

eman la zabal zazu



Universidad del País Vasco Euskal Herriko Unibertsitatea

BILBOKO INGENIARITZA ESKOLA ESCUELA DE INGENIERÍA DE BILBAO

INDUSTRIA INGENIARITZA TEKNIKOKO ATALA

SECCIÓN INGENIERÍA TÉCNICA INDUSTRIAL

--

FDO.: FECHA:	FDO.: FECHA:
-----------------	-----------------



Índice

1. Introducción	1
2. Planteamiento inicial	5
2.1. Definición del Proyecto	5
2.2. Arquitectura	6
2.3. Objetivos y funcionalidad	8
2.4. Alcance	9
2.5. Planificación temporal	14
2.6. Herramientas	24
2.7. Gestión de riesgos	27
2.8. Evaluación económica	35
3. Antecedentes	41
4. Análisis	47
4.1. Catálogo inicial de requisitos	47
4.1.1. Requisitos funcionales	47
4.1.2. Requisitos no funcionales	48
5. Diseño	51
5.1. Diagrama de casos de uso	51
5.2. Modelo de dominio	56
5.3. Diagrama de bases de datos	59
5.3.1. Diseño de maquetas	61
5.4. Diseño de pruebas	74
6. Desarrollo	77
6.1. Iteración 1	77



6.2. Iteración 2	79
6.3. Iteración 3	82
6.4. Iteración 4	84
6.5. Iteración 5	86
6.6. Iteración 6	88
6.7. Iteración 7	95
6.8. Iteración 8	96
7. Validación	98
7.1. Pruebas para la iteración 1	98
7.2. Pruebas para la iteración 2	99
7.3. Pruebas para la iteración 3	102
7.4. Pruebas para la iteración 4	103
7.5. Pruebas para la iteración 5	105
7.6. Pruebas para la iteración 6	106
7.7. Pruebas para la iteración 7	108
7.8. Pruebas para la iteración 8	108
8. Conclusiones	110
8.1. Gestión del proyecto	110
8.2. Diseño del proyecto	112
8.3. Trabajo futuro	114
8.4. Conclusión personal	116
9. Bibliografía	118



Índice de figuras

1.	Diagrama de la arquitectura del sistema	7
2.	EDT fase de planificación	14
3.	Detalle del diagrama de Gantt	20
4.	Detalle del diagrama de Gantt	21
5.	Detalle del diagrama de Gantt	22
6.	Detalle del diagrama de Gantt	23
7.	Detalle del diagrama de Gantt	23
8.	Logo software Creately	24
9.	Logo software Tom's Planner	25
10.	Logo software MoqUps	25
11.	Logo software Sublime Text 3	26
12.	Logo software XAMPP	26
13.	Logo software TeXstudio	27
14.	Figura con las prioridades de los riesgos	34
15.	Figura con las prioridades de los riesgos	34
16.	Detalle del menú de la página	42
17.	Detalle del juego de la página	42
18.	Detalle del juego de la página	43
19.	Detalle del juego de la página	44
20.	Detalle del juego de la página	44
21.	Detalle del juego de la página	45
22.	Detalle del juego de la página	46
23.	Actores que componen el sistema	52
24.	Diagrama de caso de uso relacionado al Usuario	53
25.	Diagrama de caso de uso relacionado al Profesor	54



26.	Diagrama de caso de uso relacionado al Alumno	55
27.	Diagrama de modelo de dominio	57
28.	Diagrama de Entidad-Relación	60
29.	Diagrama que define el flujo de pantallas	62
30.	Diseño de la pantalla de inicio	64
31.	Diseño de la pantalla de registro del profesor	65
32.	Diseño de la pantalla de registro del alumno	66
33.	Diseño de la pantalla de inicio de sesión	67
34.	Diseño de la pantalla de inicio del alumno	68
35.	Diseño de la pantalla de historial del alumno	69
36.	Diseño de la pantalla de juego del alumno	70
37.	Diseño de la pantalla de juego del alumno	71
38.	Diseño de la pantalla de inicio del profesor	72
39.	Diseño de la pantalla de confirmación de registros de alumnos del profesor	73
40.	Diseño de la pantalla de listado de alumnos del profesor	74
41.	Estructura de los ficheros dentro del proyecto	78
42.	Detalle del inicio de sesión en la interfaz de usuario	79
43.	Detalle del código	80
44.	Detalle del código	81
45.	Detalle del menú en la interfaz de usuario	81
46.	Detalle del menú en la interfaz de usuario	82
47.	Detalle de la página inicial de la interfaz de profesor	82
48.	Detalle de la página de información de alumno de la interfaz de profesor	83
49.	Detalle de la página de información de alumno de la interfaz de alumno	84



50. Detalle de la página de matriculación a un tema en la interfaz de alumno	85
51. Detalle de la página del listado de códigos en la interfaz del profesor	85
52. Detalle de la página de confirmación de registro de la interfaz de profesor	87
53. Detalle del cuadro de diálogo en la interfaz de profesor	87
54. Ejemplo documento con especificación QTI	89
55. Modificación de la base de datos	90
56. Detalle del código para procesar las preguntas de un tipo	91
57. Detalle del código para procesar las preguntas de un tipo	92
58. Detalle del tipo de pregunta 1	93
59. Detalle del tipo de pregunta 2	93
60. Detalle del tipo de pregunta 3	94
61. Interfaz del alumno al seleccionar un tema	95
62. Interfaz del alumno de su historial	97
63. Detalle del código	100
64. Detalle del cuadro de diálogo en la interfaz de usuario	100
65. Detalle del código	101
66. Detalle del cuadro de diálogo en la interfaz de usuario	102
67. Detalle del código	104
68. Detalle del cuadro de diálogo en la interfaz de alumno	104
69. Detalle del código	106
70. Gráfica comparativa entre horas estimadas y reales	111
71. Estructura de la base de datos definitiva	113
72. Diseño de la pantalla de Login	114
73. Pantalla definitiva de Login	114



Índice de tablas

1.	Diferencias Proyecto externo e interno	2
2.	Tabla estimación de tiempos	19
3.	Factores de probabilidad de los riesgos	30
4.	Factores de impacto de los riesgos	30
5.	Identificación de costes	36
6.	Cálculo de costes MOD	37
7.	Resumen estimación económica	39
8.	Tabla comparativa estudio de mercado	46
9.	Pruebas Inicio de sesión	99
10.	Pruebas para el registro del profesor	99
11.	Pruebas para el registro del alumno	101
12.	Pruebas para la consulta de alumnos por parte del profesor	102
13.	Pruebas para la consulta de historial por parte del alumno	103
14.	Pruebas para la matriculación a un tema por parte de un alumno	104
15.	Pruebas para la confirmación del registro de un alumno por parte de un profesor	105
16.	Pruebas para el correcto funcionamiento del juego	107
17.	Pruebas para el correcto funcionamiento de las calificaciones	109



1. Introducción

El objetivo de este proyecto es llevar a cabo el desarrollo software que solventará una aplicación web propuesta por una profesora. Con la intención de utilizarla como herramienta educativa, se ha presentado la propuesta de un juego basado en preguntas que permita a alumnos y profesores desarrollar unidades didácticas. Dado el carácter de este trabajo, se pretende llevar a cabo el proceso completo de la obtención del producto de la manera más rigurosa y cercana al mundo profesional.

Desde un punto de vista muy general, podemos diferenciar tres fases dentro de un proyecto profesional: fase de planificación, fase de ejecución y fase de entrega.

- **Fase de planificación:** en esta fase, se dota de método y estructura a las acciones conjuntas previas al desarrollo del producto. Define los tiempos y labores a llevar a cabo dentro del proceso de desarrollo, y en el caso de empresas con varios trabajadores, cuál es el perfil que debe desempeñarlo.
- **Fase de ejecución:** conjunto de tareas y actividades que suponen la realización del proyecto, es decir, la ejecución del producto. Responde a las características técnicas de cada tipo de proyecto y supone gestionar los recursos de los que se dispone de la forma adecuada para el desarrollo.
- **Fase de entrega:** para realizar la entrega al cliente y puesta en marcha del sistema, se deberá comprobar que funciona adecuadamente y responde a los requisitos definidos con el cliente en la fase de planificación del proyecto.

Estas etapas generales pueden presentar características muy diferentes según el tipo de proyecto. En este caso, diferenciaremos entre proyecto interno y externo, siendo el criterio de clasificación el tipo de cliente. Llamaremos proyecto interno aquellos en los que el cliente sea la propia empresa desarrolladora, es decir, no existe la necesidad de presentar una oferta a un cliente, dado que la adjudicación

no está en juego. En un proyecto externo, en cambio, deberán llevarse a cabo ciertas acciones para realizar una propuesta contundente al cliente, con objetivo de ser la empresa seleccionada para realizar el desarrollo. Las principales diferencias entre estos dos tipos de proyectos, aparecen en la etapa de planificación.

PROYECTO INTERNO		PROYECTO EXTERNO
1. Análisis del proyecto	FASE DE PLANIFICACIÓN	1. Elaboración de la oferta
2. Determinación de las opciones existentes		2. Adjudicación del contrato
3. Selección de la opción más conveniente para la empresa: formulación		3. Planificación detallada
4. Planificación detallada		
5. Desarrollo	FASE DE EJECUCIÓN	4. Desarrollo
	FASE DE ENTREGA	

Tabla 1: Diferencias Proyecto externo e interno

En este trabajo, simularemos un Proyecto externo, donde un cliente ajeno a la empresa hace una petición para cubrir una necesidad. En un ámbito laboral real, el primer paso es elaborar el Documento técnico para presentar la oferta del producto a dicho cliente y una vez la empresa es seleccionada para llevar a cabo el desarrollo, se realiza la planificación detallada del desarrollo. En este caso, tratándose de un ámbito académico, no es necesaria la elaboración del Documento Técnico dado que no existe un concurso real para la obtención del desarrollo.

En ese primer documento que se entrega al cliente, el Documento Técnico, se establecen la definición del proyecto, definición de los objetivos y una descripción más completa. Posteriormente, se definen los requisitos que el sistema deberá cumplir para una correcta implementación y desarrollo del proyecto. También se incluyen un mapa de navegación para que el cliente pueda familiarizarse con la interfaz del producto. Por último, para ayudar a los futuros usuarios a hacerse una idea lo más aproximada posible de la aplicación, se detallan las maquetas de las distintas



ventanas que compondrán el proyecto. De esta manera, el cliente podrá conocer de forma concisa la solución propuesta por la empresa y realizar correcciones sobre las funcionalidades o la navegación.

La creación de la propuesta será un proceso cíclico, ya que lo más habitual es que el cliente pida cambios. Una vez realizada la oferta y la presentación al cliente, si existe alguna solicitud de cambio, deberán revisarse los requisitos para cumplir la exigencia. En este proyecto, las definiciones que engloban en el Documento Técnico estarán reflejadas en los primeros apartados del documento.

Una vez el cliente haya aprobado la propuesta, se pasará a la **planificación** detallada para el desarrollo del proyecto. Esta fase incluye procesos de análisis como la planificación temporal del desarrollo, el desglose del coste económico o informe de riesgos.

Tras concluir la fase de planificación, procederemos a la **fase de elaboración** o ejecución. Con un proyecto totalmente centrado en la satisfacción del cliente, se aplicarán metodologías ágiles. De esta manera, organizaremos el desarrollo del proyecto en iteraciones incrementales que iremos mostrando al cliente para recibir su retroalimentación. Con la aplicación de esta metodología, lograremos introducir al cliente en el proceso, proporcionando un resultado completo sobre el producto final.



2. Planteamiento inicial

En este apartado se definirán todos aquellos aspectos necesarios para el análisis y desarrollo del proyecto, así como la estimación temporal y económica de todo el proceso. Se identificarán los objetivos y funcionalidades, la arquitectura, el alcance y las herramientas que serán utilizadas a lo largo de todas las fases del proyecto. De la misma manera, se llevará a cabo la gestión de riesgos que concretará un plan de contingencia.

2.1. Definición del Proyecto

El objetivo principal del proyecto es desarrollar un sistema de aprendizaje como herramienta de trabajo para un aula en la asignatura de Historia de 2º de la ESO. El cliente, en este caso, será una profesora de Historia y Geografía de secundaria. Se trata de un juego por fases conformadas por un número de preguntas a responder.

El sistema de registro de los alumnos para acceder a la aplicación, estará regulada por el profesor, de esta manera, evitaremos que usuarios ajenos a la clase accedan. Una vez registrados, el profesor tendrá acceso a la lista de sus alumnos y los progresos de cada uno, así como a la calificación obtenida. Dicha calificación vendrá definida más adelante, según las especificaciones del cliente. De la misma manera, el alumno deberá tener acceso a sus propios avances en el juego.

En cuanto al juego, estará compuesto por distintas fases que el alumno deberá superar. Cada fase contendrá un número de preguntas de distinta tipología -siempre relacionados al temario de la fase a la que pertenece- que el alumno deberá responder de forma correcta. Es condición necesaria que el alumno responda de manera correcta a todas las preguntas de la fase, para poder así superarla y acceder a la siguiente. De esta manera, un alumno podrá tener varios intentos de responder una única pregunta.

El sistema de gestión de las calificaciones estará definido en la propia aplicación, convirtiéndolo en un proceso automático. De esta manera logramos evitar incoherencias dentro de la información contenida sobre los alumnos. La fórmula para la obtención de la calificación tendrá en cuenta los intentos fallidos del alumno en cada pregunta.

Con el proyecto ya terminado, encontraremos una interfaz común a todos los usuarios en la que profesores y alumnos puedan registrarse e identificarse. Los perfiles de los profesores los gestionará el administrador de la base de datos. Una vez identificado, dependiendo del tipo de usuario que se trate, podrá realizar varias acciones:

- **Profesor:** una vez identificado, tendrá acceso a la lista de alumnos pendientes de confirmar que se han registrado a sus temarios. Tendrá acceso a otra lista de los alumnos ya confirmados, en la cual podrá acceder a cada alumno por separado para conocer su avance en el juego y su calificación hasta el momento.
- **Alumno:** los alumnos tendrán la opción de acceder a su propio historial para conocer sus avances en el juego, así como tendrán la opción de jugar aquellos temas a los que se han matriculado.

Cualquier usuario ajeno al sistema tendrá la opción de registrarse en cualquiera de las dos modalidades, aunque su identificación no tendrá validez si no existe una confirmación del perfil.

2.2. Arquitectura

La arquitectura que dará soporte a la aplicación, deberá basar su funcionamiento en un esquema de cliente-servidor, es decir, el proveedor de los recursos será el servidor y el demandante el cliente. La parte del cliente, se encarga de ofrecer una

interfaz entre el usuario y el resto del sistema, mientras que el servidor, maneja los distintos recursos, como la base de datos.

A lo largo del desarrollo inicial del proyecto, la herramienta XAMPP actuará como servidor de manera local, interpretando el código PHP y manejando la base de datos MySQL. El servidor web que incluye la herramienta es Apache.

Utilizando una arquitectura basada en la estructura cliente-servidor, el usuario se valdrá de la interfaz para realizar peticiones HTTP al servidor, como por ejemplo, conocer el historial de juego de un alumno concreto (acción realizada por un usuario de tipo *profesor*). El servidor, una vez recibida la petición, deberá procesarla y generar una respuesta para el cliente. Para poder resolver la petición, enviará la consulta SQL a la base de datos y ésta le enviará de vuelta el set de resultados. Dicho set será la respuesta a la petición HTTP inicial y la parte del cliente deberá encargarse de mostrarla en la interfaz.

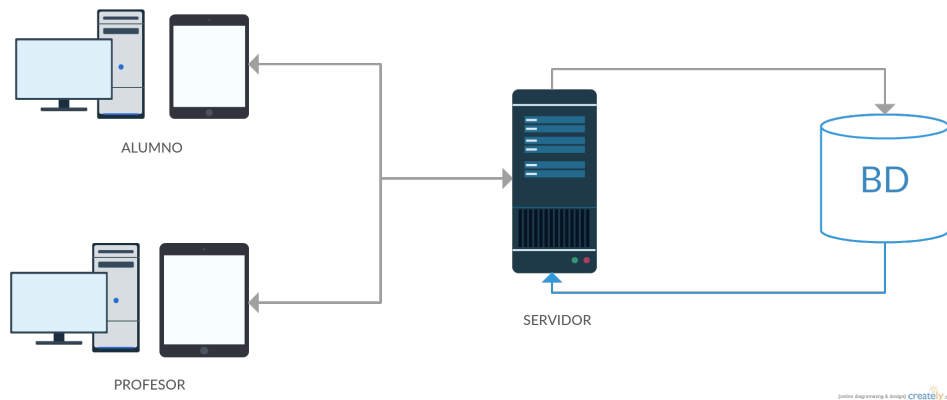


Figura 1: Diagrama de la arquitectura del sistema

Tal y como se muestra en la figura anterior, tanto el alumno como el profesor mantienen una misma comunicación con el servidor. El canal de comunicación con ambos es bidireccional para poder soportar tanto las peticiones del cliente al servidor, como las respuestas enviadas por éste. El canal de comunicación entre el

servidor y la base de datos es, de la misma manera, bidireccional.

2.3. Objetivos y funcionalidad

Una vez definido el grueso de la aplicación, pasamos a definir los objetivos y funcionalidades a cumplir por el sistema:

- Garantizar un servicio que sirva de refuerzo para la clase impartida y como herramienta de estudio
- Garantizar un acceso seguro de los alumnos, es decir, evitar usuarios ajenos a la clase
- Otorgar herramientas para la gestión del registro de los alumnos al profesor
- Poner a disponibilidad del profesor un sistema de seguimiento de sus alumnos
- De la misma manera, poner a disponibilidad del alumno sus propios avances en el juego
- Gestión automática de las calificaciones para denegar incoherencias en la información contenida

Una manera de simplificar los objetivos es la de clasificarlos por subsistemas. De esta manera transformamos los objetivos y funcionalidades en necesidades del sistema, facilitando así la posterior definición de requisitos. Con los objetivos identificados se han dividido las funcionalidades en tres subsistemas:

- **Sistema de usuarios:** define el acceso de cada tipo de usuario a la aplicación. Otorga al profesor el control en los registros de los alumnos.
- **Sistema de contenidos:** sistema gestor de los contenidos dentro de cada fase. Se encargará de gestionar la parte del juego.

- *Sistema de calificaciones*: almacena y gestiona las calificaciones de cada alumno, de modo que el profesor y el propio alumno puedan visualizarlas.

2.4. Alcance

Con fin de gestionar la complejidad del proyecto, la planificación se dividirá en diversos bloques temporales llamados iteraciones. Cada iteración solucionará algunos de los requisitos definidos posteriormente en la sección 4 Análisis, de manera incremental, obteniendo un prototipo del producto al finalizar dicha iteración. De esta manera, no quedará para el final del proyecto ningún cambio o modificación que resulte costosa para el desarrollo.

En cada iteración, el objetivo es evolucionar el prototipo obtenido de la iteración anterior, añadiendo algún requisito o mejorando alguno ya implementado. Tras la finalización del prototipo, se pasará a la fase de pruebas y documentación del mismo, concluyendo con una aprobación por parte del cliente.

Para aumentar la precisión de la definición del proyecto y ayudar en la organización del trabajo, se ha elaborado una Estructura de Desglose de Trabajo (EDT). Este tipo de mapas, define una descomposición jerárquica de las tareas a realizar desde el comienzo del proyecto, ayudando a definir el alcance total.

Siguiendo la identificación de fases generales de un proyecto, a continuación se definen todos los Paquetes de Trabajo genéricos que componen el diagrama EDT.

- Fase de planificación: engloba todos los paquetes de trabajo previos al desarrollo.
 - Gestión: éstos son los primeros pasos en el proyecto. En esta fase se comienza a realizar un esbozo de lo que queremos como proyecto, así como a informarse de las opciones existentes para la realización del mismo y los recursos disponibles.

- Reunión con el cliente: Tras las primeras reuniones iniciales para conocer las necesidades del cliente, se realizarán reuniones periódicas para obtener la validación del desarrollo. De esta manera evitaremos un error de concepto o requisito en una fase avanzada del proyecto, el cual causaría un gasto de tiempo y recursos para solucionarlo.
- Reunión con el tutor: De la misma manera que con el cliente, las reuniones periódicas con el tutor resultan ser necesarias para optimizar el desarrollo del proyecto, además de aquellas reuniones que puedan surgir para solucionar dudas concretas.
- DOP: Este primer documento recoge la definición básica del proyecto, todos aquellos aspectos necesarios para antes de comenzar con cualquiera de las fases del proyecto.
- Planificación: El siguiente paso después de la definición detallada del proyecto (DOP) es el de organizar temporalmente la evolución de las tareas del proyecto, es decir, obtener un tiempo estimado para cada tarea y conocer, mediante la suma de esas horas, el tiempo estimado para la finalización del trabajo.
- Análisis: en esta etapa se definirán las funcionalidades del producto junto al cliente. También quedarán reflejados los recursos escogidos para solventar las necesidades a lo largo del proyecto.
 - Definición de requisitos: Tras la reunión con el cliente para especificar las necesidades que deberá cubrir el sistema, deberá realizarse el documento con la definición formal de estos requisitos.
 - Análisis del contexto: Una vez conozcamos de manera detallada lo que el cliente necesita, deberemos hacer un estudio del contexto. Es decir, deberemos conocer más en profundidad el contenido que el cliente desea introducir en la aplicación así como las opciones de las



que disponemos para crear una solución.

- Selección de herramientas: Realizar un estudio sobre las herramientas que serán utilizadas en las distintas fases del proyecto ayudará en la selección de aquellas que más se ajusten a las necesidades del proyecto. Escoger la herramienta más potente y conocida no resulta ser siempre la mejor elección. El tiempo a invertir para saber obtener lo que necesitamos de esa herramienta puede resultar ser demasiado. Dedicar recursos para encontrar aquella herramienta que guarda el equilibrio entre tiempo de formación y necesidades, resulta ser una buena opción.
- Diseño: se especificará la estructura necesaria para el desarrollo del producto, así como el aspecto físico que tendrá. Junto a la definición de la estructura y maquetas, obtendremos una visión más real del proyecto dado que tenemos que cubrir todas las necesidades descritas en las funcionalidades. De esta manera, si surge la necesidad de realizar algún cambio bien en los requisitos o en la navegabilidad o aspecto de la aplicación, será menos costoso solucionarlo en esta etapa.
 - Diseño de la base de datos: Una vez definidas las necesidades del cliente, el primer paso es el diseño de la base de datos que deberá soportar los datos necesarios en la aplicación.
 - Diseño de arquitectura: En este apartado se definirá la comunicación del sistema, cómo funciona el flujo de datos entre los distintos componentes.
 - Diseño de maqueta: Esta tarea puede resultar un tanto irrelevante tratándose de una metodología ágil en la que el cliente valida cada prototipo del proyecto. Dedicar tiempo a una maqueta compleja en este caso no es viable, pero realizar una sencilla, en cambio, puede



darnos una idea general sobre el diseño gráfico de la interfaz que el cliente desea.

- Fase de ejecución: engloba los paquetes de trabajo relacionados con el desarrollo, desglosando la implementación en iteraciones.
 - Formación: es necesario adquirir práctica y conocimiento sobre las herramientas que se emplearán a lo largo del desarrollo. Aunque en etapas anteriores también se utilizarán herramientas, como por ejemplo en la gestión, la formación requerida para el uso de éstas no resulta significativo, por lo que no se han añadido más paquetes de formación.
 - Desarrollo: esta fase vendrá dividida en iteraciones. Cada iteración obtendrá como resultado un prototipo que cubrirá algunos de los requisitos definidos en la fase de análisis. A continuación se definen las iteraciones propuestas para el desarrollo, haciendo referencia al Requisito Funcional que pretenden solventar.
 - Iteración 1: [RF 4,5] inicio de sesión de cualquier usuario.
 - Iteración 2: [RF 1] registro de cualquier usuario.
 - Iteración 3: [RF 1.1] acceso por parte del profesor al listado de todos sus alumnos y sus avances en el juego.
 - Iteración 4: [RF 1.2] acceso por parte del alumno a sus propios avances.
 - Iteración 5: [RF 2,3] gestión del registro de los alumnos por parte del profesor.
 - Iteración 6: [RF 1.2] juego de preguntas al que tendrá acceso el alumno.
 - Iteración 7: [RF 4] poblar la base de datos. Esta iteración no tiene como objetivo un nuevo prototipo del producto, pero dada la carga

de trabajo que conlleva y la parte de pruebas correspondiente, se ha considerado como paquete de trabajo.

- Iteración 8: [RF 5,6,7] sistema de calificaciones.
- Pruebas: dada la metodología aplicada en la fase del desarrollo, esta fase deberá aplicarse varias veces. Cada prototipo realizado en cada iteración, deberá ser totalmente funcional, sin ningún error interno o de navegación. Por este motivo, se genera la necesidad de una fase de pruebas al finalizar cada una de las iteraciones, asegurando la calidad del producto y evitando costosas reparaciones en fases más tardías. El período de pruebas, implicará solucionar los errores que pudieran surgir.
- Fase de entrega: engloba todas aquellas tareas posteriores a la finalización del desarrollo.
 - Documentación: en esta última fase del proyecto, con la aplicación terminada y totalmente funcional, quedará terminar la memoria con los apartados posteriores al desarrollo y preparar la defensa del trabajo.

A continuación se muestra el diagrama EDT, que responde a la representación gráfica de los paquetes definidos previamente.

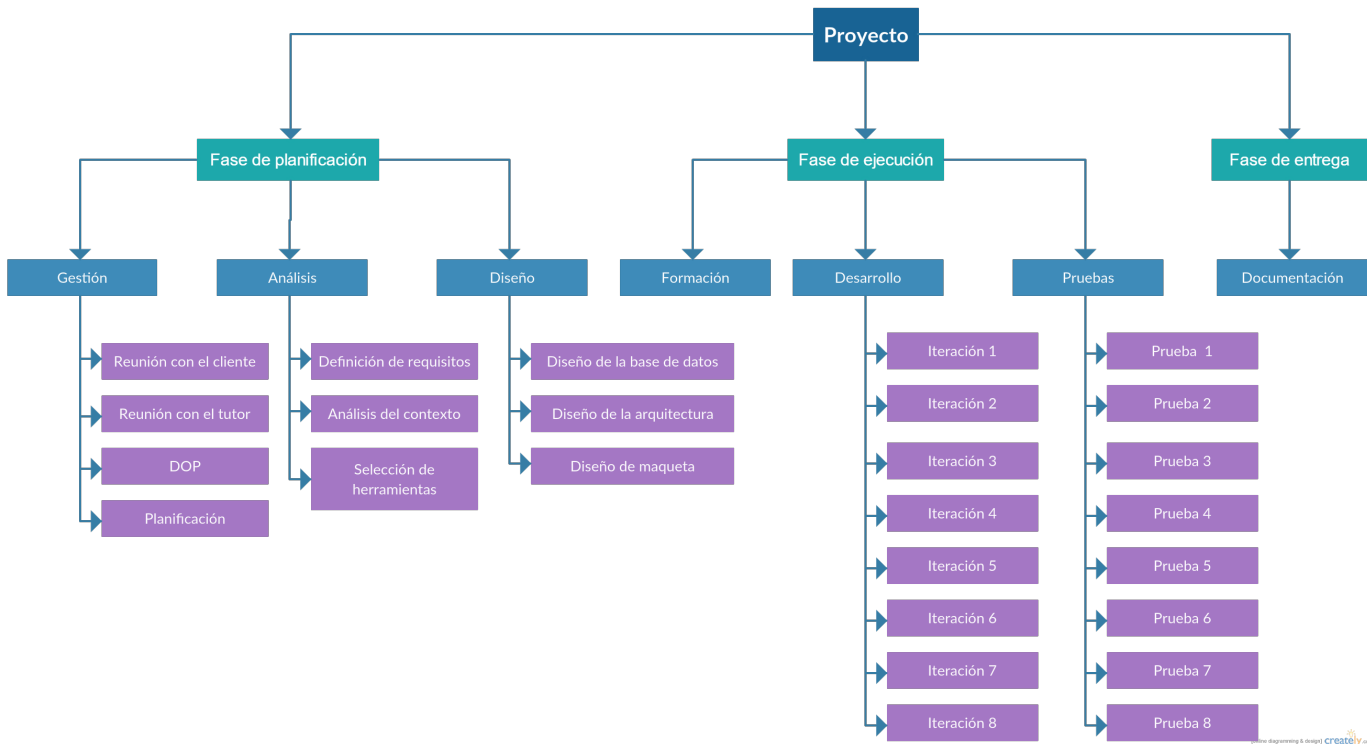


Figura 2: EDT fase de planificación

2.5. Planificación temporal

Una vez definidas de manera minuciosa las tareas a realizar a lo largo del proyecto, pasamos a asignar un tiempo a cada una de ellas y definir la secuencia de ejecución de manera que el tiempo de desarrollo sea el mínimo. En proyectos en los que existe un personal profesional cualificado para cada área (p.e. analistas, diseñadores, programadores junior, programadores senior etc) la planificación temporal se complica. Siendo el objetivo principal minimizar el tiempo estimado para llevar a cabo el proyecto, se deben tener en consideración técnicas como paralelizar tareas o asignar tareas más delicadas a los profesionales más cualificados. En este caso concreto, una única persona se encarga de desarrollar el proyecto comple-



to, simplificando la planificación. Todas las tareas deberán desarrollarse de manera secuencial y ordenada. Los hitos resultantes de las tareas serán aprobados por el tutor del proyecto y aquellos que correspondan, también por el cliente.

El primer paso para estimar el tiempo de cada tarea será definir el tiempo total a dedicar. Dado que el proyecto se trata de una asignatura de 12 créditos ECTS y el marco europeo establece que por cada crédito deberán dedicarse entre 25 y 30 horas, el tiempo a dedicar en este trabajo es de aproximadamente unas 300 horas. Tomando las tareas descritas en el apartado anterior, estimaremos un número de horas para la realización de cada una de ellas. Habrá de tomarse en consideración el desconocimiento de algunas de las herramientas que se utilizarán tanto en el diseño como en el desarrollo, así como las horas invertidas con el cliente en las validaciones de los prototipos.

A continuación se especifica el orden de desarrollo de cada tarea dentro del proyecto y en qué momento del proyecto se desarrollarán.

- Fase de planificación

- Gestión:

- Reunión con el cliente: Se deberán realizar varias reuniones con el cliente a lo largo del desarrollo del proyecto, contando con su validación para cada iteración. Se realizarán dos reuniones iniciales para definir el proyecto y los requisitos del sistema, y una tercera tras especificar los requisitos en la documentación. De esta manera, el cliente podrá realizar modificaciones sobre alguna posible mala interpretación.
- Reunión con el tutor: Tras una primera reunión con el tutor para la definición del proyecto, deberán realizarse más reuniones para hacer un seguimiento de la evolución, como por ejemplo, después

de acabar el DOP o el diseño de la base de datos. Es importante mantener al día estas reuniones, para poder evitar una grande carga de trabajo sobre el tutor y un retraso en el desarrollo.

- DOP: Dada la importancia de este primer documento requerirá la dedicación completa hasta su finalización.
- Planificación: Cuanto más completa la planificación la cabida a errores disminuye y la trazabilidad para conocer el origen de un error aumenta, por lo que resulta imprescindible dedicarle, al menos, una jornada.
- Análisis:
 - Definición de requisitos: Una vez reunidos los requisitos mediante una reunión con el cliente, la definición de éstos es importante. Tras la validación del documento de requisitos por parte del cliente, será la base para el desarrollo del proyecto.
 - Análisis del contexto: Dedicar tiempo para realizar un breve análisis sobre el contexto que rodea tanto al contenido como al tipo de producto antes de comenzar el desarrollo.
 - Selección de herramientas: Antes de proceder a realizar alguna de las tareas que requiere de una herramienta software, se dedicará tiempo a analizar los distintos productos en el mercado, para poder escoger con criterio la más adecuada para el trabajo a desempeñar.
- Diseño:
 - Diseño de la base de datos: El diseño inicial de la base de datos se realizará al comienzo de la fase de diseño, teniendo en cuenta los requisitos que deberá cumplir y los datos que deberá soportar la aplicación. Este diseño podría variar en algún momento del desarrollo.

- Diseño de arquitectura: El diseño de las comunicaciones del sistema se realiza al comienzo de la fase de diseño y habitualmente no cambia. Dada la naturaleza de este proyecto es poco probable que la arquitectura deba sufrir alguna modificación con respecto a su diseño inicial.
- Diseño de maqueta: Es importante recalcar en la importancia de la sencillez de la maqueta. En este caso concreto, el objetivo es darle una guía aproximada al cliente del aspecto que tendrá la aplicación. El diseño de la maqueta será el antecesor del primer prototipo generado en la fase de desarrollo.
- Fase de ejecución
 - Formación: La formación se dará en varios puntos del proyecto. Una vez escogidas las herramientas a utilizar (Fase de análisis. Selección de herramientas) y tengamos la necesidad de usarlas, se deberá emplear tiempo en conocerlas y aprender su uso para cubrir las necesidades que tengamos en ese punto del proyecto.
 - Desarrollo: El desarrollo se divide en Iteraciones, por lo que después de emplear el tiempo necesario en generar el prototipo en cada una, deberá realizarse una reunión con el cliente para obtener su validación.
 - Pruebas: De la misma manera, antes de contar con la opinión y validación del cliente, será necesario probar en cada iteración su prototipo.
- Fase de entrega
 - Documentación: A lo largo de todo el proyecto se irá desarrollando la memoria del proyecto, documentando cada tarea realizada hasta el momento.



Después de especificar el momento de desarrollo de cada tarea, procedemos a estimar, en horas, el tiempo a dedicar a cada una de ellas. Para cada tarea, se ha considerado el conocimiento (o desconocimiento) para el desarrollo, estimando un tiempo menor a aquellas que se presupone no existirá ningún problema en implementar. Una vez definidas las horas, comprobamos que el total no excede demasiado lo establecido para el desarrollo de 12 créditos ECTS. A continuación se muestra la estimación de horas para cada tarea.



<i>Tarea</i>		<i>Tiempo estimado (horas)</i>	
<i>Fase de planificación</i>	<i>Gestión</i>	<i>Reunión con el cliente</i>	8
		<i>Reunión con el tutor</i>	6
		<i>DOP</i>	6
		<i>Planificación</i>	4
	<i>Análisis</i>	<i>Definición de requisitos</i>	4
		<i>Análisis del contexto</i>	1
		<i>Selección de herramientas</i>	3
	<i>Diseño</i>	<i>Diseño de la base de datos</i>	2
		<i>Diseño de la arquitectura</i>	1
<i>Diseño de maqueta</i>		5	
<i>Fase de ejecución</i>	<i>Formación</i>	<i>Formación</i>	20
	<i>Desarrollo</i>	<i>Iteración 1</i>	10
		<i>Iteración 2</i>	20
		<i>Iteración 3</i>	20
		<i>Iteración 4</i>	10
		<i>Iteración 5</i>	35
		<i>Iteración 6</i>	40
		<i>Iteración 7</i>	20
		<i>Iteración 8</i>	35
	<i>Pruebas</i>	<i>Prueba 1</i>	6
		<i>Prueba 2</i>	6
		<i>Prueba 3</i>	6
		<i>Prueba 4</i>	6
		<i>Prueba 5</i>	6
		<i>Prueba 6</i>	6
<i>Prueba 7</i>		6	
<i>Prueba 8</i>		6	
<i>Fase de entrega</i>	<i>Documentación</i>	<i>Documentación</i>	60
TOTAL horas estimadas		358	

Tabla 2: Tabla estimación de tiempos

Una vez definidas las horas a dedicar en cada tarea, mostraremos de forma gráfica (1) el tiempo de dedicación previsto. Para ello, se utilizará el diagrama de Gantt, herramienta ampliamente extendida en el ámbito de la planificación de proyectos. Tomaremos como fecha de inicio la primera reunión para la definición del proyecto: 14 de febrero del 2017. A partir de ahí, con las horas ya estimadas para

cada tarea, se organizarán de manera secuencial para poder estimar el tiempo total para la finalización del proyecto. Si fuera necesario, se podría realizar algún ajuste en las horas estimadas para lograr el objetivo de entrega inicial: 5 de junio del 2017.

No se tendrán en cuenta los roles o puestos que deberán desempeñar cada tarea ya que no existe personal cualificado específicamente en cada área.

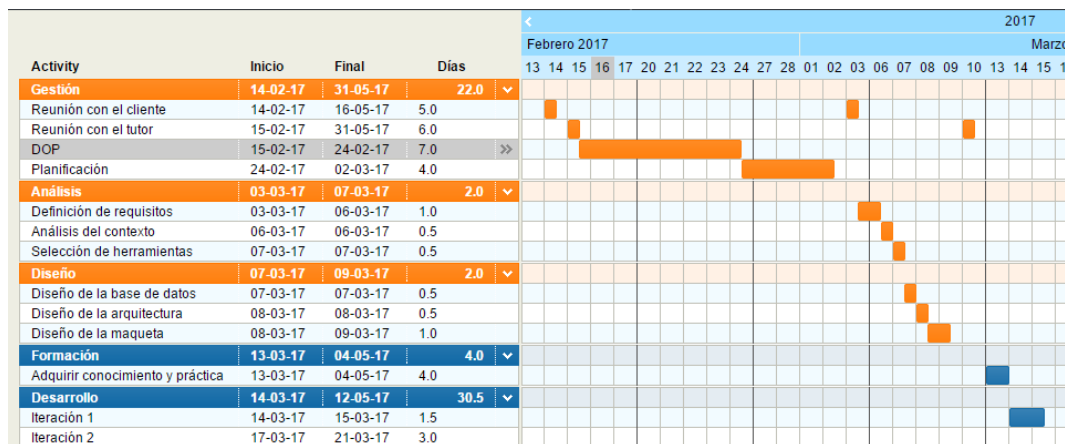


Figura 3: Detalle del diagrama de Gantt

En este primer detalle del diagrama de Gantt (Figura 3) se aprecian las jornadas dedicadas a la primera fase del proyecto: fase de planificación. Las tareas se ejecutan de manera secuencial, a excepción de las reuniones con el cliente y el tutor. Es importante recalcar la importancia de las validaciones después de un avance significativo. De no emplear tiempo en comprobar con el cliente los requisitos definidos, podríamos encontrarnos más adelante con un error difícil de corregir, dado el estado avanzado del proyecto o los recursos a emplear para solucionarlo.

Una vez finalizada la etapa de planificación, comienza la del desarrollo. Tal y como se ha comentado previamente, tras cada iteración en el desarrollo se desempeñará una tarea de prueba. En aquellas iteraciones en las que sea necesaria una tarea de formación previa, se añadirá una jornada dedicada a ello. De esta manera, se podrá conocer más en profundidad la herramienta o reto a solucionar antes de

comenzar el desarrollo.

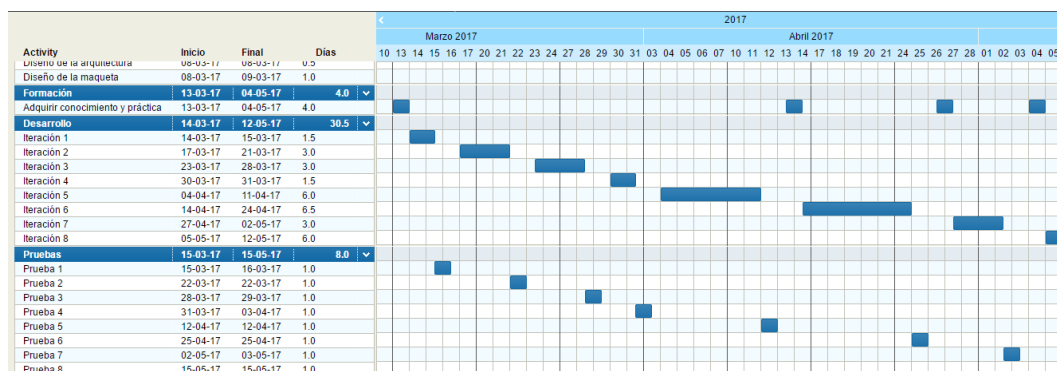


Figura 4: Detalle del diagrama de Gantt

Las horas estimadas para cada iteración y prueba, queda reflejado en este segundo detalle del diagrama (Figura 4). En esta etapa del proyecto, se han considerado necesarias cuatro tareas de formación:

- Tarea de formación previa a la primera iteración: en esta jornada se prevé un trabajo preliminar al desarrollo para obtener una idea sobre la implementación general de la parte software.
- Tarea de formación previa a la sexta iteración: esta iteración pertenece al juego en sí. Se ha previsto la necesidad de un trabajo previo al desarrollo para obtener una idea de la mejor opción. En esta tarea se pretende realizar un estudio sobre las opciones existentes y llevar alguna a la práctica para tener un criterio de elección realista.
- Tarea de formación previa a la séptima iteración: el objetivo en esta iteración es poblar la base de datos, por lo que se requerirá un estudio previo sobre las opciones existentes para hacerlo. Dado que se trata de un juego de preguntas, existen diferentes maneras de almacenar la información para después mostrarla y obtener la respuesta del jugador. Por este motivo, una jornada para



obtener información sobre las distintas opciones y poder escoger una, resulta ser de utilidad.

- Tarea de formación previa a la octava iteración: en la última iteración el objetivo es desarrollar el sistema de calificación del juego. Dedicar una tarea de formación previa para la solución de este requisito, ahorrará tiempo de desarrollo y ayudará a no cometer errores por desconocimiento en un estado ya avanzado del proyecto.

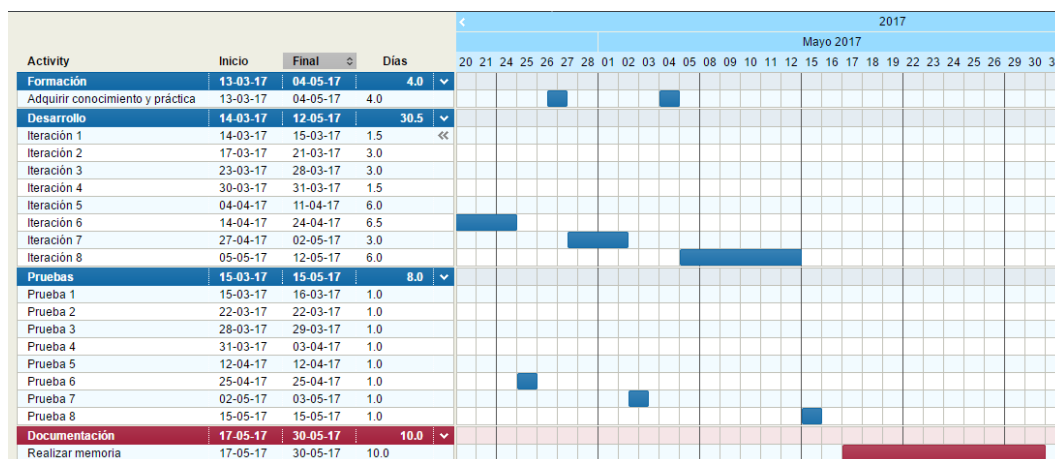


Figura 5: Detalle del diagrama de Gantt

Como se puede observar en este detalle del diagrama (Figura 5), las horas estimadas para realizar la documentación se han añadido al final del proyecto. El motivo por el cual se ha decidido insertar de esta manera la tarea de documentación, es el siguiente: aunque a lo largo del desarrollo se documentarán las tareas como las de prueba o formación, no se empleará un tiempo significativo para reflejarlo en el diagrama.

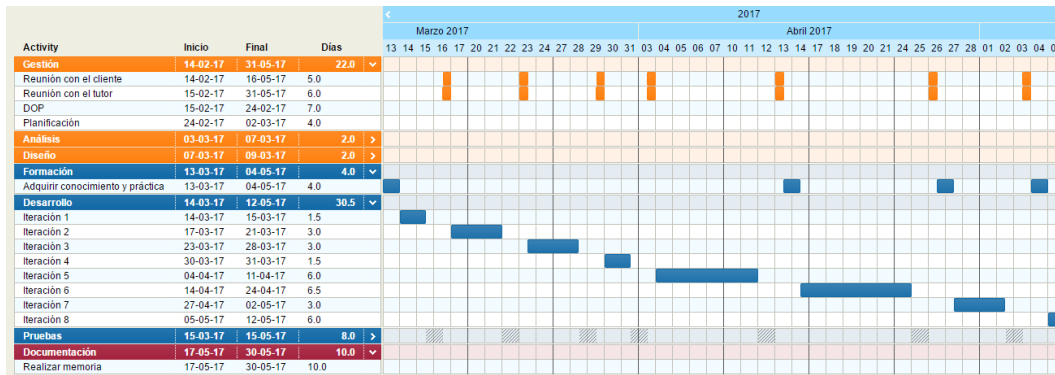


Figura 6: Detalle del diagrama de Gantt

Tras la finalización de cada iteración y su posterior tarea de pruebas, se realizará una reunión tanto con el tutor como con el cliente. En el diagrama, se han reflejado como la misma tarea, dado que es imposible prever la fecha exacta en la que sucederán dichas reuniones ya que están sujetas a la disponibilidad de terceras personas.



Figura 7: Detalle del diagrama de Gantt

Tras la finalización de la documentación se ha añadido una última tarea de reunión con el tutor (Figura 7). Con una perspectiva optimista, se ha añadido una única reunión tras la finalización de la memoria. De todas formas, con cada avance

significativo de la misma, se obtendrá un feedback del tutor con correcciones. De esta manera, evitaremos tanto al tutor como al alumno, una corrección costosa al final del proyecto.

La fecha prevista para la finalización del proyecto es el 31 de mayo del 2017, un mes antes de la fecha de entrega del proyecto. Ese margen resulta adecuado para correcciones y posibles retrasos temporales en el proyecto.

2.6. Herramientas

A lo largo del proyecto, deberán utilizarse distintas herramientas para facilitar las tareas a realizar. A continuación se enumeran dichas herramientas escogidas para cada fase del proyecto, una breve explicación de su utilidad y en caso de haber barajado otras opciones, el motivo por el cual ha sido escogida.

- Fase de planificación
 - **Creately** (2): herramienta online y gratuita para realizar cualquier tipo de diagrama. En una cuenta de google, permite almacenar un número limitado de diagramas. Contiene plantillas de muchos tipos de diagramas de los que partir, desde diagramas de flujo, hasta diagramas organizacionales. Aporta mucha libertad a la hora de construirlos y tiene una interfaz muy fluida.



Figura 8: Logo software Creately



- **Tom's planner (3):** se trata de un software online para crear diagramas de Gantt. De manera gratuita, permite crear diagramas de manera sencilla e intuitiva. Ha resultado ser una buena alternativa a programas mas toscos y de pago más tradicionales, idóneo para proyectos pequeños y sencillos para los que no se requiere un nivel de complejidad alto en este tipo de diagramas.



Figura 9: Logo software Tom's Planner

- **MoqUps (4):** se trata de un software online gratuito para crear maquetas de la interfaz del proyecto. Permite generar todo tipo de maquetas interactivas e incluso funcionales.



Figura 10: Logo software MoqUps

- Fase de ejecución

- **Sublime Text 3** (5): editor de texto para código gratuito y multiplataforma. Es una opción personal para editar el código de la web. Contiene un etiquetado sencillo para comprender a simple vista e interpreta muchos lenguajes de programación.



Figura 11: Logo software Sublime Text 3

- **XAMPP** (6): software gratuito que contiene un paquete de aplicaciones que permiten al usuario trabajar de manera local sin un servidor contratado. Permite trabajar la página web y alojar una base de datos MySQL de manera local, sin necesidad de alojarla en un servidor externo. Para este proyecto, de las aplicaciones que ofrece el software, se han utilizado el servidor web Apache, el sistema de gestión de bases de datos MySQL y el intérprete de lenguaje script PHP.



Figura 12: Logo software XAMPP

- **Google Chrome** (7): es el navegador escogido para la ejecución del código. Según las estadísticas ofrecidas por *Stat Counter Global Stats* el nave-

gador más utilizado en España en los últimos dos años es el creado por Google. Escoger adecuadamente la plataforma sobre la que se diseñará la página web, es importante para un código limpio, compatibilidad y funcionalidad de la misma.

■ Fase de entrega

- **TeXstudio (8):** herramienta gratuita para la edición de documentos. Dado el carácter de este proyecto y teniendo en cuenta las opciones que aporta esta herramienta, se ha concluido que resulta una buena opción para el desarrollo de la documentación. El diseño de la aplicación resulta un tanto complicado al principio, pero una vez se comprende lo básico para comenzar, la herramienta te permite infinidad de opciones. Aunque en diseño y usabilidad otras herramientas como Office lo superan, para documentos extensos como éste, el resultado final es más limpio y con un aspecto más técnico.

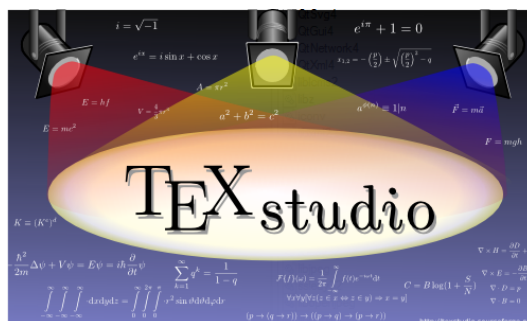


Figura 13: Logo software TeXstudio

2.7. Gestión de riesgos

Para la identificación de cada uno de los riesgos, se ha realizado un análisis al plan del proyecto. Una vez definidos, se especificará un plan para la gestión de dichos riesgos (9).

Los riesgos están agrupados por categorías y todos ellos siguen el mismo formato de enumeración:

[R X]

Donde X es el identificador numérico que corresponda.

1. Riesgos de usuario

R 1 Cambio de requisitos. Es posible que a lo largo de la duración del proyecto, el cliente decida añadir, modificar o eliminar alguno de los requisitos definidos en la fase de análisis.

2. Riesgos de los miembros (en este caso miembro) del equipo

R 2 Error en la planificación debido a la poca experiencia. Podría suceder que la poca experiencia frente a este tipo de proyectos más complejos se retrase alguno de los plazos de entrega, debido a que el tiempo invertido en la formación o investigación de algún sistema en concreto podría aumentar.

R 3 Enfermedad. Dado que el personal se compone de un único individuo, el hecho de sufrir una enfermedad pararía por completo el desarrollo de la fase en la que se encontrase el proyecto.

3. Riesgos organizativos

R 4 Sobresfuerzo. El proyecto no es la única tarea a realizar, por lo que algún imprevisto o retraso en alguna de esas otras tareas, repercutirá en el proyecto de manera negativa.

R 5 Desarrollo desorganizado. Este riesgo refleja la posibilidad de no cumplir la planificación establecida. No respetar el orden de tareas fijado, podría suponer un retraso en el avance general del proyecto.

R 6 Productividad del personal. Volviendo a la desventaja de que una única persona trabaje en el proyecto, podría darse una falta de motivación que afectase al rendimiento. Dependiendo del momento en el que se diese este efecto, podría producirse algún retraso en la duración del proyecto.

4. Riesgos de gestión del proyecto

R 7 Limitación del presupuesto. En este caso, no existe un presupuesto real adjudicado al proyecto. De todas maneras, más adelante se definirá la evaluación económica por lo que es necesario añadir este riesgo. Es probable que el cliente haga una petición de reducción de costes tras revisar el Documento Técnico.

R 8 Limitación del tiempo. Existe una fecha límite para este proyecto, pero podría requerirse un recorte en el tiempo de entrega por alguna otra circunstancia o exigencia.

5. Riesgos por causas externas

R 9 Fallos en las estaciones de trabajo. Es posible que el hardware o software utilizado no funcionen de manera correcta durante un periodo de tiempo o incluso de manera irreversible. Esto causaría un retraso en el tiempo estimado.

R 10 Descoordinación con el cliente. Una de las desventajas de la metodología escogida (desarrollo por iteraciones incrementales) es la necesidad de reunión con el cliente para obtener su aprobación en cada prototipo. La dependencia de la disponibilidad con el cliente supone un riesgo que podría afectar al correcto desarrollo y tiempos estimados del proyecto.

Estimación de los riesgos

Una vez identificados los riesgos que podrían afectar al rendimiento, es necesario estimar de qué manera y con qué probabilidad podrían suceder. Para poder cuantificar de forma numérica la prioridad de cada riesgo teniendo en cuenta su probabilidad de ocurrencia y su impacto en caso de suceder, definiremos unas tablas con coeficientes que asignaremos, uno a uno, a cada riesgo.

PROBABILIDAD	PORCENTAJE
Muy baja	0.1-0.2
Baja	0.3-0.4
Media	0.5-0.6
Alta	0.7-0.8
Muy alta	0.9

Tabla 3: Factores de probabilidad de los riesgos

IMPACTO	ESCALA
Muy bajo	0.1
Bajo	0.2
Medio	0.3
Alto	0.4
Muy alto	0.8

Tabla 4: Factores de impacto de los riesgos

Una vez definidos los coeficientes para clasificar un riesgo por su probabilidad de ocurrencia y su impacto, procederemos a asignarlos.

R 1 Cambio de requisitos.

Probabilidad: baja. El cliente ha revisado y validado el Documento Técnico con la especificación de requisitos, por lo que no se considera que éstos vuel-



van a cambiar. De todas formas, cabe la posibilidad de alguna revisión de los requisitos en la fase de desarrollo, a petición personal del cliente.

Impacto: muy alto. Un cambio de requisitos conllevaría una revisión total del proyecto, teniendo que volver a realizar las estimaciones tanto temporales como económicas.

R 2 Error en la planificación debido a la poca experiencia.

Probabilidad: muy alta. La falta de experiencia en este tipo de proyectos supone un gran riesgo para la ejecución de algunas de las tareas del proyecto.

Impacto: medio. Podría suponer un retraso en el tiempo pero no un problema trascendental en el desarrollo del proyecto.

R 3 Enfermedad.

Probabilidad: media. Es muy probable que a lo largo de los meses que componen el proyecto alguna enfermedad baje el rendimiento o incluso pare el desarrollo.

Impacto: alto. La baja por enfermedad causaría un parón completo del proyecto durante un período de tiempo. El retraso en la planificación supondría un reajuste de tiempos para poder llegar al objetivo temporal planteado.

R 4 Sobresfuerzo.

Probabilidad: medio. La carga externa al proyecto es moderada por lo que podría afectar al tiempo estimado para el desarrollo.

Impacto: alto. El hecho de no realizar de manera correcta las funciones correspondientes, no sólo supondría un retraso, si no que sería necesario una revisión del trabajo realizado en ese periodo de tiempo.

R 5 Desarrollo desorganizado.



Probabilidad: baja. La planificación ha sido minuciosa y el riesgo de que exista una descoordinación con una única persona implicada es bajo.

Impacto: medio. La desorganización en el desarrollo supondría un retraso en el tiempo, pero no generaría un problema trascendental.

R 6 Productividad del personal.

Probabilidad: media. Este riesgo hace referencia a los aspectos menos técnicos y por lo tanto menos predecibles.

Impacto: alto. La planificación no se estaría cumpliendo por lo que debería reajustarse y supondría un gasto de tiempo y recursos no contemplado.

R 7 Limitación del presupuesto.

Probabilidad: muy baja. En un proyecto real, el riesgo de la alteración del presupuesto es muy alto, pero en este caso no existe. De todas maneras, existe una estimación económica de lo que podría costar este proyecto.

Impacto: muy bajo. De nuevo, en un proyecto real la limitación de presupuesto generaría un impacto muy alto.

R 8 Limitación del tiempo.

Probabilidad: alta. La duración depende de diferentes circunstancias sujetas a posibles cambios con una probabilidad alta. Hay que tener en cuenta que cualquier problema inesperado en alguna de las tareas, afectará de manera negativa atrasando el resto de tareas consecutivas.

Impacto: muy alto. Existe una fecha establecida para la realización del proyecto. La planificación temporal se ha realizado ajustándose a dicha fecha y un aumento de tiempo o retraso no sería viable.

R 9 Fallos en las estaciones de trabajo.



Probabilidad: baja. Al tratarse de un aspecto técnico y al ser un problema temporal, no se contempla como un riesgo relevante, pudiendo paralelizar el trabajo mientras se da solución al problema.

Impacto: bajo. El tiempo estimado para solventar los problemas técnicos que pudieran causarse es bajo, por lo que podría reestructurarse el desarrollo sin más esfuerzo que cambiar de estación de trabajo.

R 10 Descoordinación con el cliente.

Probabilidad: muy alta. La dependencia con la disponibilidad del cliente es muy alta por lo que el riesgo de que alguna validación se retrase es elevado, causando un retraso general en el desarrollo.

Impacto: alto. Las pérdidas de tiempo que podrían generarse por este tipo de descoordinaciones afectarían de manera significativa al tiempo estimado.

Evaluación de los riesgos

Una vez definida la probabilidad y el impacto de cada riesgo, pasamos a clasificarlos por prioridad, de manera que podremos poner especial atención en aquellos más prioritarios.

Para asignar las prioridades se ha seguido la siguiente fórmula:

$$PRIORIDAD = PROBABILIDAD * IMPACTO$$

A continuación se muestra la tabla con las prioridades calculadas.

Código	Nombre	Probabilidad de ocurrencia	Impacto	Prioridad
R1	Cambio de requisitos	0,4	0,8	0,32
R2	Poca experiencia	0,9	0,3	0,27
R3	Enfermedad	0,5	0,6	0,3
R4	Sobresfuerzo	0,5	0,6	0,3
R5	Desarrollo desorganizado	0,4	0,3	0,12
R6	Productividad del personal	0,6	0,7	0,42
R7	Limitación del presupuesto	0,1	0,1	0,01
R8	Limitación del tiempo	0,7	0,8	0,56
R9	Fallos en las estaciones de trabajo	0,3	0,2	0,06
R10	Descoordinación con el cliente	0,9	0,5	0,45

Figura 14: Figura con las prioridades de los riesgos

De esta manera, somos capaces de priorizar los riesgos y generar así puntos de ruptura a lo largo del proyecto. En la *Figura 15*, el listado de los riesgos ordenador por la prioridad calculada.

Código
R8
R10
R6
R1
R3
R4
R2
R5
R9
R7

Figura 15: Figura con las prioridades de los riesgos

Una vez evaluados los riesgos, se establecerían tanto acciones preventivas como correctivas para intentar minimizarlos. Las acciones preventivas corresponden a aquellas medidas tomadas para procurar evitar los riesgos y las correctivas a aquellas medidas que minimizarían el impacto de los riesgos una vez sucedidos. El documento que recoge todas aquellas acciones correctivas toma el nombre de Plan de contingencia.

En este proyecto, dedicar recursos a la gestión de riesgos supondría un gasto de tiempo poco útil. En este punto, tras haber clasificado los riesgos por prioridad,

se deberían definir los puntos de ruptura dentro del proyecto, es decir, nivel en el que el rendimiento, exceso de coste, dificultades de soporte o retrasos en la planificación obligan a tomar medidas o abandonar el proyecto. Dado el carácter de este proyecto, cualquiera de los riesgos es susceptible a ocurrir en cualquier punto del desarrollo, por lo que las buenas prácticas y un riguroso seguimiento de la planificación inicial los evitarán. Para aquellos riesgos ajenos al proyecto (enfermedad, productividad etc.) que conllevarían un retraso en la entrega, existe un margen temporal antes de la fecha límite, por lo que se han tomado en consideración y su impacto no sería demasiado alto.

2.8. Evaluación económica

Realizar la evaluación económica del proyecto, permitirá determinar cuáles son sus costes directos e indirectos, definir de manera relativamente exacta los posibles beneficios, y en definitiva, si es rentable desarrollar el proyecto o no. En este caso concreto, hacer el estudio de la viabilidad o rentabilidad del proyecto, resulta innecesario. Tratándose de un proyecto social, es decir, que busca promover una mejora de ámbito social (en este caso en el ámbito educacional), no es de interés analizar métodos para rentabilizar el coste de desarrollo, pero sí analizar los costes de implementación.

El primer paso para realizar la evaluación económica es el de identificar todos los costes que participan en el proyecto (10). En función de su relación con el producto, podemos clasificar los costes de la siguiente manera:

- Costes directos: estos costes se asocian al producto final de manera directa y se pueden determinar con exactitud la cantidad que va al producto. En este tipo de costes encontramos la materia prima consumida para la construcción del producto (en este caso, al tratarse de un proyecto software, no existen este tipo de costes) y la Mano de Obra Directa (MOD).



- **Costes indirectos:** se trata de aquellos costes que no pueden ser directamente relacionados con el producto, o al menos, no de manera exclusiva. En esta clasificación, encontraríamos las licencias de los softwares utilizados en el desarrollo o los gastos de luz, dado que no existe un número concreto de coste asociado al proyecto. Para poder conocer el gasto que conlleva nuestro proyecto sobre estos costes globales, emplearemos métodos de reparto para obtener una estimación aproximada.

Los costes identificados para este proyecto han sido los siguientes:

Coste	Tipo de coste
<i>Mano de Obra Directa (MOD)</i>	<i>Directo</i>
<i>Equipo informático</i>	<i>Indirecto</i>
<i>Costes energéticos</i>	<i>Indirecto</i>
<i>Costes de licencias</i>	<i>Indirecto</i>

Tabla 5: Identificación de costes

Una vez identificados y clasificados, pasamos a definirlos y cuantificarlos.

1. Costes directos

Mano de Obra Directa (MOD)

En el ámbito en el que nos encontramos, calcular el coste generado por la mano de obra resulta sencillo. No existen diferentes rangos ni puestos entre el personal, por lo que el sueldo medio del trabajador es único. En este caso, estimaremos el sueldo en 10 euros/hora trabajada, dado que se trata de un único desarrollador junior. Para realizar una estimación un poco más cercana a la realidad, haremos una deducción de las horas de formación, que deberán correr a cargo del programador. Los datos necesarios para calcular el coste derivado de la MOD son los siguientes:

Tiempo estimado para el desarrollo total del proyecto	<i>358 horas</i>
Tiempo estimado para la formación	<i>20 horas</i>
Coste por hora	<i>20 euros/hora</i>

Tabla 6: Cálculo de costes MOD

De esta manera, aplicando un par de fórmulas, obtenemos el coste:

- Deducción de horas de formación:

$$\begin{aligned}
 \text{Horas a facturar} &= 358 \text{ (Horas totales)} - 20 \text{ (Horas dedicadas a la formación)} = \\
 &= 338 \text{ horas}
 \end{aligned}$$

- Total del coste:

$$\text{Total coste} = 338 \text{ (Horas a facturar)} * 20 \text{ euros/hora (salario)} = \mathbf{6760 \text{ euros}}$$

2. Costes indirectos

Equipo informático

Características del equipo portátil:

- Fabricante: ASUS
- Procesador: Intel Core i7 con procesador CPU a 2.3 GHz
- Memoria RAM: 8 GB
- Valor de compra: 900 euros
- Fecha de compra: diciembre del 2012

Para poder estimar el coste semidirecto derivado del equipo, es decir, hacer un reparto del coste global indirecto para obtener una cifra aproximada, necesitamos definir el tiempo de vida útil que le daremos al dispositivo y el porcentaje en el que hacemos uso de él en el periodo de desarrollo. En este



caso, se estima que el ordenador dure unos 5 años y el uso aproximado que se hará será del 60 % .

Con estos datos definidos, el cálculo del gasto generado por el equipo se realiza de la siguiente manera:

$$\text{Amortización mensual equipo informático} = 900 \text{ euros (valor inicial de compra)} / 60 \text{ meses (vida útil)} = 15 \text{ euros / mes}$$

$$\text{Coste equipo informático} = (15 \text{ euros/mes (amortización mensual equipo informático)} * 3.5 \text{ meses (duración del proyecto)}) * 0.6 \text{ (uso del equipo)} = 31.5 \text{ euros}$$

Una vez hecho el reparto, obtenemos que el coste generado por el uso del equipo informático es de 31.5 euros.

Costes energéticos

Los costes energéticos derivados del proyecto vendrán del gasto de luz necesario para poner en funcionamiento el equipo informático. Para hacer una estimación de este gasto, utilizaremos un método similar al empleado con el equipo informático. Definiremos el coste indirecto global y el uso que hacemos del recurso. Marcaremos la media de gasto mensual de luz en 60 euros y el porcentaje de uso que hacemos de éste en un 5 % .

De la misma manera que para calcular en coste del equipo informático, mediante la fórmula de reparto se obtendrá el coste energético.

$$\text{Coste energético} = (60 \text{ euros/mes (amortización mensual)} * 3.5 \text{ meses (duración del proyecto)}) * 0.05 \text{ (uso de la energía)} = 10.5 \text{ euros}$$

Una vez aplicada la fórmula en la que se tiene en consideración la amortización mensual y el porcentaje de uso que hacemos del recurso, obtenemos que el coste energético estimado para la duración del proyecto: 10.5 euros.



Costes de licencias

Una vez definidas las herramientas escogidas para el desarrollo del proyecto, ninguna de ellas tiene coste alguno. De esta manera, reducimos el gasto por licencias de software, amoldando la implementación a aquellas herramientas de software libre o con cuentas limitadas.

Una vez definidos los costes individuales, la suma de éstos será la estimación económica total del desarrollo del proyecto, recogida en el siguiente tabla.

Descripción del coste	Coste
<i>Mano de obra (MOD)</i>	6760 euros
<i>Equipo informático</i>	31.5 euros
<i>Costes energéticos</i>	10.5 euros
<i>Licencias</i>	-
TOTAL	6802 euros

Tabla 7: Resumen estimación económica

Como podemos observar, el coste total del desarrollo del proyecto, desde su inicio con las reuniones, hasta el final con la entrega de la memoria, es de 6802 euros.



3. Antecedentes

Dado que el proyecto ha sido definido por el cliente no existe una herramienta o software concreto del que partir. De todos modos, antes de comenzar con el desarrollo e incluso el análisis, conviene realizar un estudio de mercado para conocer aquellos productos similares y obtener así, un idea sobre lo que queremos y no queremos en nuestro proyecto.

A continuación se describen tres productos de mercado con una relación cercana a este proyecto.

TESTEANDO (11)

Se trata de una página web (<http://www.testeando.es/>) con un amplio repertorio de preguntas. Las preguntas se encuentran clasificadas por curso y materia. Es necesario registrarse en la página para poder acceder a las diferentes baterías de preguntas, siendo el registro de pago. Cada conjunto de preguntas respondidas obtendrá una calificación, dependiendo del número de respuestas erróneas.

- Puntos positivos
 - Interfaz intuitiva
 - Amplio repertorio de preguntas y temarios
 - Posibilidad de obtener los datos de calificación en distintos formatos
- Puntos negativos
 - Plataforma de pago
 - No describe el sistema de calificaciones
 - La manera de obtener las calificaciones no es intuitiva. La interfaz no es accesible para esta funcionalidad.



Figura 16: Detalle del menú de la página

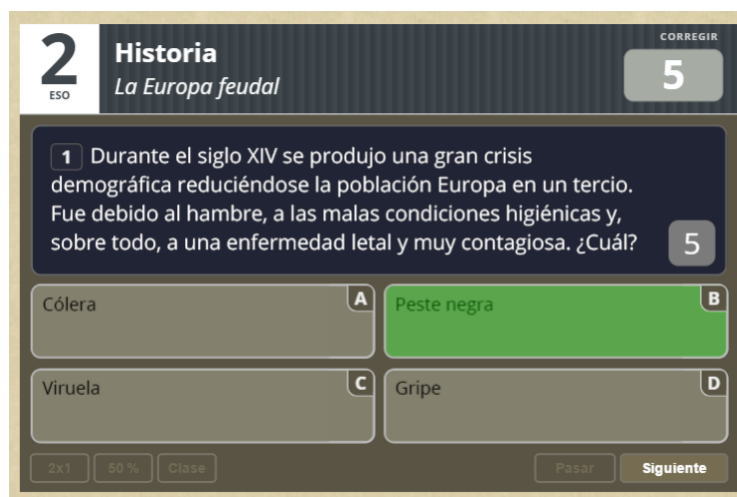


Figura 17: Detalle del juego de la página



Figura 18: Detalle del juego de la página

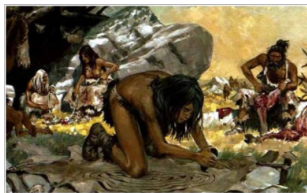
SOPASLETRAS (12)

Esta página web (<http://www.sopasletras.com/>) contiene varios tipos de juego, entre ellos, juegos de preguntas. Dentro de esta categoría, clasifica las tandas de preguntas por temas: deportes, cultura general, historia, informática etc. Dentro de cada tema, existen módulos con temas más concretos.

- Puntos positivos
 - Menú sencillo
 - Amplio repertorio de preguntas y temarios
 - Gratuita
- Puntos negativos
 - Interfaz con un diseño demasiado simple
 - La puntuación no corresponde
 - No guarda datos de sesión
 - La interfaz no es coherente

¿Cuándo Comienza la Prehistoria?

Puntuacion: 0



Aparecen los Primeros Hominidos

Con la Escritura

Despues de la Historia

Aparecen los Primeros Seres Vivos

Figura 19: Detalle del juego de la página

FINAL

Puntuacion : -6

DUEDES MEJORAR

¿Quieres volver a intentarlo?



Figura 20: Detalle del juego de la página

TRIVINET (13)

Se trata de una página web (<https://www.trivinet.com/>) con un repertorio de preguntas realizadas por usuarios. Aunque se trata de una aplicación que no pertenece al ámbito académico, se ha añadido a la lista de antecedentes dada su



similitud con las funcionalidades respecto al juego de este proyecto.

No existe una calificación como tal, se registran las respuestas correctas y el número de preguntas respondidas. El tiempo de respuesta para cada pregunta está acotada en tiempo y existe la posibilidad de parar el juego y almacenar tu historial, pudiendo retomar el juego en cualquier momento.

■ Puntos positivos

- Interfaz agradable y buen diseño
- Es muy personalizable: tiempo de pregunta, nivel de dificultad etc.
- Mediante registro, se pueden almacenar las calificaciones
- Gratuita

■ Puntos negativos

- No existe menú para elegir el temario, hay que escogerlo en las configuraciones
- La asignación de puntos (clasificación) no es clara



Figura 21: Detalle del juego de la página

Estadísticas de la partida		
Bienvenido usuario no registrado		
Hoy	Nº aciertos	17
	Nº respuestas	44
Regístrate Accede gratis a las ventajas		

Figura 22: Detalle del juego de la página

Pasamos a realizar una comparativa de las tres páginas web para obtener lo mejor y lo peor de cada una, información que podremos aprovechar para el desarrollo de este proyecto.

	TESTEANDO	SOPASLETRAS	TRIVINET
Interfaz intuitiva	✓	✓	X
Calificaciones	X	X	X
Diseño de interfaz	✓	X	✓
Diseño del juego	✓	X	✓

Tabla 8: Tabla comparativa estudio de mercado

En resumen, tanto *TESTEANDO* como *TRIVINET* son buenas fuentes de inspiración para el diseño de interfaz y del juego. Ninguno de los tres productos se ajusta en objetivos a las calificaciones que pretendemos obtener en este proyecto, por lo que no sirven como base.



4. Análisis

En el apartado de análisis, se definirá el catálogo de requisitos (9) funcionales y no funcionales, que detallará las funciones requeridas, el comportamiento y el rendimiento que se espera del producto final. En este caso, el proceso de análisis y obtención de requisitos se realizará con el cliente del proyecto.

4.1. Catálogo inicial de requisitos

A continuación se presentan los requisitos iniciales que debe disponer la aplicación.

4.1.1. Requisitos funcionales

Primero, enumeraremos los requisitos funcionales de la aplicación, es decir, las funciones que debe cumplir el software. Dividiremos las funciones siguiendo los sistemas definidos previamente.

El formato de enumeración para los Requisitos Funcionales será el siguiente:

[RF X]

Donde X será el identificador de dicho requisito.

1. Usuarios

RF 1 La aplicación tendrá dos tipos de usuarios: Alumnos y profesores.

RF 1.1 El profesor deberá realizar su registro antes que los alumnos. En su registro, se le asignará un código único que lo identifique. El profesor tendrá acceso a todos los datos de los alumnos registrados con su código: datos del perfil, avances en el juego y calificación.



- RF 1.2 El alumno deberá registrarse utilizando el código identificativo de su profesor. Tendrá acceso a sus datos y al juego.
- RF 2 El profesor deberá confirmar los registros de los alumnos para que éstos queden registrados de forma definitiva.
- RF 3 Los datos de perfil no serán modificables una vez confirmado el registro.
- RF 4 El profesor debe poder iniciar sesión y acceder a una plataforma con sus funcionalidades.
- RF 5 El alumno debe poder iniciar sesión y comenzar o continuar el juego o acceder a su historial dentro del juego.

2. Contenidos

- RF 4 Los contenidos no podrán ser modificados por ninguno de los usuarios.

3. Calificaciones

- RF 5 La aplicación tendrá el mismo sistema de calificación para las distintas fases del juego.
- RF 6 Las calificaciones no serán editables para ninguno de los usuarios.
- RF 7 El profesor tendrá acceso al desglose de las diferentes calificaciones, es decir, el proceso de obtención de la calificación.

4.1.2. Requisitos no funcionales

A continuación, los requisitos no funcionales, es decir, características del funcionamiento de la aplicación.

El formato de enumeración para los Requisitos No Funcionales será el siguiente:

[RNF X]

Donde X será el identificador de dicho requisito.

- | | |
|-------|--|
| RNF 1 | Seguridad: la aplicación utilizará un sistema de encriptación para guardar los datos personales de las cuentas. |
| RNF 2 | Accesibilidad: |
| RNF 3 | Rendimiento: el tiempo de respuesta a las acciones de un usuario no debe superar 1 segundo, si las condiciones de la conexión lo permiten. |



5. Diseño

La fase de diseño comprende varios pasos o procedimientos previos a seguir para comenzar con la implementación del código. De manera genérica, esta fase pretende establecer la estructura de datos, arquitectura general del software y representación gráfica de la interfaz.

Para poder traducir los requisitos funcionales y no funcionales del proyecto en una representación de software, utilizaremos diagramas como el modelo de casos de uso, el modelo de dominio y el diagrama de la base de datos. Con la definición de estas tres representaciones obtendremos una visión muy concreta del software a desarrollar y facilitaremos la fase de implementación, que basará su desarrollo en este apartado.

5.1. Diagrama de casos de uso

Un diagrama de casos de uso (14), o bien los casos de uso extendidos, pretende definir el comportamiento de un sistema, es decir, cómo interactúan el cliente (bien una persona o bien un sistema externo) y el sistema en cuestión.

Antes de comenzar con la definición de los diagramas de uso que definen el sistema, deberemos identificar los actores que toman un papel en él. Definimos actor como la interpretación de todos aquellos usuarios que tienen el mismo comportamiento o privilegio frente al sistema. Resulta fácil confundir los términos actor y usuario, pero es importante conocer lo que los diferencia: el usuario asume un rol cuando usa el sistema y la definición de actores hace referencia a los diferentes roles que puede adquirir el usuario.

Los actores que se han identificado en este sistema son los siguientes:

- *Usuario*: este actor ha adquirido el nombre de *usuario* para tener una mayor comprensión de su utilidad. Este rol es el que permite al usuario adquirir

cualquiera de los otros dos roles.

- *Alumno*: este es el rol que se adquiere cuando inicia el sistema con un usuario de tipo *alumno*. Podrá realizar aquellas acciones establecidas para el alumno dentro de la aplicación.
- *Profesor*: este actor corresponde a la identificación de un profesor en el sistema. Cuando se inicia la sesión con un usuario de tipo *profesor*, se adquieren los privilegios establecidos para este tipo de rol.

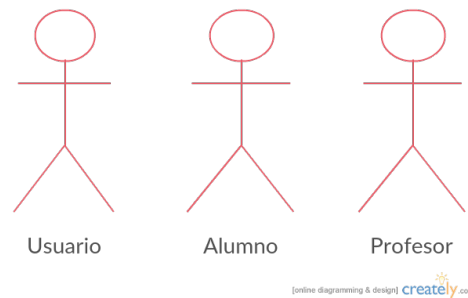


Figura 23: Actores que componen el sistema

Por cada actor identificado dentro del sistema, se realizará un diagrama de casos de uso para conocer su interacción con el software. A continuación, se muestran dichos diagramas y su correspondiente definición.

Casos de uso asociados al USUARIO

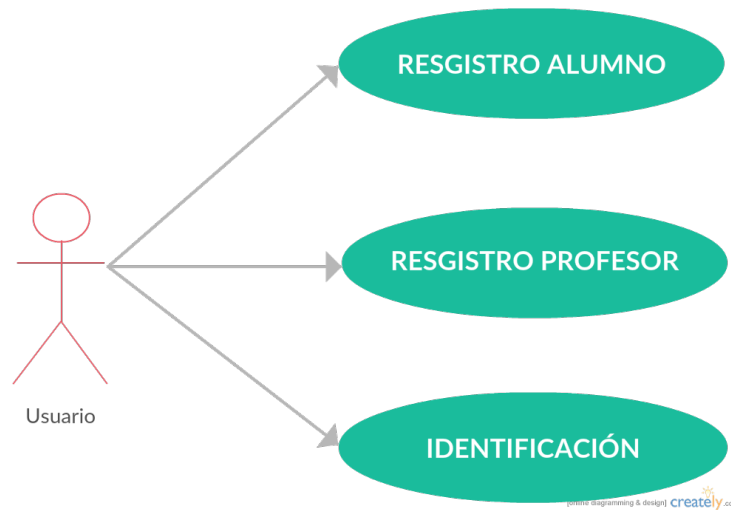


Figura 24: Diagrama de caso de uso relacionado al Usuario

■ Registro

Permite al usuario crear una cuenta para poder acceder al sistema. El registro puede hacerse como alumno o como profesor. Cada registro se tratará de manera diferente para darle acceso o no al sistema.

■ Identificación

Permite a cualquier usuario registrado (y validado) previamente acceder al sistema para poder realizar acciones definidas para su tipo de usuario, alumno o profesor.

Casos de uso asociados al PROFESOR

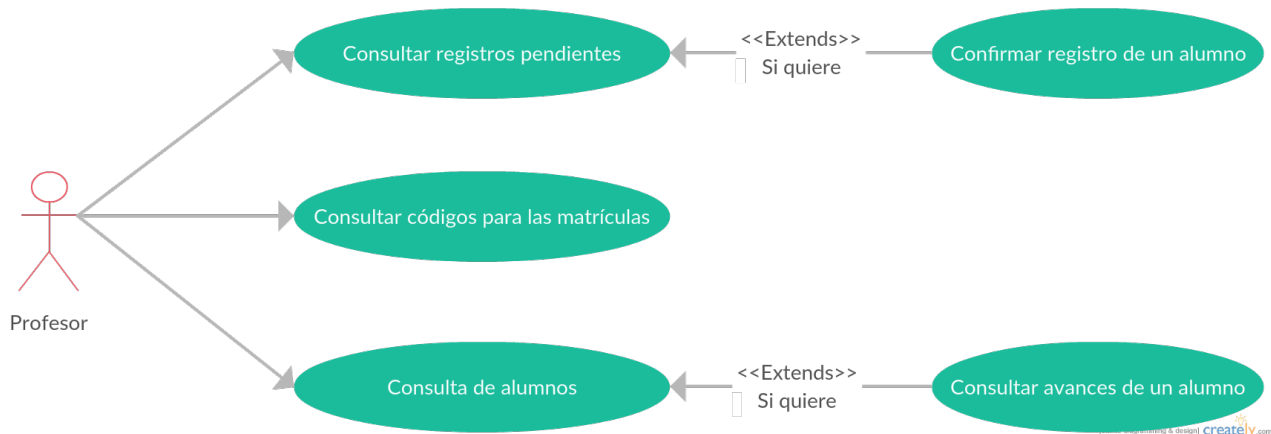


Figura 25: Diagrama de caso de uso relacionado al Profesor

■ Consultar registros pendientes

Permite al profesor visualizar el listado de alumnos pendientes de confirmar. Cuando un alumno realiza su registro, hasta que el profesor no confirma dicho registro no podrá acceder al sistema.

● Confirmar registro de un alumno

Permite al profesor confirmar el registro de un alumno del listado. En la base de datos, deberá cambiar el estado de confirmación del alumno.

■ Consultar códigos para las matrículas

Permite al profesor visualizar un listado con todos los temas asociados a su usuario y sus correspondientes códigos. Será necesario para que el alumno realice su matrícula que el profesor facilite el código del tema.

■ Consulta de alumnos

El profesor visualizará un listado con los alumnos confirmados.

- **Consultar avances de un alumno**

Permite al profesor conocer el historial de juego del alumno seleccionado.

Casos de uso asociados al ALUMNO

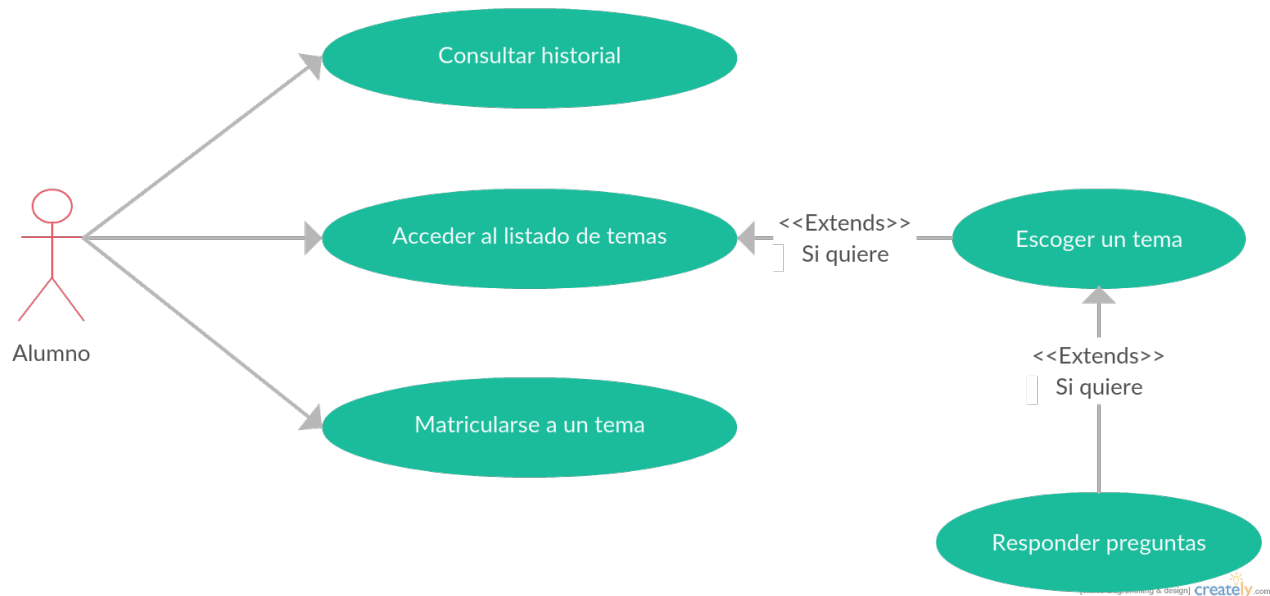


Figura 26: Diagrama de caso de uso relacionado al Alumno

- **Consultar historial**

Permite al alumno hacer una consulta sobre sus avances dentro del juego.

- **Acceder al listado de temas**

El alumno visualizará los temas a los que está matriculado.

- **Escoger tema**

Si el alumno quiere, podrá seleccionar uno de los temas a los que está matriculado para tener acceso a la batería de preguntas asociadas.



- **Responder preguntas**

Una vez accedido a las preguntas, el alumno tendrá la posibilidad de responderlas.

- **Matricularse a un tema**

El alumno tendrá la opción de matricularse a un tema insertando un código. Si el código coincide con alguno de los temas, la base de datos deberá actualizarse.

5.2. Modelo de dominio

La definición del modelo de dominio nos aporta una visión más concreta del sistema, describiendo el dominio de la aplicación por medio de un diagrama de clases que la componen. Las relaciones que existen entre dichas clases deberán quedar reflejadas, así como los atributos más importantes que las definen. A continuación, el diagrama del modelo de dominio (véase *Figura 27*).

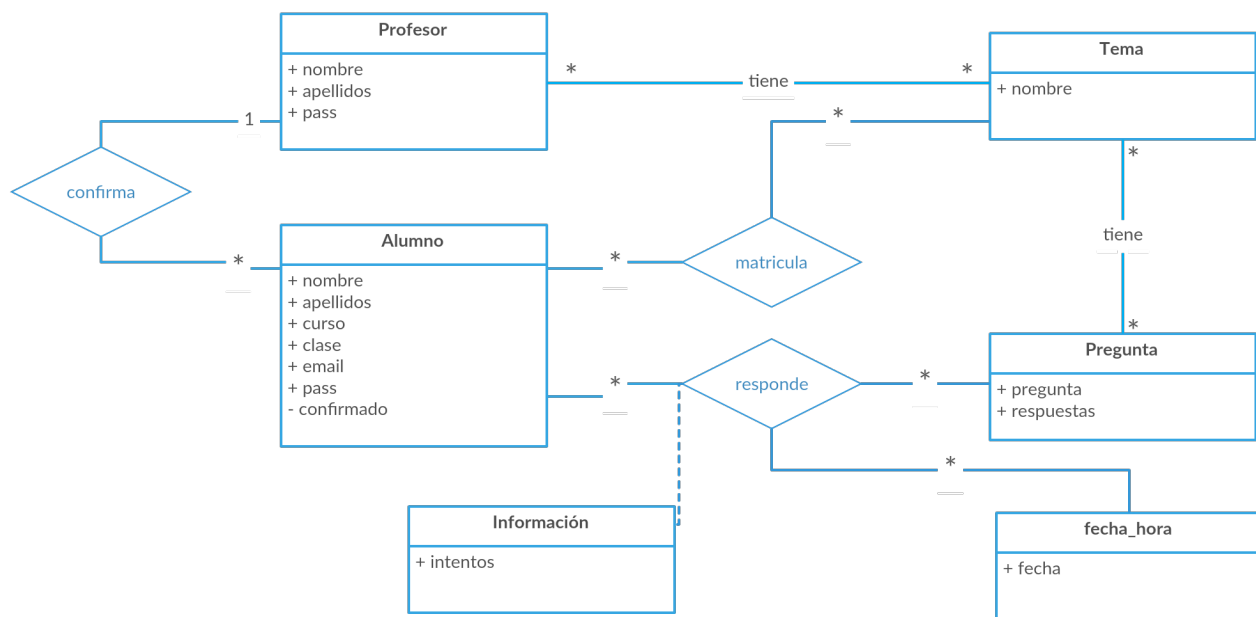


Figura 27: Diagrama de modelo de dominio

Procedemos con la explicación de todos los elementos que componen el diagrama: entidades, asociaciones y relaciones.

■ Entidades

Para un correcto almacenamiento y gestión de toda la información que requiere el sistema, se han identificado las siguientes entidades.

- **Profesor:** almacenará todos los datos del registro de un profesor.
- **Alumno:** almacenará todos los datos del registro de un alumno.
- **Tema:** almacenará todos los temas que contendrán las preguntas del juego.
- **Pregunta:** es la unidad mínima que compondrá un tema. Existen tres tipos de preguntas distintas que serán almacenadas en la misma estructura.



- **Fecha_hora**: entidad para tener un control de la fecha de respuesta a una pregunta por parte de los alumnos.

■ Asociaciones

Corresponde al objeto que almacena atributos que son comunes a más de una entidad.

- **Información**: se trata de la asociación que almacenará los intentos por parte de un alumno de responder una pregunta. Dado que es un atributo que se genera a partir de las entidades alumno y pregunta se representa dentro de una asociación en el modelo de dominio.

■ Relaciones

Las relaciones tratan de explicar la relación entre entidades, su forma de interactuar y comunicarse.

- **Alumno - Profesor (*..*)** : representa las confirmaciones de registro de alumnos que puede realizar el profesor.
- **Alumno - Tema (*..*)** : un alumno podrá matricularse a varios temas mediante un código (único para cada tema) proporcionado por el profesor. Por otro lado, un tema tendrá múltiples alumnos matriculados.
- **Profesor - Tema (*..*)** : los profesores tendrán acceso a los códigos de los distintos temas, por lo que podríamos decir que un profesor *tiene* varios temas. De la misma manera, un tema puede estar relacionado con varios profesores. Aunque en este sistema pueda parecer innecesaria esta estructura, permite futuras ampliaciones en las funcionalidades, como por ejemplo, que un profesor pueda crear un tema y sólo sus alumnos tengan acceso a él.



- **Tema - Pregunta (1..*)** : un tema estará compuesto por varias preguntas, pero una pregunta sólo podrá pertenecer a un tema. De esta manera, evitamos errores de consultas y la estructura de la base de datos resulta más intuitiva.
- **Alumno - Pregunta - Fecha_hora (*..*..*)** : cuando un alumno responde una pregunta, deberá almacenarse la fecha y la hora en la que lo ha hecho, por lo que existe la necesidad de generar esta relación. Una pregunta puede ser respondida por múltiples alumnos en una fecha y hora diferentes. De la misma manera, podría suceder que varios alumnos respondan a la misma pregunta de manera simultánea, por lo que la cardinalidad de la fecha debe ser múltiple.

5.3. Diagrama de bases de datos

En este apartado y con la información previa, se detallará el diagrama Entidad-Relación que deberá soportar los datos y requisitos del sistema. A continuación, se observa la representación gráfica del diagrama obtenido.

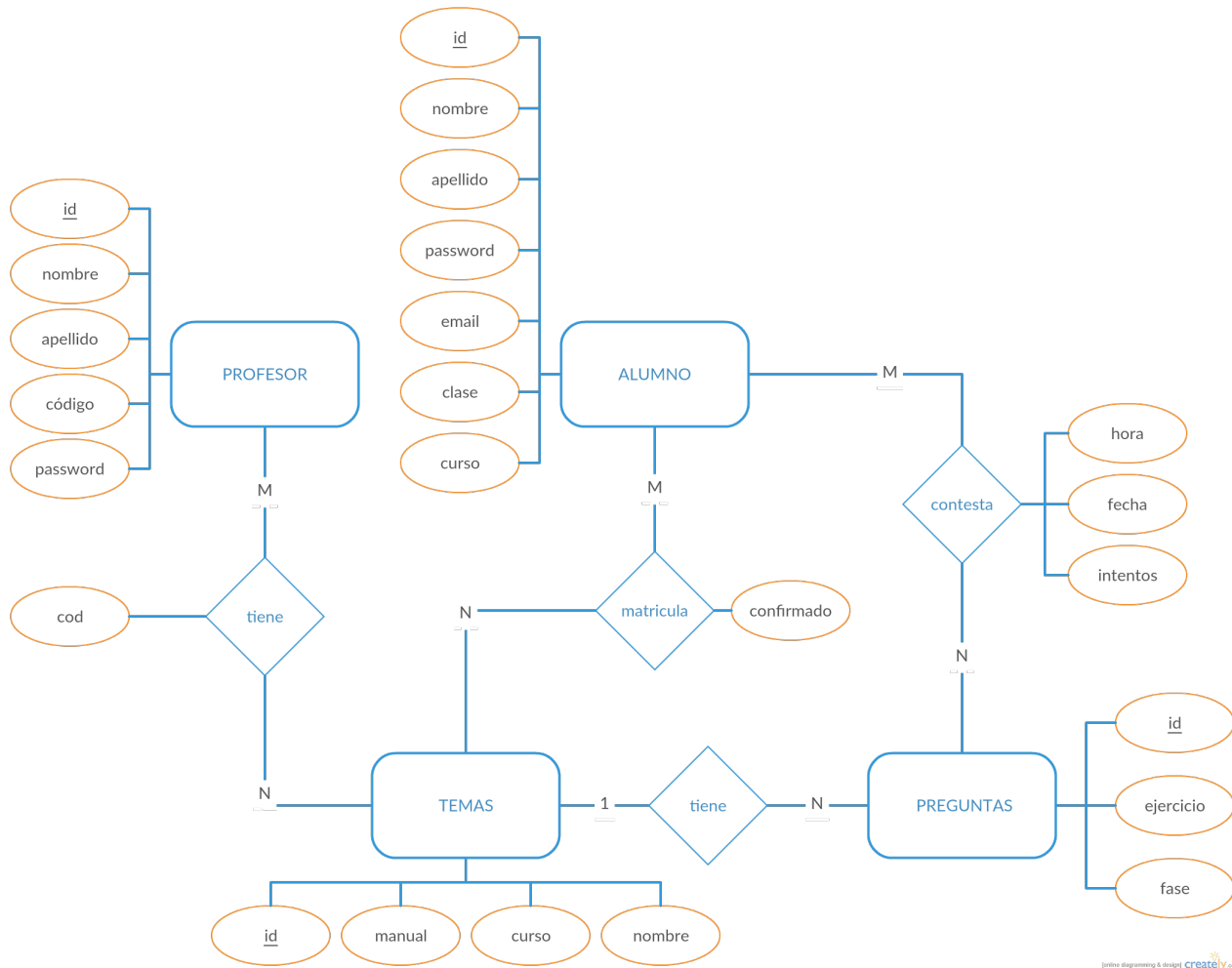


Figura 28: Diagrama de Entidad-Relación

ENTIDADES

- *Profesor*: entidad que representa a los profesores que tomarán parte en el sistema.
- *Alumno*: entidad que representa a todos los alumnos que tendrán acceso al sistema.
- *Temas*: entidad que representa aquellas unidades de preguntas que el alumno



deberá superar.

- *Preguntas*: entidad que representa una pregunta dentro de un tema.

RELACIONES

- *tiene* [Profesor-Temas]: un profesor deberá almacenar los temas que desee utilizar con sus alumnos.
- *matricula* [Alumno-Temas]: un alumno deberá tener almacenados los temas a los que puede responder.
- *tiene* [Temas-Preguntas]: un tema estará compuesto por varias preguntas.
- *contesta* [Alumno-Preguntas]: un alumno deberá almacenar las preguntas a las que haya respondido.

Este diagrama pretende solucionar las necesidades definidas con los requisitos del sistema. A lo largo del desarrollo, es probable que sufra alguna alteración para facilitar el flujo de datos o para solventar alguna necesidad que no se ha tenido en consideración.

5.3.1. Diseño de maquetas

Al tratarse de una página web, resulta de ayuda un diseño inicial de la interfaz. Para este proyecto, se ha decidido utilizar un diseño mediante maquetas el cual nos aportará una visión general de la navegación en el sistema. Podría resultar de ayuda para corregir algún error de funcionalidad o aspecto general que el cliente decida cambiar.

Como primer paso para el diseño de pantallas, definiremos el flujo de la página web, es decir, las opciones que tendremos en cada pantalla y a dónde nos lleva



cada una. A continuación, se muestra por medio de un diagrama, cada una de las pantallas que tendrá la aplicación y a qué otras pantallas nos lleva cada una.

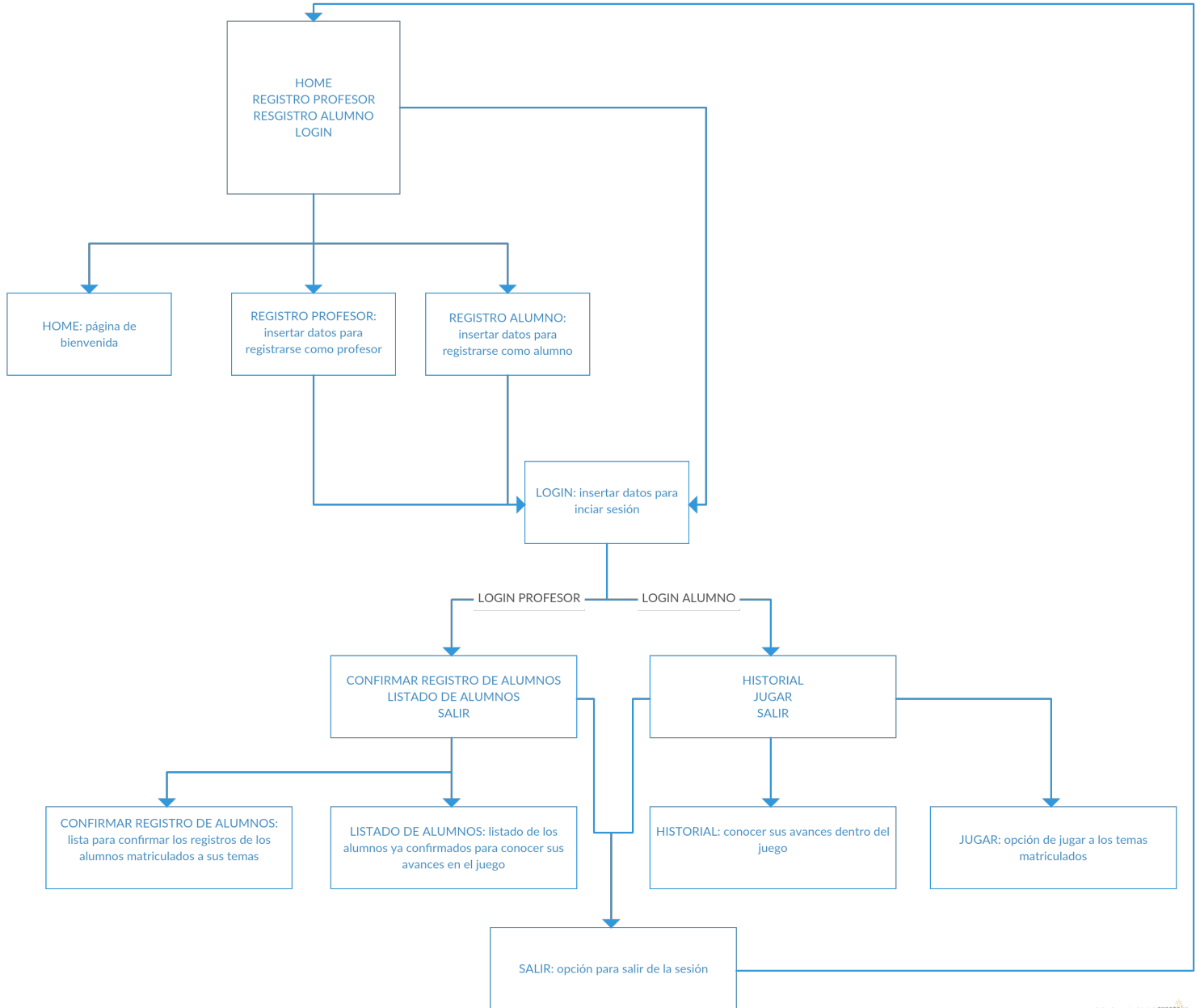


Figura 29: Diagrama que define el flujo de pantallas



Cada uno de los cuadrados en el diagrama, representa una página en el sistema, a excepción de la última llamada *SALIR*. La página inicial nos muestra un menú de inicio con varias opciones: *HOME*, *REGISTRO PROFESOR*, *REGISTRO ALUMNO* y *LOGIN*.

Al realizar el registro como alumno o profesor la pantalla a la que se redirigirá al usuario será la de *LOGIN*, para que pueda hacer el inicio de sesión y acceder a sus opciones. Desde el menú inicial también se podrá acceder al inicio de sesión, facilitando el acceso a aquellos usuarios ya registrados.

Una vez iniciada la sesión el **profesor** tendrá la opción de *CONFIRMAR REGISTRO DE ALUMNOS* o acceder al *LISTADO DE ALUMNOS* para conocer los avances en el juego de sus alumnos. Tendrá la opción de *SALIR* para cerrar su sesión dentro del sistema cuando desee.

La sesión del **alumno** nos permitirá conocer su *HISTORIAL* en el juego así como *JUGAR*. De la misma manera que el profesor, existe la opción de *SALIR* para cerrar la sesión iniciada.

Una vez definido el flujo de pantallas del sistema, pasamos a definir un diseño inicial de la interfaz del sistema. El objetivo es lograr un diseño sencillo e intuitivo para el usuario, por lo que se utilizarán menús muy visuales y fáciles de usar.

A continuación se mostrarán una a una, las maquetas diseñadas para algunas de las distintas pantallas del sistema, permitiendo crear una idea general del que será el aspecto general. Es importante remarcar que se ha empleado un tiempo limitado para generar estas maquetas, dada la metodología que se va a utilizar. Siendo una de las ventajas de esta metodología el resultado de un prototipo por cada iteración, el cliente podrá validar paso a paso el aspecto de la página web y hacer peticiones de modificación.

Para unificar el diseño y facilitar la navegabilidad en las pantallas, se creará un menú superior en todas las páginas común a todos los usuarios. Las opciones que

compartan los dos tipos de usuarios, profesor y alumno, quedarán fijas en el menú superior mientras que aquellas que sean particulares se irán mostrando en la parte central de la pantalla.

PANTALLA DE INICIO

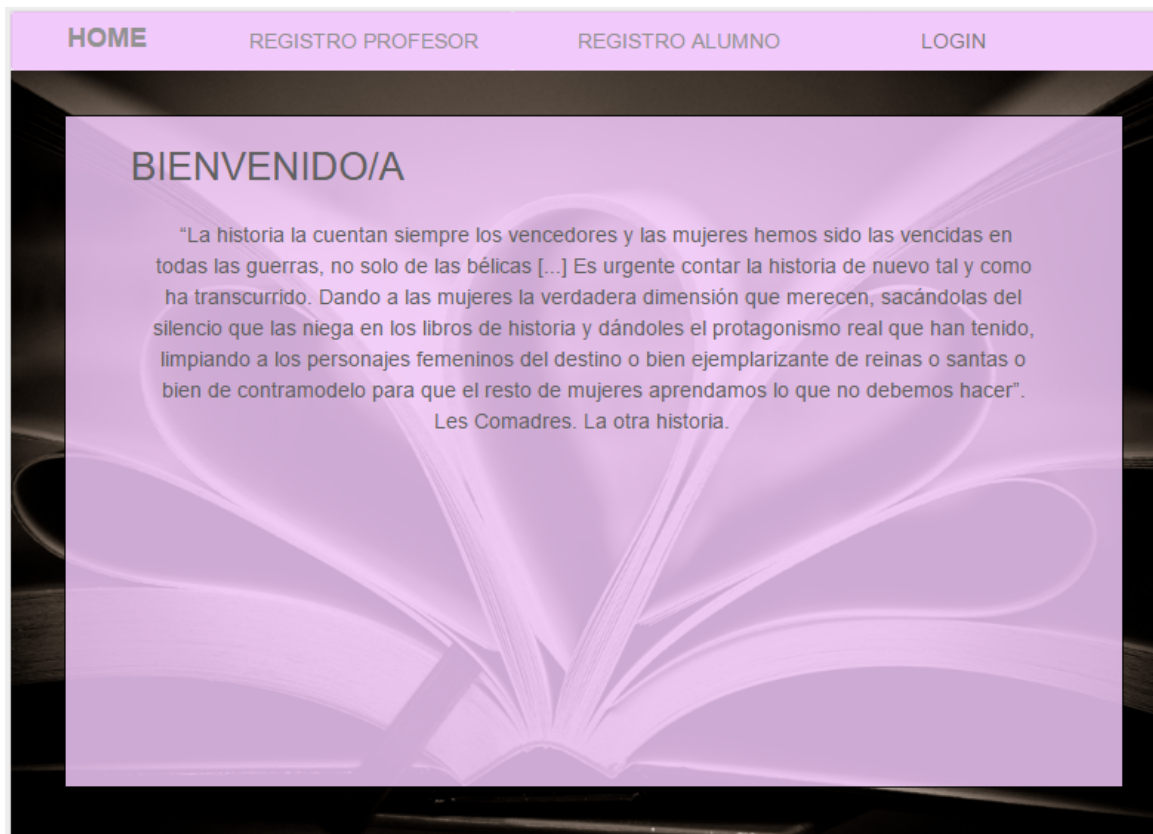


Figura 30: Diseño de la pantalla de inicio



PANTALLA DE REGISTRO DEL PROFESOR

HOME REGISTRO PROFESOR REGISTRO ALUMNO LOGIN

Registro profesor

Nombre

Apellidos

Contraseña

Confirmar contraseña

Figura 31: Diseño de la pantalla de registro del profesor

PANTALLA DE REGISTRO DEL ALUMNO



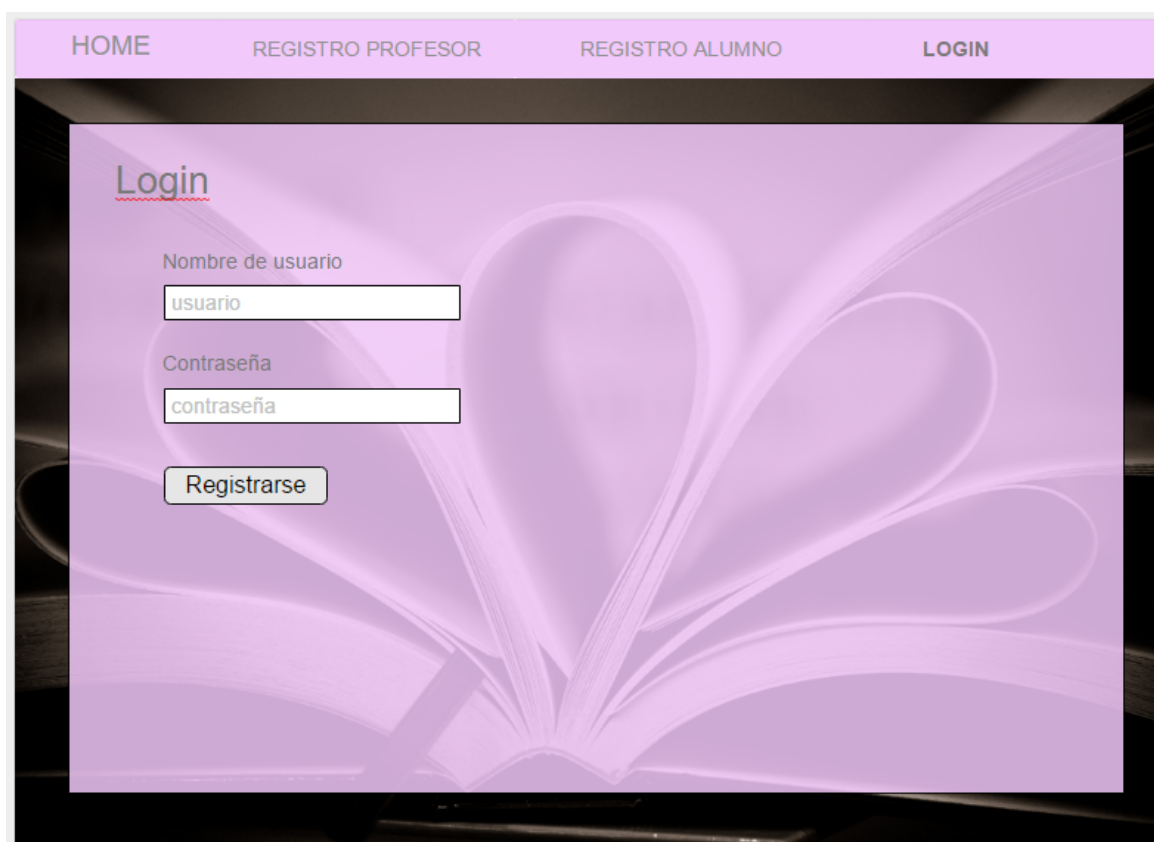
HOME REGISTRO PROFESOR **REGISTRO ALUMNO** LOGIN

Registro alumno

Nombre	<input type="text" value="nombre"/>	Correo electrónico	<input type="text" value="correo"/>
Apellidos	<input type="text" value="apellidos"/>	Curso	<input type="text" value="curso"/>
Contraseña	<input type="text" value="contraseña"/>	Clase	<input type="text" value="clase"/>
Confirmar contraseña	<input type="text" value="confirmar contraseña"/>	<input type="button" value="Registrarse"/>	

Figura 32: Diseño de la pantalla de registro del alumno

PANTALLA DE INICIO DE SESIÓN



HOME REGISTRO PROFESOR REGISTRO ALUMNO LOGIN

Login

Nombre de usuario

Contraseña

Figura 33: Diseño de la pantalla de inicio de sesión

PANTALLA DE INICIO DEL ALUMNO

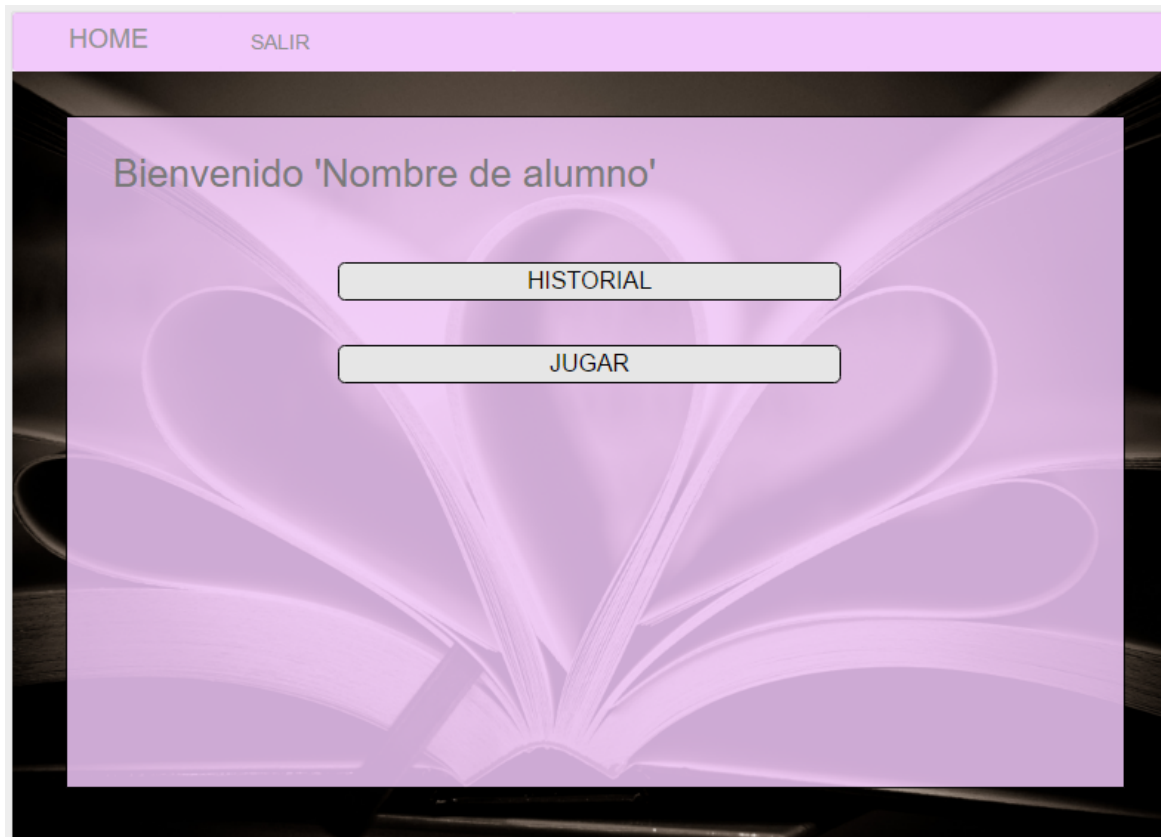
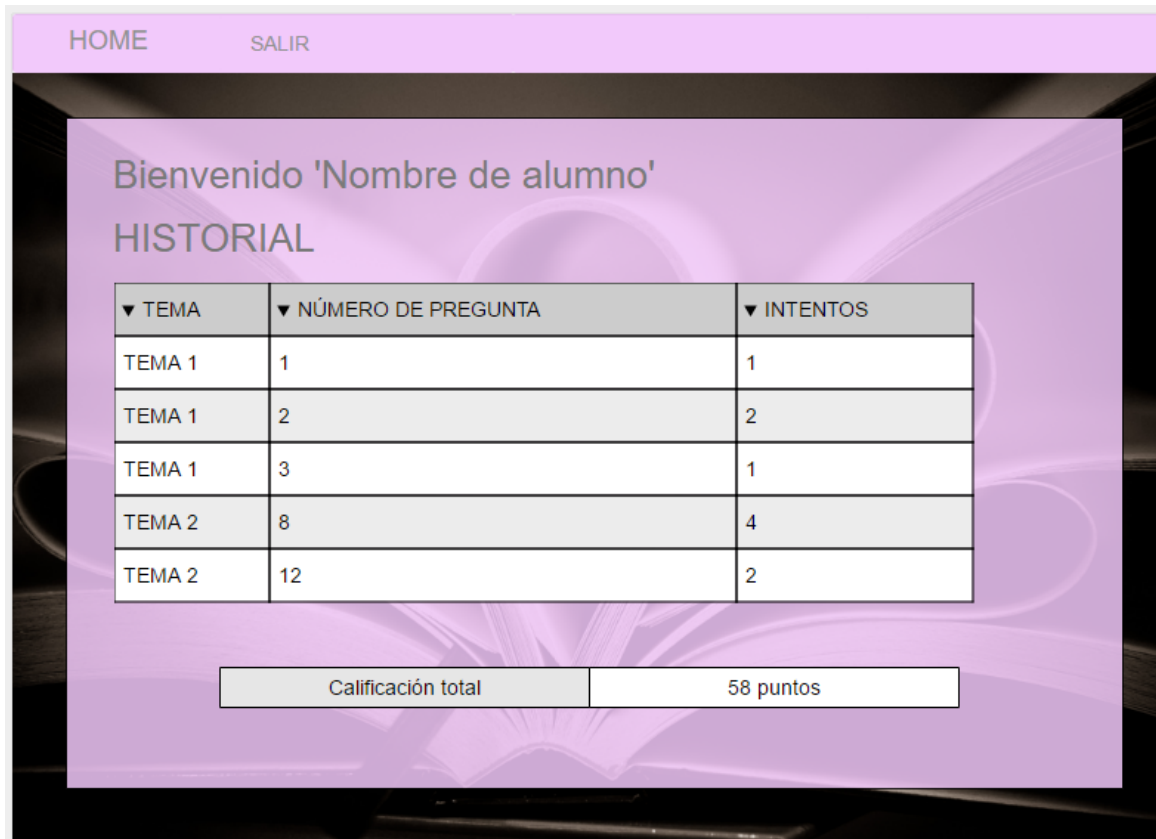


Figura 34: Diseño de la pantalla de inicio del alumno

PANTALLA DE HISTORIAL DEL ALUMNO



▼ TEMA	▼ NÚMERO DE PREGUNTA	▼ INTENTOS
TEMA 1	1	1
TEMA 1	2	2
TEMA 1	3	1
TEMA 2	8	4
TEMA 2	12	2

Calificación total	58 puntos
--------------------	-----------

Figura 35: Diseño de la pantalla de historial del alumno

PANTALLAS DE JUEGO DEL ALUMNO

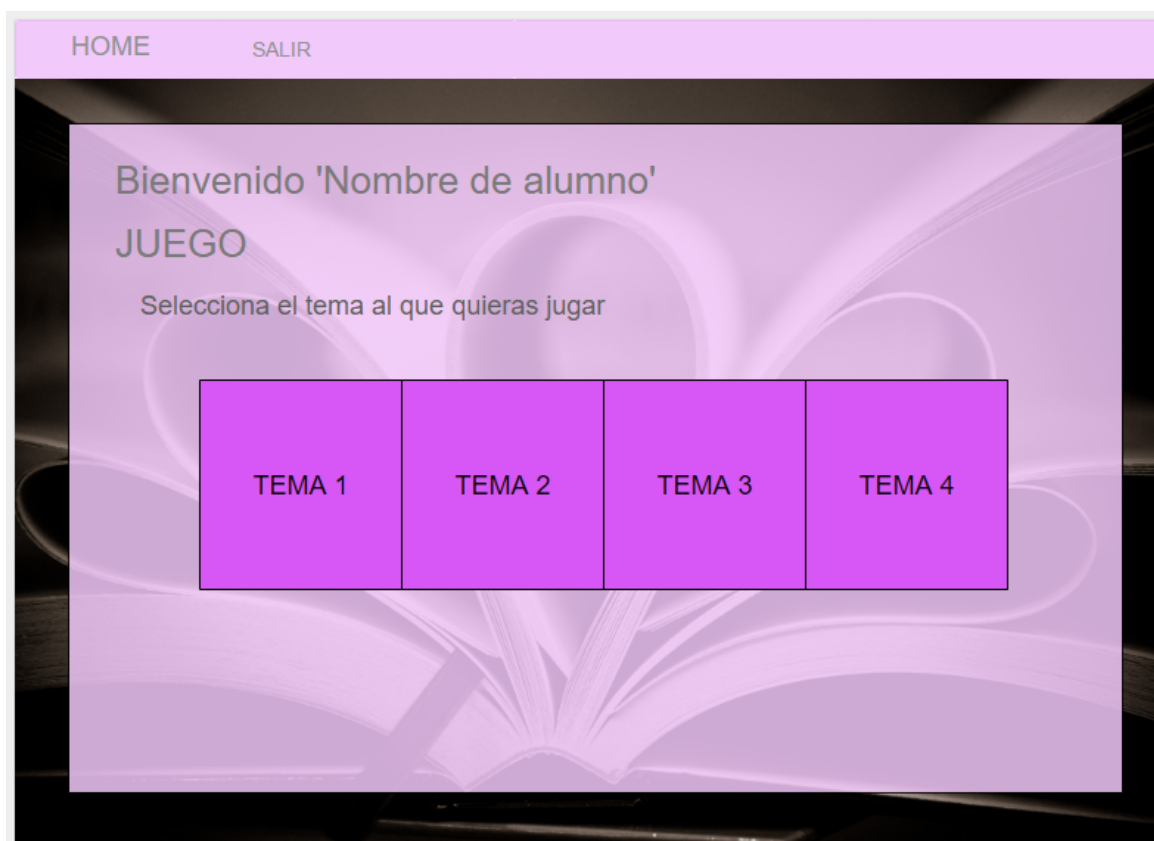


Figura 36: Diseño de la pantalla de juego del alumno



Figura 37: Diseño de la pantalla de juego del alumno

PANTALLA DE INICIO DEL PROFESOR

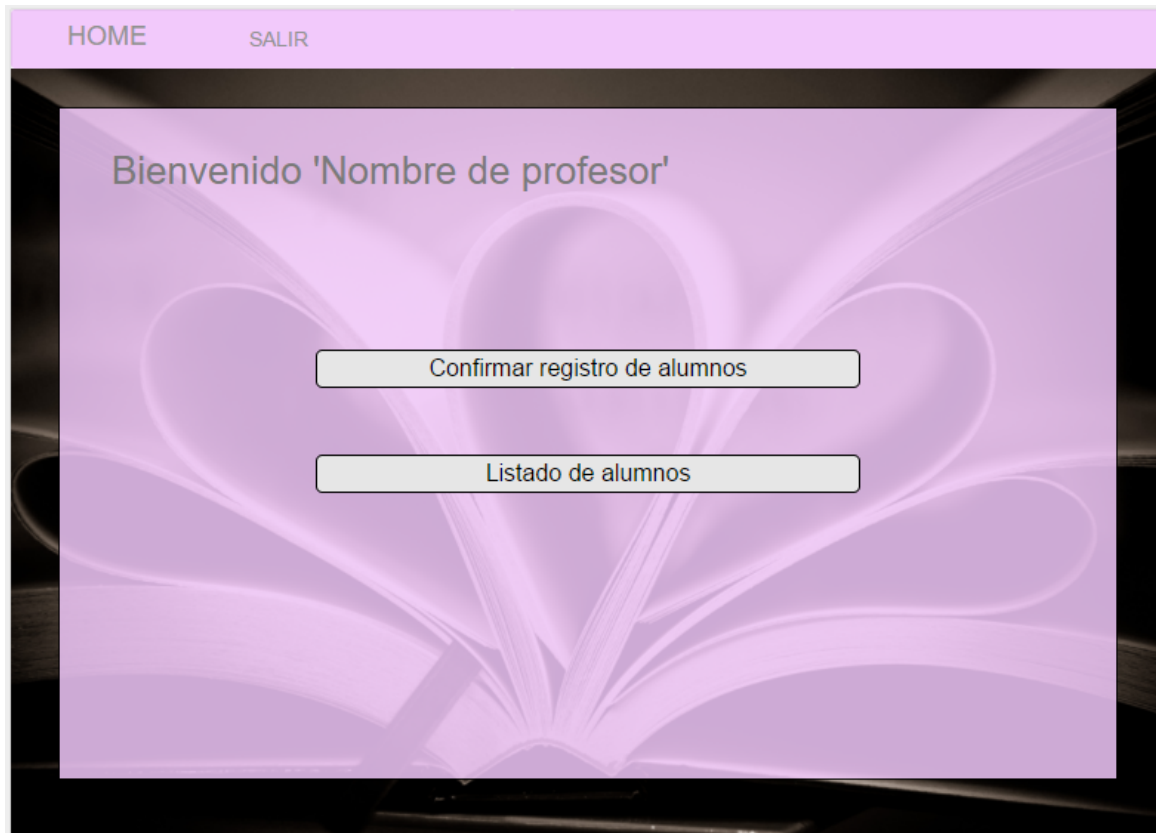


Figura 38: Diseño de la pantalla de inicio del profesor

PANTALLA DE CONFIRMACIÓN DE REGISTROS DE ALUMNOS DEL PROFESOR

HOME SALIR

Bienvenido 'Nombre de profesor'

Confirmar registro de alumnos

▼ Nombre	▼ Apellidos	▼ Correo	▼ Confirmar
Nombre 1	Apellidos	correo@correo.com	<input type="radio"/>
Nombre 2	Apellidos	correo@correo.com	<input checked="" type="radio"/>
Nombre 3	Apellidos	correo@correo.com	<input type="radio"/>
Nombre 4	Apellidos	correo@correo.com	<input checked="" type="radio"/>
Nombre 5	Apellidos	correo@correo.com	<input type="radio"/>
Nombre 6	Apellidos	correo@correo.com	<input type="radio"/>
Nombre 7	Apellidos	correo@correo.com	<input checked="" type="radio"/>

Confirmar seleccionados

Figura 39: Diseño de la pantalla de confirmación de registros de alumnos del profesor

PANTALLA DE LISTADO DE ALUMNOS DEL PROFESOR

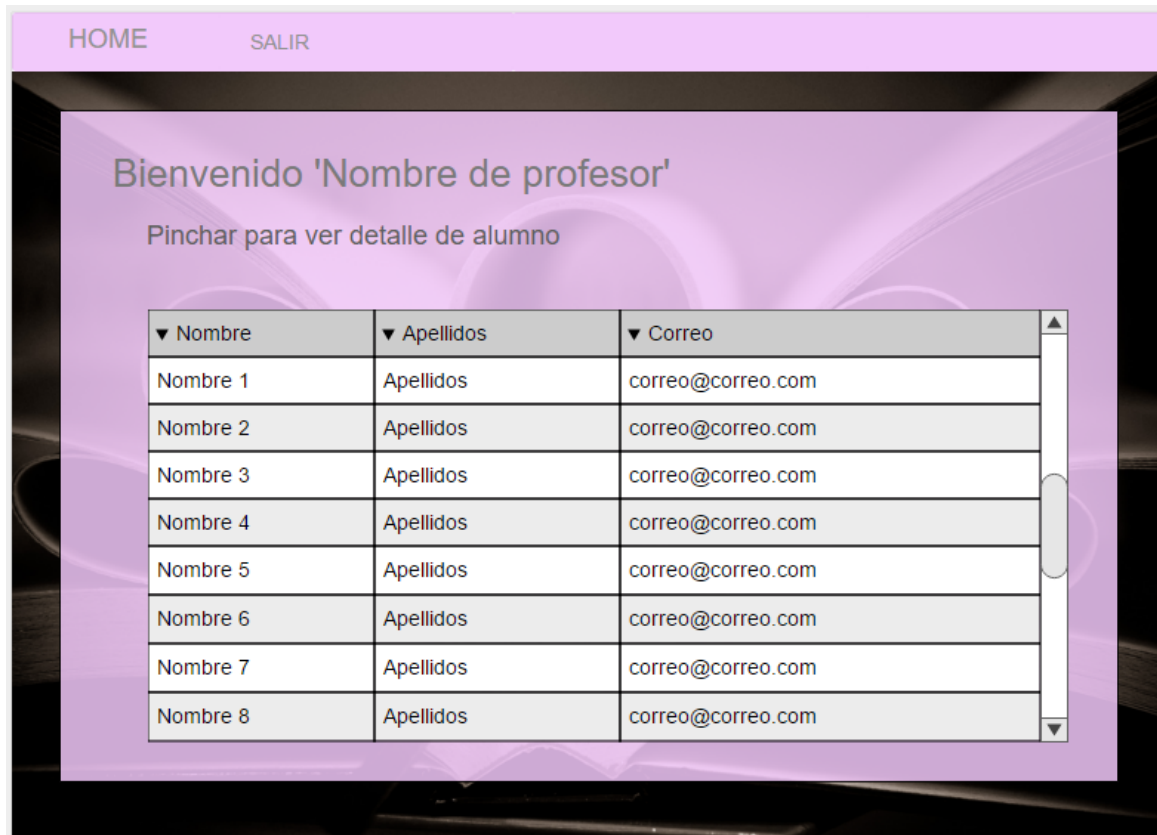


Figura 40: Diseño de la pantalla de listado de alumnos del profesor

5.4. Diseño de pruebas

En esta sección se tratará de definir el Plan de pruebas del sistema (9). Es importante identificar todas aquellas pruebas que deseamos que el sistema cumpla antes de comenzar con el desarrollo, para evitar dejar atrás comprobaciones. Esto se debe a que el desarrollo lo lleva a cabo una única persona y puede que, de manera subjetiva, defina las pruebas acotándose al funcionamiento del código ya desarrollado.

Existen diferentes niveles de pruebas dentro de un proceso software y es im-



portante identificarlos y reconocer qué tratan de comprobar dichas pruebas. Según el módulo y la funcionalidad que se desea comprobar, existen cuatro niveles de pruebas:

- **Pruebas unitarias:** conjunto de pruebas que permiten verificar el correcto funcionamiento de un módulo o unidad. Tienen como objetivo comprobar la robustez del código que cumple una funcionalidad concreta, evitando fallos en el ingreso de datos por parte del usuario y errores de navegación. En este proyecto concreto, las pruebas unitarias corresponderían con las funcionalidades definidas. Por cada una, deberán realizarse las pruebas unitarias pertinentes para asegurar el correcto funcionamiento del módulo.
- **Pruebas de integración:** se trata de aquellas pruebas que tienen como objetivo comprobar el correcto funcionamiento y comunicación entre dos bloques o módulos ya probados mediante las Pruebas unitarias. En este caso, las pruebas de integración se realizarán para comprobar que los módulos que cumplen las diferentes funcionalidades trabajan de manera correcta entre sí.
- **Pruebas de sistema:** estas pruebas suelen ser realizadas por un equipo ajeno al del desarrollo. Se pretende comprobar que el sistema cumple con lo preestablecido en el Documento de Requisitos y cubre las necesidades y expectativas del cliente. Este nivel de pruebas se realizará en el mismo entorno, o similar, al utilizado en el desarrollo para evitar problemas y fallos en el sistema donde se pretende instalar el producto. Para este proyecto, el personal dedicado a hacer estas pruebas será la tutora del proyecto.
- **Pruebas de aceptación:** este tipo de pruebas se realizan en la parte del cliente. Se trata de un conjunto de pruebas realizado por el personal al cuál se ha enfocado todo el sistema, es decir, el producto deberá instalarse en el entorno



definitivo y será probado por los individuos destinados a ser los usuarios habituales del producto. Aunque la idea de este tipo de pruebas ha sido concebida para realizarse con el producto totalmente terminado, con la metodología llevada a cabo para este proyecto, el cliente deberá validar cada prototipo resultado de cada iteración.

En la sección 7 *Validación* se identificarán aquellas pruebas que el sistema deberá superar en los distintos niveles de comprobación.



6. Desarrollo

Dado que el desarrollo se ha realizado siguiendo la planificación inicial dividida en iteraciones, se describirá el proceso de ejecución con esta misma división. De la misma manera, se mencionarán las diferentes decisiones tomadas en relación a la implementación así como a la estructuración del proyecto.

NOTA: a lo largo de esta sección se utilizarán diferentes términos para hacer referencia a las pantallas del sistema. Cuando se habla de interfaz de un *alumno* o *profesor*, se referirá a todas aquellas pantallas que pertenezcan al perfil de *alumno* o *profesor*.

6.1. Iteración 1

En esta primera iteración se pretende cubrir la funcionalidad de inicio de sesión. Para permitir este inicio de sesión, la base de datos debe estar ya creada y deberemos crear la estructura de ficheros de la página web. En estos primeros pasos es imprescindible avanzar y crear el contenido con vistas a futuro, es decir, teniendo en cuenta que con cada iteración el proyecto irá creciendo y evolucionando. Es por esto que se ha tomado la decisión de seguir una estructuración de documentos muy habitual en el ámbito Web:

- Por una parte, se almacenarán todos los ficheros relacionados con el estilo. En este caso concreto, se ha decidido utilizar Bootstrap, que dota a los elementos más comunes de la página web de un aspecto más atractivo para el usuario. [Carpeta con nombre *bootstrap*]
- Por otra parte se almacenarán los ficheros JavaScript que ayudarán en el procesamiento de ciertas funcionalidades de la página, como las validaciones en los formularios. [Carpeta con nombre *js*]



- En una carpeta almacenaremos todas las imágenes que utilizaremos. De esta manera quedarán recogidas y más accesibles. [Carpeta con nombre *img*]
- Aunque todos los ficheros que generemos tendrán la extensión *.php*, aquellos que hagan alguna actualización en la base de datos o incluso algunas consultas más complejas, se recogerán en una carpeta a parte. [Carpeta con nombre *php*]
- Por último, todos los ficheros que formen la interfaz gráfica de la página web irán ubicados en la raíz del documento.

En la siguiente figura se aprecia el directorio raíz generado para el proyecto en esta primera iteración.

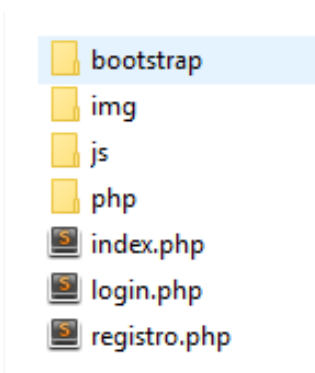
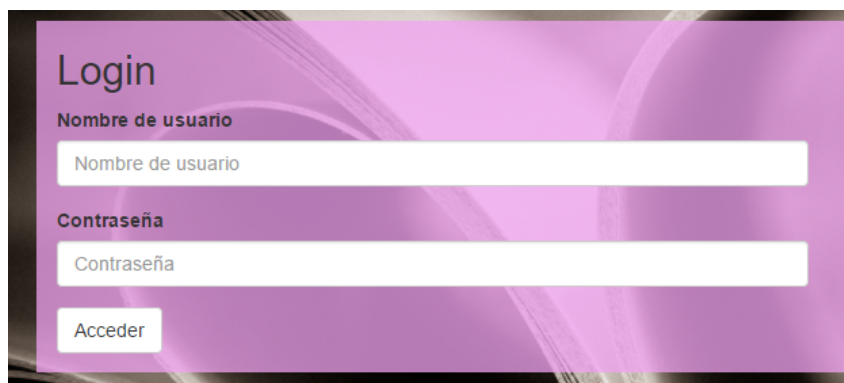


Figura 41: Estructura de los ficheros dentro del proyecto

Una vez creado el directorio de ficheros que utilizaremos a lo largo del proyecto, toca generar un inicio de sesión para cualquier usuario. Para ello, utilizaremos la base de datos generada a partir del 5.3 *Diseño de base de datos inicial* e introduciremos algún dato en la tabla *alumno* o *profesor* que nos permita probar el código de inicio de sesión.

El aspecto final de esta pantalla es el siguiente.



La imagen muestra una interfaz de usuario para el inicio de sesión. El fondo es de color morado con líneas abstractas. En la parte superior izquierda, se encuentra el título "Login". Debajo de este, hay dos secciones de entrada de texto: "Nombre de usuario" y "Contraseña", cada una con un campo de entrada blanco. Debajo de estos campos, hay un botón rectangular con el texto "Acceder".

Figura 42: Detalle del inicio de sesión en la interfaz de usuario

6.2. Iteración 2

En esta iteración se deberá generar el código que permita a cualquier usuario registrarse, tanto como *alumno* o como *profesor*. Para ello se validarán los datos insertados por el usuario con funciones de JavaScript, de esta manera logramos dividir funcionalidades del código en diferentes ficheros y hacer la lectura del mismo más sencilla.

Con esta estructuración de los ficheros, el formulario que generaremos en la interfaz del registro, deberá llamar al fichero JavaScript que valide dicho formulario. El código validador será el que decida lo que sucede si el registro es erróneo (mostrará un mensaje de error en el registro) o correcto (reenviará al usuario a la página de inicio de sesión).



```
1 <?php session_start(); ?>
2 <html>
3 <head>
4 <title>Formulario de Registro</title>
5 <link rel="stylesheet" type="text/css" href="bootstrap/css/bootstrap.min.css">
6 </head>
7 <body style="background-image: url(img/fondo.jpg); background-repeat: no-repeat; background-position: center center; background-attachment: fixed;
8 background-size: cover;">
9 <?php include "php/navbar.php"; ?>
10 <div class="container" style="background: rgba(242, 163, 242, 0.75);">
11 <div class="row">
12
13 <div class="col-md-6">
14 <h2>Registro Alumno</h2>
15
16 <form role="form" name="registro" action="php/registroAlumno.php" method="post">
17 <div class="form-group">
18 <label for="nombreAlumno">Nombre</label>
19 <input type="text" class="form-control" id="nombreAlumno" name="nombreAlumno" placeholder="Nombre">
20 </div>
21 <div class="form-group">
22 <label for="apellidoAlumno">Apellidos</label>
23 <input type="text" class="form-control" id="apellidoAlumno" name="apellidoAlumno" placeholder="Apellidos">
24 </div>
25 <div class="form-group">
26 <label for="emailAlumno">Correo Electronico</label>
27 <input type="email" class="form-control" id="emailAlumno" name="emailAlumno" placeholder="Correo Electronico">
28 </div>
29 <div class="form-group">
30 <label for="cursoAlumno">Curso al que perteneces</label>
31 <input type="text" class="form-control" id="cursoAlumno" name="cursoAlumno" placeholder="Curso al que perteneces">
32 </div>
33 </div>
34 </div>
35 </div>
36 </body>
37 </html>
```

Figura 43: Detalle del código

Como se puede observar en el detalle del código que recoge la *Figura 43*, remarcado dentro de un recuadro morado, se encuentra la llamada al código que validará cada campo.

En este punto del desarrollo, es imprescindible ampliar la base de datos para cubrir la funcionalidad y albergar todos los datos necesarios para cada tipo de usuario que se registrará.

Una vez ampliado el prototipo con el registro, resulta necesario comenzar a gestionar la sesión del usuario dentro del sistema. Cuando el usuario inicie sesión de manera satisfactoria, el sistema deberá guardar el identificador de dicho usuario para poder reconocerlo en cualquier punto de su navegación. Aquí es donde entran en juego las variables de sesión, que serán las encargadas de pasar la información necesaria entre páginas, como por ejemplo, el identificador de usuario. Como se puede apreciar en la siguiente ilustración, llamando a la función PHP *session.start()*, se iniciará o reanudará la sesión que nos permitirá almacenar las variables que necesitamos.



```
<?php
session_start();
if(!isset($_SESSION["user_id"]) || $_SESSION["user_id"]==null){
    print "<script>alert(\"Acceso invalido!\");window.location='login.php';</script>";
}
```

Figura 44: Detalle del código

Llegados a este punto del desarrollo y contando ya con varias opciones (registro alumno, registro profesor e iniciar sesión), se crea la necesidad de un menú para el usuario. La idea es implementar un menú común a todas las pantallas, con diferentes opciones, para crear un diseño uniforme a lo largo de todas las interfaces. Para ello, se ha tomado la decisión de apartar el código correspondiente al menú e invocar a dicho código en cada una de las pantallas. De esta manera, logramos un código más compacto y evitamos duplicidad y posibles fallos en el menú. El fichero que recoge el código común a todas las pantallas, se encargará de generar el menú con las opciones adecuadas en cada pantalla.

Por ejemplo, si el usuario ya ha iniciado sesión, no le deberá aparecer la opción de registro o inicio de sesión, pero sí la de acabar la sesión. Tanto el alumno como el profesor compartirán el mismo menú superior y las distintas acciones que puedan realizar se les mostrarán en su pantalla. De esta manera, logramos unificar la implementación del menú. En las siguientes ilustraciones se aprecia el aspecto de la barra de menú antes de iniciar sesión por parte de ningún usuario y después de haberla iniciado en cualquiera de los dos roles, alumno o profesor.

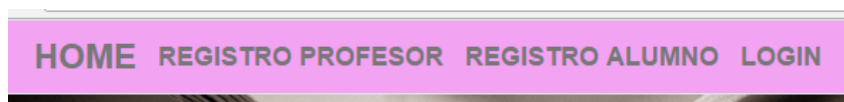


Figura 45: Detalle del menú en la interfaz de usuario

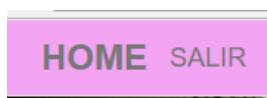


Figura 46: Detalle del menú en la interfaz de usuario

6.3. Iteración 3

Para esta iteración de deberá implementar el código que permita a un profesor acceder a una lista de todos sus alumnos y al detalle de cada alumno, de manera que conozca sus avances dentro del juego. Como decisión inicial, dado que el profesor hará más uso de esta opción que de las futuras (confirmar registros de alumnos), se ha decidido incluir la lista en la página principal. De esta manera hacemos el acceso más directo y visual.

El listado que le aparezca al profesor será de todos aquellos alumnos registrados y confirmados, ordenados por *curso* y *clase*. Para crear una interfaz más intuitiva y usable, el profesor deberá hacer click sobre el alumno que desea recibir información. A continuación, un detalle de la interfaz resultante, con el listado de alumnos ya confirmados.



CONFIRMAR ALUMNOS PENDIENTES

Listado de alumnos

Nombre	Apellido	Email	Curso	Clase
alumno 1	fvd	fdfv@fvfs.com	1	A
alumno 2	kjh	kjh@lj.com	1	B
jon	jon	jon@jon.com	4	C

Figura 47: Detalle de la página inicial de la interfaz de profesor

Una vez pinche sobre un alumno del que desea obtener información, deberán aparecer todos los datos de dicho alumno en la pantalla: temas a los que está matriculado, preguntas a las que ha respondido, fecha y hora de respuesta, intentos de responder etc. De esta manera, el profesor conocerá el avance de ese alumno concreto dentro del juego y obtendrá información valiosa como la fecha y hora del último intento de respuesta. En la siguiente ilustración se aprecia la pantalla generada con los datos de un alumno.

Progresos del alumno:

jon jon

TEMA 1

Pregunta	Fecha y hora	Intentos
1	2017-05-17 11:26:00	2
2	2017-05-17 11:32:00	1

TEMA 2

Pregunta	Fecha y hora	Intentos
----------	--------------	----------

TEMA 3

Pregunta	Fecha y hora	Intentos
----------	--------------	----------

Figura 48: Detalle de la página de información de alumno de la interfaz de profesor

En la última iteración del desarrollo (calificaciones de un alumno) deberá añadirse a esta interfaz la calificación obtenida por el alumno en cada tema.

6.4. Iteración 4

Para el desarrollo de esta iteración correspondiente al acceso por parte del alumno a sus propios avances, basaremos la implementación en la anterior. Los detalles a conocer del avance de un alumno, serán los mismos en la interfaz del *profesor* y en la del *alumno*, por lo que comparten código similar.

El aspecto que comparten ambas interfaces (la del *profesor* y la del *alumno*) son similares, con intención de unificar el diseño de toda la aplicación. A continuación, una ilustración sobre la pantalla de detalle del alumno en la interfaz del *alumno*.



La imagen muestra una interfaz de usuario con un fondo morado claro y un patrón abstracto de líneas blancas. El título principal es "Listado de preguntas respondidas". El contenido está organizado en secciones por tema, cada una con un encabezado "TEMA X" y una tabla de datos.

TEMA 1		
Pregunta	Fecha y hora	Intentos
1	2017-05-17 11:26:00	2
2	2017-05-17 11:32:00	1

TEMA 2		
Pregunta	Fecha y hora	Intentos

TEMA 3		
Pregunta	Fecha y hora	Intentos

TEMA 4		
Pregunta	Fecha y hora	Intentos

Figura 49: Detalle de la página de información de alumno de la interfaz de alumno

En esta iteración desarrollaremos otra de las funcionalidades: la matriculación del alumno a un tema. Para ello, permitiremos al alumno introducir un código de



matrícula que habrá recibido del profesor. Si el código es correcto, la matrícula se realizará y el alumno tendrá acceso a las preguntas de dicho tema.

En la siguiente imagen se aprecia el aspecto de la pantalla de matriculación del *alumno*. Se trata de un formulario sencillo con una única entrada de texto para el código proporcionado por el profesor.

Matrícula para un TEMA

Código del TEMA

MATRICULARME

Figura 50: Detalle de la página de matriculación a un tema en la interfaz de alumno

Además, para facilitar al profesor el código de cada tema, tendrá acceso a una pantalla con el listado, véase *Figura 51*.

Listado de temas y códigos

Tema	Código asociado
TEMA 1	1
TEMA 2	2
TEMA 3	3
TEMA 4	4

Figura 51: Detalle de la página del listado de códigos en la interfaz del profesor

6.5. Iteración 5

En esta iteración se pretende cubrir la necesidad de controlar los registros de todos los alumnos, para poder evitar que usuarios ajenos al sistema puedan entrar y generar alguna molestia o confusión. A estas alturas del desarrollo, deberemos plantearnos ciertas dudas a resolver:

- Un profesor puede impartir en varias clases y en diferentes cursos, por lo que no se le puede asignar un único curso y clase. ¿Cómo hacer para que le aparezcan en la lista de pendientes de confirmar sus alumnos?
 - **Decisión tomada:** confiando en la responsabilidad del profesor, se le mostrarán todos los alumnos pendientes de confirmar y confirmará aquellos que pertenezcan a sus aulas y sean correctos.
- Cuando un alumno se registra, hasta que su usuario no sea confirmado por el profesor, ¿deberá tener acceso al sistema?
 - **Decisión tomada:** para poder evitar confusiones y malos usos del sistema, hasta que el perfil de un alumno no sea confirmado por algún profesor, el inicio de sesión fallará.
- ¿Existirá la opción de echar atrás una confirmación?
 - **Decisión tomada:** la idea de otorgar el privilegio de deshacer la confirmación de un perfil de alumno al profesor ha sido descartada. Con esta limitación se pretende minimizar el impacto de un error. Si alguna confirmación fuera errónea, será el administrador quien deberá solucionarlo.

Tal y como se definió en el diseño de pantallas, la manera más intuitiva de realizar una confirmación de un estudiante por parte del profesor es un listado

interactivo. Es decir, pinchando en una de las líneas de la tabla correspondiente a cada alumno, se realizará la confirmación de dicho alumno. La siguiente imagen refleja el aspecto final de la pantalla de confirmación con el listado de alumnos pendientes de confirmar.

Nombre	Apellido	Email	Curso	Clase
alumno 1	fvd	fdfv@fvfs.com	1	A
alumno 2	kjh	kjh@lj.com	1	B
alumno3	alumno3	skjha@kjshda.com	2	F

Figura 52: Detalle de la página de confirmación de registro de la interfaz de profesor

Cuando el profesor desee confirmar a algún alumno de la lista, deberá *clickar* sobre él y confirmarlo. Como medida de seguridad y para evitar algún posible error, se ha añadido un cuadro de diálogo *JavaScript* que deberá aceptar para terminar la confirmación.

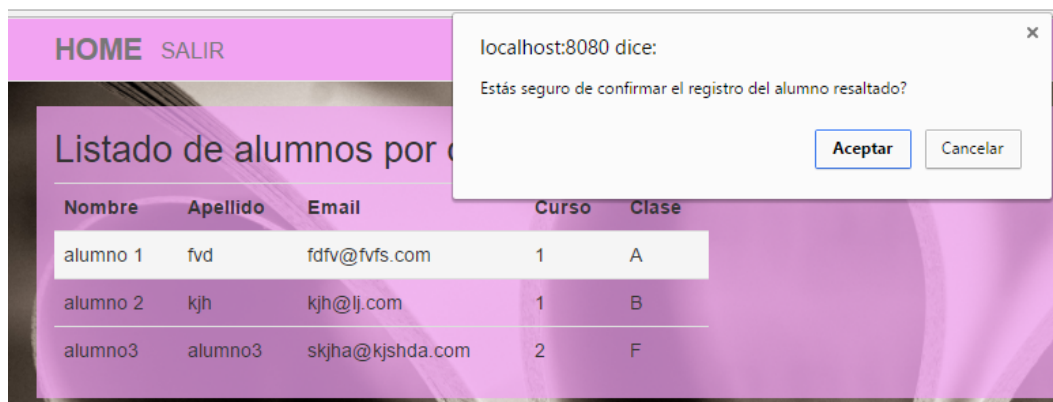


Figura 53: Detalle del cuadro de diálogo en la interfaz de profesor



Una vez confirmado, el alumno desaparecerá de la lista de pendientes y pasará a la lista de consulta de información del profesor.

6.6. Iteración 6

En esta iteración se implementará el juego de preguntas para el alumno. Para poder acceder a la pantalla de las preguntas, el usuario de tipo alumno deberá escoger uno de los temas en los que está matriculado. De esta manera, accederá únicamente a las preguntas asociadas a ese tema concreto.

El primer reto a resolver en esta iteración es el del formato de las preguntas. Existe un formato estándar para la especificación de preguntas y pruebas: *QTI*. Define el formato para la representación de contenidos y los resultados de evaluaciones. El objetivo es crear un lenguaje por etiquetas que permita interoperabilidad entre diferentes sistemas.

A continuación (véase *Figura 54*), un ejemplo de pregunta con una única respuesta a elegir. Como se puede observar, resulta inevitable pensar en la similitud con el lenguaje de marcas *XML*. De hecho, el estándar basa su lenguaje en etiquetas de manera que las preguntas y respuestas sean legibles a simple vista.

```

<itemBody>
  <choiceInteraction responseIdentifier="MR01" shuffle="true" maxChoices="10">
    <prompt>
      How to make chocolate milk. Select the combination of steps that lead to a nice glass of hot and steamy chocolate mil
    </prompt>
    <simpleChoice identifier="C01" fixed="false">Take a lighter</simpleChoice>
    <simpleChoice identifier="C02" fixed="false">Open the gas on the stove</simpleChoice>
    <simpleChoice identifier="C03" fixed="false">Light the gas</simpleChoice>
    <simpleChoice identifier="C04" fixed="false">Pour the milk in the pan</simpleChoice>
    <simpleChoice identifier="C05" fixed="false">Add 2 tea spoons of cocoa into the mug</simpleChoice>
    <simpleChoice identifier="C06" fixed="false">Add 2 tea spoons of sugar into the mug</simpleChoice>
    <simpleChoice identifier="C07" fixed="false">Add 2 spoons of water into the mug</simpleChoice>
    <simpleChoice identifier="C08" fixed="false">
      Stir the water, cocoa and sugar until the mixture is smooth
    </simpleChoice>
    <simpleChoice identifier="C09" fixed="false">Put the pan with milk on the stove</simpleChoice>
    <simpleChoice identifier="C10" fixed="false">Pour the boiling milk into the mug</simpleChoice>
    <simpleChoice identifier="C11" fixed="false">
      Put the mug with the mixture and milk into the microwave
    </simpleChoice>
    <simpleChoice identifier="C12" fixed="false">Add milk to the mug with the smooth mixture</simpleChoice>
    <simpleChoice identifier="C13" fixed="false">
      Add cold milk from the fridge into the mug with smooth mixture
    </simpleChoice>
    <simpleChoice identifier="C14" fixed="false">
      Set the microwave on 700 Watt and set the timer to 2 minutes
    </simpleChoice>
  </choiceInteraction>

```

Figura 54: Ejemplo documento con especificación QTI

Resulta ser una herramienta muy potente utilizada por grandes plataformas como *Moodle*, pero no encaja con las necesidades de este proyecto. No existe ninguna dificultad en generar un archivo utilizando el estándar *QTI* para almacenar las preguntas y respuestas del juego, el impedimento surge en la fase de conversión del lenguaje a una representación gráfica de la interfaz. Existen dos opciones para solucionar este problema:

- Generar un archivo de estilo para traducir el lenguaje por etiquetas a un código comprensible por el navegador
 - Esta opción conlleva un trabajo y esfuerzo que no están contemplados en la planificación y supondría un retraso importante en la entrega. Existen plantillas de pago que solucionan el problema.
- Utilizar herramientas que transforman el lenguaje en un fichero comprensible por el navegador

- Estas herramientas son de pago e integrarlas en el proyecto conllevaría una modificación en la planificación temporal y económica no asumible en este punto del proyecto.

Dadas las circunstancias y la imposibilidad de integrar el estándar al proyecto, se ha realizado una modificación en la base de datos para soportar la nueva modificación de estructura. Para ello, se ha añadido una tabla para cada tipo de pregunta, dado que no comparten una estructura común. De esta manera, la tabla que almacena los identificadores de las distintas preguntas, contendrá el tipo de pregunta al que pertenece.

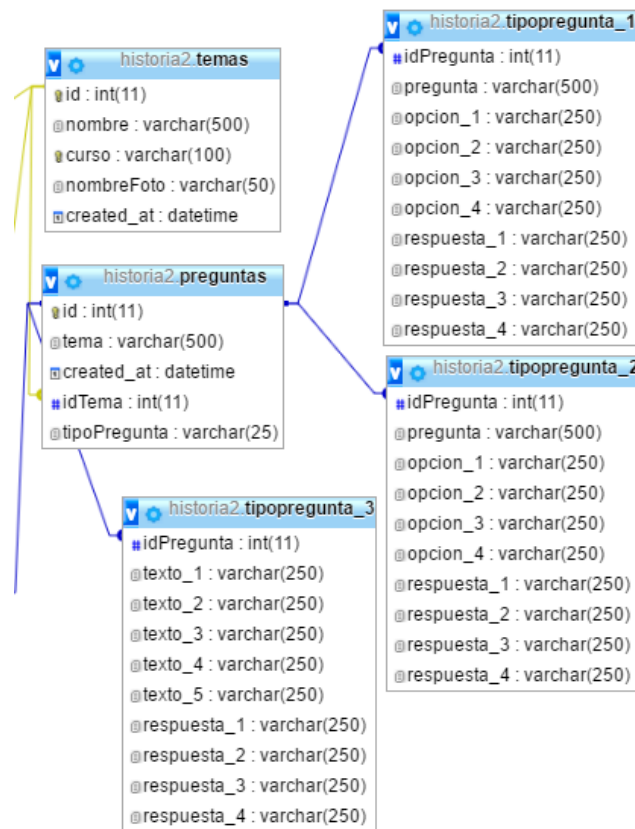


Figura 55: Modificación de la base de datos

Para cada tipo de pregunta, el sistema ejecutará un código diferente para la representación en la interfaz.

```
<?php
if($r[1] == 'tipoPregunta_1'){ //cargamos formulario para el tipo de pregunta uno, escoger una o varias opciones
}
<form method="post" id="tipoPregunta_1" onsubmit="validarPregunta1(this)">

  <div class="list-group-item" style="background-color: #D466F7;">
    <p class="h4"> <?php echo $r2[1]; ?> </p>
  </div>
  <div class="list-group-item">
    <input type="checkbox" class="checkbox" name="respuesta_1" id="respuesta_1" value="<?php echo $r2[2]; ?>" />
    <label for="question-1-answers-A"> <?php echo $r2[2]; ?> </label>
  </div>
  <div class="list-group-item">
    <input type="checkbox" class="checkbox" name="respuesta_2" id="respuesta_2" value="<?php echo $r2[3]; ?>" />
    <label for="question-1-answers-B"> <?php echo $r2[3]; ?> </label>
  </div>
  <div class="list-group-item">
    <input type="checkbox" class="checkbox" name="respuesta_3" id="respuesta_3" value="<?php echo $r2[4]; ?>" />
    <label for="question-1-answers-C"> <?php echo $r2[4]; ?> </label>
  </div>
  <div class="list-group-item">
    <input type="checkbox" class="checkbox" name="respuesta_4" id="respuesta_4" value="<?php echo $r2[5]; ?>" />
    <label for="question-1-answers-D"> <?php echo $r2[5]; ?> </label>
  </div>

  <div >
    <input type="submit" class="btn btn-block" value="Responder" />
  </div>
</form>
```

Figura 56: Detalle del código para procesar las preguntas de un tipo



```
} else if($r[1] == 'tipoPregunta_2') //cargamos formulario para el tipo de pregunta dos, ordenar las opciones
?>
<form action="corregir_2.php" method="post" id="tipoPregunta_2">

  <div class="list-group-item" style="background-color: #D466F7;">
    <p class="h4"> <?php echo $r2[1]; ?> </p>
  </div>

  <div class="list-group-item">
    <p><?php echo $r2[2]; ?></p>
    <select id="respuesta_1">
      <option value="" selected="selected">- Orden -</option>
      <option value="1">1</option>
      <option value="2">2</option>
      <option value="3">3</option>
      <option value="4">4</option>
    </select>
  </div>

  <div class="list-group-item">
    <p><?php echo $r2[3]; ?></p>
    <select id="respuesta_2">
      <option value="" selected="selected">- Orden -</option>
      <option value="1">1</option>
      <option value="2">2</option>
      <option value="3">3</option>
      <option value="4">4</option>
    </select>
  </div>

  <div class="list-group-item">
    <p><?php echo $r2[4]; ?></p>
    <select id="respuesta_3">
```

Figura 57: Detalle del código para procesar las preguntas de un tipo

Se han generado tres tipos de preguntas:

- Pregunta con múltiples respuestas a elegir. Constará de un enunciado con cuatro posibles opciones de las cuales se podrán escoger una o más.



Ada Lovelace es conocida por...

- ser la primera programadora de ordenadores.
- ser una escritora importante.
- ser una sufragista.
- luchar por los derechos de la mujer.

Responder

Figura 58: Detalle del tipo de pregunta 1

- Pregunta con cuatro respuestas a ordenar. La formarán un enunciado y cuatro opciones. Las opciones deberán ser etiquetadas con un número del uno al cuatro.

Ordena las filas de un programa escrito en ADA, nombre puesto en honor a Ada Lovelace.

```
Put_Line(Hola, mundo);  
- Orden - ▾  
  
begin  
2 ▾  
  
procedure Hello is  
- Orden - ▾  
  
end  
- Orden - ▾
```

Responder

Figura 59: Detalle del tipo de pregunta 2



- Rellenar huecos. Cuatro huecos a completar con una o varias palabras.

Rellena los huecos

Describo conceptos como el

y la

Solia definirse a si misma como

y metafisica algo bastante mas avanzado para su

Responder

Figura 60: Detalle del tipo de pregunta 3

El aspecto general de la hoja de preguntas que obtiene el alumno al seleccionar un tema viene reflejado en la *Figura 61*.

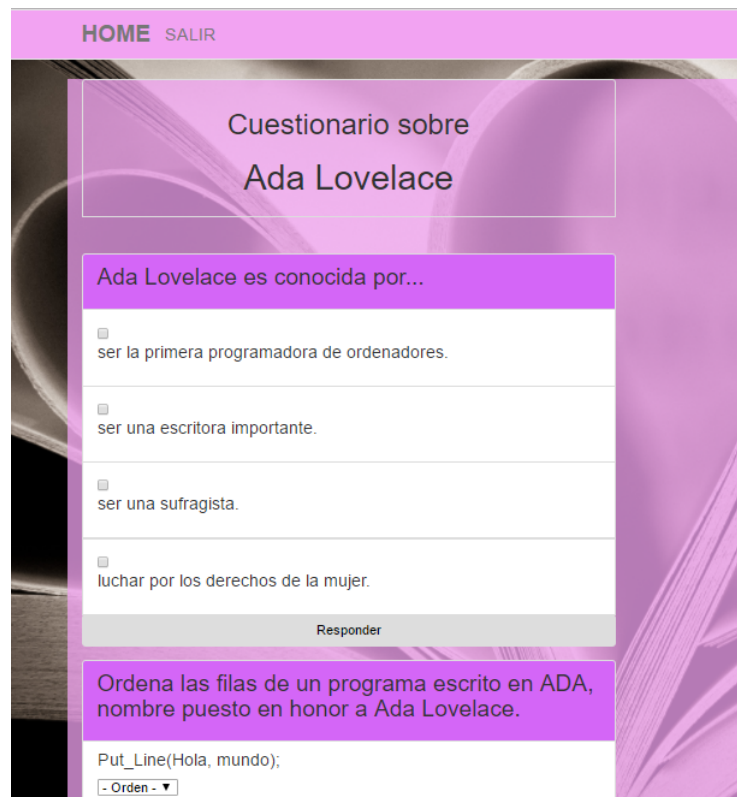


Figura 61: Interfaz del alumno al seleccionar un tema

6.7. Iteración 7

En esta iteración pretende poblarse la base de datos con temas y preguntas. A petición del cliente, la temática de la página web será la mujer a lo largo de la historia. Por este motivo, se han recopilado nombres de mujeres trascendentes a lo largo de la historia y realizado un pequeño estudio sobre sus hazañas y logros.

Los nombres escogidos para el desarrollo de preguntas son los siguientes:

- Ada Lovelace (15) (1815-1852): matemática y escritora británica, considerada la primera programadora de ordenadores.
- Emmeline Pankhurst (16) (1858-1928): activista política británica y líder del



movimiento sufragista.

- Frida Kahlo (17) (1907-1954): pintora mexicana considerada símbolo del feminismo.
- Angela Davis (18) (1944-actualidad): filósofa, política marxista, activista afroamericana y profesora de universidad.

Para realizar la población de la base de datos, se ha recogido información sobre estas cuatro mujeres y adaptado al formato de pregunta del juego.

6.8. Iteración 8

En esta iteración se implementa el código que solventa el requisito de las calificaciones. Tanto el profesor como el propio alumno, deberán conocer las calificaciones correspondientes a cada tema. Para ello, se ha creado una fórmula que se aplicará para establecer una puntuación a un alumno en un tema concreto.

$$(PUNTOS OBTENIDOS / PUNTOS TOTALES) * 10$$

Con dicha fórmula, se obtendrá una calificación sobre diez puntos. De esta manera, logramos unificar las calificaciones obtenidas haciéndolas independientes del número de preguntas que compongan cada tema.

Uno de los requisitos del cliente es el de valorar el esfuerzo del alumno a la hora de realizar los cuestionarios, por lo que existe una puntuación intermedia. Es decir, si un alumno consigue responder de manera correcta en el primer intento, recibirá un punto de la pregunta. Pero si en cambio, lo logra en el segundo intento, se le atribuirá medio punto. Una respuesta correcta con más de dos intentos no se valorará.

- 1 intento: 1 punto
- 2 intentos: 0.5 puntos
- 3 intentos o más: 0 puntos

Aunque existe la posibilidad de que un alumno deje a medias un cuestionario, las calificaciones se harán sobre el número de preguntas del tema. De esta manera, aunque las preguntas que hayan sido respondidas obtengan la puntuación total, puede no darse la misma calificación en la media del tema.

A continuación (véase *Figura 62*), el aspecto de la interfaz con un temario que no ha sido respondido en su totalidad.



HOME SALIR		
Listado de preguntas respondidas		
Ada Lovelace		
Pregunta	Fecha y hora	Intentos
1	2017-05-17 11:26:00	2
2	2017-05-17 11:32:00	1
Número de preguntas	Puntos obtenidos	Calificación
12	1.5	1.25/10
Emmeline Pankhurst		
Pregunta	Fecha y hora	Intentos

Figura 62: Interfaz del alumno de su historial

7. Validación

Tras el desarrollo de cada iteración, tal y como se especifica en la 2.5 *Planificación temporal* inicial, se procederá a realizar una fase de pruebas para comprobar la contundencia y coherencia del código. Tal y como se describe en la sección 5.4 *Diseño de pruebas*, se clasificarán las pruebas por los niveles de comprobación:

- Pruebas unitarias
- Pruebas de integración
- Pruebas de sistema
- Pruebas de aceptación

En esta sección del proyecto se recogerán aquellas pruebas que pertenezcan al nivel de Pruebas unitarias y de integración, y se dividirán por iteraciones del desarrollo.

Para unificar la definición de las pruebas, se recogerán en tablas con el identificador de prueba, descripción de la misma, resultado esperado, resultado obtenido y correcciones realizadas. De esta manera obtenemos, de un vistazo, conocer el resultado de todas las pruebas realizadas.

7.1. Pruebas para la iteración 1

Para el inicio de sesión, común a todos los tipos de usuario, tendremos dos campos de texto y un botón que compondrán el formulario. Los campos corresponden al nombre de usuario y contraseña.



Id	Descripción de la prueba	Resultado esperado	Resultado obtenido
1	El usuario introduce un nombre y contraseña correctos	Inicio de sesión correcto	✓
2	El usuario introduce un nombre correcto y una contraseña incorrecta	Fallo en el inicio de sesión	✓
3	El usuario introduce un nombre incorrecto	Fallo en el inicio de sesión	✓
4	El usuario deja sin rellenar alguno de los dos campos	Fallo en el inicio de sesión	✓
5	El usuario introduce caracteres extraños en alguno de los dos campos	Fallo en el inicio de sesión	✓

Tabla 9: Pruebas Inicio de sesión

7.2. Pruebas para la iteración 2

En esta iteración deberá cubrirse la funcionalidad de registro para los dos tipos de usuario, *alumno* y *profesor*. Las pruebas realizadas se dividirán en el registro para cada tipo de usuario.

Registro profesor

En esta pantalla, el formulario lo componen varios cuadros de texto: nombre, apellidos, contraseña y confirmación de contraseña.

Id	Descripción de la prueba	Resultado esperado	Resultado obtenido
6	El profesor inserta de manera correcta todos los datos para el registro	Registro exitoso	✓
7	El profesor deja sin rellenar alguno de los campos	Registro fallido	✓
8	El profesor rellena todos los campos pero la contraseña y su confirmación no coinciden	Registro fallido	✓
9	El profesor inserta algún carácter extraño en alguno de los campos	Registro fallido	X

Tabla 10: Pruebas para el registro del profesor

En la prueba número 9 el sistema no ha dado el resultado esperado, permite al



profesor insertar cualquier carácter en los campos. Para solucionar este comportamiento indeseado, se añaden sentencias de control y validación para los diferentes campos del formulario. En el caso de los campos *Nombre* y *Apellidos*, sólo permitiremos letras y espacios, para ello, añadimos una sentencia de control en el código (véase *Figura 63*) y un mensaje de error cuando el usuario introduzca un carácter erróneo (véase *Figura 64*).

```
//Comprobaciones
if (!preg_match("/^[a-zA-Z ]*$/", $nombreProfesor) || !preg_match("/^[a-zA-Z ]*$/", $apellidoProfesor)) {
    print "<script>alert(\"Los campos deben ser correctos. Nombre y Apellidos sólo pueden contener letras y espacios.\");window.location='../registroProfesor.php';</script>";
}
```

Figura 63: Detalle del código

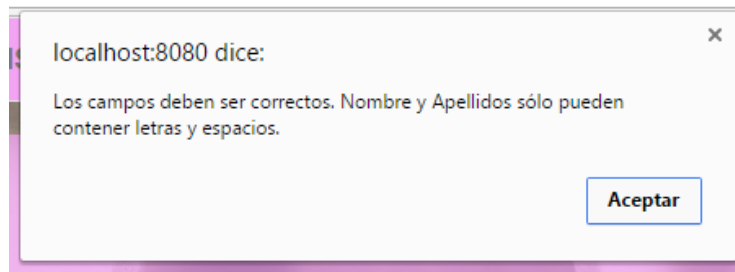


Figura 64: Detalle del cuadro de diálogo en la interfaz de usuario

De esta manera logramos solventar el error de introducir caracteres erróneos en alguno de los campos, y además, obtenemos un nivel de seguridad más alto, evitando inyecciones SQL en el registro del profesor.

Registro alumno

En esta pantalla, el formulario lo componen varios cuadros de texto: nombre, apellidos, correo electrónico, curso, clase, contraseña y confirmación de contraseña.

Id	Descripción de la prueba	Resultado esperado	Resultado obtenido
10	El alumno inserta de manera correcta todos los datos para el registro	Registro exitoso	✓
11	El alumno deja sin rellenar alguno de los campos	Registro fallido	✓
12	El alumno rellena todos los campos pero la contraseña y su confirmación no coinciden	Registro fallido	✓
13	El alumno inserta algún carácter extraño en alguno de los campos	Registro fallido	X

Tabla 11: Pruebas para el registro del alumno

En este módulo encontramos el mismo error que en el registro del profesor: los campos no han sido validados para evitar caracteres extraños. La solución esta vez también consiste en añadir sentencias de control en el código para evitar estos caracteres. En los campos *Nombre* y *Apellidos* sólo permitiremos letras y espacios, el email debe tener un formato correcto y el curso debe ser únicamente numérico. Para ello, añadimos las sentencias de validación en el código (véase *Figura 65*) y mostramos un cuadro de diálogo para cada error (véase *Figura 66*).

```

//Comprobaciones
if (!preg_match("/^[a-zA-Z ]*$/",$_POST["nombreAlumno"]) || !preg_match("/^[a-zA-Z ]*$/",$_POST["apellidoAlumno"])) {
    print "<script>alert(\"Los campos deben ser correctos. Nombre y Apellidos sólo pueden contener letras y espacios.\");window.location='../registroAlumno.php';</script>";
}
if(!filter_var($_POST["emailAlumno"], FILTER_VALIDATE_EMAIL)){
    print "<script>alert(\"El email debe tener un formato correcto.\");window.location='../registroAlumno.php';</script>";
}
if (!preg_match("/^[1-9]*$/",$_POST["cursoAlumno"]) || !preg_match("/^[a-zA-Z ]*$/",$_POST["claseAlumno"])) {
    print "<script>alert(\"Los campos deben ser correctos. El curso debe ser numérico y la clase debe contener únicamente letras.\");window.location='../registroAlumno.php';</script>";
}
  
```

Figura 65: Detalle del código

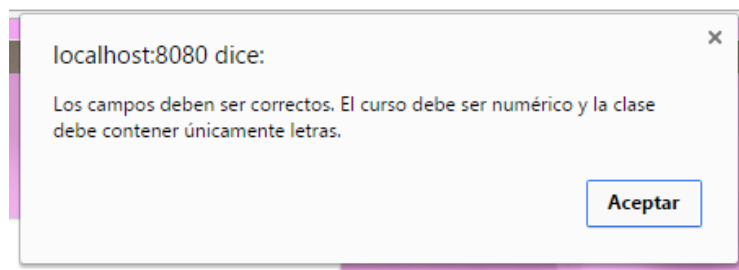


Figura 66: Detalle del cuadro de diálogo en la interfaz de usuario

7.3. Pruebas para la iteración 3

En esta iteración se ha desarrollado la consulta de alumnos para el profesor. Deberá tener acceso a un listado de alumno en su página inicial en la que pinchando sobre un alumno concreto, tenga información sobre el avance del alumno en el juego.

Id	Descripción de la prueba	Resultado esperado	Resultado obtenido
14	El profesor se encuentra en la página inicial	El listado de alumnos le aparece de manera correcta	✓
15	El profesor pincha en un alumno concreto	El detalle del alumno le aparece de manera correcta	✓
16	El alumno sobre el que ha pinchado no tiene temas matriculados	Al profesor no le aparece nada más que el nombre del alumno sobre el que ha pinchado	✓
17	El alumno tiene temas matriculados y además alguna pregunta respondida	El profesor puede ver los temas a los que está matriculado y además las preguntas a las que ha respondido	✓
18	El alumno tiene temas matriculados pero no ha respondido a ninguna pregunta	Al profesor le aparecen todos los temas a los que está matriculado el alumno, aunque no haya respondido a ninguna pregunta	✓

Tabla 12: Pruebas para la consulta de alumnos por parte del profesor

7.4. Pruebas para la iteración 4

En el desarrollo de este módulo, el alumno debe tener acceso a sus propios avances dentro del juego.

Id	Descripción de la prueba	Resultado esperado	Resultado obtenido
19	El alumno se encuentra en la página inicial	Tiene la opción de ver su historial	✓
20	El alumno pincha en su historial	Su historial se muestra de manera correcta	✓
21	El alumno no tiene temas matriculados	Al alumno no le aparece nada	✓
22	El alumno tiene temas matriculados y además alguna pregunta respondida	El alumno puede ver los temas a los que está matriculado y además las preguntas a las que ha respondido	✓
23	El alumno tiene temas matriculados pero no ha respondido a ninguna pregunta	Al alumno le aparecen todos los temas a los que está matriculado el alumno, aunque no haya respondido a ninguna pregunta	✓

Tabla 13: Pruebas para la consulta de historial por parte del alumno

Deberemos realizar las pruebas unitarias de otra funcionalidad desarrollada en esta iteración: la matriculación del alumno a un tema. El formulario para realizar la matrícula lo compone un único texto de entrada en el cual el alumno deberá insertar un código proporcionado por el profesor, correspondiente a un tema concreto.

Id	Descripción de la prueba	Resultado esperado	Resultado obtenido
24	El alumno se encuentra en la página inicial	Tiene la opción de matricularse a un tema	✓
25	El alumno pincha en la opción de matricularse	Tiene un acceso correcto para la matriculación	✓
26	El alumno inserta un código válido	Al alumno queda matriculado al tema	✓
27	El alumno inserta un código erróneo	El alumno no se matricula a ningún tema	✓
28	El alumno no inserta ningún código y acepta el formulario	El alumno no se matricula a ningún tema	✓
29	El alumno inserta algún carácter erróneo	El alumno recibe un mensaje de error	X

Tabla 14: Pruebas para la matriculación a un tema por parte de un alumno

Para solventar el error obtenido a partir de la prueba número 29, se deberá añadir una sentencia para validar el texto de entrada del alumno. El código no permitirá un código con caracteres extraños, es decir, deberá ser alfanumérico (véase *Figura 67*). Si el código introducido no coincide con las restricciones, se mostrará una ventana con el error (véase *Figura 68*).

```

//Comprobaciones
if (!preg_match("/^[a-zA-Z ]*$/",$_POST["codigoTema"]) || !preg_match("/^[1-9 ]*$/",$_POST["codigoTema"])) {
    print "<script>alert(\"El campo código debe ser alfanumérico.\");window.location='../matricularse.php';</script>";
}
  
```

Figura 67: Detalle del código

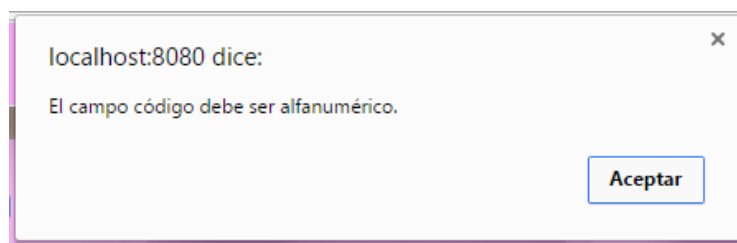


Figura 68: Detalle del cuadro de diálogo en la interfaz de alumno

7.5. Pruebas para la iteración 5

El requisito a cumplir es el de permitir al profesor confirmar registros de alumnos para que puedan tener acceso a sus perfiles. El primer paso es el de mostrar un listado de alumnos pendientes de confirmar en orden por curso y clase, de esta manera, facilitamos la búsqueda al profesor. Una vez seleccionado el alumno a confirmar, se realiza la actualización en la base de datos.

Id	Descripción de la prueba	Resultado esperado	Resultado obtenido
30	El profesor accede al listado de alumnos pendientes de confirmar	El listado corresponde con los alumnos pendientes de confirmar	✓
31	Se selecciona el alumno a confirmar	Aparece el cuadro de diálogo y el alumno sigue seleccionado en la tabla	✓
32	Si acepta la confirmación	La base de datos se actualiza de manera correcta	✓
33	Si cancela la confirmación	La base de datos no se modifica	X

Tabla 15: Pruebas para la confirmación del registro de un alumno por parte de un profesor

Se ha detectado un error en el cuadro de diálogo *JavaScript*. Si el usuario pincha en “Cancelar” el alumno es modificado en la base de datos. Para solucionarlo, se ha añadido una sentencia en el código para volver a cargar la página sin llamar al método que actualiza la base de datos.



```
<script>
function confirmarAlumno(idAlumno){
    if (confirm('Estás seguro de confirmar el registro del alumno resaltado?')) {
        // confirmamos alumno
        window.location='php/confirmarAlumnos.php?idAlumno='+idAlumno;
    } else {
        // no hacemos nada
        window.location='confirmarAlumnos.php';
    }
}
</script>
```

Figura 69: Detalle del código

7.6. Pruebas para la iteración 6

En esta iteración se pretende implementar el código que permita al alumno responder preguntas de cada tema. Las pruebas relacionadas serán en torno a la visualización y gestión de las mismas.

Id	Descripción de la prueba	Resultado esperado	Resultado obtenido
34	El alumno accede a un tema por primera vez	Visualiza todas las preguntas del tema	✓
35	El alumno accede a un tema del cual ya ha respondido alguna pregunta de manera correcta	Visualiza las preguntas pendientes de responder (o aquellas que no ha respondido de manera correcta)	✓
36	El alumno accede a un tema que ya ha superado (ha respondido a todas las preguntas de manera satisfactoria)	No aparece ninguna pregunta en pantalla	✓
37	El alumno accede a un tema para visualizar las preguntas	Las preguntas de diferentes tipos se muestran de manera correcta	✓
38	El alumno responde a una pregunta de manera correcta	La base de datos es actualizada de manera correcta y la pregunta desaparece de la pantalla	X
39	El alumno responde a una pregunta de manera incorrecta	La base de datos se actualiza de manera correcta y la pregunta permanece en pantalla	X

Tabla 16: Pruebas para el correcto funcionamiento del juego

El sistema no ha superado varias pruebas. La primera corresponde a la siguiente situación: el alumno selecciona un tema y responde a una pregunta de manera correcta. El fallo se ha detectado en la visualización de las preguntas. Una vez que la respuesta a una pregunta es correcta, dicha pregunta debe desaparecer de la pantalla. La solución ha sido volver a cargar la pantalla, de manera que la propia consulta inicial se encarga de hacer únicamente visibles las preguntas que queden por responder de manera correcta.



La segunda prueba que no ha sido superada corresponde a los intentos de una pregunta. Cada vez que el alumno respondía de manera incorrecta a una, la base de datos no era actualizada de manera correcta. Por cada respuesta incorrecta, el contador de intentos debe incrementar en uno.

7.7. Pruebas para la iteración 7

Dado que esta iteración no genera un nuevo prototipo del sistema, no se han realizado pruebas.

7.8. Pruebas para la iteración 8

Esta última iteración cumple el requisito de calificar los cuestionarios realizados. Las calificaciones se realizarán de manera automática siguiendo la fórmula definida en el apartado de desarrollo (véase *Sección 6*). Las pruebas centrarán su atención en comprobar que dicha fórmula funciona de manera correcta.



Id	Descripción de la prueba	Resultado esperado	Resultado obtenido
40	El alumno responde en el primer intento de manera correcta a todas las preguntas de un tema	El alumno obtiene la calificación máxima	✓
41	El alumno no obtiene puntuación en ninguna de las preguntas (responde de manera correcta en el tercer o mayor intento)	El alumno obtiene una calificación de cero	✓
42	El alumno obtiene puntuaciones intermedias (respuestas correctas en el segundo intento)	La calificación obtenida es correcta sobre diez puntos	✓
43	El alumno no ha respondido a alguna de las preguntas del cuestionario	La calificación es correcta sobre diez puntos	✓

Tabla 17: Pruebas para el correcto funcionamiento de las calificaciones



8. Conclusiones

En esta sección se explicarán las conclusiones obtenidas a raíz del desarrollo del Proyecto de Fin de Grado. Por un lado, se realizará un análisis sobre las diferencias entre la planificación inicial y la realidad del proceso. Por otro lado, una opinión personal sobre las aportaciones personales del desarrollo y por último, ampliaciones posibles para un futuro.

8.1. Gestión del proyecto

En este apartado se pretende hacer un análisis sobre el tiempo real dedicado al proyecto sobre el estimado al principio de este. Estimar unos tiempos concretos para el desarrollo de cada fase del proyecto me resultó una tarea complicada, dada mi inexperiencia en este tipo de tareas. Estimé la mayor parte de esfuerzo y tiempo en la fase de implementación del proyecto, nada más lejos de la realidad.

Tras finalizar el proyecto he podido comprobar cómo aquellas partes que estaban más desarrolladas en la fase de planificación, análisis y diseño, han sido más sencillas de implementar. Dedicar un mayor tiempo a las fases tempranas del proyecto, resta un tiempo considerable de la etapa de implementación.

Teniendo en cuenta el tiempo reducido para realizar el trabajo (comienzo 14/02/2017 - finalización 5/06/2017), aunque me he exigido unas jornadas de trabajo largas, era predecible un retraso en la entrega. Comentar que la dedicación no ha sido exclusiva al proyecto, ha sido compatibilizado con un trabajo a media jornada. Los tiempos estimados para cada tarea se ajustaron lo más posible en la fase de planificación para poder realizar la entrega y presentación del proyecto en la convocatoria de junio. Es importante entender la necesidad de realizar la entrega en dicha convocatoria, dada la intención de cursar un máster de la *UPV/EHU* que tiene comienzo en septiembre.

Una vez concluidas todas las fases del proyecto, pasamos a realizar la comparativa. Mediante una representación gráfica pondremos a la par las horas estimadas para cada tarea junto a las reales (aproximación) y podremos observar el desajuste entre ellas.

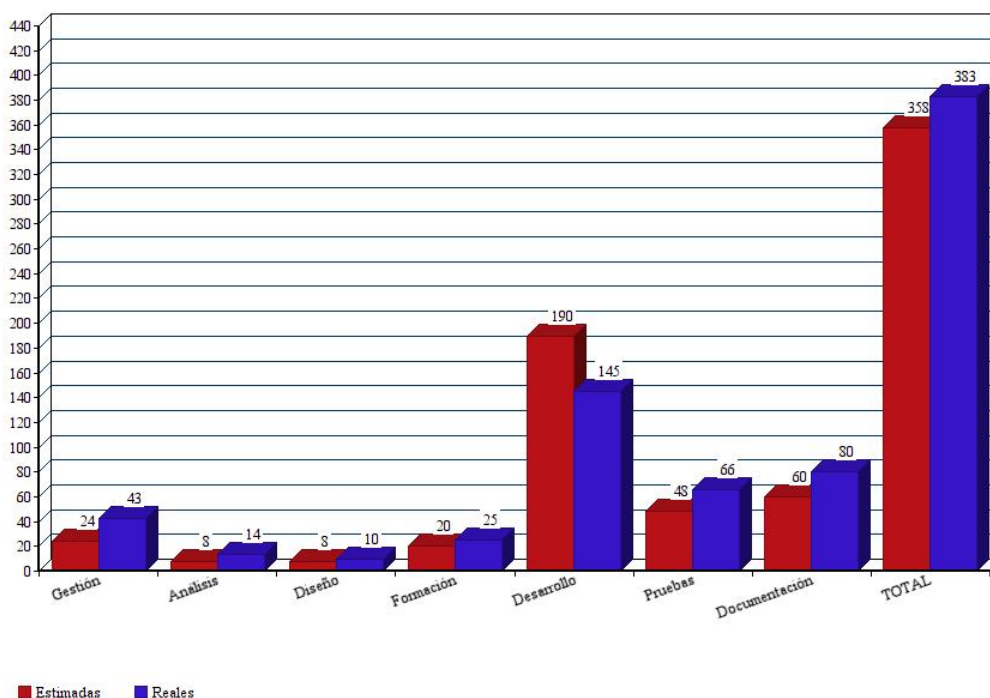


Figura 70: Gráfica comparativa entre horas estimadas y reales

Como podemos observar en la gráfica (véase Figura 70), tal y como se explica al principio de este apartado, las horas reales en las fases tempranas aumentan considerablemente, en la de desarrollo, en cambio, disminuyen. Para la fase de pruebas se habían estimado 6 horas por cada iteración pero tal y como la gráfica refleja, las reales han sido mayores.

Si observamos la descompensación en el total de las horas, no existe una diferencia significativa. Si bien es verdad que las reales superan a las estimadas, tan



sólo en 25 horas por lo que la estimación inicial no fue demasiado lejana a la realidad.

8.2. Diseño del proyecto

Resulta interesante realizar una comparativa sobre el diseño inicial realizado y el resultado final del mismo. En este caso concreto, analizaremos la evolución del diseño de la base de datos. Con los requisitos y funcionalidades definidas se realizó el diseño de la base de datos que soportaría los datos del sistema.

Las modificaciones en la estructura no han sido demasiado significativas más allá de añadir atributos, a excepción de la gestión de preguntas. En un principio se contempló la posibilidad de utilizar estándares de formato de preguntas como el *QTI* pero tras una fase de formación y pruebas se descartó. En lugar de utilizar ningún tipo de estándar, se añadió a la estructura de la base de datos lo necesario para soportar tres tipos de preguntas.

A continuación, la estructura de la base de datos definitiva (véase *Figura 71*).

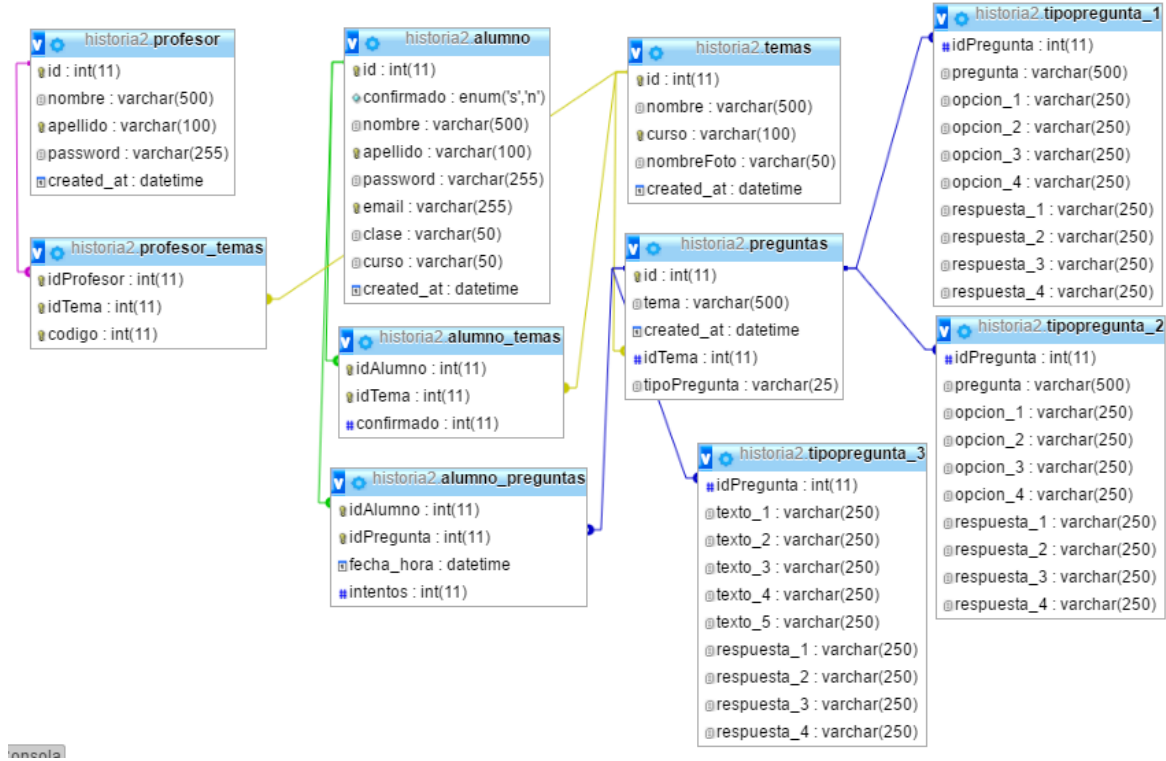


Figura 71: Estructura de la base de datos definitiva

En cuanto al diseño de pantallas, el aspecto y diseño del sistema no ha variado demasiado más que para añadir funcionalidades y opciones. Esto se debe a la participación del cliente a lo largo de todo el proyecto. Una vez realizado el diseño del proyecto fue validado por el cliente, con la petición expresa de no modificar demasiado el aspecto de las pantallas. El diseño ha sido pulido y ajustado a las necesidades de la aplicación, obteniendo un buen resultado.

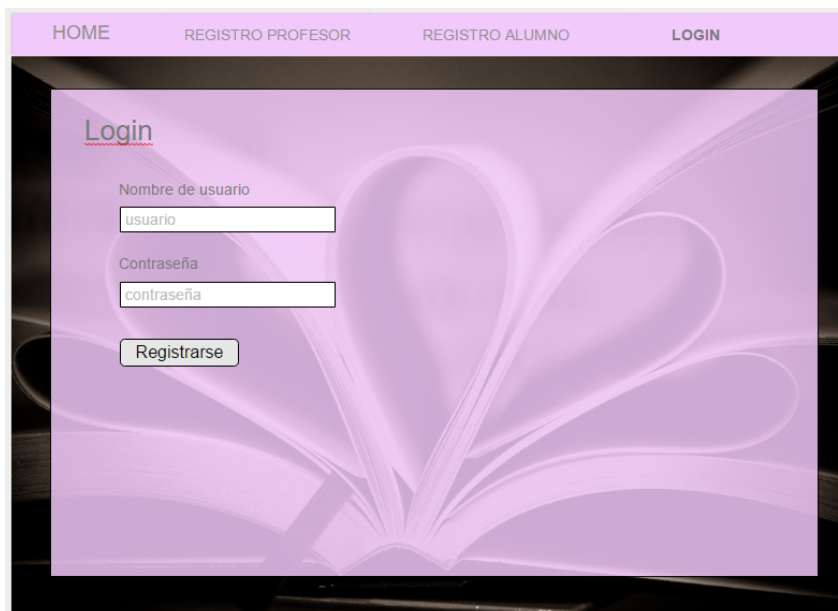


Figura 72: Diseño de la pantalla de Login

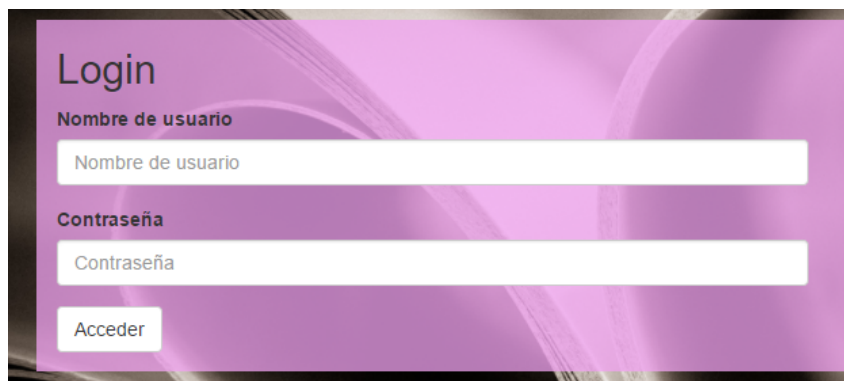


Figura 73: Pantalla definitiva de Login

8.3. Trabajo futuro

En este apartado se describirán aquellas funcionalidades que no han sido implementadas en este proyecto pero mejorarían el sistema de manera significativa. La estructura generada ha sido pensada para poder incluir estas ampliaciones, así

como otras posibles modificaciones.

La primera ampliación más natural sería crear una aplicación móvil para su uso en el ámbito educacional. Cada vez más aulas complementan sus metodologías de estudio con aparatos móviles como tabletas. Esto es una realidad por lo que ampliar el mercado de este producto al mundo del dispositivo móvil sería una buena opción. El funcionamiento sería exactamente el mismo que el de la página web, todos los usuarios tendrían las mismas funcionalidades y roles dentro de la aplicación móvil.

Una de las funcionalidades con la que no he quedado muy acuerdo es la gestión de los alumnos por parte del profesor. Tal y como está ahora, los alumnos realizan sus registros y deben añadir su *curso* y *clase* al formulario. De esta manera, cuando un profesor entra en la opción de *Confirmar registros de alumnos* debe buscar en una lista de alumnos aquellos que sean de su clase. Para solucionar este aspecto se han identificado dos opciones:

- Crear filtros de búsqueda en la lista. Esta es la opción más sencilla de implementar pero, tal vez, no la más adecuada. De esta manera la página debe cargar todos los alumnos y los filtros de búsqueda tardarían unos segundos en realizar la consulta en la base de datos y mostrarla en la interfaz.
- El profesor puede crear clases. Con esta opción, el profesor tendrá una nueva funcionalidad para crear aulas virtuales en su perfil. Los alumnos deberán unirse a estas aulas, mediante una matriculación o en el mismo registro. Como medida de seguridad el profesor deberá validar la entrada de los alumnos en su aula, para verificar la autenticidad de los usuarios.

Otra mejora en la que podemos pensar es la de añadir preguntas y temas por parte del profesor. En un primer momento se pensó en añadir la funcionalidad al catálogo de este proyecto pero la problemática generada por el formato de pregun-

tas hizo que se descartara. Conservando el formato de las preguntas tal y como está en el proyecto, añadir preguntas por parte del profesor sería una tarea sencilla. Generar un formulario diferente por cada tipo de pregunta y rellenarlo para añadir una pregunta a un tema concreto. Para usuarios más avanzados o mediante plataformas y herramientas, se podrían considerar los ficheros *xml* como opción de inserción de preguntas.

8.4. Conclusión personal

Una vez concluido el proyecto toca hacer una reflexión sobre las aportaciones y experiencias obtenidas. Como primer desarrollo de un proyecto real la experiencia ha sido totalmente positiva. La elección del uso de metodologías ágiles ha sido, sin duda, todo un acierto. Me ha permitido conocer en todo momento la opinión del cliente sobre el producto evitando posibles mal entendidos y correcciones en fases más avanzadas del proyecto.

La necesidad de desarrollo de las fases más tempranas del proyecto como el análisis y el diseño, ha confirmado una de las bases que se repiten a lo largo del grado: resultan ser imprescindibles e incluso más importantes que otras fases del un proyecto. La realización de una buen análisis del problema (y su posterior validación con el cliente) y un diseño contundente, conlleva a una implementación mucho más sencilla y coherente. Comenzar la implementación del sistema sin una buena base previa induce a un código menos claro y un gasto de recursos innecesario.

Sin lugar a duda, la realización de este proyecto me ha ayudado a ganar una experiencia que me acerca más al mundo laboral, donde la elaboración de una buena documentación aumenta la calidad del producto.

Como conclusión más personal describiría mi aumento de confianza para afrontar retos importantes. Hasta ahora, no me creía capaz de realizar un proyecto de



inicio a fin, de una manera más formal y profesional. Tras la finalización de este, echar un vistazo atrás y comprobar que el fruto de un esfuerzo y trabajo duros ha sido satisfactorio, incrementa mi realización como profesional.



9. Bibliografía

- [1] Diagramas de Gantt: explicación teórica
<http://es.ccm.net/contents/580-diagrama-de-gantt>
- [2] Herramienta: Creately
<https://creately.com/>
- [3] Herramienta: Tom's planner
<https://www.tomsplanner.es/>
- [4] Herramienta: MoqUps
<https://moqups.com/>
- [5] Herramienta: Sublime Text 3
<https://www.sublimetext.com/>
- [6] Herramienta: Xampp
<https://www.apachefriends.org/es/index.html>
- [7] Herramienta: Chrome
<https://www.google.es/>
- [8] Herramienta: TeXstudio
<http://www.texstudio.org/>
- [9] Roger S. Pressman [2001] *Ingeniería del software: un enfoque práctico* University of Connecticut: S.A. MCGRAW-HILL / INTERAMERICANA DE ESPAÑA
- [10] Análisis de costes directos e indirectos
<http://www.elderecho.com/tribuna/contable/costesdirectoscostesindirectos11685180004.html>



- [11] Antecedentes: Testeando
<http://www.testeando.es/>
- [12] Antecedentes: Sopasletras
<http://www.sopasletras.com/>
- [13] Antecedentes: Trivinet
<https://www.trivinet.com/>
- [14] Craig Larman [2002] *UML Y PATRONES Una introducción al análisis y diseño orientado a objetos y al proceso unificado* Pearson Education, Inc
- [15] Ada Lovelace
<http://www.mujaresenlahistoria.com/2011/01/la-encantadora-de-numeros-ada-lovelace.html>
- [16] Emmeline Pankhurst
<http://www.mujaresenlahistoria.com/2014/06/la-sufragista-inglesa-emmeline.html>
- [17] Frida Kahlo
<http://www.mujaresenlahistoria.com/2012/02/pintando-su-propia-vida-frida-kahlo.html>
- [18] Angela Davis
<http://www.history.com/topics/black-history/angela-davis>