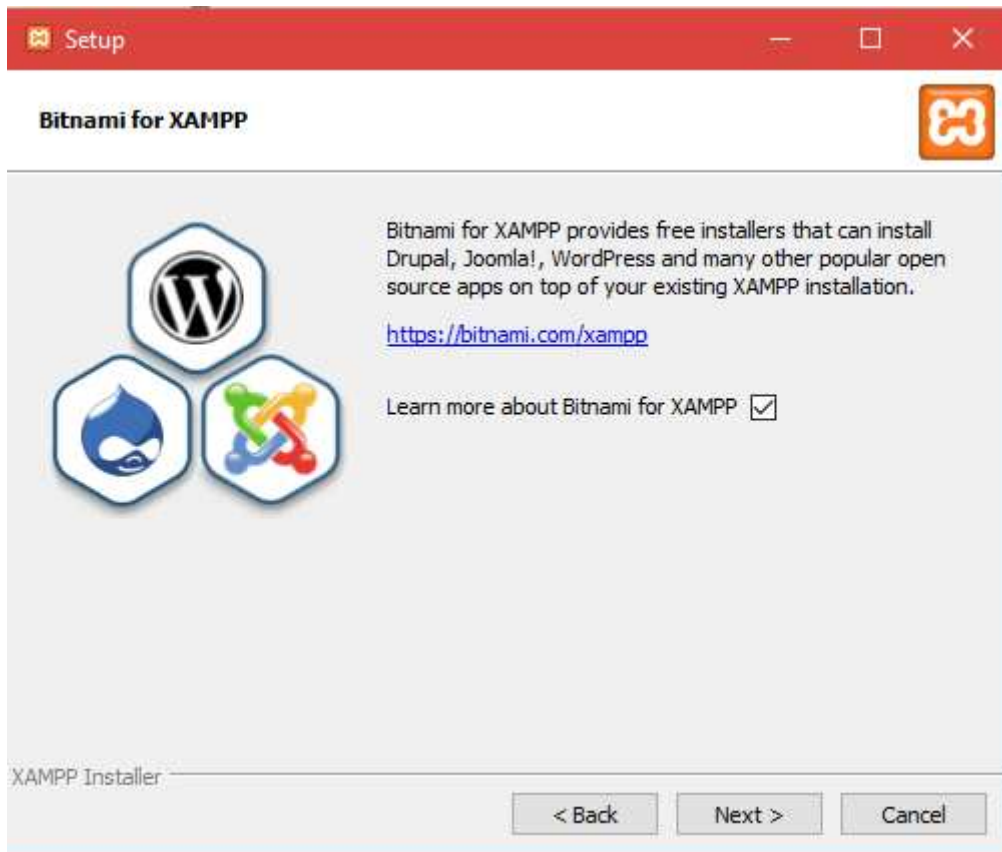
Pulsando el botón “Next” aparecerá una ventana con todas las opciones que se pueden instalar, lo dejamos según aparece y pulsamos el botón “Next” de nuevo:



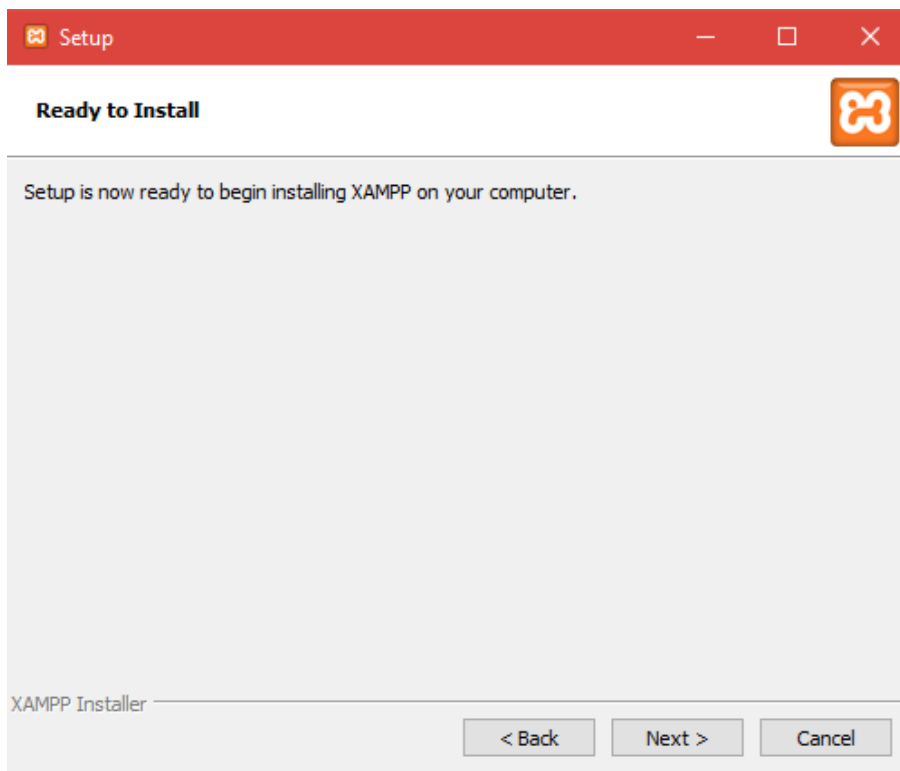
En la siguiente ventana aparecerá la dirección de la carpeta destino donde se alojará este programa. En este caso, se ha escogido la carpeta que viene por defecto, es decir, “c:\xampp”, aunque se podría cambiar pulsando el botón con el icono de carpeta.



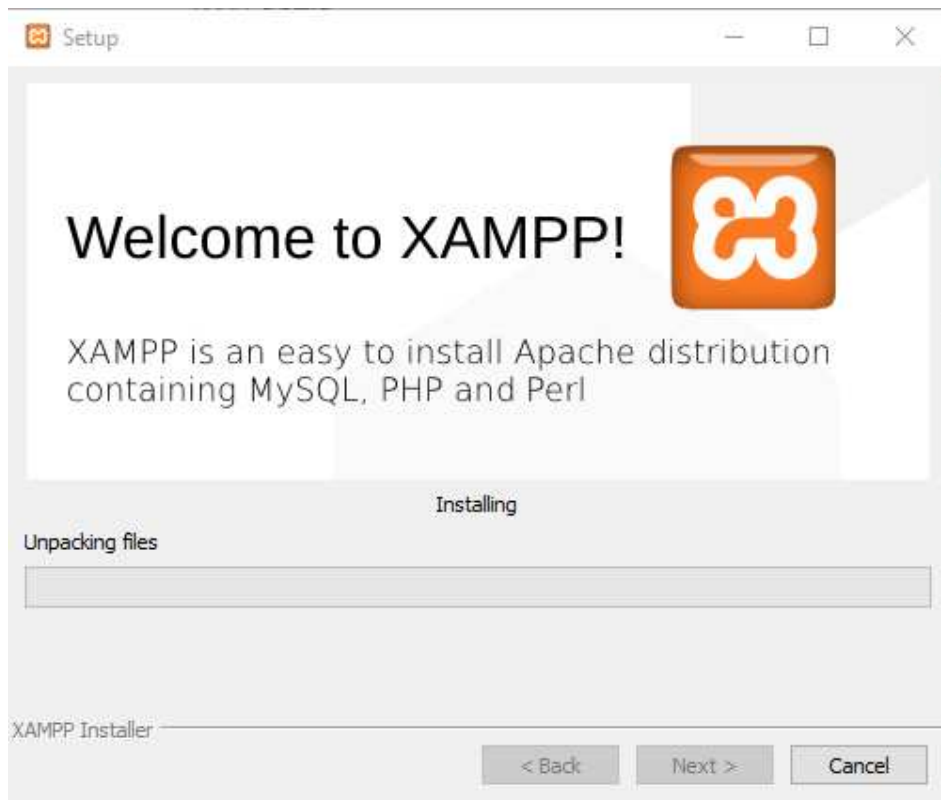
Una vez elegido el directorio, pulsamos “Next” y aparecerá la siguiente ventana:



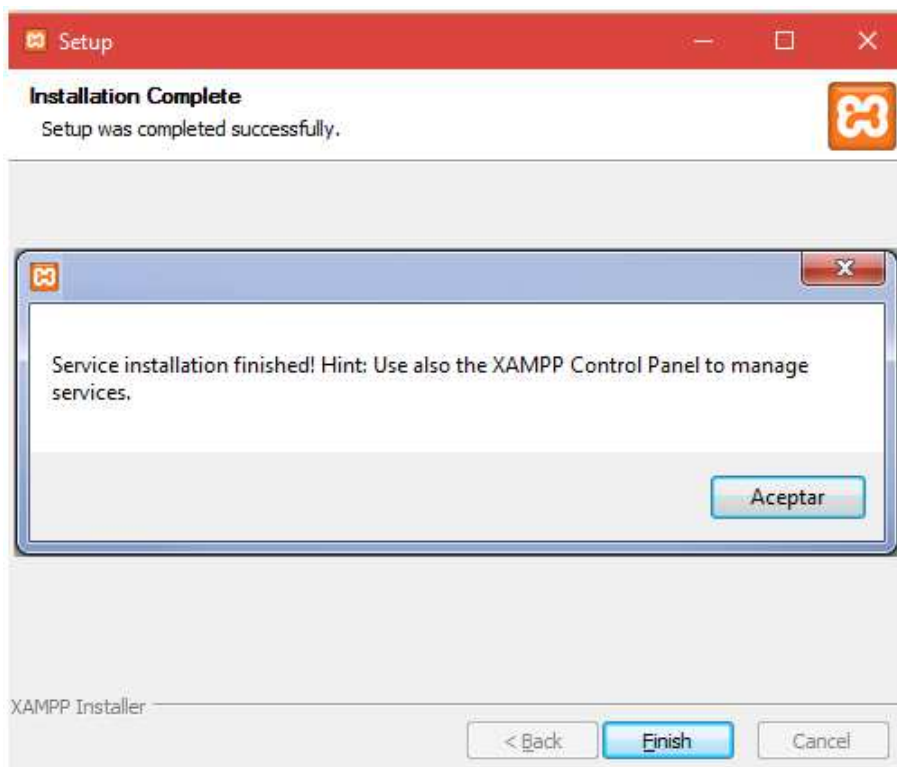
Pulsamos de nuevo el botón “Next” obtenemos una ventana donde nos avisa que XAMPP va ser instalado:



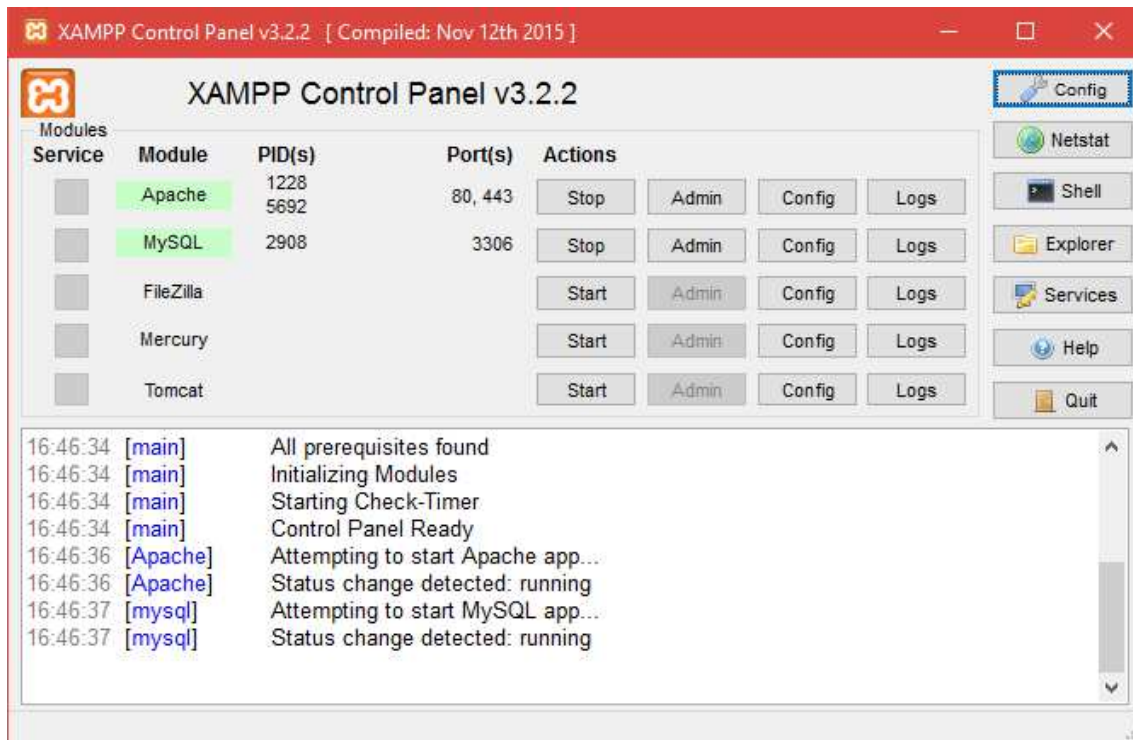
Pulsamos “Next” y en este momento empezará a instalarse el programa. Esto tardará unos minutos:



Una vez terminada la instalación aparecerá una ventana que informará que la instalación se ha realizado con éxito, y para finalizar habrá que hacer click en “Aceptar” y después en el botón "Finish":



Posteriormente nos muestra un mensaje, en el que se nos pregunta si deseamos iniciar el panel de control de XAMPP, en el cual haremos click sobre el botón "Sí" apareciendo el siguiente panel:



Como se puede observar, tanto Apache como MySQL, que son las aplicaciones que necesitamos, están ejecutándose.

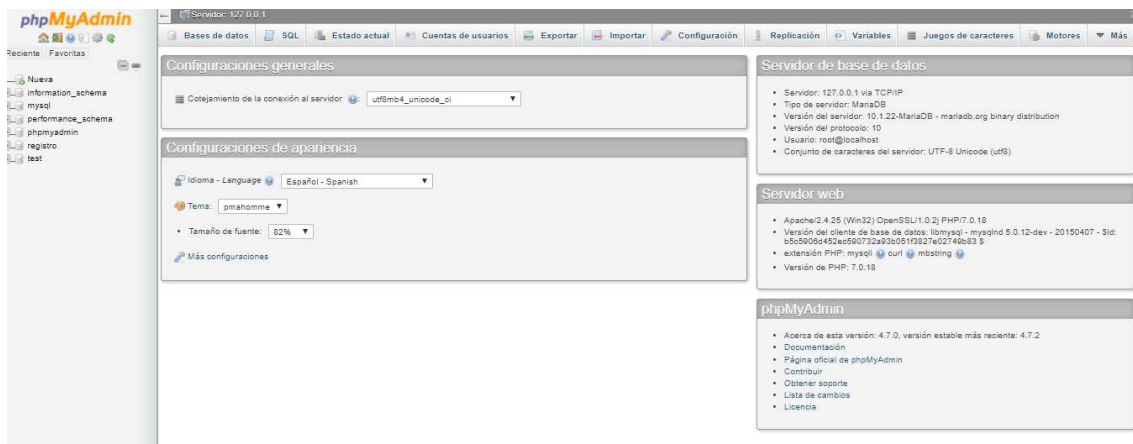
Cada vez que iniciemos el ordenador, debemos iniciar estos dos programas desde el panel de control de XAMPP pulsando el botón "Start" correspondiente. Si en algún momento se desea parar el funcionamiento de estos programas habrá que pulsar en el botón "Stop" del proceso que se desea parar.

Los programas estarán en correcto funcionamiento cuando sobre el nombre de Apache y MySQL aparezca un fondo verde.

Si queremos saberlo de una forma más fiable, podemos hacer la prueba con el funcionamiento de MySQL, haciendo click aquí:

<http://localhost/phpmyadmin>

Si todo está correctamente, nos aparecerá una pantalla de este tipo:



En caso contrario (si MySQL no está ejecutándose), aparecerá una pantalla que nos indicará el tipo de error que se ha producido:

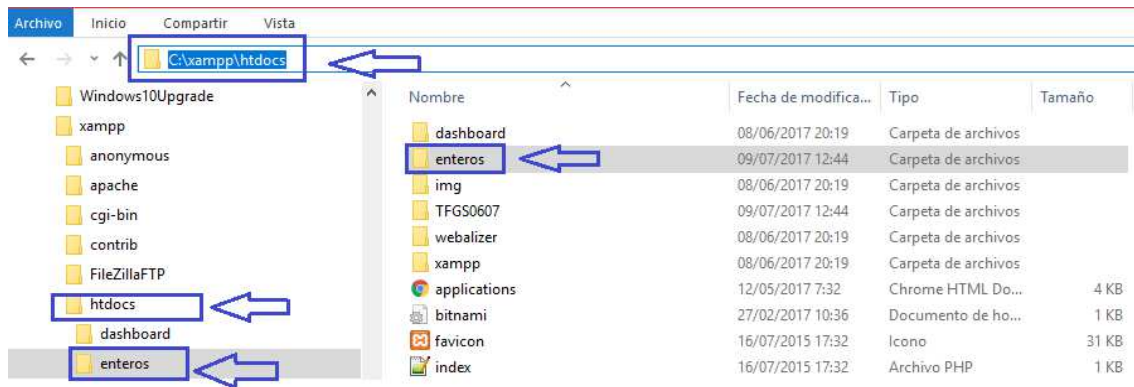


Con todos estos pasos realizados habremos terminado la configuración del paquete XAMPP, por lo tanto la aplicación funcionará correctamente de forma local.

7.2.1.2 Instalación de la aplicación web "Aprendiendo los números enteros"

Para que la aplicación web pueda funcionar correctamente de forma local, es necesario copiar la carpeta "enteros", que se encuentra dentro del CD-ROM en la parte de "Código Fuente" y pegarla en la carpeta "htdocs".

La carpeta "htdocs" la podemos encontrar en el directorio raíz de XAMPP. En este caso, la carpeta de destino se encuentra en "C:\xampp". Por tanto la carpeta "enteros" se tendrá que copiar dentro de "C:\xampp\htdocs" quedando de la siguiente forma:



De igual modo, si se cambió la carpeta de destino de XAMPP en la instalación, se deberá copiar el directorio mencionado anteriormente en la carpeta "htdocs" perteneciente a XAMPP.

7.2.1.3 Importación de la base de datos a MySQL

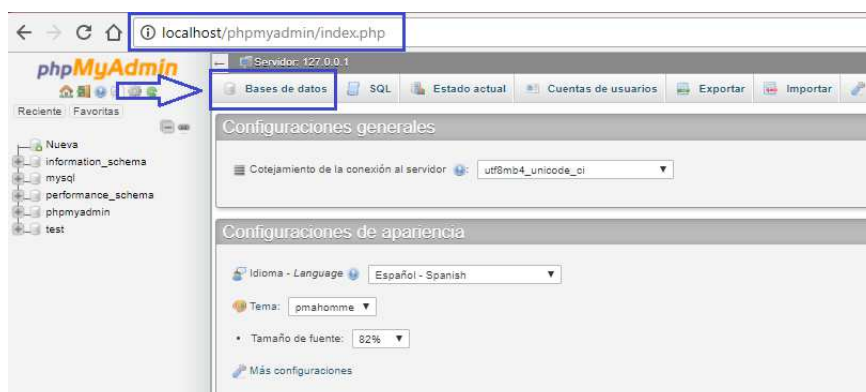
Para poder visualizar correctamente todos los contenidos de la web, será necesario importar una pequeña base de datos de ejemplo a MySQL.

Mediante unos pocos pasos muy sencillos lo conseguiremos.

Abrimos nuestro navegador, y escribimos en la barra de direcciones lo siguiente:

<http://localhost/phpmyadmin>

Nos aparece una ventana en la cual deberemos hacer click en la parte superior en "Bases de datos":



Una vez hecho esto, nos aparece una nueva ventana. En esta ventana deberemos añadir una nueva base de datos. Para ello donde pone "Crear nueva base de datos" escribimos la palabra "**registro**" y posteriormente pulsamos el botón "Crear".



Como podemos observar, nos aparece en el menú de la derecha, la nueva base de datos, y debemos hacer click sobre ella.



Posteriormente deberemos importar la base de datos. Para ello en la parte superior de la nueva ventana pulsamos sobre el texto "Importar":



En la nueva pantalla, deberemos pulsar sobre el botón examinar, y buscar en el "CD-ROM" la base de datos. Esta se encuentra en la carpeta de "Fichero para la base de datos" y su nombre es: "registro.sql", la seleccionamos y pulsamos en el botón abrir.

Una vez seleccionado el archivo, pulsamos en el botón "Continuar", como podemos ver en la imagen siguiente:

The screenshot shows a web browser window with the address bar displaying "Servidor: 127.0.0.1" and the page title "Base de datos: registro". The navigation menu includes "Estructura", "SQL", "Buscar", "Generar una consulta", "Exportar", "Importar", and "Op".

Importando en la base de datos "registro"

Archivo a importar:

El archivo puede ser comprimido (gzip, bzip2, zip) o descomprimido. Un archivo comprimido tiene que terminar en `[formato].[compresión]`. Por ejemplo: `.sql.zip`

Buscar en su ordenador: (Máximo: 102MB)

También puede arrastrar un archivo en cualquier página.

Conjunto de caracteres del archivo:

Importación parcial:

Permitir la interrupción de una importación en caso que el script detecte que se ha acercado al límite de tiempo PHP

Omitir esta cantidad de consultas (en SQL) desde la primera:

Otras opciones:

Habilite la revisión de las claves foráneas

Formato:

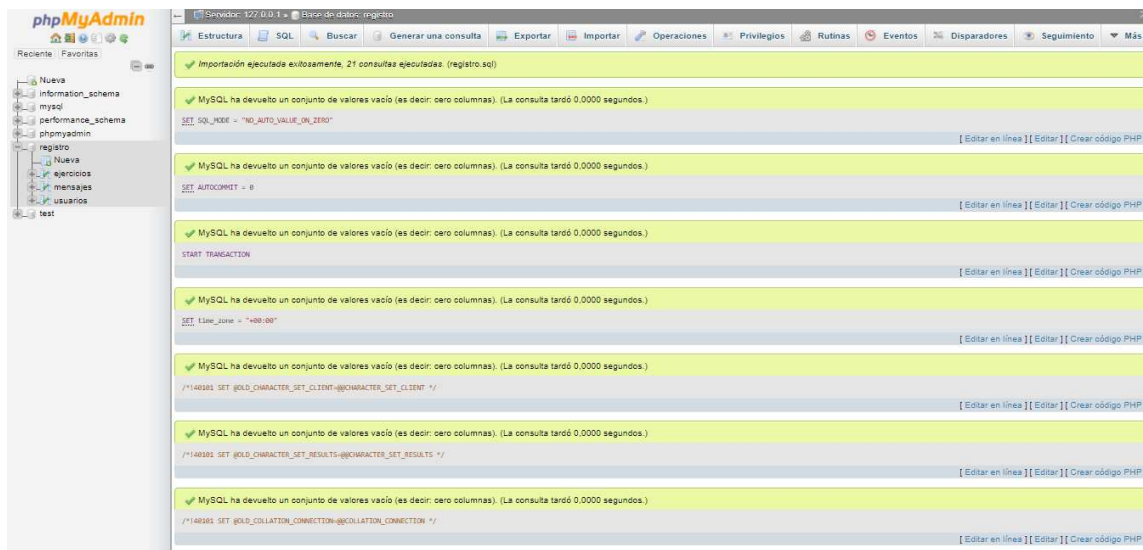
Opciones específicas al formato:

Modalidad SQL compatible:

No utilizar `AUTO_INCREMENT` con el valor 0

A blue box highlights the "Continuar" button, and a blue arrow points to it from the right.

Si todo se ha realizado correctamente nos deberá mostrar el siguiente mensaje:



7.2.2 Instalación de la aplicación

7.3 Manual de usuario

Esta parte del manual está destinada a todo usuario que desee utilizar la aplicación.

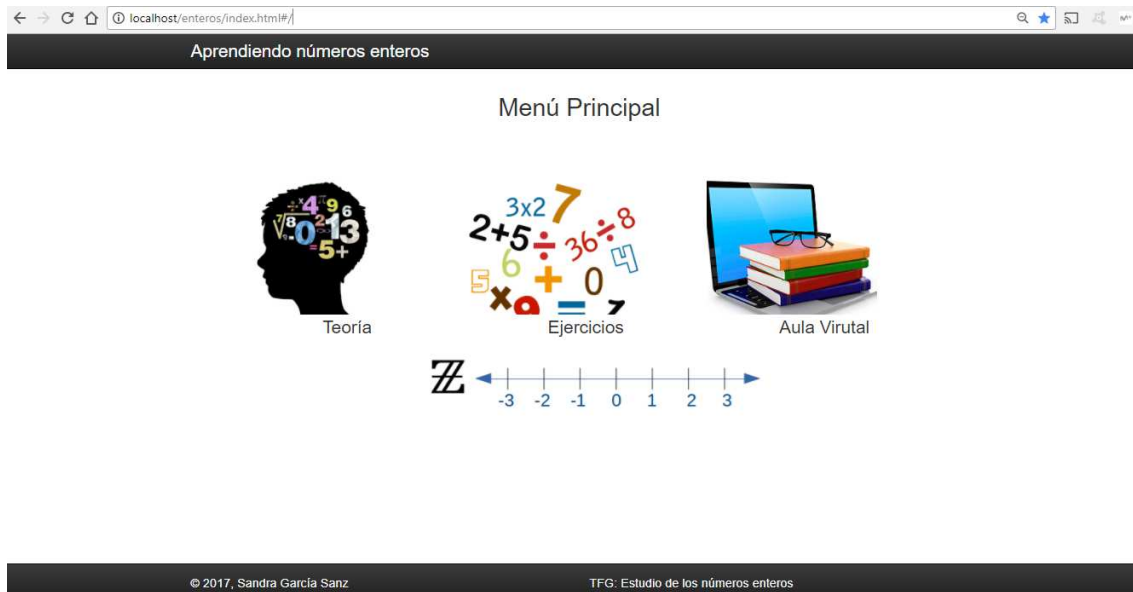
Funcionamiento

Esta aplicación se puede ejecutar de dos modos, una de forma local y la otra, a través de Internet. Independientemente en qué modo se ejecute, el funcionamiento será el mismo. A lo largo de todo el desarrollo y de este manual se trabajará de forma local.

Para poder ejecutar la aplicación de forma local, tendremos que abrir el navegador e introducir en la barra de direcciones lo siguiente:

<http://localhost/enteros/index.html#/>

Esto nos mostrará la página de la aplicación:



En todas las pantallas nos vamos a encontrar con tres partes diferenciadas:

➤ **Cabecera.**

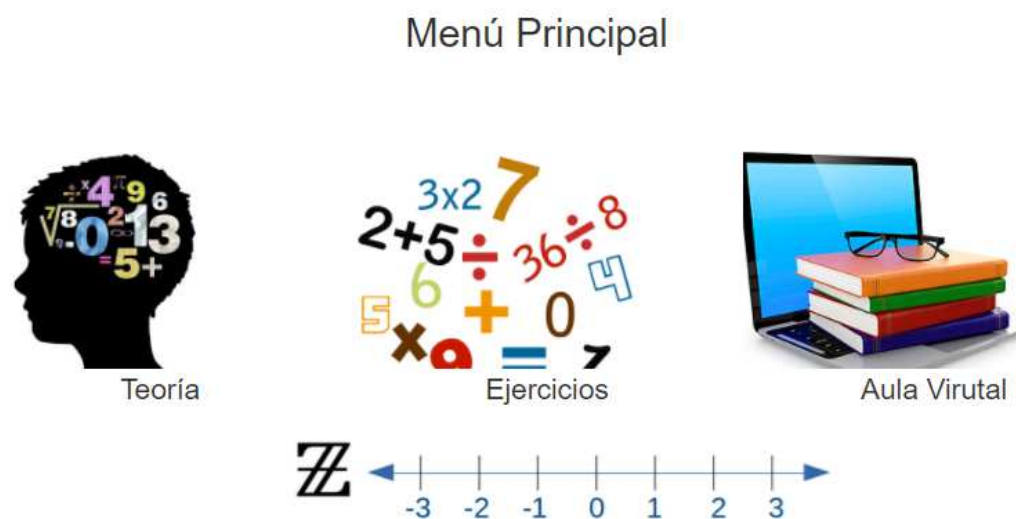
Muestra la sección en la que nos encontramos, vemos un par de ejemplos:

Aprendiendo números enteros

Aprendiendo números enteros | Teoría

➤ **Contenido.**

Aparecerá la información de la sección en la que estemos, como ejemplo, vemos la página de bienvenida:



➤ **Pie de página.**

Aparece el nombre del Trabajo Fin de Grado y el autor.
Esto no va a variar nunca.

© 2017, Sandra García Sanz

TFG: Estudio de los números enteros

NOTA: tanto al menú principal, como a la teoría y a los ejercicios podrá acceder cualquier tipo de usuario. Al aula virtual podrán acceder usuarios que se quieran dar de alta o que ya tengan credenciales para acceder a la parte privada de la aplicación.

Menú principal

Al entrar en la aplicación web veremos el menú principal. Este menú permite navegar por las diferentes secciones del portal. Si hacemos click sobre cada una de las imágenes nos llevará a su sección correspondiente.

Menú Principal



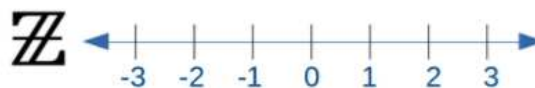
Teoría



Ejercicios



Aula Virtual



Teoría

En esta sección se permite a un usuario consultar todos los temas disponibles sobre los números enteros.

localhost/enteros/index.html#/teoria

Aprendiendo números enteros | Teoría

Temario de la asignatura

Volver

Tema	Nombre	Descripción Breve	Consultar
0	Introducción	Introducción a los números enteros	Consultar
1	Tema 1	Ordenar y comparar números enteros	Consultar
2	Tema 2	Valor absoluto y Opuesto	Consultar
3	Tema 3	Suma y diferencia de enteros	Consultar
4	Tema 4	Producto y división de enteros	Consultar

Además de consultar los diferentes temas, tendrá la opción de imprimir el tema consultado.

The screenshot shows a web browser window with a print dialog on the left and a page titled 'Aprendiendo los números enteros' on the right. The print dialog is for 'PDFCreator' and shows 'Total: 2 hojas de papel'. The page content includes a section 'Tema 1 Ordenar y comparar números enteros' with a diagram of a number line labeled 'Recta Numérica' showing integers from -9 to 9. The page also contains text explaining how to order integers and compare them.

Ejercicios

En esta sección, el usuario tendrá la posibilidad de realizar algunos ejemplos para afianzar conceptos y también, realizar ejercicios por tema:

The screenshot shows a web browser window with a header 'Aprendiendo números enteros | Aula Virtual' and a section titled 'Ejercicios'. There is a 'Volver' button in the top right. The main content lists exercises under two categories: 'Práctica con estos ejemplos:' and 'Ejercicios de Temas:'. The exercises include 'Ordenar números enteros', 'Calcula el Opuesto', 'Operaciones: Suma y Resta', 'Operaciones: Multiplicación y División', 'Introducción', 'Tema 1', 'Tema 2', 'Tema 3', and 'Ejercicio Completo'. The footer contains copyright information: '© 2017, Sandra García Sanz' and 'TFG: Estudio de los números enteros'.

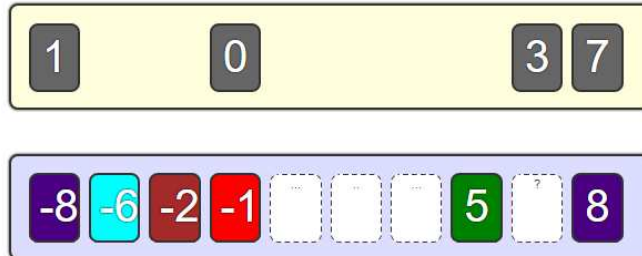
A continuación se muestra uno de los ejemplos, como se puede observar es intuitivo y fácil:

localhost/enteros/index.html#/ejercicios/ordenar

Aprendiendo números enteros | Aula Virtual

Ordena estos números de menor a mayor

Volver



Una vez finalizado el ejemplo, la aplicación muestra el resultado, ya sea correcto o erróneo. Además si es correcto pero hubo errores, también se indica:

localhost/enteros/index.html#/ejercicios/ordenar

Aprendiendo números enteros | Aula Virtual

Ordena estos números de menor a mayor

Volver



Si se hace click en el botón “Probar de nuevo”, se genera otro ejercicio con distintos datos para seguir practicando.

Una vez afianzado conceptos, mostramos uno de los ejercicios por tema (en este caso, del Tema 1).

Ejercicios de Temas:

Introducción

Tema 1 

Tema 2

Tema 3

Ejercicio Completo

Siempre tendremos un listado con todos los posibles ejercicios:

Aprendiendo números enteros | Teoría

Temario de la asignatura

[Volver](#)

Tema	Nombre	Consultar
Tema 1	Ordenación de números enteros	Consultar
Tema 1	Ordenación	Consultar

Haciendo click sobre Consultar, accedemos al ejercicio concreto:

- Siempre vamos a tener la posibilidad en todos los ejercicios de poder imprimirlos pulsando el botón “Imprimir”.
- Todos los ejercicios tiene comprobación automática mediante el botón “Comprobar”.
- Si nos hemos equivocado, podemos limpiar el cuadro de texto con el botón “Limpiar”.

localhost/enteros/index.html#/ejerciciosTemas/Tema_1

Aprendiendo números enteros | Teoría

[Volver](#)**Ejercicio: Ordenación de números enteros (Tema 1)**

Ordena de mayor a menor:

1. +6, -5, -10, +12
2. +4, -20, -7, -4

 Imprimir

Puedes comprobar los resultados aquí:

Solución 1 Correcto
Solución 2 Incorrecto

[Comprobar](#) [Limpiar](#)

✓ Has tenido 1 acierto

Aula virtual

En esta sección el usuario tendrá dos opciones (registrarse o iniciar sesión):

localhost/enteros/index.html#/aulaVirtual

Aprendiendo números enteros | Aula Virtual



Inicia sesión:

[Registrar](#) [Crear Cuenta](#) [Iniciar sesión](#)

[Acceder](#)

[Volver](#)

© 2017, Sandra García Sanz

TFG: Estudio de los números enteros

➤ Registrarse.

Pulsamos el botón “Crear Cuenta” y accederemos al siguiente formulario:



Nuevo Usuario:

Nombre

Apellidos

DNI

Tipo de Usuario Profesor
 Alumno


Correo Electrónico

Contraseña

* Todos los campos son obligatorios

- Si el usuario que se quiere registrar es un profesor, marcará en el tipo de usuario “profesor”.
 - El profesor debe rellenar los datos del formulario y pinchar sobre el botón “Crear Cuenta”:

Aprendiendo números enteros | Aula Virtual

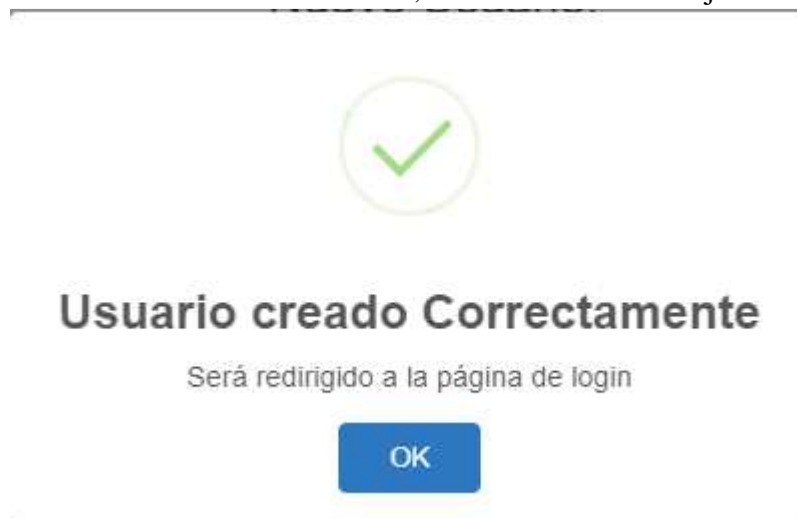


Nuevo Usuario:

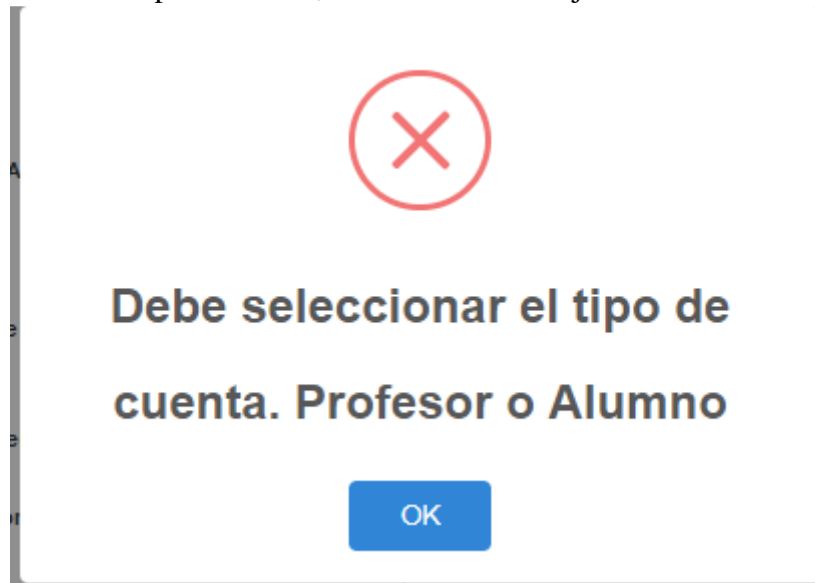
Nombre	<input type="text" value="Prueba Profesor"/>
Apellidos	<input type="text" value="Prueba Profesor"/>
DNI	<input type="text" value="70242424X"/>
Tipo de Usuario	<input checked="" type="radio"/> Profesor <input type="radio"/> Alumno
Correo Electrónico	<input type="text" value="prueba@profesor.com"/>
Contraseña	<input type="password" value="....."/>

* Todos los campos son obligatorios

- Si todos los datos son correctos, se mostrará un mensaje de éxito:

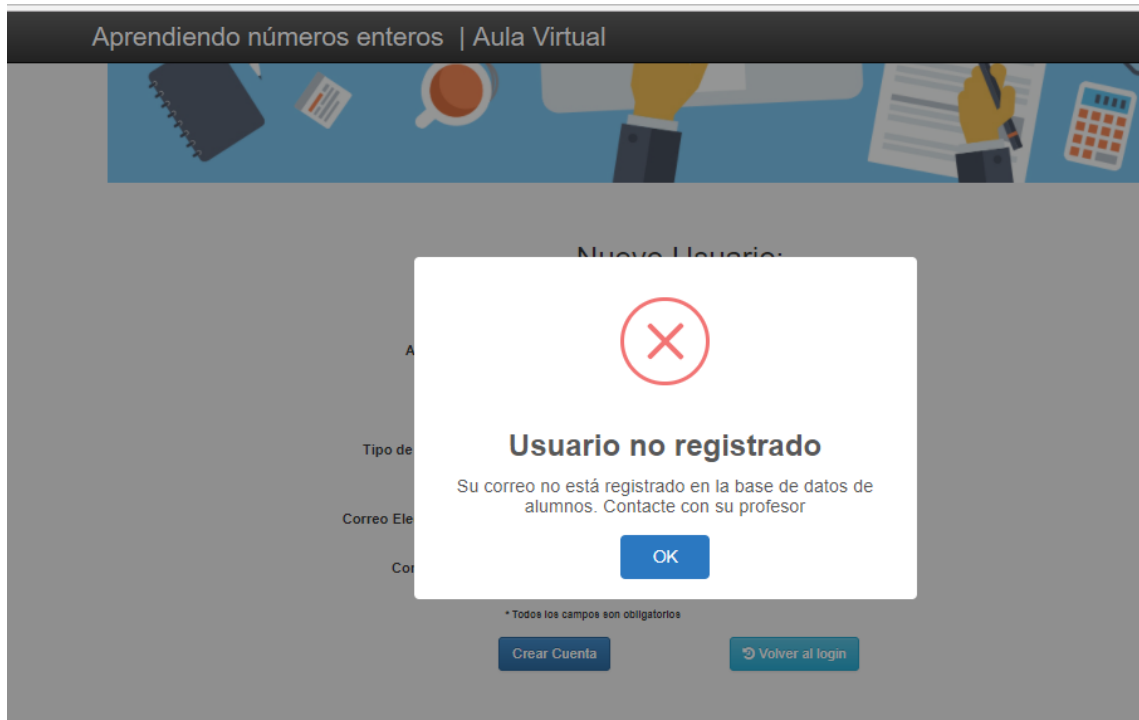


- Si algún dato no se rellena correctamente, por ejemplo si no se rellena el tipo de cuenta, mostrará un mensaje de error:

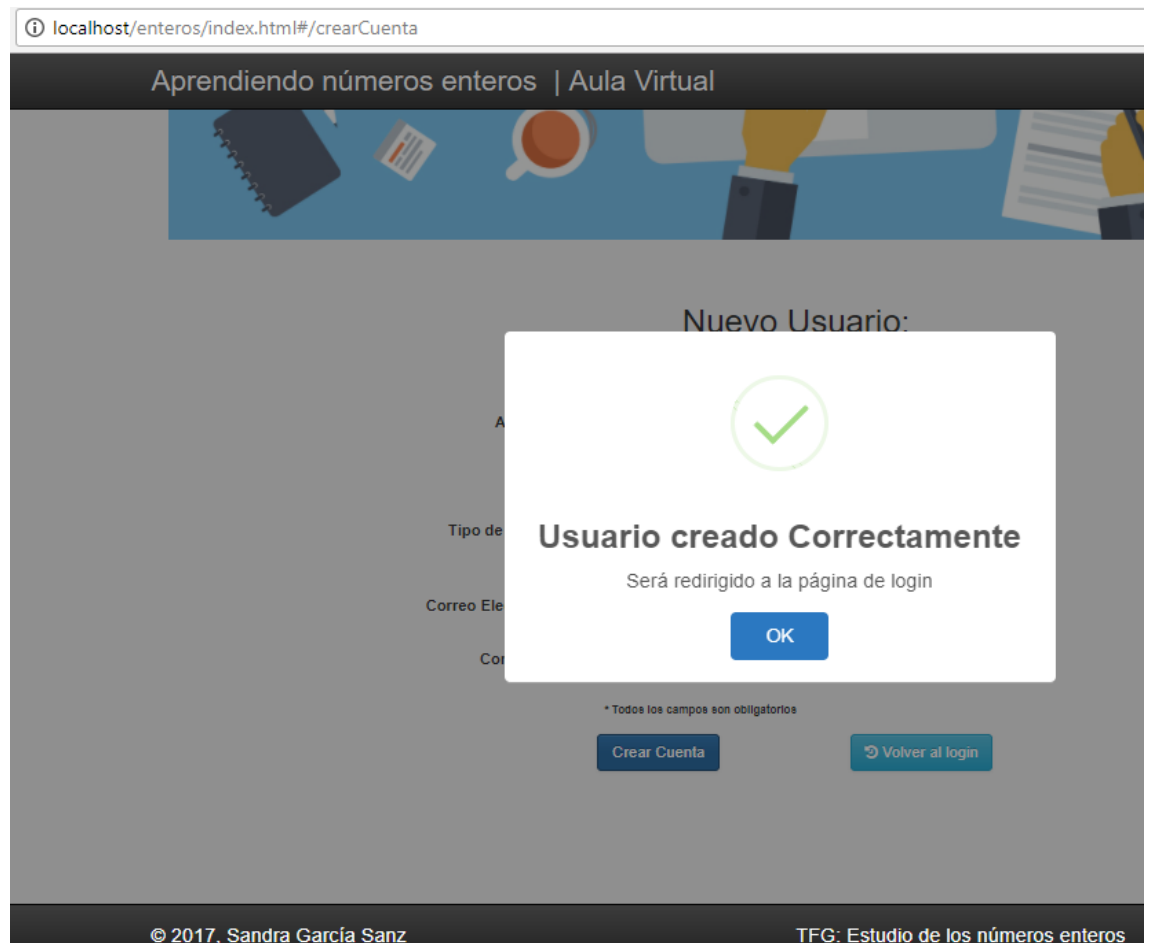


- Si el usuario que se quiere registrar es un alumno, marcará en el tipo de usuario “alumno”, además este alumno tiene que estar dado de alta en la lista de alumnos del profesor.
A continuación, se muestra cómo sería el registro de un alumno que no está dado de alta en el listado del profesor:
 - El alumno debe rellenar los datos del formulario y pinchar sobre el botón “Crear Cuenta”:

ocalhost/enteros/index.html#/crearCuenta



Si el alumno sí está dado de alta en el listado del profesor y todos los datos de registro son correctos:



➤ Iniciar sesión.

Una vez registrado el usuario, con sus credenciales (correo electrónico y contraseña) podrá acceder a la zona privada de cada tipo de usuario.

Si no se rellena algún dato, la aplicación mostrará mensajes informativos al usuario:



Menú Profesor

En esta sección, el profesor una vez logado accederá a su menú de administración:



Menú Profesor – Temario

El profesor va a poder gestionar todo lo relacionado con el temario: dar de alta un tema, modificarlo y eliminarlo.

Aprendiendo números enteros | Aula Virtual

Editor de Temario

Volver

Cerrar sesión

Nombre	Tema	Descripción breve	¿Visible?	
<input type="checkbox"/> Introducción	0	Introducción a los números enteros	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> Tema 1	1	Ordenar y comparar números enteros	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> Tema 2	2	Valor absoluto y Opuesto	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> Tema 3	3	Suma y diferencia de enteros	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> Tema 4	4	Producto y división de enteros	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> Tema 1 - Ejemplos	5	Ejemplos de Ordenación y comparación	<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> Tema 2 - Ejemplos	6	Ejemplos de valor absoluto y opuesto	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> Tema 3 - Ejemplos	7	Ejemplos de operaciones con números enteros	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>

Pág. 1 < Anterior > Siguiente Añadir Tema Borrar Tema

Guardar Cambios

Tanto para dar de alta un tema como para editor, se facilita un editor de texto en el que podrá incluir imágenes, tablas, fórmulas, cambiar el formato, etc:

[Guardar Cambios](#)

Editando tema: Tema 2

[Guardar Cambios](#)

Menú Profesor – Alumnos

En esta sección, el profesor va a poder realizar una gestión sobre los alumnos, podrá dar de alta un alumno, modificarlo y eliminarlo.

En la imagen se muestra cómo sería si el profesor quisiera dar de alta un alumno y eliminar otro.

Aprendiendo números enteros | Aula Virtual

Editor de Alumnos

[Volver](#) [Cerrar sesión](#)

Nombre	Apellidos	Contacto	Teléfono	¿Usuario activo?	
<input type="checkbox"/> Javier	López Riesgo	javier@gmail.com	666555666	<input checked="" type="checkbox"/>	✎ ✖
<input type="checkbox"/> Ramón	Freixa-Vicente	ramonFreixa88@gmail.com	914665562	<input type="checkbox"/>	✎ ✖ ✖
<input type="checkbox"/> Sandra	Alonso Ayuso	sandra.alonso.ayuso@hotmail.com	654789654	<input checked="" type="checkbox"/>	✎ ✖ ✖
<input type="checkbox"/>				<input type="checkbox"/>	✎ ✖ ✖

(*) Puede contactar con un alumno pulsando sobre su nombre

Pág. 1 [< Anterior](#) [> Siguiente](#) [Añadir Alumno](#) [Borrar Alumno](#)

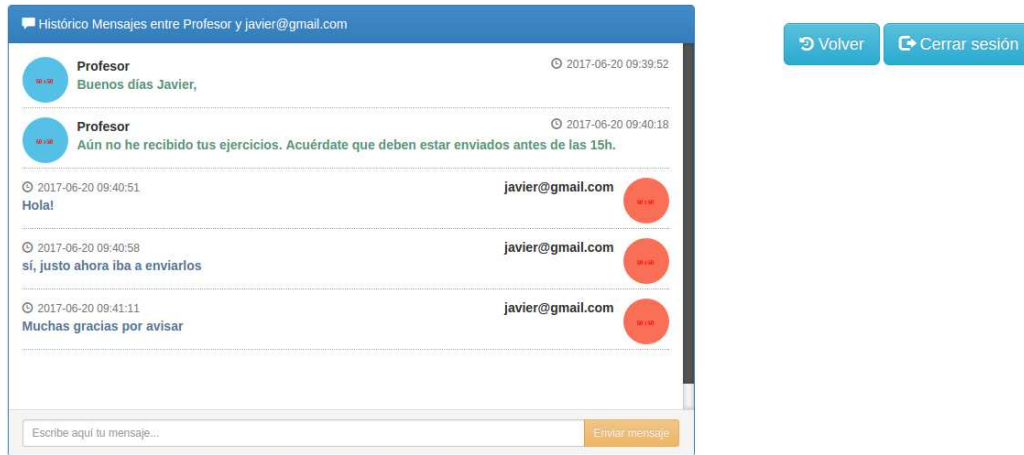
[Guardar Cambios](#)

[Añadir alumno](#)

[Eliminar](#)

Por otra parte, si el profesor pincha sobre el nombre de un alumno podrá contactar con él dejándole un mensaje:

Aprendiendo números enteros | Contacto



Menú Profesor – Ejercicios

Este apartado está dedicado a la gestión de ejercicios, el profesor podrá dar de alta ejercicios, modificarlos y eliminarlos.

Aprendiendo números enteros | Aula Virtual



➤ Editar ejercicio:

En una sola pantalla se va a tener la posibilidad de editar los ejercicios, por ejemplo, una solución errónea.

Aprendiendo números enteros

Volver

Cerrar sesión

Editor de Ejercicios

Nombre	Tema	1ª Solución	2ª Solución	3ª Solución	4ª Solución	¿Ejercicio visible?	
<input type="checkbox"/> Ejercicio introducción	Introducción	-3	-234	2455	-5	<input checked="" type="checkbox"/>	
<input type="checkbox"/> Ordenación de números enteros	Tema 1	12,6,-5,-10	4,-4,-7,-20			<input checked="" type="checkbox"/>	
<input type="checkbox"/> Primeros pasos	Introducción	-11				<input checked="" type="checkbox"/>	
<input type="checkbox"/> Ordenación	Tema 1	-6,-5,-4,-2,0,3,4,7,8	-22,-17,-17,-1,0,2,5,7,9	-9,-7,-5,-4,0,1,7,8,9,11	-17,-9,-8,-1,1,2,3,6,9	<input checked="" type="checkbox"/>	
<input type="checkbox"/> Valor absoluto	Tema 2	4,6,2,1,5,0,9				<input checked="" type="checkbox"/>	
<input type="checkbox"/> Opuestos	Tema 2	4,-6,2,-1,5,0,-9				<input checked="" type="checkbox"/>	
<input type="checkbox"/> Operaciones: Sumas y restas	Tema 3	2	12	-2		<input checked="" type="checkbox"/>	
<input type="checkbox"/> Ejercicio Completo	Ejercicio Completo					<input checked="" type="checkbox"/>	

Pág. 1 < Anterior > Siguiente

Guardar Cambios

➤ Eliminar ejercicio:

Volver

Cerrar sesión

Editor de Ejercicios ¿Hay cambios? No

Nombre	Tema	1ª Solución	2ª Solución	3ª Solución	4ª Solución	¿Ejercicio visible?	
<input type="checkbox"/> Ejercicio introducción	Introducción	-3	-234	2455	-5	<input checked="" type="checkbox"/>	
<input type="checkbox"/> Ordenación de números enteros	Tema 1	12,6,-5,-10	4,-4,-7,-20			<input checked="" type="checkbox"/>	
<input type="checkbox"/> Primeros pasos	Introducción	-11				<input checked="" type="checkbox"/>	
<input type="checkbox"/> Ordenación	Tema 1	-6,-5,-4,-2,0,3,4,7,8	-22,-17,-17,-1,0,2,5,7,9	-9,-7,-5,-4,0,1,7,8,9,11	-17,-9,-8,-1,1,2,3,6,9	<input checked="" type="checkbox"/>	
<input type="checkbox"/> Valor absoluto	Tema 2	4,6,2,1,5,0,9				<input checked="" type="checkbox"/>	
<input type="checkbox"/> Opuestos	Tema 2	4,-6,2,-1,5,0,-9				<input checked="" type="checkbox"/>	
<input type="checkbox"/> Operaciones: Sumas y restas	Tema 3	2	12	-2		<input checked="" type="checkbox"/>	
<input type="checkbox"/> Ejercicio Completo	Ejercicio Completo					<input checked="" type="checkbox"/>	

Pág. 1 < Anterior > Siguiente Borrar Tema

Guardar Cambios

Menú Profesor – Carga de Ficheros

El profesor tiene la posibilidad en esta sección de poder subir archivos a la aplicación, por ejemplo, ejercicios de apoyo. Puede elegir varios archivos de su carpeta local y después dar al botón “Subir archivos” para realizar la subida.



Menú Profesor – Generar Test

El profesor va a poder generar un test a partir de los ejercicios creados previamente.

Lo primero que se va a encontrar es con un formulario donde podrá dar un título, una fecha y una breve descripción al test:

Aprendiendo números enteros | Aula Virtual

Campos opcionales a completar

Fecha:

Nombre de la prueba:

Instrucciones de la prueba:

Puede enviar los ejercicios resueltos antes del día dd/mm/aaaa. Todos los ejercicios tienen la misma puntuación

Continuar

Pulsando el botón “Continuar”, tendrá la posibilidad de elegir los ejercicios que quiera para incluirlos en el test:

Aprendiendo números enteros | Aula Virtual

Selecciona los ejercicios que desea añadir al test

- ▣ Introducción: Ejercicio Ejercicio introducción
- ▣ Tema 1: Ejercicio Ordenación de números enteros
- ▣ Introducción: Ejercicio Primeros pasos
- ▣ Tema 1: Ejercicio Ordenación
- ▣ Tema 2: Ejercicio Valor absoluto
- ▣ Tema 2: Ejercicio Opuestos
- ▣ Tema 3: Ejercicio Operaciones: Sumas y restas
- ▣ Ejercicio Completo: Ejercicio Ejercicio Completo

Previsualizar

Guardar

Antes de guardar, el profesor puede hacer una primera visualización del test pulsando el botón “Previsualizar”. Se mostrará una ventana de este tipo:

Aprendiendo números enteros | Aula Virtual

Selecciona

- ▣ Introducción:
- ▣ Tema 1: Ejerc
- ▣ Introducción:
- ▣ Tema 1: Ejerc
- ▣ Tema 2: Ejerc
- ▣ Tema 2: Ejerc
- ▣ Tema 3: Ejerc
- ▣ Ejercicio Com

Previsualizar

Así quedará el test
✕

Puede enviar los ejercicios resueltos antes del día dd/mm/aaaa. Todos los ejercicios tienen la misma puntuación

Ejercicio 1: Ordenación (Tema 1)

Ordenar, en sentido creciente, representar gráficamente, y calcular los opuestos y valores absolutos de los siguientes números enteros:

- a) 8, -6, -5, 3, -2, 4, -4, 0, 7
- b) -15, 0, 5, -17, 9, 2, 7, -3, -1, -22
- c) 7, 8, -9, 0, -5, -7, -4, 1, -11
- d) 1, -1, 3, -8, 6, -9, 9, 2, -17

Solución: _____

Cerrar

Cerrar sesión

Si pulsa el botón “Guardar”, la aplicación mostrará un mensaje informativo:


Aprendiendo números enteros | Aula Virtual

Selecciona los ejercicios que desea añadir al test

[Volver](#)
[Cerrar sesión](#)

- Introducción: Ejercicio Ejercicio introducción
- Tema 1: Ejercicio Ordenación de números enteros
- Introducción: Ejercicio Primeros pasos
- Tema 1: Ejercicio Ordenación
- Tema 2: Ejercicio Valor absoluto
- Tema 2: Ejercicio Opuestos
- Tema 3: Ejercicio Operaciones:
- Ejercicio Completo: Ejercicio Ejercicio Completo

[Previsualizar](#)
[Guardar](#)



Test Guardado

Será redirigido al menú de administración

[OK](#)

Menú Profesor – Calificaciones

Esta sección está destinada para que el profesor añada notas a cada alumno. Podrá añadir una nota o varias, tendrá visible el histórico de notas así como la media de cada alumno.

Aprendiendo números enteros | Aula Virtual

[Volver](#)
[Cerrar sesión](#)

Alumnos	Consulta sus ficheros	Calificación	Histórico de Notas	Media
Herranz, Javier	Consultar	<input type="text"/>	<input type="text" value="5-5-8"/>	6.00
Lopez, Manuel	Consultar	<input type="text"/>	<input type="text" value="7-8-7"/>	7.33
Garcia, Luis	Consultar	<input type="text"/>	<input type="text" value="9-1"/>	5.00
Alonso, Sandra	Consultar	<input type="text"/>	<input type="text" value="7-8"/>	7.50
#####, #####	Consultar	<input type="text"/>	<input type="text"/>	

Las notas con decimales deben llevar un punto, ejemplo: 7.5

[Anterior](#)
[Siguiente](#)
[Guardar Notas](#)

Pulsando el botón “Guardar Notas”, la aplicación guardará dicha nota y mostrará un mensaje informativo:

The screenshot shows the 'Aprendiendo números enteros | Aula Virtual' interface. At the top right, there are buttons for 'Volver' and 'Cerrar sesión'. Below this is a table with columns: 'Alumnos', 'Consulta sus ficheros', 'Calificación', 'Histórico de Notas', and 'Media'. The table lists students: Herranz, Javier; Lopez, Manuel; Garcia, Luis; Alonso, Sandra; and a row with masked names. A modal dialog box is overlaid on the table, displaying a green checkmark icon and the text 'Notas guardadas Correctamente' with an 'OK' button. At the bottom of the interface, there are navigation buttons: 'Anterior', 'Siguiente', and 'Guardar Notas'.

Alumnos	Consulta sus ficheros	Calificación	Histórico de Notas	Media
Herranz, Javier	Consultar		5-5-8	6.00
Lopez, Manuel	Consultar		7-8-7	7.33
Garcia, Luis	Consultar			5.00
Alonso, Sandra	Consultar			7.50
██████████, ██████████	Consultar			NaN

Las notas con decimales deben llevar un punto, ejemplo: 7.5

Por otra parte, se le da la posibilidad al profesor de consultar los archivos que cada alumno han ido subiendo a la aplicación, por ejemplo, un test resuelto:

The screenshot shows the 'Aprendiendo números enteros | Aula Virtual' interface. At the top, it says 'Ficheros subidos por Sandra Alonso'. There are buttons for 'Volver' and 'Cerrar sesión'. Below this is a table with columns: 'Nombre del Fichero' and 'Descargar'. The table lists a file named 'documento Control.pdf' with a 'Descargar' button next to it.

Nombre del Fichero	Descargar
documento Control.pdf	Descargar

Menú Profesor – Página Principal

Si el profesor pincha en esta sección, le devuelve a la página principal.

Menú Alumno

En esta sección, el alumno registrado accederá a su menú:



Menú Alumno – Test

El alumno va a poder visualizar e imprimir si lo desea el test generado por el profesor.

Aprendiendo números enteros | Aula Virtual

Test Actual:

[Volver](#) [Cerrar sesión](#)

Puede enviar los ejercicios resueltos antes del día dd/mm/aaaa. Todos los ejercicios tienen la misma puntuación

Ejercicio 1: Ordenación (Tema 1)

Ordenar, en sentido creciente, representar gráficamente, y calcular los opuestos y valores absolutos de los siguientes números enteros:

- a) 8, -6, -5, 3, -2, 4, -4, 0, 7
- b) -15, 0, 5, -17, 9, 2, 7, -3, -1, -22
- c) 7, 8, -9, 0, -5, -7, -4, 1, -11
- d) 1, -1, 3, -8, 6, -9, 9, 2, -17

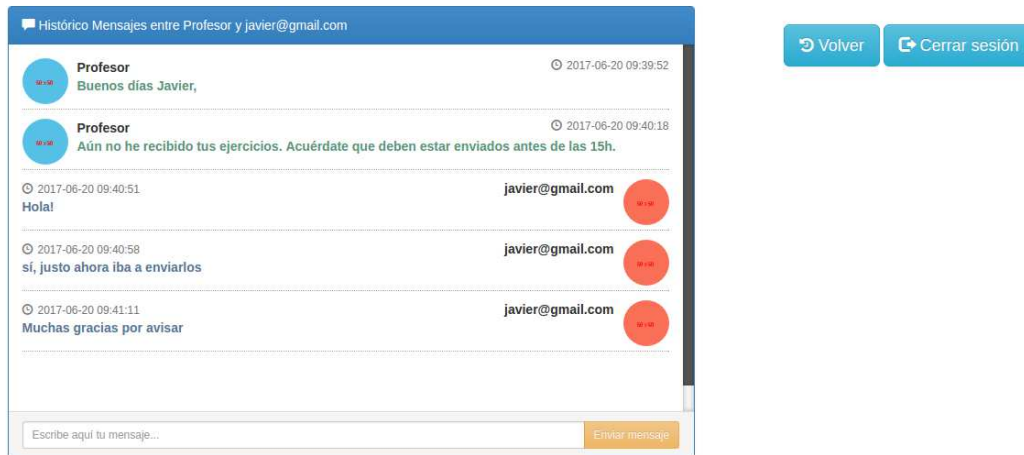
Solución: ____

[Imprimir](#)

Menú Alumno – Contactar con el profesor

Un alumno puede dejar un mensaje al profesor o leer si éste le ha dejado algún mensaje:

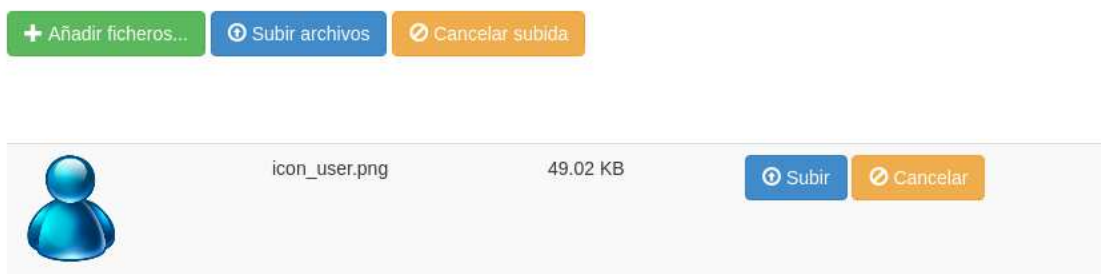
Aprendiendo números enteros | Contacto



Menú Alumno – Envía Fichero

Un alumno puede enviar o subir un fichero, esto es útil para cuando el profesor ha creado un test y tiene que entregarlo vía “Aula Virtual”.

El funcionamiento es el mismo que la subida de ficheros del menú de Profesor:



Menú Alumno – Consulta tus notas

Esta sección mostrará al alumno las notas disponibles que ha ido obteniendo hasta el momento.

En verde aparecerá la nota actual y en azul las notas del resto de ejercicios.

Aprendiendo números enteros | Aula Virtual



Última Nota: 9

Notas Anteriores:

1º Nota: 4

2º Nota: 6

Volver Cerrar sesión

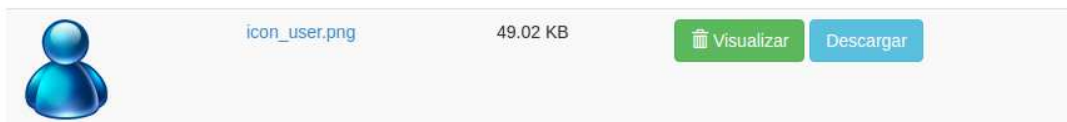
Menú Alumno – Descarga Ficheros

El alumno podrá descargar ficheros que el profesor ha subido anteriormente, por ejemplo una lista de ejercicios para reforzar.

Podrá visualizarlo o descargárselo:

Aprendiendo números enteros | Aula Virtual

Ficheros Disponibles:



icon_user.png 49.02 KB Visualizar Descargar

Menú Alumno – Página Principal

Si el alumno pincha en esta sección, le devuelve a la página principal.

7.4 Requisitos del sistema

Para que esta aplicación funcione correctamente se deben de cumplir una serie de requisitos mínimos. El rendimiento y ejecución van a depender en gran medida de la configuración hardware del equipo al que vaya destinada.

➤ Requisitos hardware

REQUISITOS MÍNIMOS	
Procesador	Intel, AMD...
Memoria	512 Mb o superior.
Disco Duro	20 Gb o superior.
Monitor	Resolución de al menos 1024x800px

Tabla 111: Requisitos mínimos hardware.

➤ Requisitos software

REQUISITOS MÍNIMOS	
Sistema operativo	Windows 7 o superior. Ubuntu 12.04 o superior

Tabla 112: Requisitos mínimos software.

➤ Resolución óptima

La página está optimizada para una resolución de 1360x768px, pese a ello se puede usar cualquier otra resolución.

➤ Navegador recomendado

La aplicación ha sido comprobada para los siguientes navegadores:

- Internet Explorer 11.
- Google Chrome 59.
- Mozilla Firefox 49.

NOTA: El navegador recomendado para ejecutar la aplicación es Google Chrome ya que cumple todas las operaciones a la perfección. El resto de navegadores son realmente competentes en casi todas las operaciones, pudiendo dar algún problema en el 1% de los casos.

BLOQUE IV
BIBLIOGRAFÍA,
WEBGRAFÍA
Y
ANEXOS

8. BIBLIOGRAFÍA

- **ANGULAR JS:**
 - Green, Brand: *AngularJS*. O'reilly, 2013.
- **PHP:**
 - Lockhart, Josh: *Modern PHP: New Features and Good Practices*. O'reilly, 2015.
 - Charte Ojeda, Francisco: *Proyectos Profesionales PHP5*. Anaya Multimedia, 2004.
- **Documentación de las asignaturas estudiadas en la Adaptación al Grado:**
 - Gestión de Proyectos basados en las TI.
 - Plataformas Software Empresariales.

9. WEBGRAFÍA

- **XAMPP:**
http://www.mclibre.org/consultar/php/otros/in_php_instalacion.html
- **PHP:**
<http://php.net/manual/es/index.php>
- **ANGUJAR JS:**
<https://docs.angularjs.org/guide>
- **JavaScript y HTML:**
<https://www.w3schools.com/>
- **Para resolver dudas a lo largo de la implementación:**
<https://stackoverflow.com/>
- **Ejemplos de Matemáticas:**
<http://www.edu365.cat/eso/muds/matematiques/edad/eso1/1quincena3/1quincena3.pdf>
http://www.vitutor.com/di/e/a_1.html
- **MySQL:**
<https://dev.mysql.com>

10. ANEXOS

10.1 Índice de tablas

- Tabla 1: Identificación de puntos de función no ajustados.
Tabla 2: Valores de evaluación de complejidad.
Tabla 3: Factores de complejidad.
Tabla 4: Equivalencia entre líneas de código y puntos de función.
Tabla 5: Constantes de COCOMO.
Tabla 6: Factores de ajuste de COCOMO.
Tabla 7: Relación interacción-peso COCOMO.
Tabla 8: Relación interacción-transacciones-peso COCOMO.
Tabla 9: Valoraciones posibles en cálculo del TCF COCOMO.
Tabla 10: Factores de complejidad técnica COCOMO.
Tabla 11: Factores de entorno COCOMO.
Tabla 12: Estimación por fases.
Tabla 13: Salarios trabajadores.
Tabla 14: Recursos Software / coste.
Tabla 15: Presupuesto total estimado.
Tabla 16: Salarios trabajadores.
Tabla 17: Presupuesto total real.
Tabla 18: Comparación LDC estimadas/reales.
Tabla 19: Objetivo 1 - Gestión de usuarios.
Tabla 20: Objetivo 2 - Gestión de teoría
Tabla 21: Objetivo 3 - Gestión de ejercicios.
Tabla 22: Objetivo 4 - Gestión de test.
Tabla 23: Objetivo 5 - Gestión de ficheros.
Tabla 24: Objetivo 6 - Gestión de calificaciones.
Tabla 25: Objetivo 7 - Gestión de alumnos.
Tabla 26: Objetivo 8 - Contacto entre alumnos y profesores.
Tabla 27: Requisitos de información de usuarios.
Tabla 28: Requisitos de información de teoría.
Tabla 29: Requisitos de información de ejercicios.
Tabla 30: Requisitos de información de test.
Tabla 31: Requisitos de información de ficheros.
Tabla 32: Restricción: unicidad de usuario.
Tabla 33: Restricción: validar alumno.
Tabla 34: Actor alumno no registrado.
Tabla 35: Actor alumno registrado.
Tabla 36: Actor profesor.
Tabla 37: Caso de Uso 1 – Registro.
Tabla 38: Caso de Uso 2 – Validar registro alumno
Tabla 39: Caso de Uso 3 – Login de usuario.
Tabla 40: Caso de Uso 4 – Logout de usuario.
Tabla 41: Caso de Uso 5 – Consultar tema.
Tabla 42: Caso de Uso 6 – Consultar temario.
Tabla 43: Caso de Uso 7 – Imprimir.
Tabla 44: Caso de Uso 8 – Consultar ejercicio.
Tabla 45: Caso de Uso 9 – Resolver ejercicio.
Tabla 46: Caso de Uso 10 – Crear tema.

- Tabla 47: Caso de Uso 11 – Editar tema.
Tabla 48: Caso de Uso 12 – Eliminar tema.
Tabla 49: Caso de Uso 13 – Crear alumno.
Tabla 50: Caso de Uso 14 – Editar alumno.
Tabla 51: Caso de Uso 15 – Eliminar alumno.
Tabla 52: Caso de Uso 16 – Contacto profesor - alumno.
Tabla 53: Caso de Uso 17 – Contacto alumno - profesor.
Tabla 54: Caso de Uso 18 – Crear ejercicio.
Tabla 55: Caso de Uso 19 – Editar ejercicio.
Tabla 56: Caso de Uso 20 – Eliminar ejercicio.
Tabla 57: Caso de Uso 21 – Generar test.
Tabla 58: Caso de Uso 22 – Añadir notas.
Tabla 59: Caso de Uso 23 – Descargar archivos de alumno.
Tabla 60: Caso de Uso 24 – Subir archivo de profesor.
Tabla 61: Caso de Uso 25 – Previsualizar ficheros.
Tabla 62: Caso de Uso 26 – Descargar ficheros subidos por profesor.
Tabla 63: Caso de Uso 27 – Subir fichero de alumno.
Tabla 64: Caso de Uso 28 – Eliminar fichero.
Tabla 65: Caso de Uso 29 – Consultar notas.
Tabla 66: Caso de Uso 30 – Consultar test.
Tabla 67: Requisito no funcional 1 – Disponibilidad.
Tabla 68: Requisito no funcional 2 – Mantenibilidad.
Tabla 69: Requisito no funcional 3 – Interoperabilidad.
Tabla 70: Requisito no funcional 4 – Interfaz.
Tabla 71: Requisito no funcional 5 – Usabilidad.
Tabla 72: Requisito no funcional 6 – Escalabilidad.
Tabla 73: Requisito no funcional 7 – Seguridad.
Tabla 74: Requisito no funcional 8 – Accesibilidad.
Tabla 75: Diccionario de datos: entidad usuarios.
Tabla 76: Diccionario de datos: entidad mensajes.
Tabla 77: Diccionario de datos: entidad ejercicios.
Tabla 78: Diseño de interfaz 1 – Estructura común.
Tabla 79: Diseño de interfaz 2 – Página principal.
Tabla 80: Diseño de interfaz 3 – Teoría.
Tabla 81: Diseño de interfaz 4 – Ejercicios.
Tabla 82: Diseño de interfaz 5 – Aula virtual.
Tabla 83: Diseño de interfaz 6 – Crear cuenta.
Tabla 84: Diseño de interfaz 7 – Menú profesor.
Tabla 85: Diseño de interfaz 8 – Menú alumno.
Tabla 86: Diseño de interfaz 9 – Menú profesor – edición temario.
Tabla 87: Diseño de interfaz 10 – Menú profesor – edición temario – añadir tema.
Tabla 88: Diseño de interfaz 11 – Menú profesor – edición ejercicios.
Tabla 89: Diseño de interfaz 12 – Menú profesor – edición ejercicios – añadir ejercicio.
Tabla 90: Diseño de interfaz 13 – Menú profesor – cargar ficheros.
Tabla 91: Diseño de interfaz 14 – Menú profesor – generar test.
Tabla 92: Diseño de interfaz 15 – Menú profesor – calificaciones.
Tabla 93: Diseño de interfaz 16 – Menú profesor – alumnos.
Tabla 94: Diseño de interfaz 17 – Contactar con el profesor.
Tabla 95: Caso de prueba 1 - Crear usuario.
Tabla 96: Caso de prueba 2 - Iniciar sesión en la aplicación.

- Tabla 97: Caso de prueba 3 - Cerrar sesión en la aplicación.
- Tabla 98: Caso de prueba 4 - Añadir tema.
- Tabla 99: Caso de prueba 5 - Modificar tema.
- Tabla 100: Caso de prueba 6 - Eliminar tema.
- Tabla 101: Caso de prueba 7 - Añadir ejercicio.
- Tabla 102: Caso de prueba 8 - Eliminar ejercicio.
- Tabla 103: Caso de prueba 9 - Modificar ejercicio.
- Tabla 104: Caso de prueba 10 - Añadir alumno.
- Tabla 105: Caso de prueba 11 - Eliminar alumno.
- Tabla 106: Caso de prueba 12 - Modificar alumno.
- Tabla 107: Caso de prueba 13 - Cargar/Enviar archivo.
- Tabla 108: Caso de prueba 14 - Generar test.
- Tabla 109: Caso de prueba 15 - Añadir calificaciones.
- Tabla 110: Caso de prueba 16 - Consulta de notas.
- Tabla 111: Requisitos mínimos hardware.
- Tabla 112: Requisitos mínimos software.

10.2 Glosario

SW:

Abreviatura de Software.

UC:

Abreviatura de Caso de Uso. Describen que es lo que el usuario debe hacer, y lo que el sistema debe responder.

UML:

Lenguaje Unificado de Modelado. Lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

UAW:

Factor de peso de los actores sin ajustar.

UUCW:

Factor de peso de los casos de uso sin ajustar.

UCP:

Puntos de caso de uso ajustados.

UUCP:

Puntos de caso de uso sin ajustar.

TCF:

Factor de complejidad técnica.

EF:

Factor de entorno.

HTML:

HyperText Markup Language («lenguaje de marcado de hipertexto»). Es un lenguaje para la elaboración de páginas web. Es usado para describir la estructura y contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

XML:

EXtensible Markup Language ('lenguaje de marcas extensible'). Da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información.

CSS:

Cascading Style Sheets u hojas de estilo en cascada. Es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.

BBDD:

Abreviatura de bases de datos.

PDF:

Abreviatura de portable document format (formato de documento portátil).

MD5:

MD5 es uno de los algoritmos de reducción criptográficos.

MySQL:

Sistema de gestión de bases de datos relacional, multihilo y multiusuario.

