



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

SISTEMA DE CUESTIONARIOS PARA ESTUDIANTES DE ORTOPEDIA

Alumno: Alberto Delgado del Castillo

Tutor: Prof. D. Arturo Montejo Ráez
Dpto: Departamento de Informática

Julio, 2023

(Página intencionalmente en blanco)



Universidad de Jaén

Departamento de Informática

Don Arturo Montejo Ráez, tutor del Trabajo Fin de Grado titulado: '**Sistema de cuestionarios para estudiantes de ortopedia**', que presenta Don Alberto Delgado del Castillo, otorga el visto bueno para su entrega y defensa en la Escuela Politécnica Superior de Jaén.

Jaén, julio de 2023

El alumno:

El tutor:

Alberto Delgado del Castillo

Arturo Montejo Ráez

(Página intencionalmente en blanco)

Agradecimientos

En primer lugar, me gustaría agradecer a mi familia y amigos por todo el apoyo que me han brindado durante este tiempo para seguir adelante.

También quisiera dar las gracias a mi tutor del TFG, Arturo Montejo Ráez, por aconsejarme y guiarme en todo el camino, con especial atención y dedicación en todo momento.

Finalmente, me gustaría dar mis agradecimientos a todos los profesores que he tenido a lo largo de la carrera, los cuales me han aportado con su trabajo los conocimientos necesarios para la realización de este proyecto.

FICHA DEL TRABAJO FIN DE TÍTULO	
Titulación	Grado en Ingeniería Informática
Modalidad	Proyecto de Ingeniería
Especialidad (solo TFG)	Tecnologías de la Información
Mención (solo TFG)	Sin mención
Idioma	Español
Tipo	General
TFT en equipo	No
Autor/a	Alberto Delgado del Castillo
Fecha de asignación	26/09/2022
Descripción corta	El objetivo de este proyecto es el análisis, diseño e implementación de una plataforma de realización de test adaptados al usuario para el estudio del temario de los cursos de ortopedia de la empresa Cursocot, fomentando una forma interactiva de estudio. La plataforma debe incluir gestión de usuarios y preguntas y pretende implantarse en una empresa real, por lo que el desarrollo estará basado en las necesidades de la misma.

NORMAS APLICADAS EN ESTE DOCUMENTO	
LOCALES	
TFT-UJA:2017	Normativa de Trabajos Fin de Grado, Fin de Máster y otros Trabajos Fin de Título de la Universidad de Jaén (Normativa marco UJA aprobada en Consejo de Gobierno)
TFT-EPSJ:2017	Normativa sobre Trabajos Fin de Grado y Fin de Máster en la Escuela Politécnica Superior de Jaén (Normativa EPSJ aprobada en Junta de Escuela)
TFT-EPSJ	Criterios de evaluación y normas de estilo para TFG y TFM de la Escuela Politécnica Superior de Jaén
NACIONALES E INTERNACIONALES	
ISO 2145:1978	Documentación - Numeración de divisiones y subdivisiones en documentos escritos
UNE 50132:1994	Traducción de la ISO 2145
APA 6ª edición	Estilo de referencias y citas de APA (American Psychological Association)

NORMAS UTILIZADAS COMO BASE O REFERENCIA	
NACIONALES	
UNE 157001:2014	Criterios generales para la elaboración formal de los documentos que constituyen un proyecto técnico
UNE 157801:2007	Criterios generales para la elaboración de proyectos de sistemas de información
<i>Estas normas se han utilizado como base o referencia para la inclusión de algunos contenidos y definiciones sobre elaboración de proyectos, entendiendo como proyecto la documentación consensuada entre una empresa y un cliente, que da lugar al perfeccionamiento de un contrato para la elaboración de una obra o la prestación de un servicio. Por consiguiente, no debe esperarse la aplicación de estas normas en cuanto a la completitud de los contenidos ni a la organización de los mismos.</i>	

Contenido

1	Especificación del trabajo	12
1.1	Introducción.....	12
1.2	Objetivos del trabajo	13
1.3	Antecedentes y estado del arte.....	13
1.4	Descripción de la situación de partida.....	15
1.4.1	Descripción del entorno actual.....	15
1.4.2	Resumen de las deficiencias y carencias identificadas	15
1.5	Requisitos iniciales.....	16
1.6	Alcance	17
1.7	Hipótesis y restricciones.....	17
1.8	Estudio de alternativas y viabilidad	17
1.8.1	Frontend	17
1.8.2	Backend.....	19
1.9	Tecnologías utilizadas.....	21
1.10	Metodología de desarrollo de software	27
1.11	Estimación del tamaño y esfuerzo	28
1.12	Planificación temporal	28
1.13	Presupuesto	29
1.13.1	Costes hardware.....	29
1.13.2	Costes software	29
1.13.3	Costes indirectos	31
1.13.4	Costes de personal	31
1.13.5	Resumen del presupuesto	32
2	Análisis	32
2.1	Análisis preliminar	32
2.2	Especificaciones del sistema	33
2.3	Casos de uso	35
2.4	Historias de usuario	45
3	Diseño inicial.....	48
3.1	Diagramas de secuencia.....	48
3.1.1	Inicio de sesión	48
3.1.2	Realización de test.....	49
3.1.3	Añadir nuevo usuario	50
3.2	Diagrama de clases	51
3.2.1	Paquete Entidades	52
3.2.2	Paquete Componentes	53
3.3	Diseño de interfaces	54
3.3.1	Menú administrador	54
3.3.2	Inicio de sesión	56
3.3.3	Panel principal	57
3.3.4	Realización de test.....	57
3.3.5	Editar perfil.....	58
3.3.6	Ranking.....	59
3.4	Diseño del algoritmo de creación de test	60
3.5	Arquitectura de la aplicación	63

3.5.1	Frontend	63
3.5.2	Backend.....	64
4	Desarrollo	66
4.1	Primera Iteración	67
4.1.1	Planificación.....	67
4.1.2	Implementación	67
4.1.3	Pruebas y validación.....	69
4.2	Segunda iteración	70
4.2.1	Planificación.....	70
4.2.2	Implementación	71
4.2.3	Pruebas y validación.....	71
4.3	Tercera iteración	72
4.3.1	Planificación.....	72
4.3.2	Implementación	76
4.3.3	Pruebas y validación.....	76
4.4	Cuarta iteración.....	77
4.4.1	Planificación.....	77
4.4.2	Implementación	78
4.4.3	Pruebas y validación.....	78
5	Conclusiones y trabajos futuros	79
6	Apéndices	81
6.1	Instalación y configuración del sistema	81
6.2	Manuales de usuario	83
7	Definiciones y abreviaturas	87
8	Bibliografía	89

Índice de ilustraciones

Ilustración 1: Angular vs React vs Vue	18
Ilustración 2: Comparación de un servidor backend tradicional con Firebase	20
Ilustración 3: Estructura de una base de datos de Firebase	22
Ilustración 4: Estructura de un proyecto en Trello	24
Ilustración 5: Estructura de un proyecto en Visual Studio	25
Ilustración 6: Ejemplo de proyecto en SmartGit	25
Ilustración 7: Ejemplo de diseño en Moqups	26
Ilustración 8: Ejemplo de diagrama en Visual Paradigm	27
Ilustración 9: Diagrama de Gantt del proyecto	28
Ilustración 10: Planes de pago de Firebase	30
Ilustración 11: Diagrama de secuencia de inicio de sesión	49
Ilustración 12: Diagrama de secuencia de realización de test	50
Ilustración 13: Diagrama de secuencia de añadir nuevo usuario	51
Ilustración 14: Diagrama de clases del paquete entidades	52
Ilustración 15: Diagrama de clases del paquete componentes	53
Ilustración 16: Diseño inicial de la sección de usuarios	55
Ilustración 17: Diseño inicial de la vista Añadir usuario	55
Ilustración 18: Diseño inicial de la sección de preguntas	56
Ilustración 19: Diseño inicial de la vista de inicio de sesión	56
Ilustración 20: Diseño inicial de la vista del panel principal	57
Ilustración 21: Diseño inicial de la vista de realización de test	57
Ilustración 22: Diseño inicial de la vista de corrección de test	58
Ilustración 23: Diseño inicial de la vista de editar perfil	59
Ilustración 24: Diseño inicial de la vista Ranking	59
Ilustración 25: Diagrama arquitectónico del frontend	64
Ilustración 26: Funcionamiento de las comunicaciones en Firebase	65
Ilustración 27: Diagrama arquitectónico de la plataforma	66
Ilustración 28: Reglas de seguridad del proyecto en Firebase	68
Ilustración 29: Diseño inicial de la cabecera	69
Ilustración 30: Diseño del panel inicial tras los cambios de la primera iteración	70
Ilustración 31: Diagrama de secuencia de importar preguntas desde Excel	74
Ilustración 32: Diseño de la sección usuarios tras los cambios de la segunda iteración	75
Ilustración 33: Diseño de la sección preguntas tras los cambios de la segunda iteración	76
Ilustración 34: Diseño de la vista de test tras los cambios de la tercera iteración	77
Ilustración 35: Página de instalación de NodeJS	81
Ilustración 36: Instrucciones a incluir en la consola	82
Ilustración 37: Mensaje en la consola al ejecutar	82
Ilustración 38: Panel principal	83
Ilustración 39: Realización de test	84
Ilustración 40: Corrección de test	84
Ilustración 41: Ranking	85
Ilustración 42: Sección de usuarios	85
Ilustración 43: Sección de preguntas	86
Ilustración 44: Editar perfil	87

Índice de tablas

Tabla 1: Costes hardware del proyecto	29
Tabla 2: Costes software del proyecto	30
Tabla 3: Costes indirectos del proyecto.....	31
Tabla 4: Costes de personal del proyecto	32
Tabla 5: Resumen del presupuesto.....	32
Tabla 6: Caso de uso Registrar usuario	36
Tabla 7: Caso de uso Iniciar sesión.....	36
Tabla 8: Caso de uso Recuperar contraseña.....	38
Tabla 9: Caso de uso Cerrar sesión	38
Tabla 10: Caso de uso Editar perfil	39
Tabla 11: Caso de uso Indicador de progreso	40
Tabla 12: Caso de uso Realización de test y corrección	41
Tabla 13: Caso de uso Ranking.....	41
Tabla 14: Caso de uso Eliminar usuario.....	42
Tabla 15: Caso de uso Editar usuario	43
Tabla 16: Caso de uso Añadir pregunta.....	43
Tabla 17: Caso de uso Editar pregunta.....	44
Tabla 18: Caso de uso Eliminar pregunta	45
Tabla 19: Historias de usuario	48
Tabla 20: Caso de uso Importar preguntas desde Excel	73
Tabla 21: Historia de usuario Importar preguntas desde Excel	73

1 ESPECIFICACIÓN DEL TRABAJO

En este capítulo se presenta la especificación del trabajo, con una estructura y contenidos **inspirados** en los criterios y recomendaciones que establece la norma UNE 157801:2007 - “*Criterios Generales para la elaboración de proyectos de Sistemas de Información*”.

A lo largo del documento se utilizarán términos y acrónimos cuya descripción aparecen en el apartado 7 (Definiciones y abreviaturas).

1.1 Introducción

El presente trabajo consiste en el análisis, diseño e implementación de una aplicación web para una empresa real de cursos de ortopedia que ofrezca test adaptados a cada usuario según las preguntas que este ya haya contestado. El objetivo principal es que los alumnos del curso aprendan el temario a través de la realización de test cortos, contando con una amplia batería de preguntas. De esta forma, será necesario crear un algoritmo que cree test automáticamente y de forma eficiente para el alumno basándose en su historial de test anteriores, de manera que el usuario avanzará en su progreso por cada pregunta que responda correctamente de un test.

Por tanto, la aplicación web pretende adaptar las formas de estudio a los nuevos tiempos e implicar al alumno de forma activa en el estudio. Se opta por un sistema de estudio basado en la realización constante de test, lo cual implica más al estudiante y puede convertir la tarea en algo mucho más ameno y que aumente el nivel de concentración en él.

Además, para incentivar y motivar más al alumnado del curso, se pretende incluir gamificación mediante la inclusión de un ranking en el que los alumnos compitan por ser el que más preguntas responde correctamente.

En el presente documento se detallan todos los aspectos relativos al proyecto. Se trata el objetivo y funcionalidades del proyecto, la motivación, descripción del entorno (empresa) en el que se integrará el proyecto... además de todos los aspectos técnicos sobre el mismo.

La empresa contratante es Cursocot, una empresa de cursos de ortopedia con cientos de alumnos por toda España. Si el resultado del proyecto es satisfactorio, StudyGame será incluido en la página oficial de la empresa, siendo usado por sus alumnos y obteniendo una rentabilidad del mismo.

1.2 Objetivos del trabajo

Los objetivos principales del proyecto son los siguientes:

- Estudio de un algoritmo que permita realizar test inteligentes de forma automática dada una batería de preguntas que se adapten a cada alumno y a su historial de respuestas, agilizando lo máximo posible la tarea de estudio activa mediante la realización de test.
- Realización de una aplicación web completa que aplique este algoritmo y presente de forma intuitiva y clara los test a los alumnos y su progresión.
- Inclusión de un ranking basado en el historial de test de cada alumno que permita la competición entre ellos.
- Inclusión de un menú para administradores que permita la gestión total de usuarios y preguntas.

1.3 Antecedentes y estado del arte

Observando los objetivos para los que el proyecto es creado, se podría llegar a pensar en la utilización de alguna plataforma de test ya creada y la adaptación a las necesidades del curso, para no tener que crear una plataforma nueva.

A continuación, vamos a explicar por qué esta opción no sería válida para cumplir con los objetivos de la plataforma. Para ello vamos a hablar de Moodle, la plataforma web para realizar test más popular de la actualidad, y las diferencias y funcionalidades extra que se aplicarán en el proyecto respecto a esta plataforma.

Moodle es una plataforma de aprendizaje diseñada para proporcionar a educadores, administradores y estudiantes un sistema integrado único, robusto y seguro para crear ambientes de aprendizaje personalizados. La actividad de examen es un módulo dentro de la plataforma Moodle para evaluar a los alumnos mediante cuestionarios. Según la documentación de Moodle [1] los pasos a realizar para crear un examen son:

1. En un curso, con la edición activada, elija "Examen" en el Selector de actividad
2. Dele un nombre y, si se requiere, una descripción.
3. Expanda las otras secciones para seleccionar las configuraciones que desea. Con las configuraciones predeterminadas, los estudiantes pueden repetir el examen, moverse libremente entre las preguntas, cada pregunta en una página diferente. No hay límite de tiempo y el puntaje y la retroalimentación se muestran una vez que hayan completado el examen.
4. Para cambiar cualquiera de estas configuraciones predeterminadas, y para más información acerca de cómo configurar su examen, vea Configuraciones del examen.
5. Haga clic en Guardar y mostrar.
6. Haga clic en Editar examen.
7. Haga clic en Añadir y después haga clic en '+ una nueva pregunta' (Si usted ya había hecho preguntas dentro del Banco de preguntas entonces haga clic en '+ del banco de preguntas' o si desea añadir una pregunta elegida aleatoriamente de entre una categoría de preguntas, haga clic en '+ una pregunta aleatoria').

Como podemos ver, en Moodle es necesario que el profesor cree un examen para que los alumnos lo completen. Es necesario que el profesor cree una actividad de examen y elija las preguntas que se mostrarán en él.

La carencia encontrada en la plataforma Moodle por la que se ha decidido implantar un sistema nuevo radica en la creación de los test. Como ya se ha explicado en el apartado anterior, en Moodle el profesor crea el test con unas preguntas concretas, para todos los usuarios, sin fijarse en el progreso de cada uno. En StudyGame, el administrador no creará los test, tan solo suministrará un banco de preguntas sobre cada tema. Cuando un alumno seleccione realizar un test sobre un tema, el sistema realizará el test automáticamente y las mostrará al usuario. Además, estas preguntas no serán seleccionadas aleatoriamente, ya que se trata de test inteligentes: El test estará compuesto por preguntas que el alumno aún no ha respondido correctamente, o las ha respondido correctamente, pero hace demasiado

tiempo de ello. De esta forma, cada respuesta correctamente respondida reflejará un avance en su aprendizaje.

Por tanto, el valor añadido que pretende aportar la plataforma StudyGame es un algoritmo de creación de test inteligentes basados en el usuario. Además, también se creará un ranking basado en el número de preguntas contestado correctamente en total, a deseo de la empresa contratante.

1.4 Descripción de la situación de partida

1.4.1 Descripción del entorno actual

Actualmente, en la empresa contratante (Cursocot) ya existe una plataforma llamada StudyGame que fue creada con los mismos objetivos que este proyecto: una plataforma de creación de test adaptados al usuario, suministrando al sistema un banco de preguntas, con un ranking. Sin embargo, la plataforma cuenta con una serie de carencias que han motivado a la propuesta de creación de una plataforma nueva que pretenda subsanar todas estas deficiencias.

1.4.2 Resumen de las deficiencias y carencias identificadas

Las deficiencias encontradas en la actual plataforma StudyGame que han motivado a la empresa a la creación de una nueva plataforma son:

- **Falta de adaptabilidad de los test:** Esta es la característica más importante que le falta a la vigente plataforma. Actualmente, los test que realizan los alumnos están compuestos por preguntas seleccionadas aleatoriamente, de manera que quizá un alumno responda a la misma pregunta en distintos test correctamente mientras otras aún no le han salido. La empresa contratante quiere optimizar el aprendizaje teniendo en cuenta el historial de respuestas del usuario en la creación de un test, algo que aún no ocurre. Para ello será necesario el diseño de un algoritmo de creación de test.
- **Altos tiempos de carga:** Algunas secciones de la plataforma, como el Ranking, tardan más de diez segundos en cargarse, debido a que no fue planificada correctamente para la alta carga de usuarios. Además, dispone de un foro que no carga correctamente.

- **Interfaz poco intuitiva y manejable:** Según han reportado la empresa, la interfaz de la actual plataforma no proporciona una buena experiencia de usuario, y a los alumnos les cuesta entender el funcionamiento cuando toman contacto inicialmente con la web.

Debido a estas carencias, se va a crear una plataforma desde cero que pretenda incorporar todas las especificaciones de la empresa contratante y subsanar las carencias de la actual plataforma.

1.5 Requisitos iniciales

El proyecto se adhiere a los siguientes objetivos:

- Debe incluir un sistema de autenticación con recuperación de contraseña.
- Debe permitir la gestión total de usuarios dentro de la propia página por parte del administrador. Esto incluye añadir, editar y eliminar usuarios.
- Debe permitir la gestión total de preguntas dentro de la propia página por parte del administrador. Esto incluye añadir, editar y eliminar preguntas.
- Debe dividir las preguntas en seis temas para la realización de test: Columna, trauma, miembro superior, miembro inferior, ciencias básicas y trauma y ortopedia infantil.
- Debe mostrar al usuario el progreso porcentual que tiene en cada uno de los seis temas en el panel principal, basándose en el porcentaje de preguntas que ha respondido correctamente de cada tema.
- Debe mostrar una clasificación semanal y total de los usuarios que han obtenido mejor puntuación según sus resultados en los test.
- Debe mostrar al usuario una corrección al final del test con su nota en el mismo y la respuesta correcta de las preguntas acompañadas de una explicación.

1.6 Alcance

La plataforma será creada con compatibilidad para todos los navegadores más usados en la actualidad (Google Chrome, Safari, Microsoft Edge, Mozilla Firefox, Opera, Vivaldi y Brave [2]).

El número de usuarios una vez implantado en la empresa será entre 200 y 500. Sin embargo, el diseño es escalable y soporta un mayor número de usuarios.

El sistema es también escalable respecto al número de preguntas. El número total de preguntas que ha creado la empresa ronda las 3500.

Los entregables una vez terminado el proyecto serán:

- La aplicación web preparada para su uso.
- Documentación final o memoria.
- Manual de usuario con todos los pasos para la instalación y replicación de la plataforma web en cualquier ordenador.

1.7 Hipótesis y restricciones

El TFT se define como una asignatura de 12 créditos, lo que supone que la duración total del proyecto será de 300 horas, incluyendo todas las etapas del ciclo de vida, con la excepción del mantenimiento. Por consiguiente, la principal restricción aplicable es la limitación de la duración del trabajo.

1.8 Estudio de alternativas y viabilidad

1.8.1 Frontend

Para el desarrollo del código frontend de la aplicación, en la actualidad coexisten muchos entornos de trabajo de JavaScript que ofrecen un amplio abanico de opciones a la hora de elegir, cada uno con ciertas ventajas, por lo que fue necesario un estudio para determinar el más apropiado.

Para acortar el estudio, de entre todos los frameworks disponibles se seleccionaron los tres más populares en la actualidad: React, Vue y Angular. Al escoger estos tres, se podrían estudiar las características más a fondo para elegir con un mejor criterio.

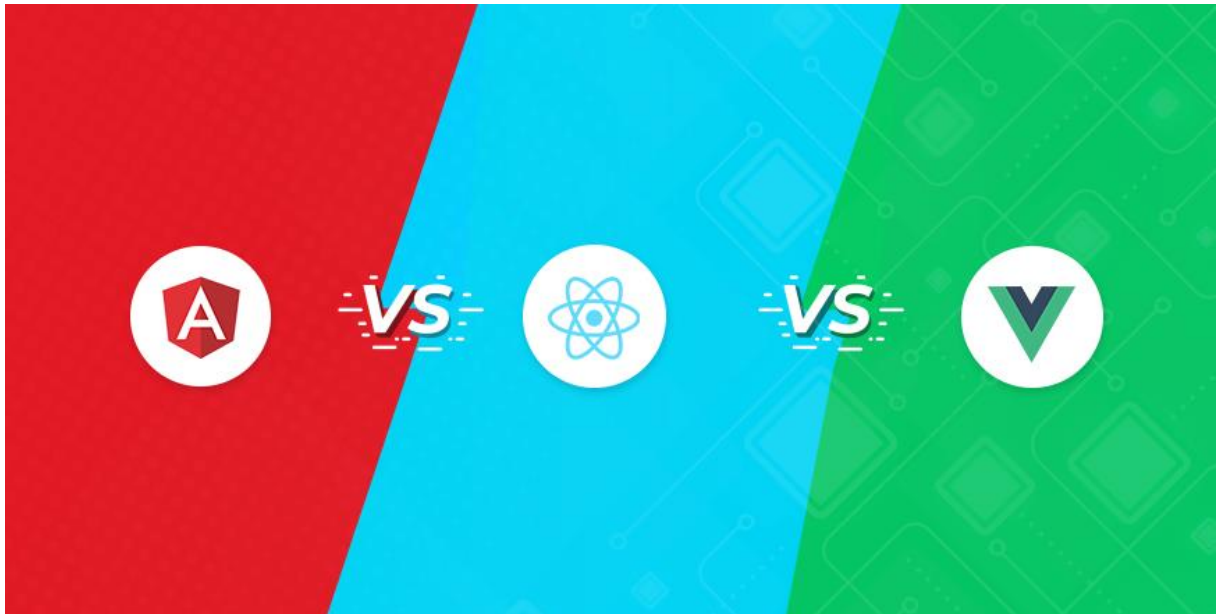


Ilustración 1: Angular vs React vs Vue

Los factores considerados para la elección de un framework entre los tres son [3]:

- **Rendimiento:** Para este punto, Angular es el claro perdedor, porque usa DOM real, lo cual disminuye su rendimiento y capacidad para crear aplicaciones de software dinámicas. Tanto Vue como React utilizan un DOM virtual que elimina estos problemas de rendimiento.
- **Apoyo de la comunidad:** Angular fue presentado y es usado por Google, mientras que React fue desarrollado por Facebook, que sigue trabajando en él. Por lo tanto, son frameworks confiables y con mucho apoyo a día de hoy. Por otro lado, Vue no está respaldado por ninguna empresa de alto nivel, pero tiene una gran popularidad en la comunidad de código abierto. Por tanto, para esto punto los tres frameworks son válidos.
- **Tamaño del framework:** El tamaño de Angular es de unos 500 KB, mientras que el de React y Vue es de unos 100 KB. Por tanto, para aplicaciones ligeras, como es el caso de este proyecto, es más recomendable React o Vue que Angular en este aspecto.
- **Curva de aprendizaje:** Debido a que el tiempo para la realización del proyecto no es demasiado amplio, este factor tiene una gran importancia. Para los desarrolladores, React es el preferido para usar y aprender,

seguido de Vue y luego Angular. Angular tiene una dificultad añadida que es el uso de TypeScript, un superconjunto de JavaScript.

- **Flexibilidad:** Angular es el menos flexible, ya que ofrece todo tipo de funcionalidades, lo cual se puede considerar bueno o malo, ya que no permite añadir herramientas de terceros. React ofrece menos herramientas, lo que le da más flexibilidad para integrar soluciones de terceros y permite crear aplicaciones más dinámicas. Vue es más flexible que Angular, pero menos que React.

Tras estudiar estas variables, el primer framework descartado fue Angular, debido al menor rendimiento y flexibilidad y mayor dificultad para aprender de inicios. La decisión entre React y Vue fue más difícil, pero finalmente la solución elegida fue React, debido a su mayor flexibilidad, su mejor curva de aprendizaje y estar respaldado por una gran empresa como Meta.

1.8.2 Backend

Para el desarrollo del Backend hay varias opciones, una de ellas es usar un framework del lado servidor como, por ejemplo, CodeIgniter (<https://codeigniter.com/>). Sin embargo, la decisión elegida finalmente fue el uso de Firebase, una plataforma en la nube para el desarrollo de aplicaciones web y móvil. Las razones que motivaron la elección fueron:

- Facilita enormemente la creación del lado servidor (backend) tanto en recursos como en tiempo, ya que ofrece un servidor completo con unas funcionalidades que llevarían demasiado tiempo implementar de forma manual.
- Las funcionalidades son muy sencillas de controlar. Las de desarrollo se pueden incluir con unas pocas líneas de código y otras, como las analíticas, se pueden comprobar desde un panel.
- Es muy útil para un envío de notificaciones rápido y eficiente
- Ofrece analíticas con métricas muy detalladas sobre toda la actividad en nuestra aplicación.

- Es muy escalable, ofrece servicios gratuitos, pero permite ir adaptándose a las necesidades de la aplicación con diferentes planes de pago.
- Ofrece seguridad en distintas capas.
- Google ofrece numerosos documentos y tutoriales para que su uso sea mucho más fácil.

Dentro de las desventajas, se podría considerar que la más importante es el coste. Pese a que Firebase ofrece un inicio con un plan Spark que es gratuito, este tiene limitaciones, por lo que cuando el número de usuarios y espacio necesario de almacenamiento es alto resulta necesario adquirir un plan de pago. Aun así, esto solo habrá que hacerlo en aplicaciones profesionales pretendiendo obtener beneficios mucho mayores de lo que cuestan estos planes, así que no debería suponer un gran problema.

En conclusión, Firebase es una plataforma muy completa y adaptable que termina dejando en segundo plano las pocas desventajas que presenta. Es muy útil en los inicios de un proyecto, ya que sus funciones están pensadas para acompañar en este proceso, y permite centrarse en el código del lado cliente (frontend).

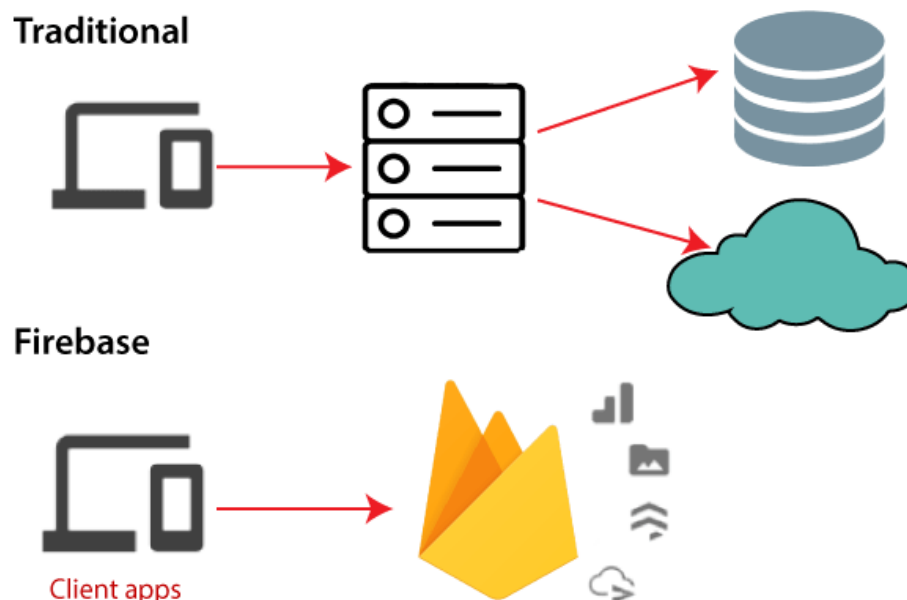


Ilustración 2: Comparación de un servidor backend tradicional con Firebase

1.9 Tecnologías utilizadas

Las tecnologías utilizadas en la realización de este proyecto son:

- **Framework frontend:** La tecnología usada es React (versión 4.25.0), por las razones descritas en el [apartado 1.8](#).
- **Framework backend:** La tecnología usada es Firebase, por las razones descritas en el [apartado 1.8](#). Firebase se compone de un conjunto de extensiones o SDKs, cada uno orientado a una función que se suele llevar a cabo en el backend de la aplicación, de manera que los desarrolladores pueden usar las extensiones que vean convenientes para el correcto funcionamiento del sistema. Además, provee una API REST para la comunicación de la aplicación con sus servicios. Para este proyecto, las extensiones que se van a utilizar son [4]:
 - Base de datos en tiempo real: Cloud Firestore es la herramienta de bases de datos en Firebase. Se tratan de bases de datos en tiempo real NoSQL y que almacena los datos como JSON. El hecho de que la base de datos sea NoSQL presenta varias ventajas. Una de ellas es que la estructuración de los datos queda a gusto de cada desarrollador. A través de colecciones y documentos, se pueden crear jerarquías para almacenar datos relacionados y recuperar los datos que se necesiten mediante consultas expresivas de manera sencilla. Esto, además, permite una base de datos muy escalable que tolera cantidades de datos muy altas. Una gran ventaja de estas bases de datos es que no es necesario crear un servidor propio para nuestra aplicación, ya que Cloud Firestore se distribuye con un conjunto completo de reglas de seguridad que permiten acceder a la base de datos alojada en la nube sin necesidad de crear un servidor propio. Se puede acceder a la base de datos y realizar las modificaciones pertinentes desde cualquier biblioteca cliente (es decir, Node, Python, Go o Java). Otra ventaja de Cloud Firestore es que se pueden sincronizar automáticamente los cambios en distintos dispositivos, ya que el servidor notifica los cambios en los datos a medida que ocurren, lo cual permite crear experiencias colaborativas y apps en tiempo real con facilidad. Los usuarios pueden acceder y

realizar cambios incluso sin conexión. Además, esta base de datos presenta un sólido sistema de seguridad, ya que se integra con la autenticación de usuarios.



Ilustración 3: Estructura de una base de datos de Firebase

- Autenticación de usuarios: Este proyecto necesita un mecanismo de autenticación de usuarios. Firebase simplifica enormemente este trabajo, ya que provee un sistema de autenticación seguro que se puede integrar fácilmente en nuestra aplicación. Además, permite implementar autenticación con todos estos proveedores:
 - Cuenta de correo electrónico.
 - Autenticación telefónica.
 - Acceso mediante cuenta de Google, Facebook, Play Juegos, Game Center, Apple, GitHub, Microsoft, Twitter y Yahoo.

Estos sistemas de autenticación se pueden combinar a gusto del desarrollador, por lo que quedan cubiertas todas las formas posibles de identificarse restando mucho esfuerzo de desarrollo de backend y de forma completamente segura.

- Almacenamiento en la nube: Firebase cuenta con un sistema de almacenamiento, donde los desarrolladores pueden guardar los ficheros de sus aplicaciones y sincronizarlos. Esto permite tratar los ficheros subidos por los usuarios, como fotografías o vídeos, de forma rápida y segura. Es altamente escalable, ya que se basa en la misma tecnología que respalda Spotify y Google Photos. Está pensado para cargas y descargas robustas con conectividad móvil por lo que, en caso de pérdida y recuperación de conexión, las transferencias se detienen y reanudan de forma automática. Además, contiene un fuerte mecanismo de seguridad, ya que se integra con la autenticación para proporcionar un control de acceso intuitivo y sencillo. En el caso de este proyecto, se usará para almacenar imágenes.
- **Lenguajes de programación:** Los lenguajes de programación usados están condicionados por el framework frontend y la naturaleza del proyecto. Al tratarse de un proyecto web, será necesario el uso del lenguaje de marcado HTML, junto con el lenguaje de estilo gráfico CSS. Por otro lado, elegir React como framework frontend implica el uso del lenguaje JavaScript para el diseño de la interactividad de la plataforma web y la conexión con el backend.

Respecto al backend, no se usarán lenguajes de programación debido a que, como se ha explicado en apartados anteriores, Firebase no requiere de elaboración de código de lado servidor, lo que simplifica mucho las tareas para los desarrolladores.
- **Planificación de tareas:** Para la planificación de tareas, la tecnología que se ha decidido utilizar es Trello [5]. Se trata de una plataforma de formato web a través de la cual podemos organizar las tareas de forma simple e intuitiva, de manera que permite una clarificación de las ideas y facilita el trabajo. Está basado en la metodología Kanban cuya esencia radica en la utilización de tarjetas de trabajo en un tablero llevando de este modo una “línea de producción” de tareas con sus estatus correspondientes. Estos tableros permiten, por ejemplo, la ejecución de un diagrama de Gantt que

facilite la planificación. Pese a que Trello está enfocado al trabajo en grupo, también ofrece muchas ventajas en la planificación de trabajos individuales, de las cuales nos beneficiaremos en el desarrollo del proyecto.

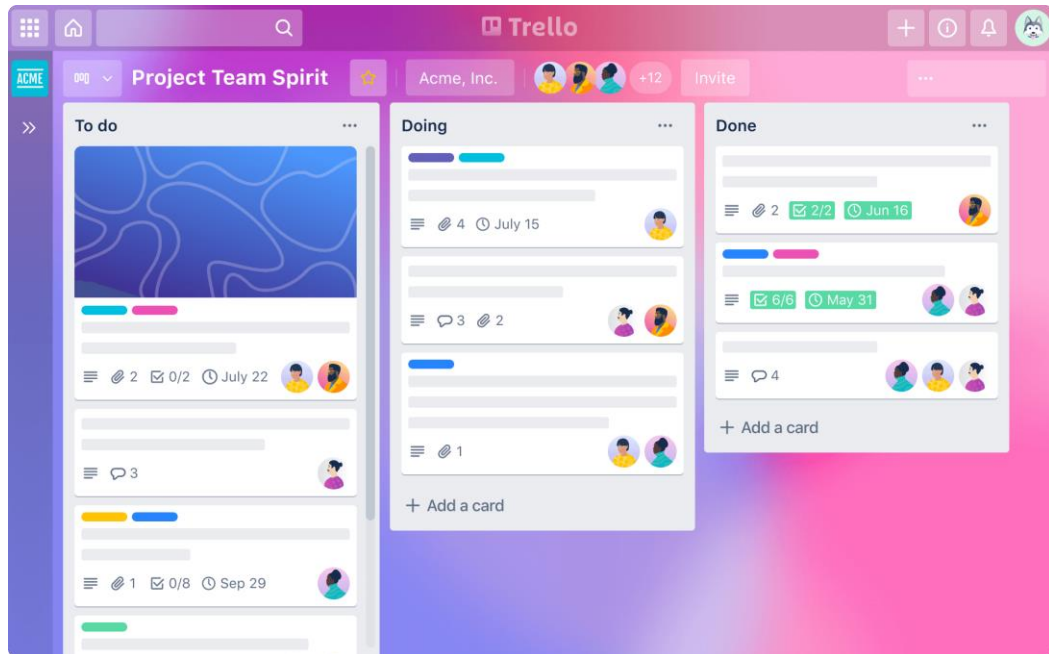


Ilustración 4: Estructura de un proyecto en Trello

- **Editor de código:** El software elegido para escribir y editar el código fuente de la aplicación es Visual Studio 2022 [6]. Las razones por las que fue elegido son varias: la primera, que al ser un software conocido por el autor no fue necesario un proceso de familiarización con la plataforma. Otra razón de peso es que tiene un buen mecanismo de depuración que avisa de cualquier error que se dé en la escritura de código tanto HTML como JavaScript, lo que ahorra una gran cantidad de tiempo de búsqueda de errores. La tercera razón es la facilidad y comodidad que ofrece el software para desplazarse entre los distintos archivos, incluir nuevos... algo que es muy importante para este proyecto debido a la reutilización de componentes, una de las características principales de React. Será necesario trabajar a la vez con muchos ficheros y Visual Studio facilita esta tarea.

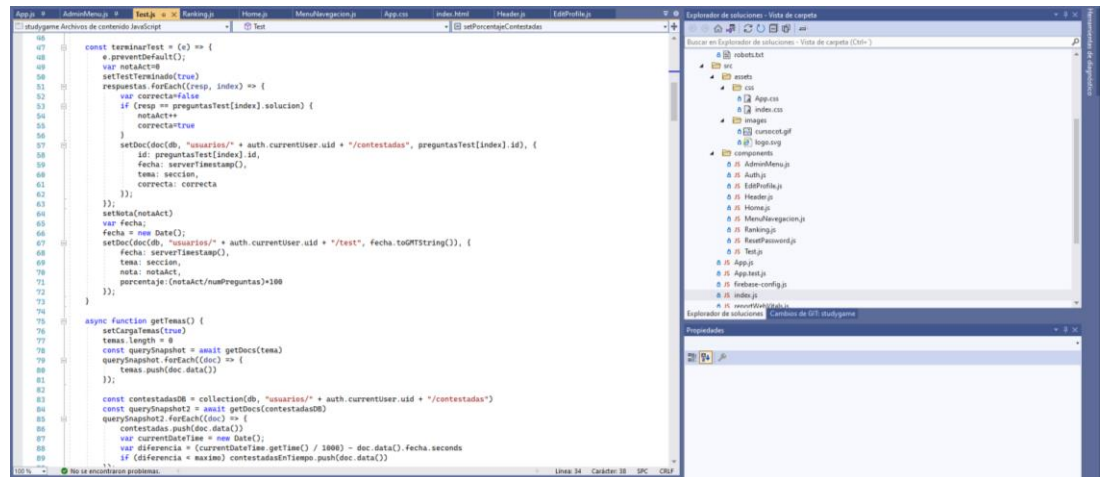


Ilustración 5: Estructura de un proyecto en Visual Studio

- **Control de versiones:** Para el control de versiones y el desarrollo distribuido del sistema se va a utilizar Git, y en concreto el programa SmartGit [7] para su manejo. La elección de este software radica en que es muy intuitivo y apenas suma tiempo a la realización del trabajo para los procesos relacionados con Git. La plataforma Git que mantendrá el repositorio en la nube es GitLab, ofrecida por el grupo de investigación SINAI de la Universidad de Jaén en el servidor <https://gitlab.ujaen.es>.

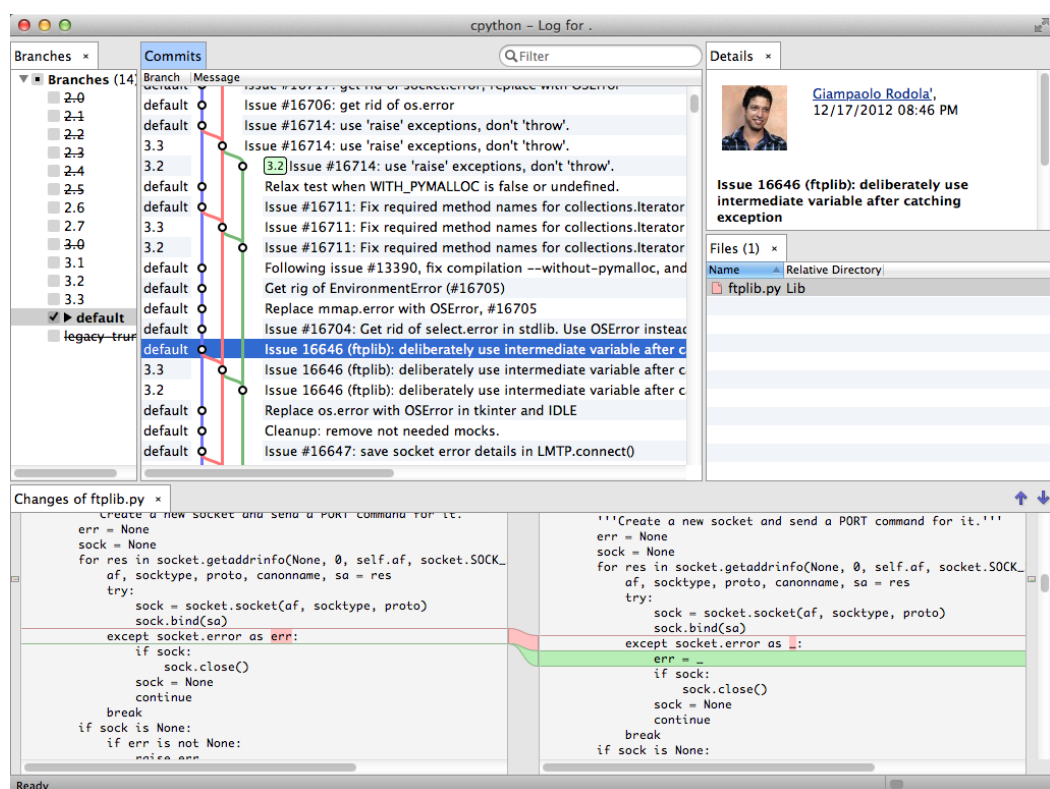


Ilustración 6: Ejemplo de proyecto en SmartGit

- **Diseño de interfaces:** La tecnología usada para el diseño inicial de las interfaces de la aplicación es Moqups (<https://moqups.com/es/>).

Moqups es una aplicación web simplificada que ayuda a crear y colaborar en tiempo real en esquemas, maquetas, diagramas y prototipos. Las dos razones por las que se ha elegido esta plataforma son su sencillez y variedad. Sencillez porque es una plataforma altamente intuitiva, se tarda muy poco tiempo en plasmar una idea de la mente porque es muy manejable y flexible, se entiende a la perfección cómo se usa en cuestión de minutos. También tiene una alta variedad: dispone de todo tipo de iconos, formas, botones, márgenes, plantillas... de forma que cualquier idea se puede plasmar en un boceto en esta plataforma, porque ofrece todo tipo de opciones de diseño.

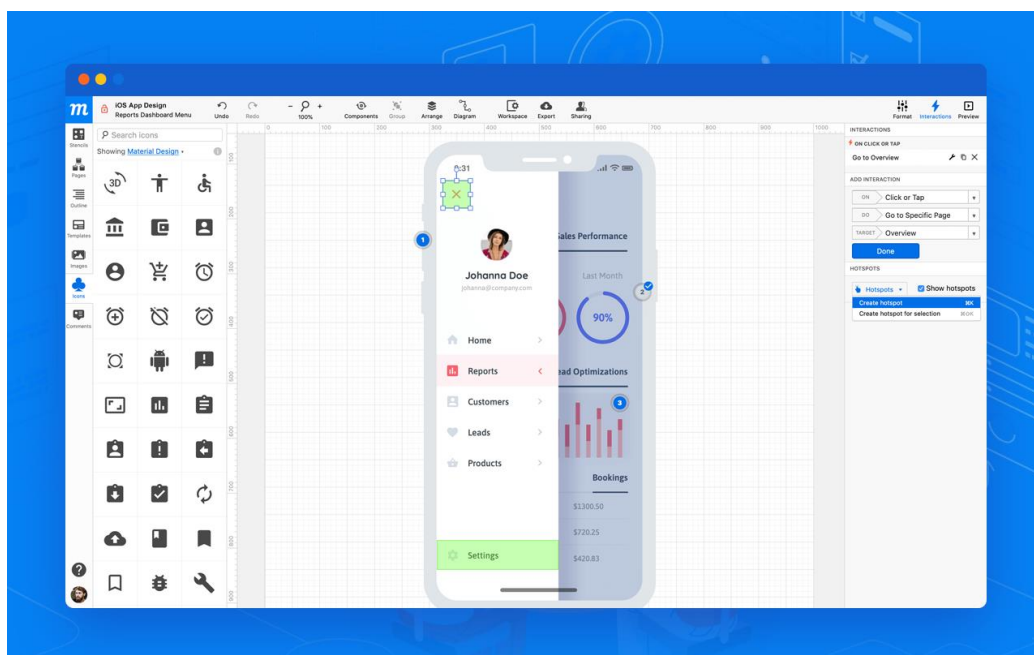


Ilustración 7: Ejemplo de diseño en Moqups

- **Diseño de diagramas:** Para el diseño del sistema será necesaria la inclusión de diagramas, tales como diagramas de secuencia, de clases... La tecnología usada para el diseño de estos diagramas es Visual Paradigm (<https://www.visual-paradigm.com/>). La razón por la que se ha elegido esta plataforma para el diseño de los diagramas es la experiencia con la misma, debido a que ya se había hecho uso de este software para otras asignaturas y se consideró que presenta una variedad de diagramas que ocupan todas

las necesidades de diseño de este proyecto, además de ser simple e intuitivo.

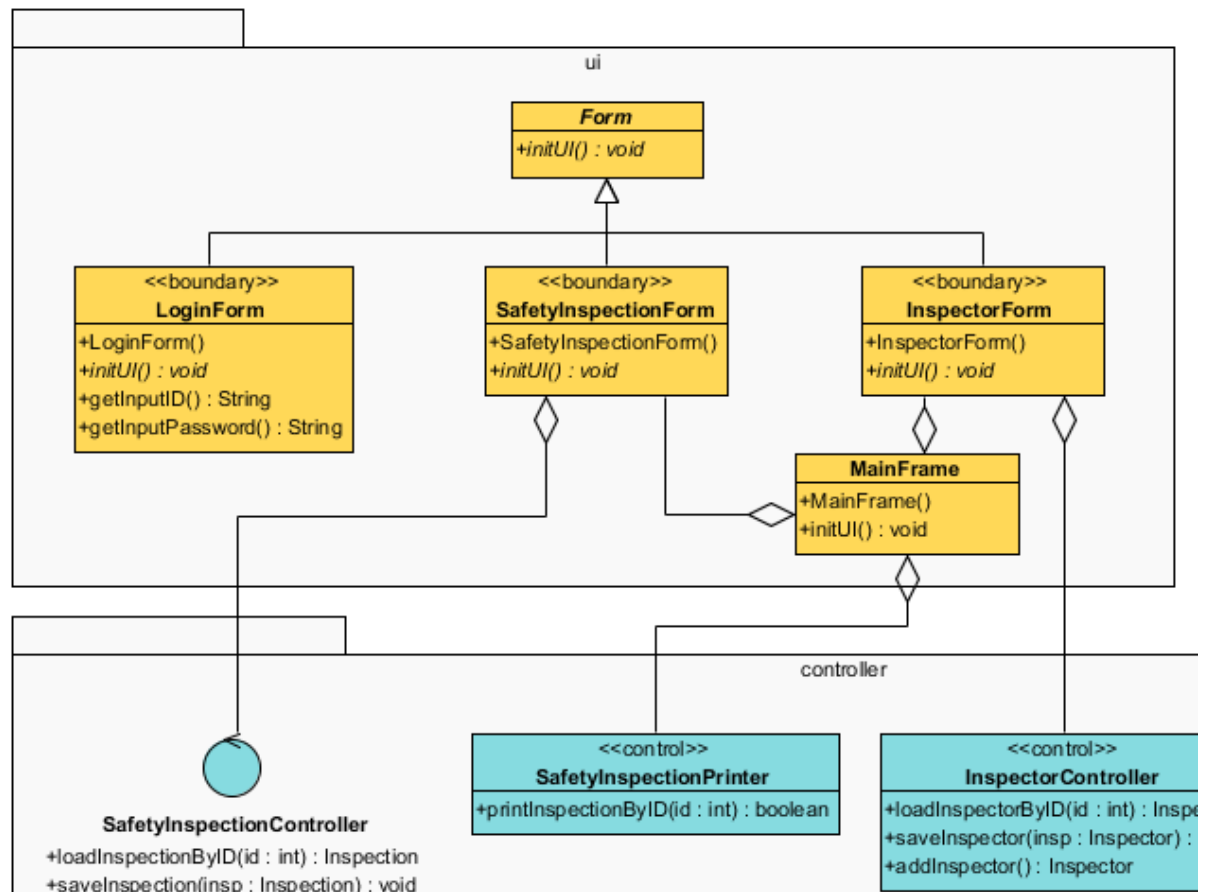


Ilustración 8: Ejemplo de diagrama en Visual Paradigm

1.10 Metodología de desarrollo de software

La metodología de desarrollo elegida para este proyecto es una metodología ágil, basada en iteraciones. Aunque se trata de un proyecto individual y no en grupo, se ha considerado óptimo este enfoque por varias razones:

- Mejorar la calidad del producto: durante el desarrollo del proyecto va a haber contacto constante con la empresa contratante (Cursocot), de manera que durante las iteraciones se va a buscar constantemente la adaptación a los requisitos de la empresa, con un constante intercambio de ideas que creen el producto final que el cliente necesita.
- Descubrir errores de forma temprana: se van a realizar validaciones y pruebas constantemente en cada iteración, de manera que los errores pueden ser descubiertos y corregidos rápidamente.

- Mejorar la satisfacción del cliente: debido al constante intercambio de ideas con la empresa contratante, se garantiza que el producto final va a estar adaptado a las necesidades de la misma.

1.11 Estimación del tamaño y esfuerzo

Ya que el presente proyecto es un TFT, no existen restricciones de tipo económico, sino de tipo temporal (un número aproximado de horas). Por consiguiente, los cálculos de tamaño del proyecto están supeditados el tiempo disponible, en este caso, unas 300 horas aproximadamente.

En cuanto al esfuerzo, se dispone de tan un solo efectivo, el autor del proyecto.

1.12 Planificación temporal

El proyecto dio comienzo el 19 de septiembre de 2022 y la finalización estimada es el 15 de febrero de 2023, pues precisa unos cinco meses de trabajo. En base a esto se ha realizado una estimación de los tiempos necesarios para cada fase del sistema, quedando la siguiente planificación temporal:

- Análisis inicial del sistema: 19/09/2022 – 10/10/2022.
- Diseño inicial del sistema: 11/10/2022 – 1/11/2022
- Implementación mediante iteraciones: 7/11/2022 – 2/2/2023
- Documentación del proyecto: 19/9/2022 – 15/2/2023

Quedando como resultado el siguiente diagrama de Gantt:

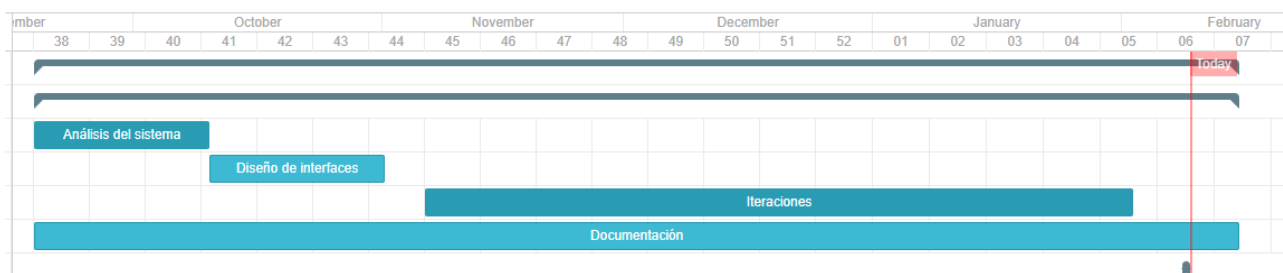


Ilustración 9: Diagrama de Gantt del proyecto

1.13 Presupuesto

En este apartado se va a hablar del presupuesto total para la realización de la plataforma, desglosado en apartados.

1.13.1 Costes hardware

El ordenador en el que se está llevando a cabo el proyecto se trata de un portátil Lenovo Yoga 530 con procesador Intel Core i7 de 8ª generación, Windows 10, pantalla táctil HD de 35,56 cm (14") y resolución 1366 × 768, memoria DDR4 de 8 GB y unidad SSD PCIe de 128 GB. El coste del ordenador es 600 €, con un periodo de amortización de 6 años aproximadamente, el coste sería de 100 €/año. Puesto que el tiempo de realización del proyecto es de unos 5 meses, el coste del ordenador sería 42 €.

Por otro lado, también es necesario un móvil para realizar las pruebas de usabilidad en dispositivos móviles. El móvil usado para ello es el Google Pixel 6, con un coste de 400 € y un periodo de amortización aproximado de 3 años, por lo que el coste del móvil para el proyecto sería de 56 € aproximadamente.

Herramienta	Coste
Ordenador	42 €
Móvil	56 €
TOTAL	98 €

Tabla 1: Costes hardware del proyecto

1.13.2 Costes software

Los costes de los programas y aplicaciones software detalladas en el [apartado 1.9](#) para este proyecto son:

Herramienta	Coste
Visual Studio	0 €
SmartGit	0 €
GitLab	0 €
Firebase	0 €*

Trello	0 €
Moqups	0 €
TOTAL	0 €

Tabla 2: Costes software del proyecto

Como se puede ver, para todos los programas/aplicaciones se usan versiones gratuitas o, en el caso de SmartGit, una licencia de estudiante, por lo que los costes software son de 0 €.

*Con respecto a Firebase, pese a que el coste en desarrollo es fijo, este podría variar en la etapa de producción cuando el producto esté disponible para un gran número de usuarios. Esto se debe a que el proyecto Firebase de nuestra aplicación se adhiere al plan de pago Blaze (*Pay as you go, paga sobre la marcha*) de forma que el precio a pagar por la aplicación depende del uso que se haga de la misma. Como en la etapa de desarrollos la cantidad de usuarios y actividad es baja (solo para pruebas) no hay coste, pero en el momento en el que la aplicación sea lanzada podría ser necesario realizar pagos [8].

Products	Spark Plan Generous limits for hobbyists Free	Flame Plan Fixed pricing for growing apps \$25/month	Blaze Plan Calculate pricing for apps at scale Pay as you go ✓ Free usage from Spark plan included*
Free Products A/B Testing, Analytics, App Indexing, Authentication (except Phone Auth), Cloud Messaging (FCM), Crashlytics, Dynamic Links, Invites, Performance Monitoring, Predictions, and Remote Config.	✓ Included	✓ Included Free	✓ Included Free
Realtime Database Simultaneous connections ? GB stored GB downloaded Multiple databases per project	100 1 GB 10 GB/month ✗	100k 2.5 GB 20 GB/month ✗	100k/database \$5/GB \$1/GB ✓
Cloud Firestore Stored data Bandwidth Document writes	1 GB total 10GB/month 20K/day	2.5 GB total 20GB/month 100K/day	\$0.18/GB Google Cloud Pricing \$0.18/100K

Ilustración 10: Planes de pago de Firebase

1.13.3 Costes indirectos

Estos costes incluyen electricidad y acceso a Internet.

Respecto a la electricidad, teniendo en cuenta que el ordenador consume unos 72 Wh, el precio de la luz medio en España es de 0,233 €/kWh y la duración del proyecto es de aproximadamente 300 horas, el precio de la electricidad ronda un precio de 5.04 €.

Para la conexión a Internet se usa una tarifa de 30 € al mes, con seis dispositivos conectados normalmente y una duración de trabajo de cinco meses aproximadamente, el coste de la conexión a Internet para la realización del proyecto ronda los 25 €.

Herramienta	Coste
Electricidad	5,04 €
Conexión a Internet	25 €
TOTAL	30,04 €

Tabla 3: Costes indirectos del proyecto

1.13.4 Costes de personal

Para realizar los cálculos de costes de personal, se va a suponer que a cargo del proyecto hay un arquitecto de software junior y un programador junior, basando su sueldo en la media española según su puesto y experiencia.

El arquitecto software es el encargado de toda la planificación del proyecto, realizar el análisis y diseño y trasladarlo al programador. Su tiempo de trabajo sería aproximadamente el 40% del tiempo total del proyecto, 120 horas. El sueldo medio de un arquitecto software junior en España es de 14.42 €/hora, por tanto, su sueldo total sería de 1730.40 €.

El programador es el encargado de la implementación del proyecto mediante la escritura del código fuente, la puesta en marcha del backend y las pruebas. Su tiempo de trabajo sería aproximadamente el 60% del tiempo total del proyecto, 180 horas. El sueldo medio de un programador junior en España es de 10.98 €/hora, por tanto, su sueldo total sería de 1976.40 €.

Personal	Coste
----------	-------

Arquitecto software	1730,40 €
Programador	1976,40 €
TOTAL	3706,80 €

Tabla 4: Costes de personal del proyecto

1.13.5 Resumen del presupuesto

De acuerdo con los apartados anteriores, la tabla del presupuesto global del proyecto queda así:

Herramienta	Coste
Costes hardware	98 €
Costes software	0 €
Costes indirectos	30,04 €
Costes de personal	3706,80 €
TOTAL	3834,84 €

Tabla 5: Resumen del presupuesto

2 ANÁLISIS

2.1 Análisis preliminar

Se pretende implementar un sistema de test interactivos adaptados al usuario. Para ello, es necesario suministrar un banco de preguntas que se dividirá en 6 temas. Cuando un usuario seleccione un tema, el sistema creará automáticamente un test adaptado a él de 10 preguntas basándose en preguntas que el usuario aún no ha contestado correctamente o, en su defecto, preguntas que haya contestado correctamente hace un tiempo determinado.

El acceso a los test será a través de un panel principal en el que se muestran los seis temas y el porcentaje de progreso en cada uno, de manera que el usuario pueda comprobar rápidamente y de forma visual en qué temas tiene mayor conocimiento.

También habrá un ranking mensual y total en el que se muestran los mejores cinco usuarios en cada apartado.

Será necesario un sistema de autenticación con recuperación de contraseña y un menú de administrador con apartado de usuarios (para crear, editar y eliminar usuarios) y preguntas (para añadir, eliminar y editar preguntas).

2.2 Especificaciones del sistema

En este punto se presentan de forma resumida los requisitos funcionales y no funcionales del proyecto, los cuales se detallarán mejor en los próximos dos puntos.

Los requisitos funcionales del sistema son:

1. **Registrar usuario:** para poder realizar todas las funcionalidades del sistema, el usuario debe estar registrado. El registro es llevado a cabo por parte del administrador en el menú administrador. Conlleva el envío de un correo electrónico al nuevo usuario para que establezca una contraseña.
2. **Iniciar sesión:** los usuarios que hayan sido registrados por el administrador podrán hacer uso de sus credenciales para acceder al sistema y sus funcionalidades.
3. **Recuperar contraseña:** en caso de que un usuario haya olvidado su contraseña, puede pedir que se le envíe un email para restablecerla.
4. **Cerrar sesión:** el usuario debe poder cerrar la sesión iniciada en todo momento.
5. **Editar perfil:** el usuario debe poder editar la información básica de su perfil en todo momento, tanto su nombre como su contraseña.
6. **Panel de temas:** cuando el usuario vaya al panel principal de la aplicación, se le mostrará un panel con los seis temas disponibles y el porcentaje de su avance en cada uno de ellos. Podrá clicar en cualquiera de ellos para acceder a un test del tema seleccionado.
7. **Indicador de progreso:** al pulsar sobre un tema, se deben dar al usuario estadísticas simples sobre su progreso en el tema, como el número de preguntas contestadas en total, avance respecto a la media...

8. **Test:** cuando un usuario clique en un tema, se dispondrá un test de 10 preguntas con 5 opciones cada una sin tiempo máximo sobre el que podrá navegar y responder las preguntas en el orden deseado.
9. **Corrección:** al finalizar el test, se mostrará al usuario una corrección del mismo con las preguntas que ha contestado correctamente en verde y las que ha contestado mal en rojo y mostrando la respuesta correcta junto con una explicación.
10. **Ranking:** un usuario puede consultar en todo momento un ranking mensual y total en el que se muestran los mejores cinco usuarios en cada apartado y la posición del usuario que consulta.
11. **Eliminar usuario:** un administrador debe poder eliminar al usuario que desee en todo momento.
12. **Editar usuario:** un administrador debe poder editar el nombre del usuario que desee en todo momento, además de convertirlo en administrador o negarle este rol.
13. **Añadir pregunta:** un administrador debe poder añadir una pregunta al sistema perteneciente a cualquiera de los seis temas.
14. **Editar pregunta:** un administrador debe poder editar cualquier aspecto relativo a una pregunta excepto su id.
15. **Eliminar pregunta:** un administrador debe poder eliminar cualquier pregunta de la base de datos.

Los requisitos no funcionales del sistema son:

1. **Interfaz usable:** la interfaz de usuario debe ser vistosa y agradable para el mismo en todo momento, además de que las funcionalidades disponibles deben estar accesibles con facilidad.
2. **Bajos tiempos de carga:** los tiempos de carga en cualquier funcionalidad de la aplicación deben ser de dos segundos como máximo.
3. **Multiplataforma:** el diseño web debe adaptarse a pantallas de todas las dimensiones sin perder usabilidad ni resentirse la calidad del diseño.

4. **Seguridad general de la web:** deben tomarse las precauciones de diseño necesarias para evitar ataques de seguridad clásicos.
5. **Seguridad de los datos:** debe contener una capa de seguridad robusta que no permita el acceso a las preguntas o a la información de los usuarios a ningún usuario no administrador.

2.3 Casos de uso

En este apartado se va a entrar más en detalle respecto a los requisitos funcionales del sistema, mostrando mediante tablas de casos de uso la comunicación y el comportamiento del sistema mediante la interacción con usuarios.

Caso de uso Registrar usuario:

Caso de uso	Registrar usuario
Actor	Administrador
Descripción	El administrador quiere añadir un nuevo usuario a la base de datos
Precondición	Que el usuario esté registrado y tenga rol de administrador
Escenario principal	1. El administrador accede a la sección de usuarios del menú de administrador 2. El sistema muestra una lista de usuarios 3. El administrador accede al apartado de nuevo usuario 4. El sistema devuelve un formulario para ingresar correo y nombre del usuario 5. El administrador introduce los datos y pulsa Aceptar 6. El sistema valida los datos introducidos 7. El sistema registra al usuario en el sistema 8. El sistema muestra la lista de usuarios con un mensaje de aprobación y envía un correo al nuevo usuario
Escenario alternativo	5. El administrador pulsa Cancelar 5.1 El sistema muestra la lista de usuarios sin crear un nuevo usuario 6. El sistema valida los datos introducidos 6.1 El sistema comprueba que ya existe un usuario con ese correo 6.2 El sistema vuelve a la lista de usuarios con un mensaje de error y no crea el usuario

Tabla 6: Caso de uso Registrar usuario

Caso de uso Iniciar sesión:

Caso de uso Iniciar sesión	
Actor	Usuario
Descripción	El usuario quiere iniciar sesión para acceder a las funcionalidades de la plataforma
Precondición	Que el usuario no esté registrado en la plataforma
Escenario principal	1. El usuario accede al sistema sin estar registrado 2. El sistema muestra un formulario para que introduzca su correo y contraseña 3. El usuario introduce su correo y contraseña y pulsa en enviar 4. El sistema valida los datos introducidos 5. El sistema inicia sesión para el usuario y le muestra el panel principal
Escenario alternativo	4. El sistema valida los datos introducidos 4.1 El sistema no encuentra una cuenta con el correo introducido o la contraseña no es correcta 4.2 Se vuelve a mostrar el formulario de inicio de sesión al usuario junto con un mensaje de error de validación de datos

Tabla 7: Caso de uso Iniciar sesión

Caso de uso Recuperar contraseña:

Caso de uso		Recuperar contraseña
Actor		Usuario
Descripción		El usuario ha olvidado su contraseña y quiere recuperarla
Precondición		Que el usuario no esté registrado en la plataforma
Escenario principal		1. El usuario accede al sistema sin estar registrado 2. El sistema muestra un formulario para que introduzca su correo y contraseña 3. El usuario pulsa sobre el botón de olvido de contraseña 4. El sistema muestra un formulario para que el usuario introduzca su correo 5. El usuario introduce su correo y pulsa enviar 6. El sistema valida los datos introducidos 7. El sistema envía un correo al usuario con un enlace para recuperar contraseña 8. El usuario accede al enlace 9. El sistema muestra al usuario un formulario para que introduzca la nueva contraseña 10. El usuario introduce la nueva contraseña 11. El sistema valida la nueva contraseña 12. El sistema muestra un mensaje de aprobación
Escenario alternativo		6. El sistema valida los datos introducidos 6.1 El sistema comprueba que no existe ninguna cuenta en el sistema con ese correo 6.2 El sistema vuelve a mostrar el formulario al usuario con un mensaje de error 11. El sistema valida la nueva contraseña 11.1 El sistema comprueba que la contraseña tiene menos de seis dígitos

	11.2 El sistema vuelve a mostrar el formulario al usuario con un mensaje indicando la longitud mínima de la contraseña
--	--

Tabla 8: Caso de uso Recuperar contraseña

Caso de uso Cerrar sesión:

Caso de uso	Cerrar sesión
Actor	Usuario
Descripción	El usuario quiere cerrar sesión en la plataforma
Precondición	Que el usuario esté registrado en la plataforma
Escenario principal	1. El usuario pulsa en el botón de perfil -> cerrar sesión 2. El sistema muestra una pregunta de confirmación con las opciones cancelar y confirmar 3. El usuario pulsa confirmar 4. El sistema muestra el formulario de iniciar sesión
Escenario alternativo	3. El usuario pulsa cancelar 3.1 El sistema muestra la vista en la que el usuario estaba anteriormente

Tabla 9: Caso de uso Cerrar sesión

Caso de uso Editar perfil:

Caso de uso	Editar perfil
Actor	Usuario
Descripción	El usuario quiere editar la información básica de su perfil
Precondición	Que el usuario esté registrado en la plataforma
Escenario principal	1. El usuario pulsa en el botón de perfil -> editar perfil

	2. El sistema muestra un formulario con su nombre y un campo vacío para cambiar la contraseña si es necesario 3. El usuario rellena los campos deseados y pulsa enviar 4. El sistema verifica los datos introducidos 5. El sistema edita la información del usuario en la base de datos 6. El sistema muestra el panel principal con un mensaje de confirmación
Escenario alternativo	3. El usuario pulsa cancelar 3.1 El sistema muestra el panel principal sin realizar cambios 4. El sistema verifica los datos introducidos 4.1 El sistema comprueba que la contraseña tiene menos de seis dígitos 4.2 El sistema vuelve a mostrar el formulario al usuario con un mensaje indicando la longitud mínima de la contraseña

Tabla 10: Caso de uso Editar perfil

Caso de uso Indicador de progreso:

Caso de uso	Indicador de progreso
Actor	Usuario
Descripción	El usuario quiere obtener estadísticas más detalladas de su progreso y conocimiento en un tema concreto
Precondición	Que el usuario esté registrado en la plataforma
Escenario principal	1. El usuario accede al panel principal de la plataforma 2. El sistema muestra al usuario los seis temas y una barra con el porcentaje de progreso en cada uno 3. El usuario selecciona un tema

	4. El sistema muestra su porcentaje de progreso, el número de respuestas que ha contestado correcta y erróneamente, su avance respecto a la media...
--	--

Tabla 11: Caso de uso Indicador de progreso

Caso de uso Realización de test y corrección:

Caso de uso	Realización de test y corrección
Actor	Usuario
Descripción	El usuario quiere realizar el test de un tema en concreto para aumentar su conocimiento sobre el mismo
Precondición	Que el usuario esté registrado en la plataforma
Escenario principal	1. El usuario accede al panel principal de la plataforma 2. El sistema muestra al usuario los seis temas y una barra con el porcentaje de progreso en cada uno 3. El usuario selecciona un tema 4. El sistema muestra su porcentaje de progreso, el número de respuestas que ha contestado correcta y erróneamente, su avance respecto a la media... 5. El usuario pulsa en Realizar test 6. El sistema obtiene 10 preguntas adaptadas al usuario 7. El sistema muestra las preguntas al usuario permitiéndole navegar por el test 8. El usuario responde las preguntas y pulsa en Terminar 9. El sistema corrige las respuestas, recoge las respuestas del usuario y las almacena en la base de datos 10. El sistema muestra una corrección con las preguntas que el usuario ha contestado correctamente en verde y las que ha

	contestado mal en rojo mostrando la respuesta correcta junto con una explicación
--	--

Tabla 12: Caso de uso Realización de test y corrección

Caso de uso Ranking:

Caso de uso	Ranking
Actor	Usuario
Descripción	El usuario quiere conocer la clasificación de los mejores usuarios y su posición
Precondición	Que el usuario esté registrado en la plataforma
Escenario principal	1. El usuario pulsa en el botón de Ranking 2. El sistema muestra una clasificación mensual y total de los cinco mejores usuarios

Tabla 13: Caso de uso Ranking

Caso de uso Eliminar usuario:

Caso de uso	Eliminar usuario
Actor	Administrador
Descripción	El administrador quiere eliminar un usuario de la plataforma
Precondición	Que el usuario esté registrado y tenga rol de administrador
Escenario principal	1. El administrador accede a la sección de usuarios del menú de administrador 2. El sistema muestra una lista de usuarios 3. El administrador accede a la fila del usuario que quiere eliminar y pulsa en el botón Eliminar

	4. El sistema muestra un mensaje de confirmación con las opciones Cancelar y Eliminar 5. El usuario pulsa en Eliminar 6. El sistema muestra la lista de usuarios con un mensaje de aprobación y sin el usuario recientemente eliminado
Escenario alternativo	5. El usuario pulsa en Cancelar 5.1 El sistema muestra la lista de usuarios sin eliminar ninguno

Tabla 14: Caso de uso Eliminar usuario

Caso de uso Editar usuario:

Caso de uso Editar usuario	
Actor	Administrador
Descripción	El administrador quiere editar la información básica de un usuario
Precondición	Que el usuario esté registrado y tenga rol de administrador
Escenario principal	1. El administrador accede a la sección de usuarios del menú de administrador 2. El sistema muestra una lista de usuarios 3. El administrador accede a la fila del usuario que quiere editar y pulsa en el botón Editar 4. El sistema devuelve un formulario para cambiar el nombre de usuario 5. El administrador introduce los datos y pulsa Aceptar 6. El sistema muestra la lista de usuarios con un mensaje de aprobación
Escenario alternativo	5. El administrador pulsa Cancelar 5.1 El sistema muestra la lista de usuarios sin editar el usuario

Tabla 15: Caso de uso Editar usuario

Caso de uso Añadir pregunta:

Caso de uso Añadir pregunta	
Actor	Administrador
Descripción	El administrador quiere añadir una pregunta al sistema
Precondición	Que el usuario esté registrado y tenga rol de administrador
Escenario principal	1. El administrador accede a la sección de preguntas del menú de administrador, seleccionando el tema del que quiera agregar la pregunta 2. El sistema muestra el listado de preguntas de ese tema 3. El administrador pulsa en Añadir pregunta 4. El sistema muestra un formulario con enunciado, opciones, explicación y solución de la pregunta 5. El administrador asigna los atributos de la pregunta y pulsa Aceptar 6. El sistema añade la pregunta y muestra el listado de preguntas de ese tema con la nueva pregunta añadida
Escenario alternativo	5. El usuario pulsa Cancelar 5.1 El sistema muestra el listado de preguntas de ese tema sin realizar cambios

Tabla 16: Caso de uso Añadir pregunta

Caso de uso Editar pregunta:

Caso de uso Editar pregunta	
Actor	Administrador
Descripción	El administrador quiere editar los atributos de una pregunta

Precondición	Que el usuario esté registrado y tenga rol de administrador
Escenario principal	1. El administrador accede a la sección de preguntas del menú de administrador, seleccionando el tema del que quiera editar la pregunta 2. El sistema muestra el listado de preguntas de ese tema 3. El administrador busca la pregunta que quiere editar y pulsa Editar 4. El sistema muestra un formulario con enunciado, opciones, explicación y solución de la pregunta 5. El administrador cambia los atributos que considere necesarios y pulsa Aceptar 6. El sistema modifica la pregunta y muestra el listado de preguntas de ese tema con los cambios realizados
Escenario alternativo	5. El usuario pulsa Cancelar 5.1 El sistema muestra el listado de preguntas de ese tema sin realizar cambios

Tabla 17: Caso de uso Editar pregunta

Caso de uso Eliminar pregunta:

Caso de uso Eliminar pregunta	
Actor	Administrador
Descripción	El administrador quiere eliminar una pregunta
Precondición	Que el usuario esté registrado y tenga rol de administrador
Escenario principal	1. El administrador accede a la sección de preguntas del menú de administrador, seleccionando el tema del que quiera agregar las preguntas 2. El sistema muestra el listado de preguntas de ese tema 3. El administrador busca la pregunta que quiere eliminar y pulsa Eliminar

	4. El sistema muestra un mensaje de confirmación con las opciones Cancelar y Eliminar 5. El usuario pulsa en Eliminar 6. El sistema elimina la pregunta y muestra el listado de preguntas de ese tema
Escenario alternativo	5. El usuario pulsa Cancelar 5.1 El sistema muestra el listado de preguntas de ese tema sin realizar cambios

Tabla 18: Caso de uso Eliminar pregunta

2.4 Historias de usuario

Las historias de usuario son tareas de desarrollo que se suelen expresar como "persona + necesidad + propósito". Se tratan de explicaciones generales e informales de una función software desde el punto de vista del usuario final. Mientras que los casos de uso del apartado anterior muestran de forma concreta cómo debe funcionar el sistema ante cualquier acción que se pueda realizar en el mismo, las historias de usuario solo muestran objetivos finales desde el punto de vista del usuario.

Debido a la metodología ágil a la que se adhiere este proyecto, las historias de usuario resultarán de gran ayuda, ya que permiten determinar objetivos y clasificarlos en iteraciones de forma que la realización del proyecto sea ordenada y clara, llegando a los objetivos finales. Simbolizan pequeños retos que se irán cumpliendo a lo largo del desarrollo.

Cada historia de usuario tiene las siguientes características:

- Nombre de la historia de usuario.
- Rol: perfil del usuario que va a realizar la acción en el software. Puede ser administrador o cualquier usuario.
- Funcionalidad: qué acción es la que el usuario pretende realizar.
- Resultado: cuál es la finalidad que el usuario pretende conseguir mediante la realización de la acción.
- Puntuación: se va a puntuar cada una de las historias de usuario en base al esfuerzo estimado para implantarlas en el sistema. Para ello se va a utilizar

una escala creciente de Fibonacci (1, 2, 3, 5, 8, 13). Esto permitirá repartir el esfuerzo entre las iteraciones del desarrollo.

La tabla final de las historias de usuario queda así:

Nombre	Rol	Funcionalidad	Resultado	Puntuación
Registrar usuario	Como administrador	Quiero añadir un nuevo usuario	Para que pueda acceder a la plataforma	5
Iniciar sesión	Como usuario	Quiero iniciar sesión	Para usar las funcionalidades	3
Recuperar contraseña	Como usuario	Quiero recuperar mi contraseña	Para poder volver a iniciar sesión	5
Cerrar sesión	Como usuario	Quiero cerrar sesión	Para que mi sesión no quede iniciada en el dispositivo	1
Editar perfil	Como usuario	Quiero editar mi perfil	Para cambiar mi nombre y/o contraseña	2
Panel de temas	Como usuario	Quiero seleccionar un tema	Para ver mi progreso y/o realizar un test sobre él	8
Indicador de progreso	Como usuario	Quiero ver mi progreso detallado	Para saber mi conocimiento sobre un tema	8
Test	Como usuario	Quiero realizar un test del	Para aumentar mi	13

		tema seleccionado	conocimiento sobre un tema	
Corrección	Como usuario	Quiero ver la corrección de mi test	Para saber en qué preguntas he fallado, cuál era la respuesta y por qué	3
Ranking	Como usuario	Quiero ver el ranking mensual y general	Para saber quiénes son los mejores usuarios y mi posición	8
Eliminar usuario	Como administrador	Quiero eliminar usuarios	Para que un usuario que no está ya en el curso no pueda acceder a las funcionalidades	2
Editar usuario	Como administrador	Quiero editar usuarios	Para cambiar el nombre de un usuario concreto	2
Añadir pregunta	Como administrador	Quiero añadir preguntas a la base de datos	Para que esté disponible en los test	2
Editar pregunta	Como administrador	Quiero editar preguntas de	Para cambiar atributos de	2

		la base de datos	preguntas erróneas	
Eliminar pregunta	Como administrador	Quiero eliminar preguntas de la base de datos	Para eliminar preguntas erróneas	2

Tabla 19: Historias de usuario

3 DISEÑO INICIAL

Dado que el desarrollo de este proyecto será mediante una metodología ágil, en este apartado se va a mostrar la idea de diseño inicial, mientras que las modificaciones en funcionalidades, interfaces o distribución de la información serán detalladas en el apartado de desarrollo.

3.1 Diagramas de secuencia

A continuación, se van a mostrar diagramas de secuencia sobre algunos de los procesos que se han considerado más importantes, de forma que se pueda entender a grandes rasgos el funcionamiento de la aplicación en cuanto a comunicación para cada proceso.

3.1.1 Inicio de sesión

Este diagrama muestra el funcionamiento de la aplicación para el inicio de sesión de un usuario registrado.

El usuario introduce su información de inicio de sesión en el formulario que se le suministra en la vista. Una vez enviado, el contenido del formulario es procesado por el código JavaScript del lado frontend, de forma que si el formato no es correcto (por ejemplo, no se ha introducido la dirección de correo correctamente), el usuario recibe directamente un mensaje de error. En caso contrario, el código cliente manda una petición POST a la API REST con la información de inicio de sesión. La API comprueba que ese usuario se encuentra en el sistema y la contraseña es correcta y

en caso afirmativo, envía un mensaje al sistema para que quede registrada su sesión en el navegador, devolviendo al usuario la vista de la página principal. En caso de que la contraseña no sea correcta o el correo no esté registrado, se devuelve al usuario la vista de inicio de sesión con un mensaje de error.

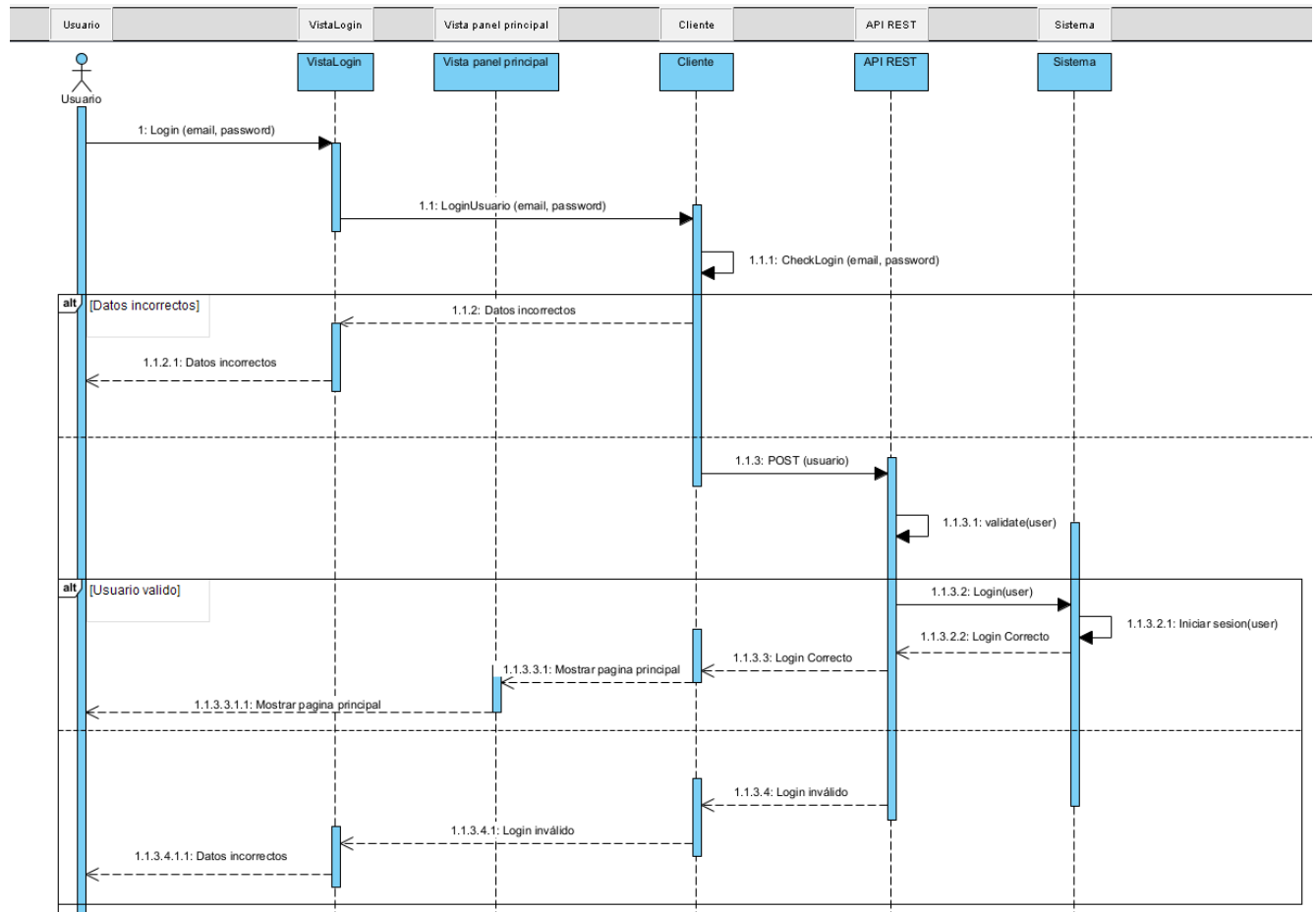


Ilustración 11: Diagrama de secuencia de inicio de sesión

3.1.2 Realización de test

Este diagrama muestra las comunicaciones que se dan en el sistema para que el usuario pueda hacer un test y se le muestre la corrección. Este proceso incluye la creación del test adaptado al usuario.

En primer lugar, el usuario está en la vista del panel principal y selecciona el tema del que quiere hacer el test. Al pulsar sobre un tema recibe información sobre el mismo, y pulsa sobre el botón de realizar test sobre ese tema. Entonces, la vista manda la petición al código del lado cliente.

Para crear el test, en primer lugar, el lado cliente solicita a la API REST todo el banco de preguntas sobre un tema. La API obtiene estas preguntas de la base de

datos y las envía al cliente. En segundo lugar, el cliente hace otra petición GET, en este caso, de la información de respuestas pasadas del usuario a distintos test. Esta información es muy importante para que el test creado esté adaptado al usuario. La API obtiene de nuevo esta información de la base de datos y la manda al cliente. El código cliente se encarga de crear un test con un número determinado de preguntas basándose en el banco total de preguntas y en las respuestas pasadas del usuario.

Una vez que ha creado el test, lo muestra al usuario en modo formulario, a través de la vista creada para ello. El usuario responderá a las preguntas y pulsará en “Terminar test”. Cuando esto ocurra, la vista mandará las respuestas al código del lado cliente para que lo corrija, y se devolverá una corrección mostrando al usuario las respuestas correctas y equivocadas, junto con una explicación.

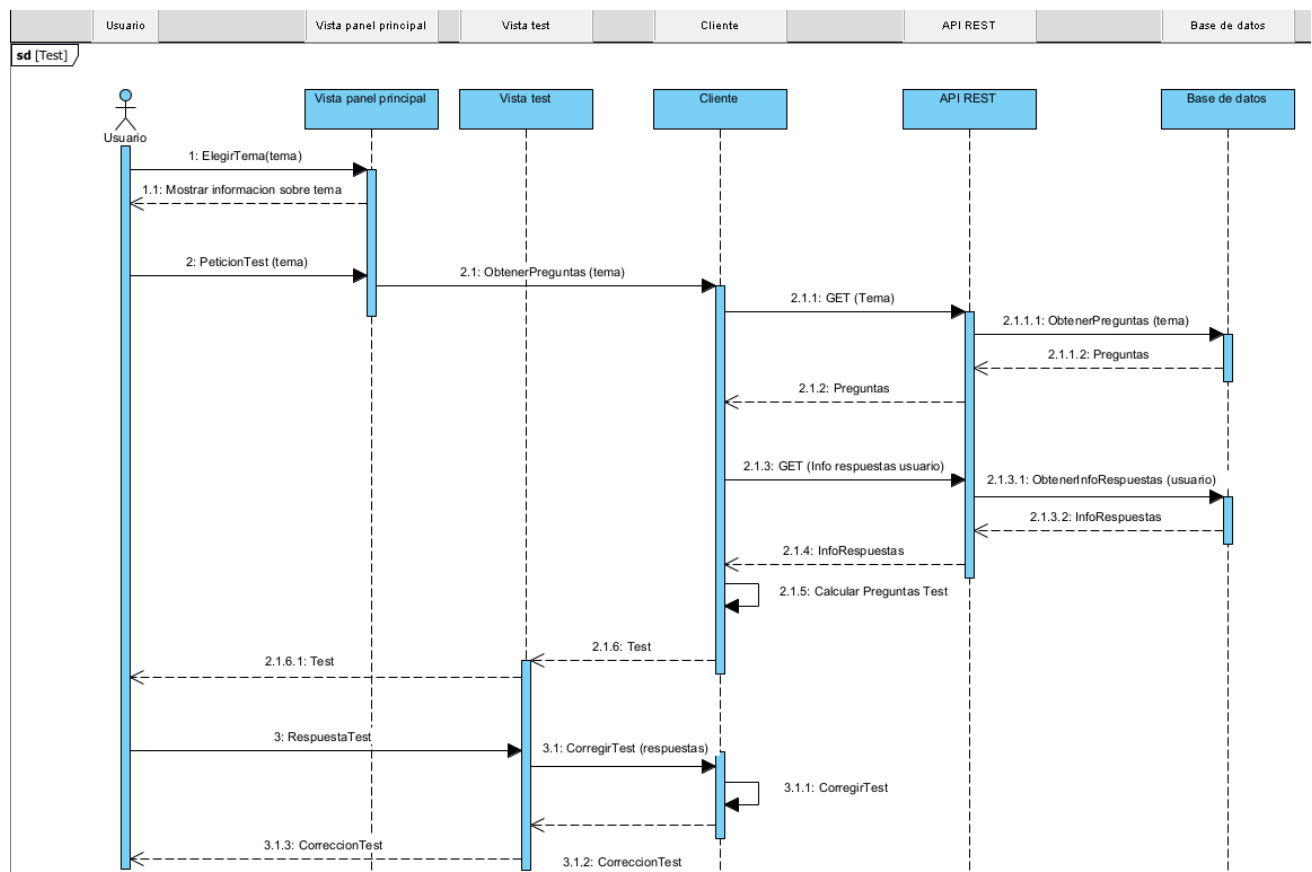


Ilustración 12: Diagrama de secuencia de realización de test

3.1.3 Añadir nuevo usuario

Este diagrama muestra las comunicaciones ocurridas en el sistema para añadir un nuevo usuario. Los usuarios son añadidos por el administrador de la plataforma

siempre, a través del menú suministrado para ello. El administrador rellena un formulario con los datos del nuevo usuario y los envía. El código del lado cliente se comunica con la API REST para añadirlo a la base de datos del sistema. Se comprueba que no haya ya un usuario registrado con ese mismo correo electrónico y que el formato de correo electrónico es correcto. Si ocurre alguno de estos dos errores, se devuelve al administrador un mensaje de error. En caso contrario, se devuelve un mensaje de éxito y el sistema envía automáticamente al nuevo usuario un correo para que este establezca una contraseña en su cuenta, a través de un enlace.

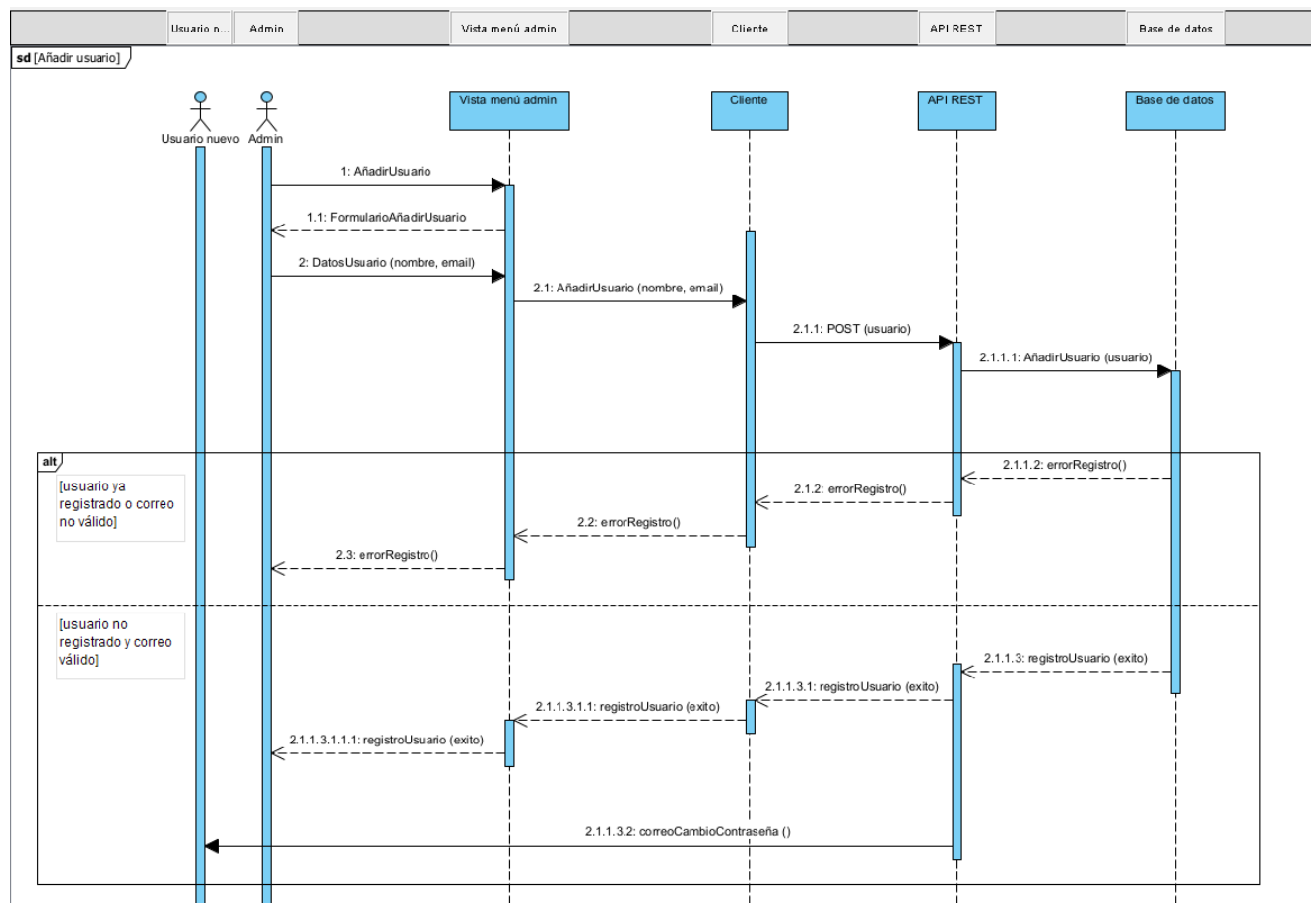


Ilustración 13: Diagrama de secuencia de añadir nuevo usuario

3.2 Diagrama de clases

En este apartado se van a exponer los diagramas de clases para mostrar una información más detallada de la plataforma.

Como la aplicación es desarrollada en React, vamos a tener dos paquetes de clases bien diferenciados. Uno está compuesto por las **entidades**, que conformarán

los documentos en las bases de datos, y otro por los **componentes**, encargados de realizar acciones usando la información de las entidades.

3.2.1 Paquete Entidades

Este paquete conforma las entidades con las que trabaja el servidor. Una entidad representa un objeto real o abstracto acerca del cual se almacena información por ser relevante para el sistema. Por tanto, este diagrama muestra cómo está compuesta la base de datos del sistema, que es el que almacena la información relevante a largo plazo.

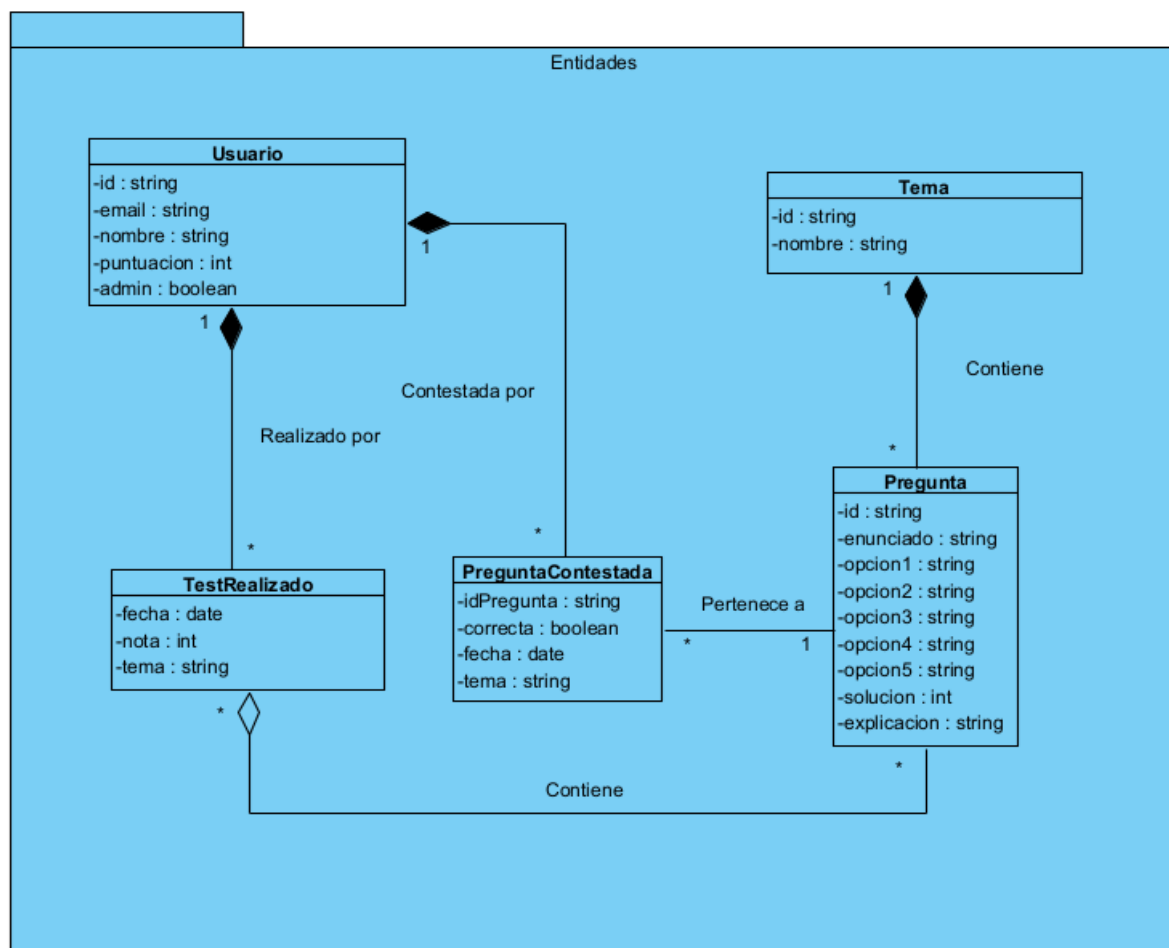


Ilustración 14: Diagrama de clases del paquete entidades

Como se puede ver, por cada test que realice un usuario se almacenará una entidad del test con su nota y fecha, y también n instancias de la entidad PreguntaContestada (siendo n el número de preguntas de un test) de manera que se tenga constancia de las preguntas a las que el usuario ha respondido bien y las que

no. La entidad pregunta contiene toda la información sobre las preguntas para realizar los test. Las preguntas se organizan en temas y los test son siempre sobre un tema.

3.2.2 Paquete Componentes

Este paquete está compuesto por los componentes de React, que son los encargados de tratar la información del paquete Entidades a través de funciones, procesarla y mostrarla al usuario. Una de las grandes ventajas de React es que los componentes son reutilizables en diferentes vistas, por lo que componentes como Header o MenuNavegacion son escritos una vez y se pueden usar en todas las vistas de la plataforma.

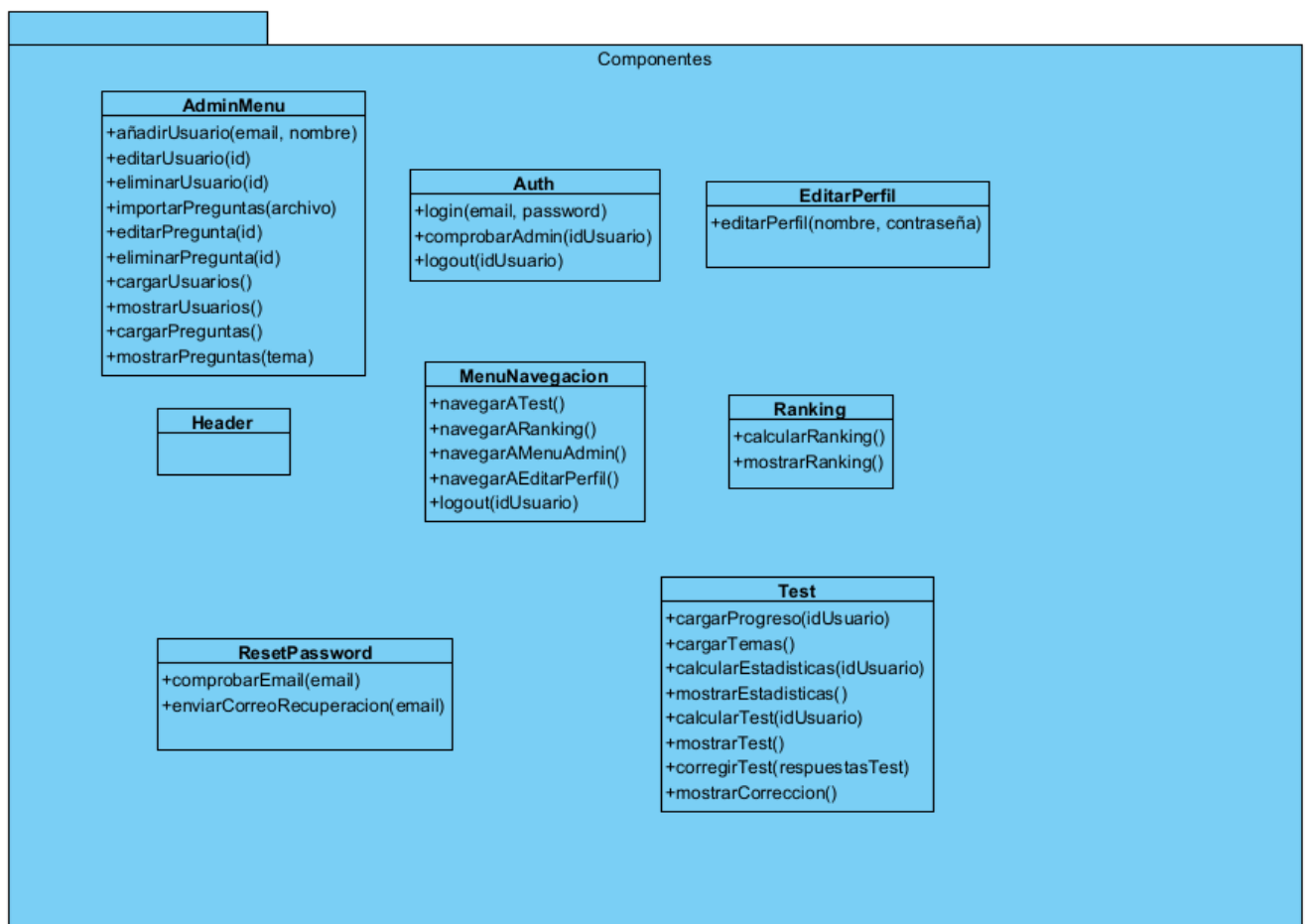


Ilustración 15: Diagrama de clases del paquete componentes

- El menú de administrador (componente AdminMenu) se encarga del tratamiento de usuarios y preguntas.

- El componente Auth muestra la ventana de inicio de sesión y comprueba si el usuario registrado es administrador, para evitarle acceder a las funciones de administrador en caso de que no lo sea.
- El componente MenuNavegacion permite que el usuario se mueva cómodamente por las distintas secciones de la plataforma.
- El componente Ranking se encarga de calcular el ranking semanal y total y mostrarlo al usuario en tablas.
- El componente ResetPassword está creado para que un usuario pueda recuperar su contraseña en caso de olvidarla, mediante el envío de un correo de recuperación de contraseña.
- El componente Test es el que se encarga de mostrar progreso y estadísticas al usuario, calcular las preguntas del test adaptadas al historial de respuestas del usuario, y finalmente mostrar su corrección.

3.3 Diseño de interfaces

En este apartado se va a mostrar el diseño inicial pensado para cada una de las interfaces de la plataforma.

3.3.1 Menú administrador

El menú administrador es el menú a través del cual se pueden consultar, añadir, editar y eliminar usuarios y preguntas. Por tanto, este menú tiene dos secciones.

La sección de usuarios se mostraría así:

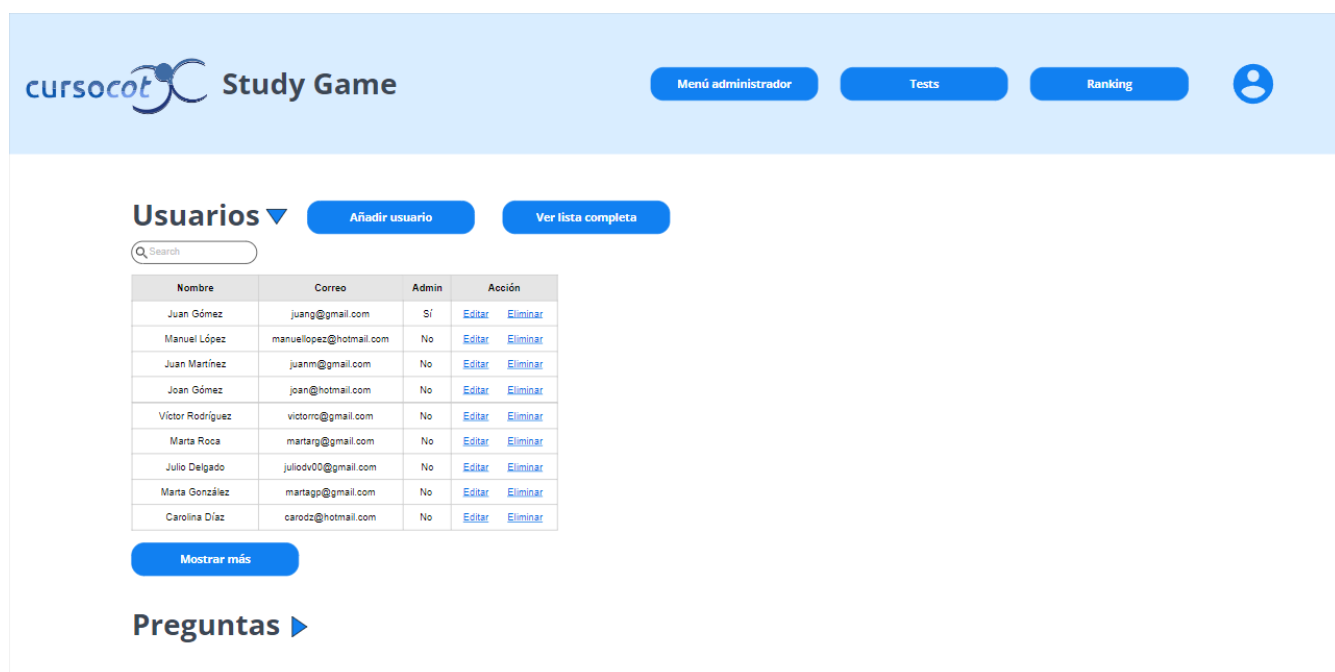


Ilustración 16: Diseño inicial de la sección de usuarios

Esta sería la vista para añadir un nuevo usuario, con un modal que se superpone al contenido de la página:



Ilustración 17: Diseño inicial de la vista Añadir usuario

Respecto a la otra sección, la sección de preguntas, esta sería la vista para consultar preguntas, seleccionando antes el tema de las preguntas que se quieren mostrar:

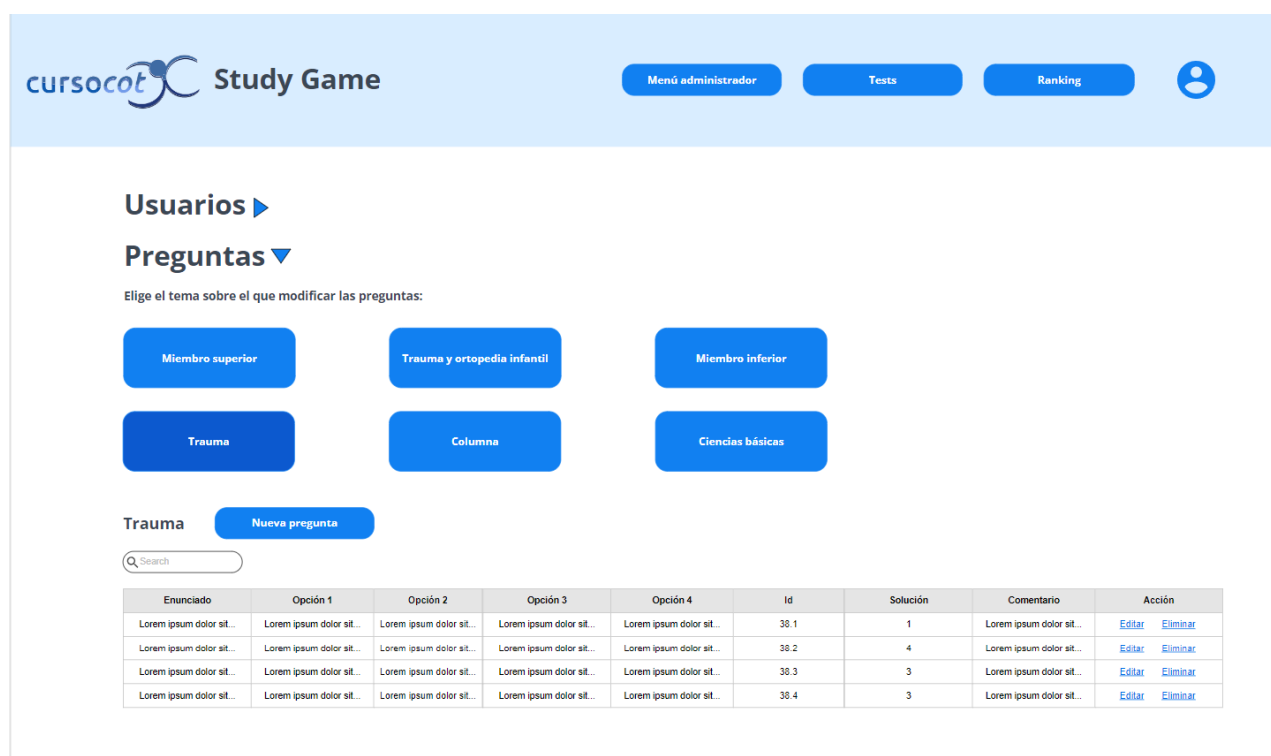


Ilustración 18: Diseño inicial de la sección de preguntas

3.3.2 Inicio de sesión

Este es el diseño de la vista de inicio de sesión:

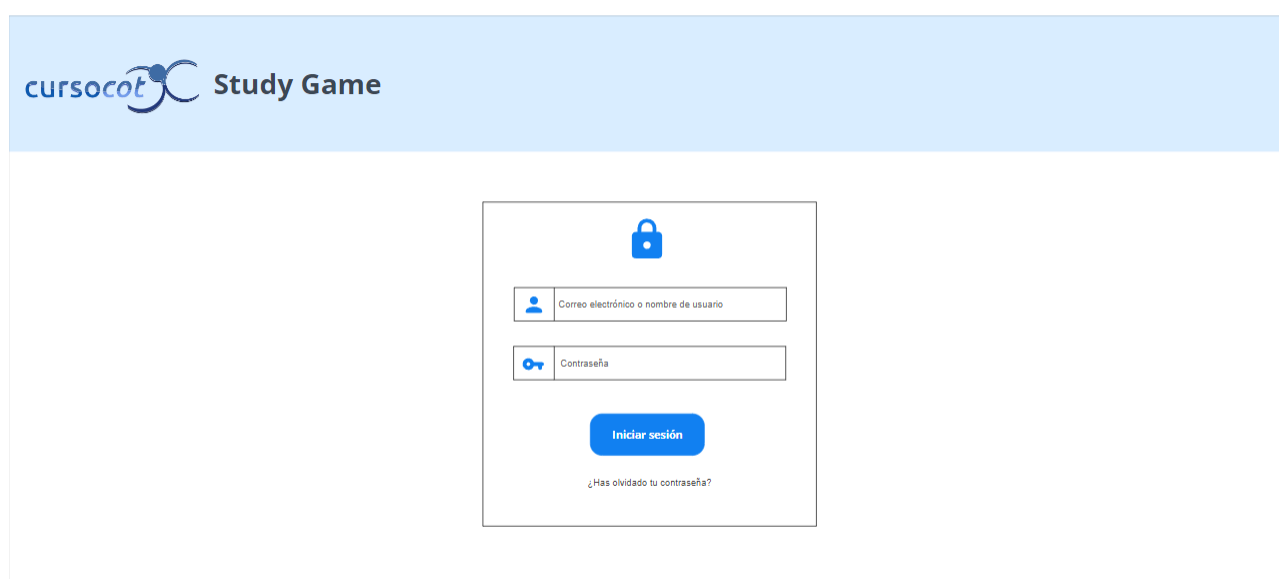


Ilustración 19: Diseño inicial de la vista de inicio de sesión

3.3.3 Panel principal

El panel principal muestra los temas y el progreso que el usuario tiene de cada tema, de forma que seleccione uno para hacer un test. La vista de este panel es la siguiente:



Ilustración 20: Diseño inicial de la vista del panel principal

3.3.4 Realización de test

Este es el diseño inicial para la vista de la realización de un test. Cada pregunta tiene un tiempo, cuando acaba ese tiempo o contesta a la pregunta, pasa a la pregunta siguiente:

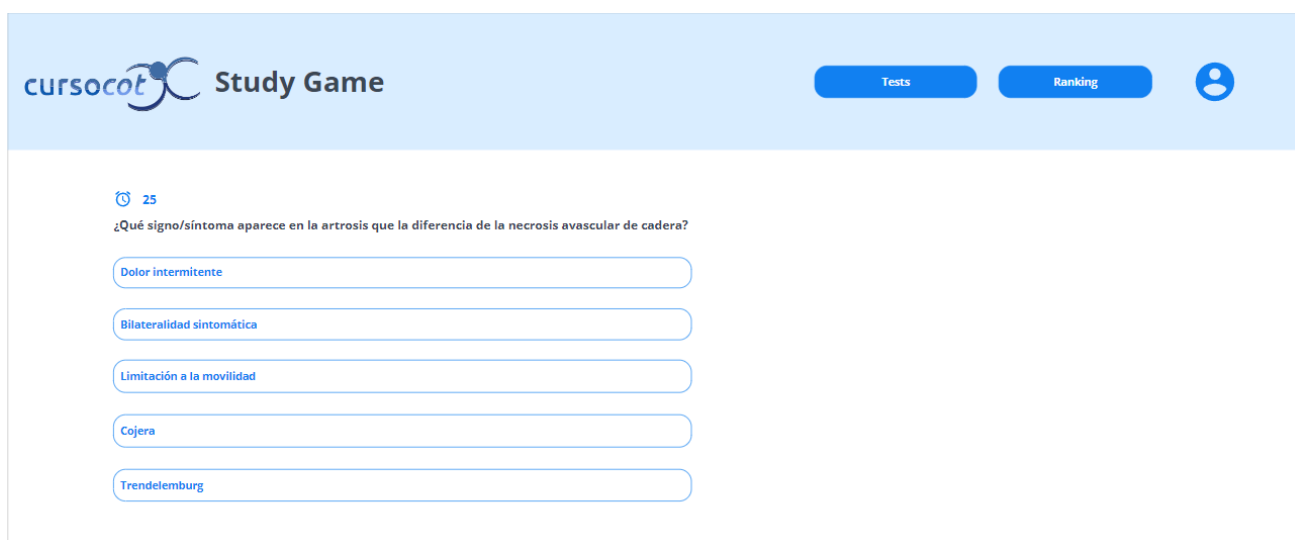


Ilustración 21: Diseño inicial de la vista de realización de test

Una vez el usuario ha terminado de contestar a todas las preguntas, se muestra una corrección. La vista de la corrección se vería así:

cursocot Study Game

Tests Ranking

FIN DEL TEST

9/10

Hacer Nuevo Test

¿Qué signo/síntoma aparece en la artrosis que la diferencia de la necrosis avascular de cadera?

Dolor intermitente

Bilateralidad sintomática

Limitación a la movilidad

Cojera

Trendelenburg

¿Qué signo/síntoma aparece en la artrosis que la diferencia de la necrosis avascular de cadera?

Dolor intermitente

Bilateralidad sintomática

Limitación a la movilidad

Ilustración 22: Diseño inicial de la vista de corrección de test

3.3.5 Editar perfil

Esta sección permite al usuario registrado cambiar su nombre y/o contraseña. Su diseño inicial es:

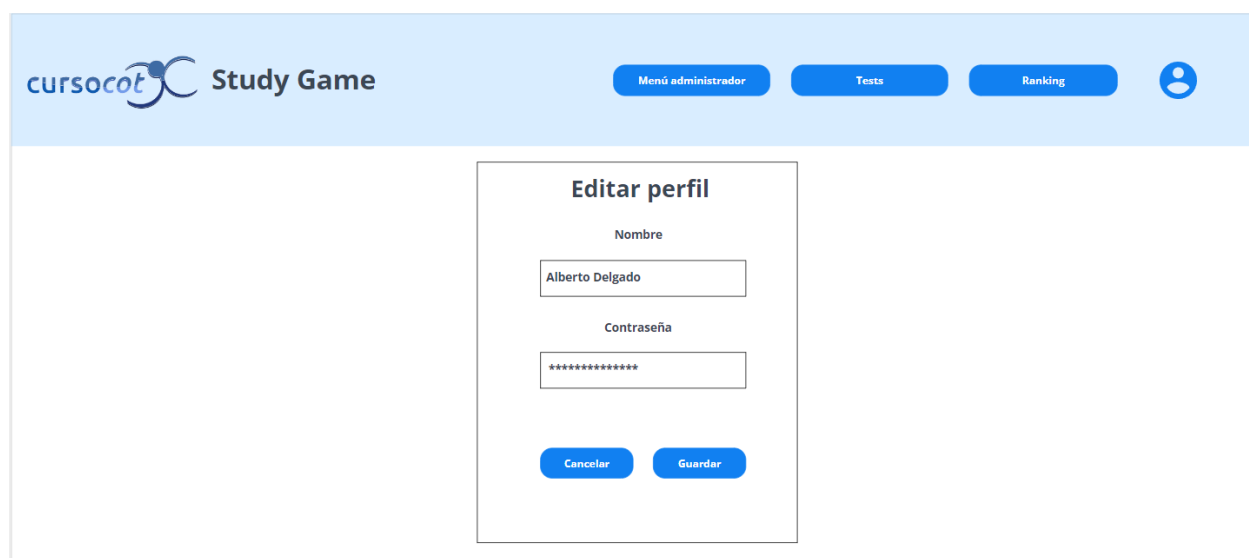
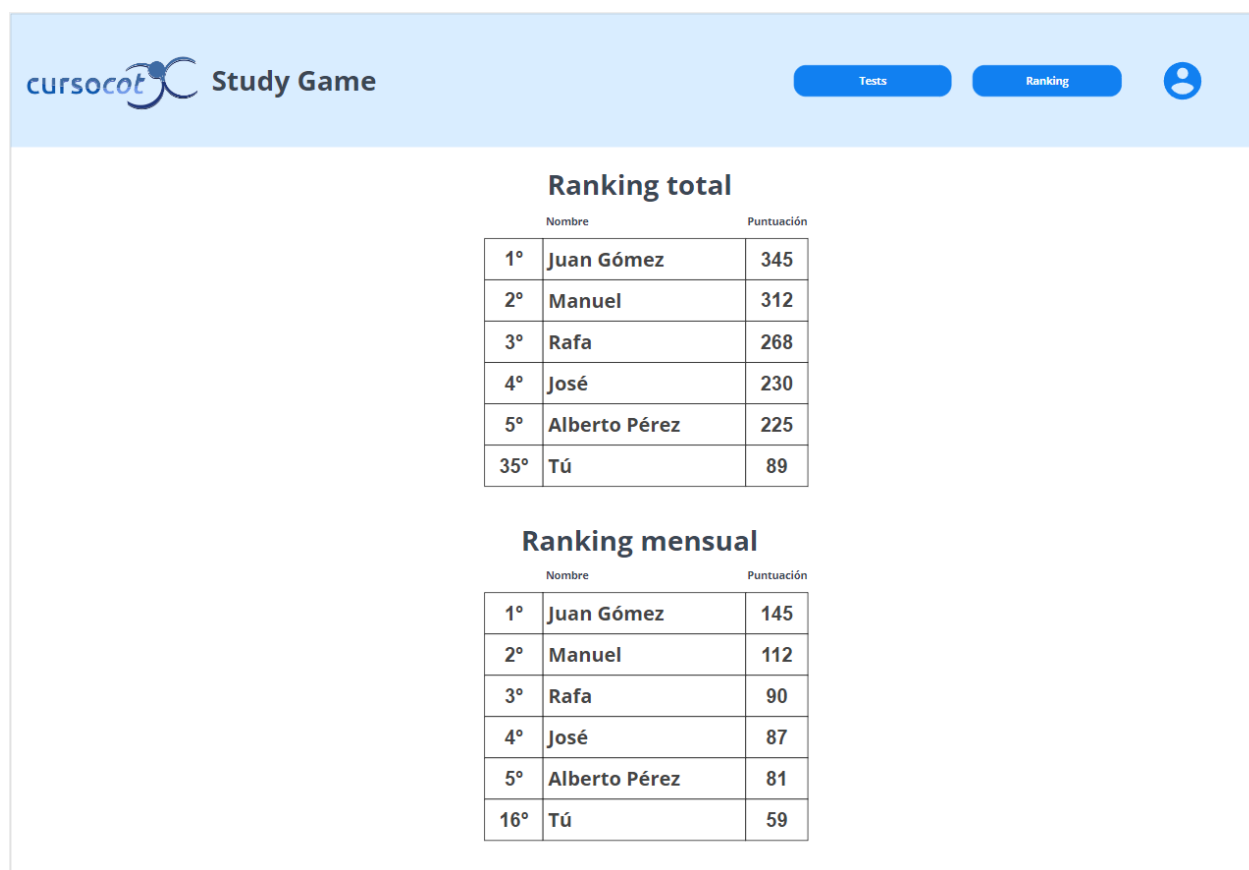


Ilustración 23: Diseño inicial de la vista de editar perfil

3.3.6 Ranking

La última sección que queda por mostrar el diseño es el ranking total y semanal. Esta sería la vista:



Ranking total		
	Nombre	Puntuación
1º	Juan Gómez	345
2º	Manuel	312
3º	Rafa	268
4º	José	230
5º	Alberto Pérez	225
35º	Tú	89

Ranking mensual		
	Nombre	Puntuación
1º	Juan Gómez	145
2º	Manuel	112
3º	Rafa	90
4º	José	87
5º	Alberto Pérez	81
16º	Tú	59

Ilustración 24: Diseño inicial de la vista Ranking

3.4 Diseño del algoritmo de creación de test

En este punto se va a hablar de una parte fundamental del desarrollo de este proyecto: la creación de test adaptados al usuario.

Es necesario crear un algoritmo que, teniendo el banco total de preguntas y la información de realización de test pasados de un usuario, sea capaz de crear para un usuario un test que contenga preguntas que el usuario aún no haya contestado bien, o que haya contestado bien, pero desde su respuesta ya ha pasado un tiempo que se considera límite (fijado por la empresa contratante, dos meses, por ejemplo). De esta manera, se garantiza que cuando un usuario realiza un test, está obteniendo preguntas que, si contesta bien, reflejarán un avance en su progreso.

Cuando un usuario accede al panel principal, se muestran los temas junto con una barra de progreso y un porcentaje. Este porcentaje se calcula de la siguiente manera:

$$\text{Porcentaje progreso} = \frac{n^{\circ} \text{ de preguntas contestadas correctamente}}{n^{\circ} \text{ total de preguntas del tema}} \times 100$$

De esta forma, cuando un usuario realiza un test, cualquier pregunta que responda correctamente significará un avance en su progreso, ya que no le saldrán preguntas repetidas que ya ha contestado bien gracias al algoritmo de creación de test. Un usuario puede estar haciendo test hasta haber contestado correctamente todas las preguntas de un tema, es decir, su progreso llegue al 100%.

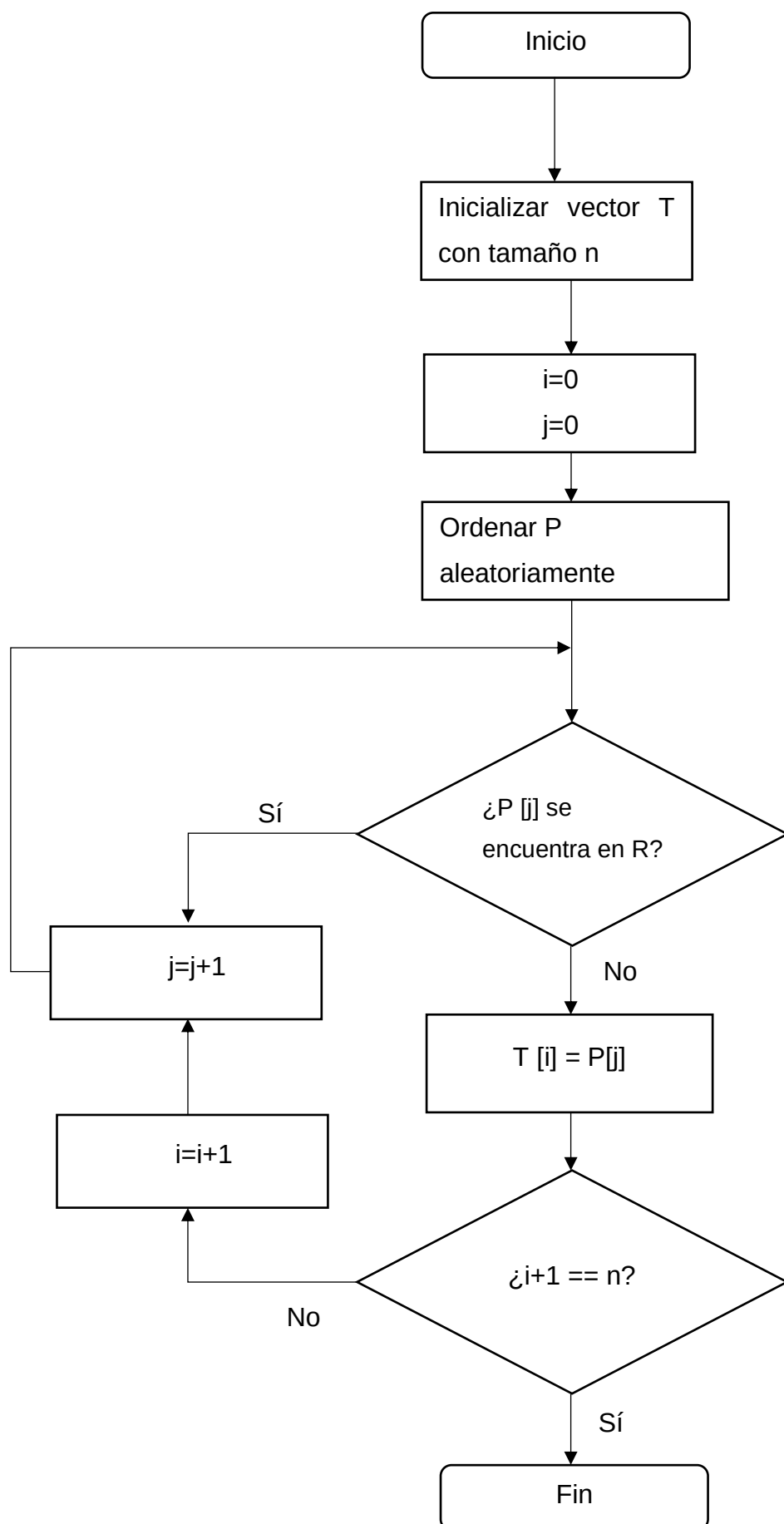
El algoritmo diseñado para crear test de forma eficiente funciona de la siguiente manera:

1. Obtener todas las preguntas de un tema de la base de datos.
2. Obtener el id de las preguntas que el usuario ha contestado bien en test anteriores.
3. Crear una colección de preguntas para el test de tamaño n.
4. Mientras no se haya recopilado el número necesario de preguntas para un test:
 1. Elegir una pregunta aleatoriamente.
 2. Comprobar si esa pregunta ha sido correctamente contestada por el usuario en test anteriores.

3. Si no ha sido contestada correctamente aún, se añade a la colección de preguntas del test. En caso contrario, se descarta.

De esta manera, cuando finalice el algoritmo se habrá recopilado un número n de preguntas para mostrarlas al usuario en el test.

Teniendo en cuenta que P es el vector con todas las preguntas sobre un tema, R es el vector con todas las preguntas respondidas correctamente por el usuario en un determinado plazo, T es el vector con las preguntas del test y n el número de preguntas que debe tener un test, este es el diagrama de flujo del algoritmo de creación de test:



3.5 Arquitectura de la aplicación

En este apartado se va a mostrar la arquitectura del sistema, para mostrar los diferentes elementos constituyentes y la relación de comunicación que hay entre ellos para el correcto funcionamiento del mismo. Para ello, se emplearán diagramas arquitectónicos.

3.5.1 Frontend

Como se ha explicado en el [apartado 1.8](#), el framework elegido para el desarrollo del código cliente de la aplicación es React. Por tanto, la arquitectura del sistema está ligada a este framework.

A diferencia de los marcos tradicionales de Modelo-Vista-Controlador (MVC), React utiliza un patrón de diseño llamado Modelo-Vista-Controlador simplificado, también conocido como Modelo-Vista-Vista de Modelo (MVVM) o Modelo-Vista-Presentador (MVP).

En el patrón MVC, el Modelo representa los datos y la lógica empresarial de la aplicación, la Vista es responsable de mostrar los datos al usuario y el Controlador maneja las interacciones del usuario y coordina la actualización del Modelo y la Vista. En React, el Modelo y el Controlador están combinados en los componentes de React, y la Vista se representa en el árbol de componentes de React.

En React, cada componente tiene su propio estado y maneja sus propias actualizaciones. Los cambios en el estado de un componente pueden provocar una actualización en la interfaz de usuario, y React utiliza su propio algoritmo para actualizar solo los componentes que necesitan ser actualizados en lugar de renderizar toda la interfaz de usuario nuevamente.

De esta forma, se tiene un funcionamiento más eficiente que con el patrón MVC tradicional, pues solo se recargan los componentes que sufren cambios, además de que los componentes son reutilizables, lo cual ahorra bastante tiempo de desarrollo, ya que se puede usar el mismo componente en distintas vistas. Los componentes reutilizables hacen que la aplicación sea también más escalable, fácil de mantener, con mayor calidad de código y más consistencia.

El diagrama arquitectónico del patrón Modelo-Vista-Presentador es el siguiente:

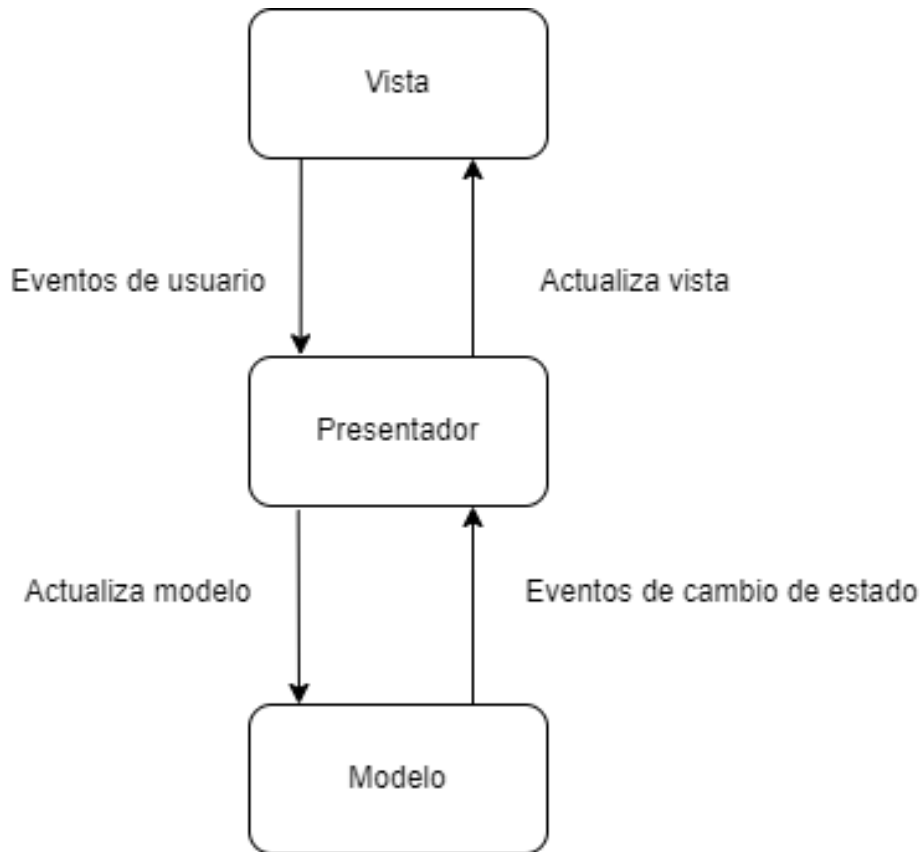


Ilustración 25: Diagrama arquitectónico del frontend

En este diagrama, la Vista es el componente de React que representa la interfaz de usuario. La Vista puede enviar acciones al Presentador para actualizar el estado de la aplicación. El Presentador actúa como intermediario entre la Vista y el Modelo. Recibe las acciones de la Vista, actualiza el estado de la aplicación en el Modelo y luego actualiza la Vista con los nuevos datos.

El Modelo es la capa que contiene los datos y la lógica empresarial. En este caso, el modelo es la API de Firebase que conecta la aplicación React con la base de datos en la nube. El Presentador se comunica con el Modelo para actualizar el estado de la aplicación en función de las acciones de la Vista.

3.5.2 Backend

Como se ha explicado en el [apartado 1.8](#), la plataforma encargada de albergar los datos de la aplicación y actuar como servidor es una instancia de Firebase.

Por tanto, el Modelo de la aplicación explicado en el apartado anterior, que es el encargado de acceder a la capa de almacenamiento de datos, lo hará a través de

la API que Firebase facilita para acceder a sus servicios, entre ellos, la base de datos en tiempo real.

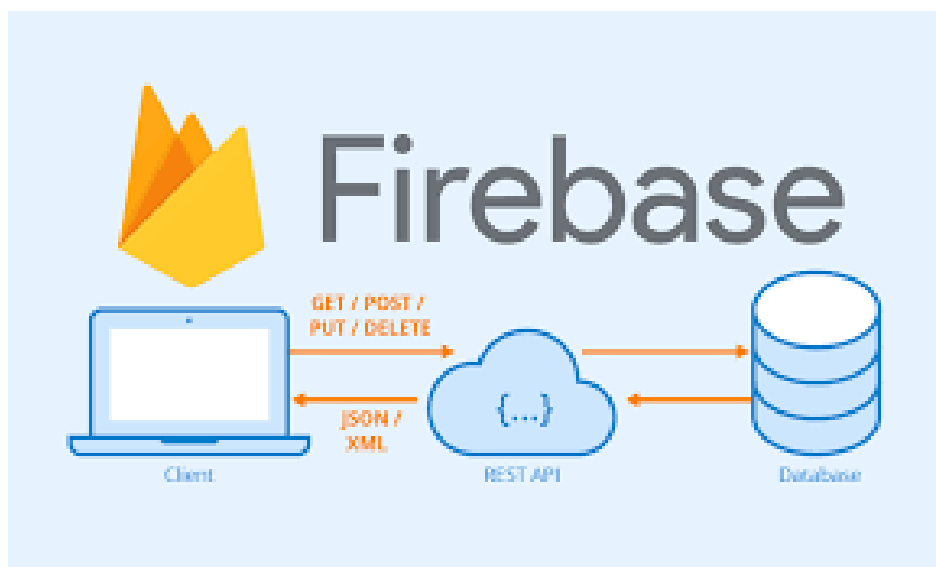


Ilustración 26: Funcionamiento de las comunicaciones en Firebase

Además, Firebase no solo proporciona una base de datos en tiempo real, si no que cuenta con una amplia gama de servicios. Como se ha explicado en el [apartado 1.8](#), los servicios de Firebase que se van a utilizar son: base de datos en tiempo real, autenticación de usuarios y almacenamiento en la nube.

Como ventaja, no es necesaria una API REST distinta para la comunicación con cada uno de estos tres servicios, sino que Firebase proporciona una API unificada que da acceso a los tres SDKs.

Por lo tanto, teniendo en cuenta el patrón Modelo-Vista-Presentador explicado en el apartado anterior y el funcionamiento de la API de Firebase, este es el diagrama arquitectónico de la aplicación completa, tanto cliente como servidor:

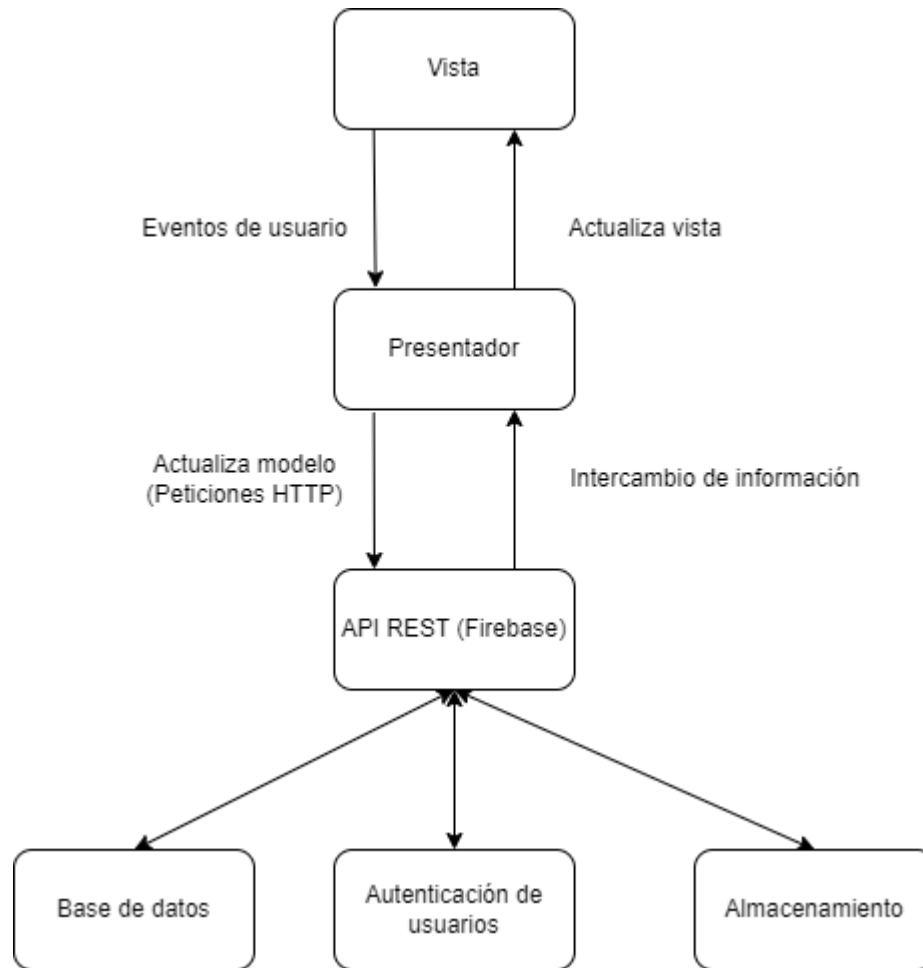


Ilustración 27: Diagrama arquitectónico de la plataforma

4 DESARROLLO

Este apartado contiene todo el proceso de desarrollo del sistema. Debido al enfoque ágil que se ha decidido para el desarrollo del software, la implementación se realizará por iteraciones. En este apartado se detalla el trabajo realizado en cada una de las iteraciones junto con las revisiones, pruebas y cambios que se decidan implementar tras la evaluación para iteraciones futuras.

Pese a que el trabajo realizado en cada iteración irá variando a lo largo del desarrollo debido a los cambios que se decidan implementar en el diseño del sistema, se ha realizado una planificación inicial de las iteraciones. Como se muestra en el [apartado 1.12](#), el tiempo estimado para las iteraciones es entre el 7/12/2022 y el 2/2/2023, por lo que el tiempo estimado para la implementación es de 87 días. Se ha planificado que el proceso de implementación conste de cuatro iteraciones, de forma

que se puedan realizar suficientes evaluaciones y validaciones. Por tanto, el tiempo estimado para cada iteración es de unas tres semanas. La planificación inicial de tareas para cada iteración es la siguiente:

- Primera iteración: Implementación de la estructura de la página (cabecera), puesta a punto del servidor Firebase e implementación de funciones básicas del usuario (iniciar sesión, editar perfil y recuperación de contraseña).
- Segunda iteración: Implementación del menú administrador, tanto de preguntas como de usuarios.
- Tercera iteración: Implementación del mecanismo de creación de test y realización del mismo por el usuario, junto con una corrección.
- Cuarta iteración: Implementación del ranking semanal y total.

Esta planificación podría variar tras las pruebas de validación al final de cada iteración, de forma que en las iteraciones siguientes se deban añadir cambios a las funciones ya implementadas. Podría ser necesaria una quinta iteración tras la implementación del ranking, si existen errores o la empresa contratante sugiere cambios.

4.1 Primera Iteración

4.1.1 Planificación

Debido a que se trata de la primera iteración, el análisis y diseño se corresponde con el análisis y diseño inicial, detallado en los apartados 2 y 3.

4.1.2 Implementación

El primer objetivo de esta iteración es instalar en el ordenador de trabajo la herramienta para el desarrollo de la página y configurar el servidor.

Para instalar la biblioteca React, lo primero fue instalar NodeJS, y a continuación usar el comando para crear una aplicación React desde la consola. La aplicación es creada rápidamente y ya contamos con el conjunto de ficheros necesario para empezar con el desarrollo.

A continuación, es necesario crear un proyecto de Firebase y sincronizarlo con el proyecto. Para ello se accede a la consola de Firebase y se crea un nuevo proyecto.

Firebase suministra unas credenciales que hay que añadir al código frontend para realizar la conexión y poder trabajar con él.

Se incluyeron en Firebase unas reglas de seguridad para impedir el acceso a la base de datos a usuarios no registrados:

```
1  rules_version = '2';
2  service cloud.firestore {
3    match /databases/{database}/documents {
4      match /{document=**} {
5        allow read, write: if
6          request.auth != null;
7      }
8    }
9  }
```

Ilustración 28: Reglas de seguridad del proyecto en Firebase

Una vez realizada la conexión con Firebase desde el código frontend, comienza el trabajo en el desarrollo de la página. El primer paso es desarrollar la cabecera, un elemento común en la interfaz de cualquier sección, tal y como se muestra en el [apartado de diseño de interfaces](#).

Fue diseñada una cabecera siguiendo el modelo del diseño inicial, como se muestra:

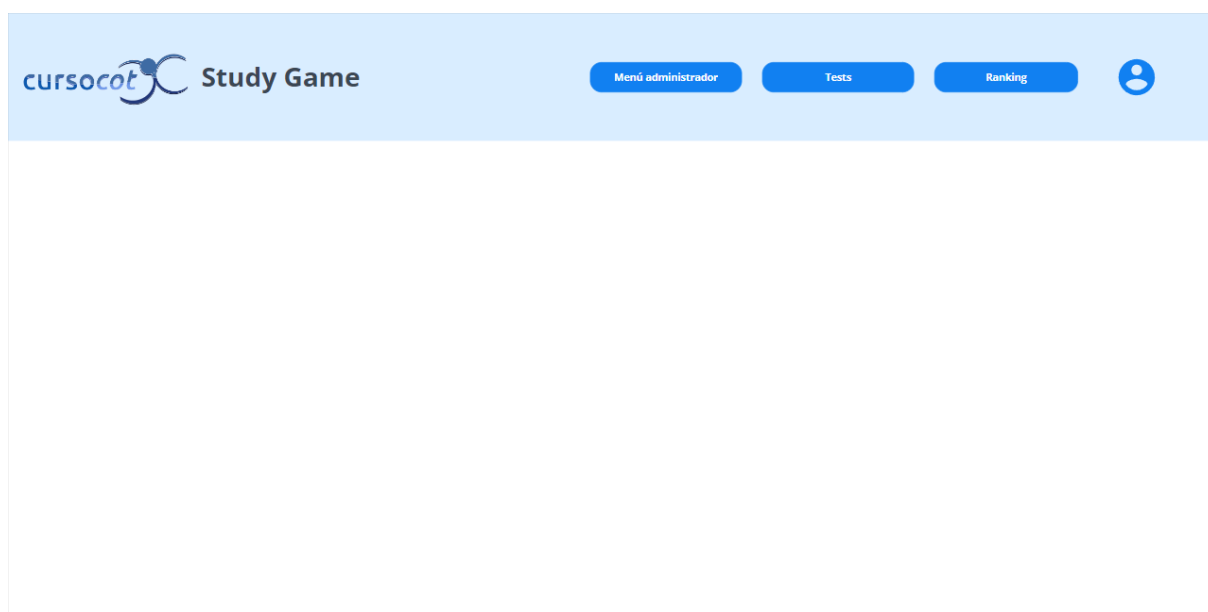


Ilustración 29: Diseño inicial de la cabecera

Una vez realizado este diseño inicial, se continuó con las funciones básicas de usuario: inicio de sesión, editar perfil y recuperación de contraseña. Se realizó el diseño de estas funcionalidades basándose en el diseño inicial, realizando conexión con el SDK de autenticación de React para el inicio de sesión y recuperación de contraseñas.

La recuperación de contraseña es una funcionalidad que envía un correo electrónico al usuario en caso de haber olvidado la contraseña. Primero se comprueba en el sistema de autenticación de Firebase si el usuario existe en el sistema y, en caso de que sí, Firebase envía un correo de recuperación. Este mecanismo de envío de correo electrónico fue fácil de implementar debido a que Firebase lo simplifica con una función de una línea.

4.1.3 Pruebas y validación

Una vez implementadas todas las funciones pensadas para esta iteración, se pasa a la fase de pruebas y validación.

En la fase de pruebas, se determinó que todas las funcionalidades (inicio de sesión, editar perfil, recuperación de contraseña) funcionaban correctamente, se adaptaban a los casos de uso y a las historias de usuario, provocando un resultado satisfactorio.

Sin embargo, en la fase de validación con el tutor, se llegó a la conclusión de que había ciertos aspectos de diseño que corregir para una mejor usabilidad:

- Estructurar el acceso a las secciones mediante botones en la cabecera no es lo más adecuado: conviene crear un menú lateral de navegación que dé acceso a todas las secciones para una mejor gestión de la información.
- La cabecera es demasiado grande, debe ocupar menos espacio en la interfaz.

Por tanto, estos cambios fueron anotados para crear la planificación de la siguiente iteración.

4.2 Segunda iteración

4.2.1 Planificación

Como se ha visto en la fase de pruebas y validación, es necesario realizar un cambio de diseño, reduciendo el tamaño de la cabecera y eliminando los botones que se encuentran en ella, creando un menú de navegación lateral con acceso a estas secciones.

Por tanto, se realizaron cambios en el diseño de interfaces, tal y como se muestra:



Ilustración 30: Diseño del panel inicial tras los cambios de la primera iteración

La cabecera tendrá el mismo tamaño en el resto de secciones, y el menú de navegación estará presente en todas las secciones excepto en la de iniciar sesión (ya que no hay un usuario registrado aún) aunque la opción de “Menú admin” solo estará visible para aquellos usuarios que sean administradores de la página.

Este es el único cambio de diseño llevado a cabo para esta iteración. Para lo demás, se mantiene el análisis y diseño inicial. En esta iteración se llevará a cabo la implementación de estos cambios y el menú administrador al completo.

4.2.2 Implementación

En primer lugar, se implementaron en el código frontend los cambios plasmados en el apartado anterior, de la cabecera y el menú.

A continuación, se realizó la implementación del menú administrador completo: un menú en el que se gestionaran usuarios y preguntas. Este menú debe permitir añadir, editar y eliminar usuarios y preguntas.

Se realizó el diseño según lo plasmado en el [apartado de diseño de interfaces](#), realizando conexiones con la base de datos de Firebase para realizar los cambios tal y como se muestra en el diagrama de secuencia del menú administrador (añadir usuario).

También se añadió un mecanismo de seguridad para comprobar que todo el usuario que entre en la sección de menú administrador, efectivamente esté registrado como administrador en la base de datos del sistema. En caso contrario, se muestra un mensaje de error, impidiendo acceder a las funcionalidades del mismo.

4.2.3 Pruebas y validación

Las pruebas de las funcionalidades implementadas dieron un resultado positivo, adaptándose a lo especificado en los casos de uso.

Sin embargo, en la validación, tanto con el tutor como con la empresa contratante, se decidieron diseñar cambios para el menú administrador.

En primer lugar, se llegó a la conclusión junto con la empresa contratante de que incluir las preguntas una a una iba a resultar un proceso muy largo y tedioso, pues el banco total de preguntas tenía un gran volumen. Por tanto, es necesario crear una funcionalidad que permita **importar preguntas de forma masiva desde Excel**.

En segundo lugar, el diseño inicial del menú administrador se consideró poco usable, y se optó por la alternativa de separarlo en dos secciones completamente diferentes: menú administrador de usuarios y menú administrador de preguntas.

En la planificación de la próxima iteración se trabajarán estos cambios, con su correspondiente análisis y diseño.

4.3 Tercera iteración

4.3.1 Planificación

La planificación de esta iteración consiste en el análisis y diseño de los cambios propuestos en la etapa de validación de la iteración anterior: incluir el requisito de importar preguntas desde Excel y rediseñar el menú administrador.

Para importar preguntas desde Excel, se adaptaría la plataforma al formato de tablas de preguntas que la empresa contratante ya había creado, de forma que con los ficheros Excel ya existentes las preguntas se puedan importar directamente.

El resumen de la especificación es:

- **Importar preguntas desde Excel:** un administrador debe poder importar preguntas de un determinado tema desde una hoja de cálculo con el formato definido previamente por la empresa.

El nuevo caso de uso “Importar preguntas desde Excel” queda así:

Caso de uso		Importar preguntas desde Excel
Actor		Administrador
Descripción		El administrador quiere importar preguntas de forma masiva desde una hoja de cálculo
Precondición		Que el usuario esté registrado y tenga rol de administrador
Escenario principal		1. El administrador accede a la sección de preguntas del menú de administrador, seleccionando el tema del que quiera agregar las preguntas 2. El sistema muestra el listado de preguntas de ese tema

	3. El administrador pulsa en Importar preguntas desde Excel 4. El sistema muestra un formulario para agregar el archivo 5. El administrador sube el archivo y pulsa en Aceptar 6. El sistema valida el contenido de la hoja de cálculo 7. El sistema agrega las preguntas de la hoja de cálculo al sistema 8. El sistema muestra el listado de preguntas de ese tema, incluyendo las preguntas que se acaban de añadir
Escenario alternativo	5. El administrador pulsa en Cancelar 5.1 El sistema muestra el listado de preguntas de ese tema sin modificar el contenido 6. El formato de la hoja de cálculo o su contenido no es correcto 6.1 El sistema muestra el listado de preguntas de ese tema con un mensaje de error sin modificar el contenido

Tabla 20: Caso de uso Importar preguntas desde Excel

La historia de usuario correspondiente es:

Nombre	Rol	Funcionalidad	Resultado	Puntuación
Importar preguntas desde Excel	Como administrador	Quiero importar preguntas a la base de datos desde Excel	Para incluir preguntas en la base de datos de forma masiva	8

Tabla 21: Historia de usuario Importar preguntas desde Excel

Por último, pasando a la fase del diseño, se ha creado un diagrama de secuencia para modelizar las comunicaciones del sistema en el proceso:

En primer lugar, el usuario importa el archivo Excel a través de un formulario en la vista del menú administrador. Este archivo es procesado por el código JavaScript

del lado cliente, comprobando que el formato es correcto y convirtiendo el contenido del archivo en un array de preguntas. En caso de que el formato del archivo sea incorrecto se manda un mensaje de error al usuario. En caso contrario, se envía una petición POST a la API REST para que añada las preguntas a la base de datos. Pero, antes de enviar las preguntas, la API manda una consulta a la base de datos para comprobar si el usuario que quiere introducir las preguntas es administrador. Si no es administrador, se envía un mensaje de error. En caso contrario, se envían las preguntas a la base de datos para que sean añadidas.

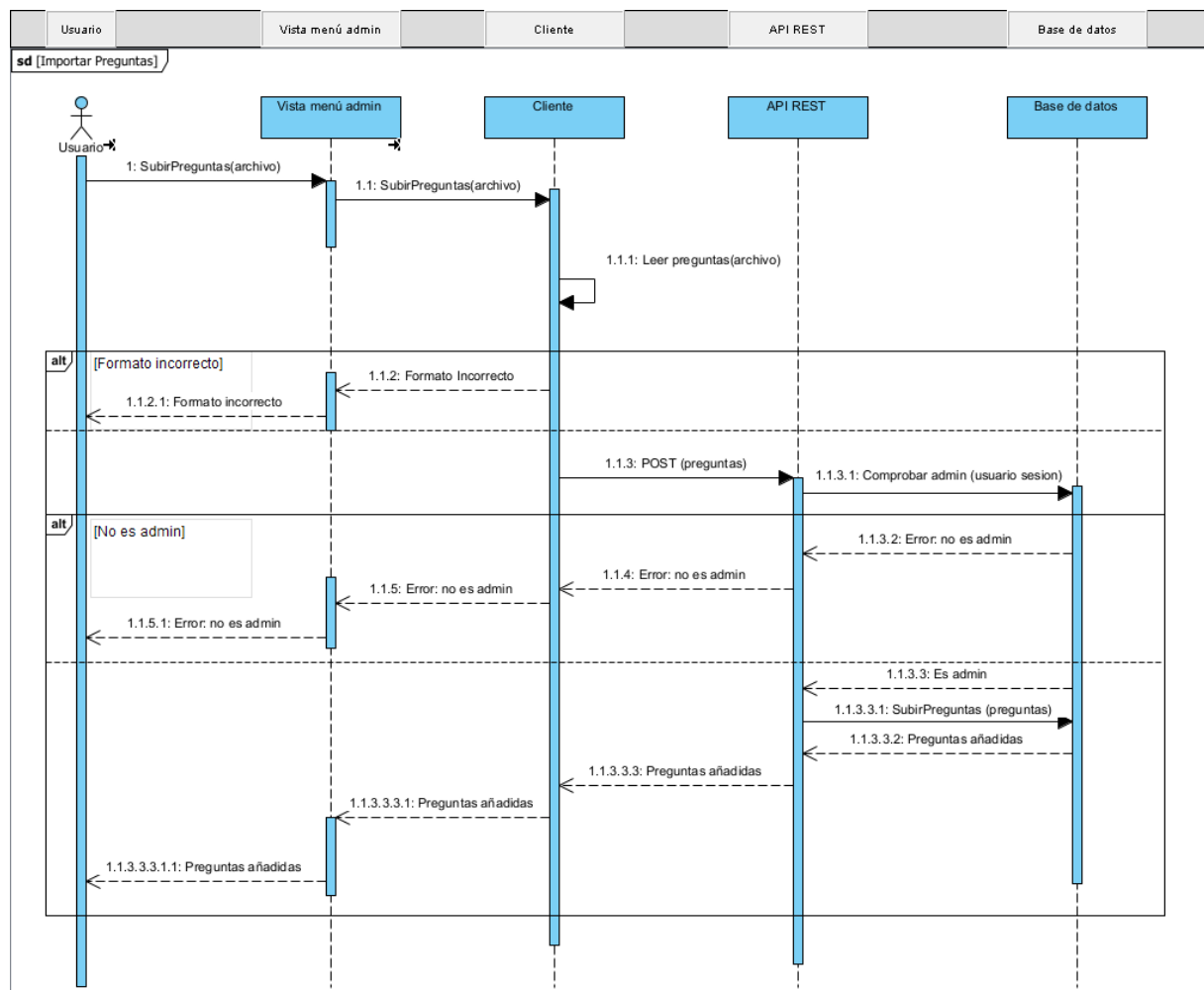
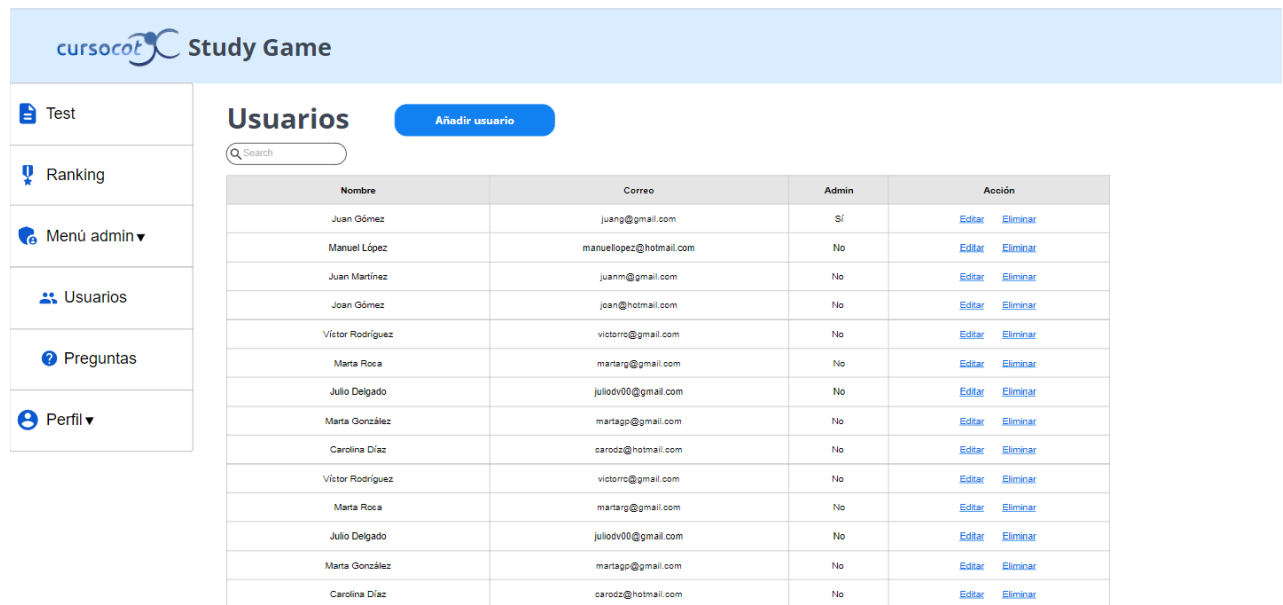


Ilustración 31: Diagrama de secuencia de importar preguntas desde Excel

Una vez detallado todo el análisis y diseño relativo a la inclusión de esta especificación en el sistema, pasamos al segundo objetivo de esta fase: rediseñar el menú administrador.

Como se ha detallado en la etapa de validación de la iteración anterior, para mejorar la usabilidad es necesario separar el menú administrador en dos secciones diferenciadas: usuarios y preguntas. Esto significa un cambio en el diseño de las interfaces, quedando así:



Usuarios [Añadir usuario](#)

Search

Nombre	Correo	Admin	Acción
Juan Gómez	juang@gmail.com	Sí	Editar Eliminar
Manuel López	manuellopez@hotmail.com	No	Editar Eliminar
Juan Martínez	juanm@gmail.com	No	Editar Eliminar
Joan Gómez	joan@hotmail.com	No	Editar Eliminar
Víctor Rodríguez	victorro@gmail.com	No	Editar Eliminar
Marta Roca	martarg@gmail.com	No	Editar Eliminar
Julio Delgado	juliodv00@gmail.com	No	Editar Eliminar
Marta González	martagp@gmail.com	No	Editar Eliminar
Carolina Díaz	carodz@hotmail.com	No	Editar Eliminar
Víctor Rodríguez	victorro@gmail.com	No	Editar Eliminar
Marta Roca	martarg@gmail.com	No	Editar Eliminar
Julio Delgado	juliodv00@gmail.com	No	Editar Eliminar
Marta González	martagp@gmail.com	No	Editar Eliminar
Carolina Díaz	carodz@hotmail.com	No	Editar Eliminar

Ilustración 32: Diseño de la sección usuarios tras los cambios de la segunda iteración

Study Game

Preguntas [Importar desde excel](#)

Elige el tema de consulta:

Trauma

Search

	Enunciado	Opción 1	Opción 2	Opción 3	Opción 4	Id	Solución	Comentario	Acción
▼	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	38.1	1	Lorem ipsum dolor sit...	Editar Eliminar
▼	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	38.2	4	Lorem ipsum dolor sit...	Editar Eliminar
▼	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	38.3	3	Lorem ipsum dolor sit...	Editar Eliminar
▼	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	38.4	3	Lorem ipsum dolor sit...	Editar Eliminar
▼	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	38.5	1	Lorem ipsum dolor sit...	Editar Eliminar
▼	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	38.6	4	Lorem ipsum dolor sit...	Editar Eliminar
▼	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	38.7	3	Lorem ipsum dolor sit...	Editar Eliminar
▼	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	38.8	3	Lorem ipsum dolor sit...	Editar Eliminar
▼	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	38.9	1	Lorem ipsum dolor sit...	Editar Eliminar
▼	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	38.10	4	Lorem ipsum dolor sit...	Editar Eliminar
▼	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	38.11	3	Lorem ipsum dolor sit...	Editar Eliminar
▼	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	Lorem ipsum dolor sit...	38.12	3	Lorem ipsum dolor sit...	Editar Eliminar

Ilustración 33: Diseño de la sección preguntas tras los cambios de la segunda iteración

Como vemos, separar el menú administrador en dos secciones hace que también se modifique el menú de navegación lateral.

En la próxima fase de implementación se harán efectivos estos cambios, además de implementar la sección de test y corrección en base al diseño inicial.

4.3.2 Implementación

En esta fase se llevó a cabo la implementación en código de los cambios de diseño del menú administrador especificados en el apartado anterior y el código JavaScript del [algoritmo de creación de test](#), basado en el diseño especificado inicialmente. También se implementó la interfaz que muestra el test al usuario y la corrección.

4.3.3 Pruebas y validación

La fase de pruebas de esta iteración fue superada satisfactoriamente, sin errores. Pese a que para la creación de un test son necesarias varias conexiones a Firebase, además de la ejecución del algoritmo para la creación del mismo, el tiempo

de carga fue muy bajo, incluso en test con un gran número de preguntas. Esto sirve como prueba de eficiencia del algoritmo de creación de test.

En cuanto a la validación, la empresa contratante dio unas cuantas consideraciones: según el diseño inicial, no se puede navegar a través de un test, cuando contestas una pregunta pasa directamente a la siguiente sin poder volver atrás, y además hay un tiempo por pregunta. Tras la realización de varios test de prueba en esta etapa, se consideró que al ser unos test que los usuarios realizan para estudiar, lo más recomendable es que no exista tiempo límite y que esté permitida y accesible la navegación a través de las preguntas del test. Por tanto, estos cambios serán diseñados e implementados en la cuarta iteración.

4.4 Cuarta iteración

4.4.1 Planificación

En esta fase es necesario diseñar los cambios ideados para la interfaz de realización de test, de la iteración anterior: es necesario permitir la navegación entre preguntas de forma accesible en todo momento, y eliminar el tiempo máximo.

Por lo tanto, se rediseñó la interfaz de realización de test con estos cambios, quedando el siguiente resultado de diseño:

The screenshot displays the 'Study Game' interface. On the left is a sidebar menu with options: 'Test', 'Ranking', 'Menú admin', and 'Perfil'. The main area contains a question: '¿Qué signo/sintoma aparece en la artrosis que la diferencia de la necrosis avascular de cadera?'. Below the question are five radio button options: 'Bilateralidad sintomática' (selected), 'Limitación a la movilidad', 'Cojera', 'Trendelenburg', and 'Dolor intermitente'. To the right of the question is a 'Navegar por el test' section with a grid of 10 numbered circles (1-10). Circles 1, 2, 3, and 4 are blue, while 5 through 10 are light blue. Below the grid is a 'Terminar test' link. At the bottom of the main area are two blue buttons: '<< Anterior' and '>> Siguiente'.

Ilustración 34: Diseño de la vista de test tras los cambios de la tercera iteración

En la navegación por el test podemos ver que se muestran de forma distinta las preguntas contestadas de las no contestadas aún, para que sea más fácil el manejo para el usuario.

4.4.2 Implementación

En esta fase se implementaron en el código frontend los cambios de diseño de la interfaz de realización de test, y también se realizó la implementación total del mecanismo de ranking, tanto el cálculo del mismo como mostrarlo al usuario.

Como en la base de datos se mantiene por cada usuario una colección de las preguntas que ha contestado bien y la fecha, para la implementación del ranking se obtiene esa información y se suman las preguntas correctas, siendo cada respuesta correcta un punto más para el usuario. La fecha en la que la pregunta fue correctamente contestada es útil para el ranking semanal, pues solo se contarán las preguntas de la última semana. Tras calcular el ranking este se muestra al usuario en forma de tablas, con el nombre de los cinco primeros clasificados y la puntuación, tal y como se muestra en el diseño.

4.4.3 Pruebas y validación

Las pruebas de las nuevas funcionalidades de esta iteración fueron superadas satisfactoriamente. La navegación por el test no presentó problemas y el ranking fue calculado correctamente y en tiempo real con usuarios de prueba. Se vio que el tiempo de carga para la sección de ranking era algo mayor que para el resto de secciones, pues tenía que realizar un mayor número de conexiones a la base de datos de Firebase. Sin embargo, el tiempo de carga se consideró tolerable al no ser demasiado elevado (entre 2 y 3 segundos).

Respecto a la validación, la empresa contratante consideró implementada de forma correcta la sección de ranking, por tanto, no hubo objeciones. No sería necesaria una quinta iteración, por tanto, el desarrollo de la plataforma se puede dar por finalizado.

5 CONCLUSIONES Y TRABAJOS FUTUROS

Para finalizar la memoria se van a presentar unas conclusiones sobre el trabajo realizado, analizando el nivel de cumplimiento de los objetivos iniciales y si ha sido realizado satisfactoriamente.

Se van a repasar los objetivos iniciales detallados en el [apartado 1.2](#) para analizar su cumplimiento y cambios. Los objetivos iniciales eran:

- *Estudio de un algoritmo que permita realizar test inteligentes de forma automática dada una batería de preguntas que se adapten a cada alumno y a su historial de preguntas respondidas, agilizando lo máximo posible la tarea de estudio activa mediante la realización de test.*

Este objetivo fue superado de forma satisfactoria, ya que se ha diseñado un algoritmo que teniendo la información necesaria puede crear test adaptados al usuario con un tiempo de carga bajo. También se ha estructurado la base de datos (entidades) de forma que el algoritmo pueda obtener esa información de forma eficiente.

- *Realización de una aplicación web completa que aplique este algoritmo y presente de forma intuitiva y clara los test a los alumnos y su progresión.*

Este objetivo ha sido cumplido a lo largo del desarrollo de todo el proyecto, tanto con el análisis y diseño iniciales como con su posterior implementación, fiel a este diseño, pero con cambios a lo largo de las iteraciones para adaptarse a los requerimientos de la empresa contratante.

- *Inclusión de un ranking basado en el historial de test de cada alumno que permita la competición entre ellos.*

Este objetivo fue cumplido en la cuarta iteración, basado en el diseño inicial, de forma que se muestra de forma clara en tablas los alumnos con mejor puntuación, además de la implementación de un algoritmo de cálculo de puntuación sencillo.

- *Inclusión de un menú para administradores que permita la gestión total de usuarios y preguntas.*

Este objetivo fue satisfactoriamente cumplido, tras unos cambios tanto en análisis y diseño entre la primera y segunda iteración, se dispone de un

menú administrador completamente funcional para gestionar usuarios y preguntas, además de la función nueva que permite importar preguntas de forma masiva desde un archivo Excel con el formato dado por la empresa.

Una vez revisados los objetivos, podemos decir que la implementación de la plataforma ha sido completada de forma satisfactoria con la empresa contratante, el tutor del trabajo y los objetivos fijados. Esto ha sido posible en parte gracias a la metodología ágil del desarrollo del mismo, que ha permitido que el proyecto esté en continua revisión y mejora, adaptándose a los requerimientos y mejorando en usabilidad.

Se espera que finalmente la plataforma pase a estar en funcionamiento en la empresa contratante, Cursocot, y pueda fomentar el aprendizaje activo de todos sus alumnos, tal y como se fijó en el principal objetivo de la plataforma por parte de la empresa.

En cuanto a futuras mejoras, se proponen dos puntos que podrían mejorar la plataforma:

- Inclusión de un modo multijugador, en el que dos usuarios puedan competir en tiempo real mediante la realización de test. Esto podría aumentar el enfoque competitivo de la plataforma.
- Mejora de la eficiencia del cálculo del ranking: la sección ranking es la que tiene un mayor tiempo de carga de la plataforma, aunque presenta un tiempo de carga aceptable según las pruebas. Sin embargo, en caso de que con el aumento de usuarios también aumente el tiempo de carga, se podría replantear la metodología del cálculo de ranking y, en vez de que sea en tiempo real como actualmente, hacer que se actualice cada cierto tiempo.

Como conclusión a nivel personal, la realización de este proyecto ha sido un gran aprendizaje para mí, ya que después de cinco meses de trabajo he aprendido mucho sobre el manejo de tecnologías que pueden ser muy importantes en mi futuro laboral, además de mejorar mis metodologías de desarrollo de software, algo que es imprescindible. Ha supuesto un gran crecimiento a nivel personal en la habilidad de aprender a resolver problemas individualmente.

Por tanto, este trabajo supone un gran culmen y el punto final a cuatro años de constante aprendizaje, con un esfuerzo y trabajo que me han permitido obtener unos conocimientos fundamentales en el campo de la informática. Ahora empieza una

nueva etapa, donde dejo la carrera atrás, pero seguiré en constante evolución y aprendizaje, aplicando todas las habilidades que he aprendido y desarrollando otras nuevas.

6 APÉNDICES

6.1 Instalación y configuración del sistema

Este apartado está destinado a explicar cómo replicar la plataforma en cualquier ordenador. Para ello, hay que seguir estos pasos:

1. Instalar NodeJS, que es un entorno de tiempo de ejecución de JavaScript que permite la ejecución de proyectos desarrollados con React. La instalación es muy sencilla, simplemente consiste en acceder a la página oficial de NodeJS (<https://nodejs.org/en/>) y pulsar el botón de descarga, lo cual descargará un archivo .msi. La versión más recomendable es la LTS.

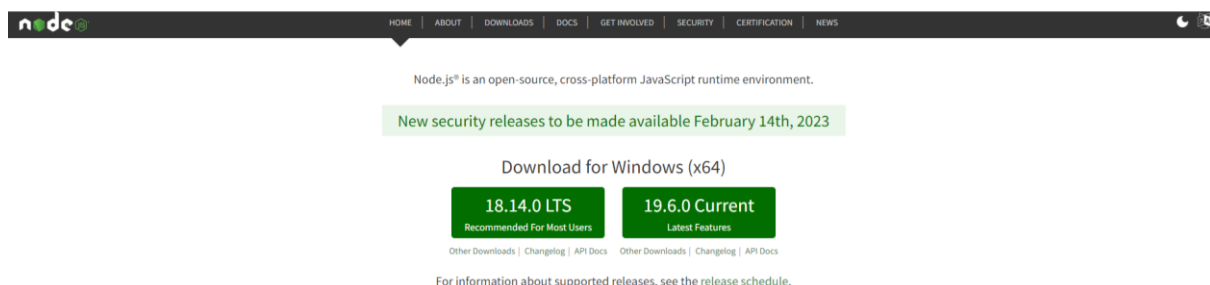
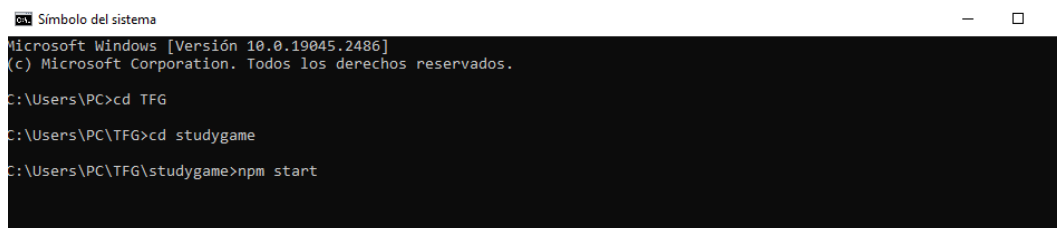


Ilustración 35: Página de instalación de NodeJS

Una vez descargado, se abre el archivo y se siguen unos breves pasos de instalación. Con esto, ya está NodeJS instalado en el sistema.

2. A continuación, hay que descomprimir el archivo .zip adjunto a la entrega del TFG, que contiene una carpeta llamada “studygame” con el código ejecutable del proyecto.

- Posteriormente, se abre la consola del ordenador (CMD) y se accede a la ubicación de la carpeta “studygame” donde se haya descomprimido. Una vez en la ruta de la carpeta, se ejecuta la orden “npm start”.

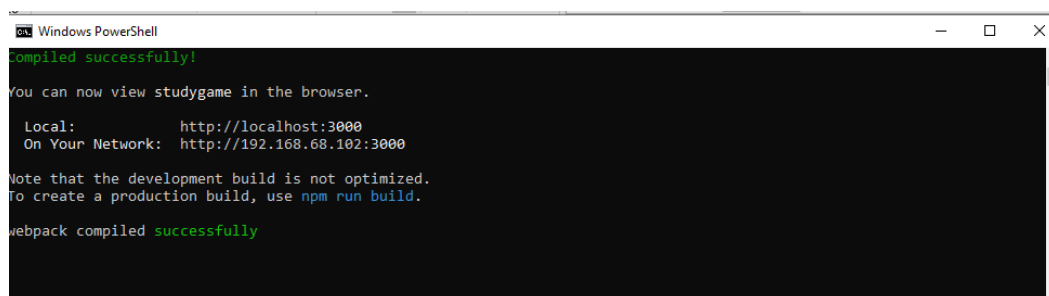


```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.2486]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\PC>cd TFG
C:\Users\PC\TFG>cd studygame
C:\Users\PC\TFG\studygame>npm start
```

Ilustración 36: Instrucciones a incluir en la consola

- Una vez ejecutada la orden, se pondrá en marcha la ejecución de la aplicación. Al ser la primera ejecución, la orden podría tardar en cargar alrededor de un minuto, dependiendo de la máquina en que se ejecute. Cuando se haya ejecutado, aparecerá un mensaje como el que se muestra en la consola y también se abrirá directamente una página en el navegador con la plataforma web.



```
Windows PowerShell
compiled successfully!
You can now view studygame in the browser.

Local:      http://localhost:3000
On Your Network:  http://192.168.68.102:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Ilustración 37: Mensaje en la consola al ejecutar

- La plataforma ya está lista para ser probada. La página que aparecerá al iniciar la plataforma será la de inicio de sesión, ya que sin estar registrado en el sistema no se puede acceder a las demás. Como el registro de usuarios lo debe realizar un administrador, se ha creado una cuenta de prueba para poder probar todas las funciones de la página. Las credenciales de esta cuenta son:

Correo electrónico: prueba@ejemplo.com

Contraseña: contraseña

Introduciendo esas credenciales, se puede acceder al contenido de la plataforma y hacer pruebas de su contenido al completo.

6.2 Manuales de usuario

Este apartado sirve como manual de explicación del funcionamiento de la página, para poder probar sus funcionalidades.

Al iniciar el servidor, se muestra la página de inicio de sesión. Se accede con una cuenta de prueba, con las credenciales explicadas en el apartado anterior:

Correo electrónico: prueba@ejemplo.com

Contraseña: contraseña

Una vez se ha iniciado sesión, ya se puede acceder a todas las secciones de la plataforma. La primera sección que nos aparece es la principal, la que muestra el progreso que el usuario registrado tiene en cada tema.



Ilustración 38: Panel principal

Las barras de progreso están vacías, pues el usuario de prueba aún no ha realizado ningún test. Para realizar un test, se pulsa cualquier tema y a continuación, sobre “Realizar test”. Esto mostrará una vista con un test sobre el tema, que se puede rellenar aleatoriamente, mostrando una corrección al final.

Study Game

Test

Ranking

Menú admin

Perfil

Señale la opción VERDADERA sobre el tratamiento de la estenosis de canal lumbar:

- ☐ No existe evidencia de que ningún tratamiento conservador altere la historia natural del proceso
- ☐ Los fármacos anticonvulsivantes son útiles
- ☒ La utilización de ortesis no contribuye a la mejoría clínica
- ☐ El tratamiento quirúrgico debe indicarse en todos los casos sintomáticos
- ☐ Los espaciadores interespinosos son una opción de tratamiento menos invasiva para pacientes jóvenes

Navegar por el test

1 2 3 4 5 6 7 8 9 10

Terminar test

Anterior Siguiente

Ilustración 39: Realización de test

Study Game

Test

Ranking

Menú admin

Perfil

Tu nota: 1/10

¿Cuál de los siguientes NO es un inhibidor de la fusión ósea vertebral?

- ☐ Ibuprofeno
- ☒ Etoricoxib
- ☐ Alendronato
- ☐ Tabaquismo
- ☐ Ciprofloxacino

Respuesta correcta: Etoricoxib

Explicación: Factores que inhiben la fusión ósea: (ver tema 12) ➤ Los AINEs reducen las tasas de fusión espinal, pero esto no se ha demostrado con los inhibidores selectivos de la ciclooxigenasa 2 (celecoxib, etoricoxib...), hasta la fecha. ➤ El uso de alendronato (bisfosfonatos en general) y calcitonina ha demostrado disminuir las tasas de fusión en animales. Deben suspenderse en el periodo postoperatorio. ➤ Tabaquismo (recomendar dejar de fumar). ➤ Otros: Irradiación, tumores, inestabilidad mecánica por fallo en la fijación, infección, corticoides, quimioterapia, diabetes, malnutrición proteica, enfermedad metabólica ósea, enfermedades infecciosas (HIV), quinolonas (Ciprofloxacino), fármacos antivitaminas K (sintrom).

Señale la opción FALSA sobre la estenosis de canal raquídeo lumbar:

- ☐ Suele presentarse sin listesis
- ☐ La espondilolistesis degenerativa no suele presentar estenosis de canal
- ☐ La estenosis central está típicamente causada por la combinación de láminas cortas con abultamiento discal e hipertrofia facetaria
- ☒ La estenosis de canal radicular se suele causar por un abultamiento posterolateral del disco e hipertrofia de la faceta superior
- ☐ La estenosis foraminal es el resultado de la pérdida de altura discal e hipertrofia facetaria

Respuesta correcta: La espondilolistesis degenerativa no suele presentar estenosis de canal

Explicación: • La estenosis de canal se presenta frecuentemente sin listesis (deslizamiento de una vértebra sobre otra) degenerativa. • La espondilolistesis degenerativa casi siempre presenta estenosis de canal. • Estenosis central: Típicamente causada por combinación de láminas cortas con abultamiento discal e hipertrofia de las facetas, del ligamento amarillo y de la cápsula articular de las facetas. Estos factores causan una disminución de la superficie transversal del canal espinal y compresión circunferencial del saco tecal asociado a trastornos vasculares microarteriales y venosas de las raíces nerviosas. Generalmente se cree que es la causa de la claudicación neurogénica. Basándose en el diámetro mediosagital del canal central cuyo umbral inferior es de 13 mm, se puede hablar de estenosis central relativa y absoluta: o Estenosis central relativa: Diámetro mediosagital de 10 a 13 mm. o Estenosis central absoluta: ídem < 10 mm. • Estenosis de canales radiculares: Produce compresión de la raíz que lo atraviesa desde el punto en que sale del saco dural hasta que entra en el agujero foraminal. Generalmente es causada por la compresión de la raíz entre un abultamiento posterolateral del disco y una hipertrofia de la faceta superior o interna. • Estenosis foraminal: Causada por disminución en la dimensión cráneo-caudal de dicho foramen. Resulta de una pérdida de altura discal y una hipertrofia facetaria. Material del disco a nivel foraminal también puede contribuir a la estenosis.

Ilustración 40: Corrección de test

Pulsando sobre “Ranking” en el menú de navegación se podrá observar el ranking semanal y total. El contenido de este va a ser solo de usuarios de prueba, pues aún la plataforma no está en uso por la empresa contratante.

Study Game			
Test	Ranking		
Ranking			
Menú admin			
Perfil			
		Histórico	
		Nombre	Puntuación
		Pedro	20
		Alberto	2
		Marta	0
		David	0
		Alberto	0
		María	0
		Semanal	
		Nombre	Puntuación
		Alberto	2
		Pedro	2
		Marta	0
		David	0
		Alberto	0
		María	0

Ilustración 41: Ranking

Pulsando sobre “Menú admin” tenemos acceso al menú de administrador, dividido entre usuarios y preguntas.

El menú administrador de usuarios dispone de un buscador para filtrar la búsqueda de un usuario concreto. Las acciones permitidas son añadir, editar y eliminar usuario, las tres con un formulario que se despliega automáticamente para añadir datos y confirmar la acción.

Study Game			
Test	Usuarios Nuevo usuario		
Ranking	Buscar...		
Menú admin			
Perfil			
		Nombre	Correo
		Admin	Acción
		Usuario de prueba	prueba@ejemplo.com
		Manuel	prueba7@ejemplo.com
		Juan	prueba2@ejemplo.com
		Pedro	perico@hotmail.com
		Enrique	prueba8@ejemplo.com
		Alberto	adc00026@red.ujaen.es
		Nuria	prueba10@ejemplo.com
		Rafael	prueba9@ejemplo.com
		Francisco	prueba4@ejemplo.com
		María	prueba6@ejemplo.com
		Alberto	albertodd00@gmail.com
		David	prueba5@ejemplo.com
		Marta	prueba3@ejemplo.com

Ilustración 42: Sección de usuarios

El menú administrador de preguntas dispone de un botón para importar preguntas desde un archivo Excel. El formato de las tablas tiene que ser el establecido por la empresa para que las preguntas se importen correctamente, especificado en el modal. Junto a la entrega se adjuntará un archivo Excel creado por la empresa con preguntas para poder adjuntarlo a la plataforma directamente. Además, también permite la consulta de las preguntas, eligiendo un tema se despliega una tabla con las preguntas de ese tema. Las acciones permitidas son editar y eliminar pregunta, ambas con un formulario que se despliega automáticamente para añadir datos y confirmar la acción.

Study Game										
Test	Preguntas									
Ranking	Importar desde Excel									
Menú admin	Categoría: Columna									
Perfil										
Ampliar	Id	Enunciado	Opción A	Opción B	Opción C	Opción D	Opción E	Respuesta	Comentario	Acción
	79.2	¿Cuál es la localización típica del dolor provocado por la inestabilidad L5-S1?	Parte posterior de la pierna	Parte lateral de la pierna	Cara lateral del muslo	Cara posterior del muslo	Cara anterior del muslo	5	» La inestabilidad L4-L5 (más frecuente) puede producir dolor inguinal que, en ocasiones, se irradia al trocánter y a la cara externa del muslo. Puede confundirse con dolor de cadera o trocanteritis. » La inestabilidad L5-S1 suele provocar dolor referido a la cara anterior del muslo.	 
	79.3	¿En qué consiste la ll...	Dolor lumbar que ...	Radiculopatía que ...	Maniobra de levan...	Sensación de inest...	Imagen radiológic...	1	• Clínica: » Lo más frec...	 
	79.6	¿Cuál de las siguiente...	El tratamiento cons...	La tasa de fusión c...	El patrón oro actu...	El tratamiento pref...	Todas las opción...	1	Tratamiento: No bien de...	 
	79.9	Señale la opción FALS...	Suele presentarse s...	La espondilolistesi...	La estenosis centra...	La estenosis de can...	La estenosis foram...	2	• La estenosis de canal s...	 
	79.14	Un paciente que pres...	No presenta esten...	I - Leve	II - Moderado	III - Severo	La RMN no sirve p...	2	Mediante RM las esteno...	 
	79.15	Una de las siguientes...	Radiografías dinám...	Radiografías simpl...	RM sin contraste	TC con mielografía	EMG	5	• Radiografías simples: o...	 
	79.16	Señale la opción VER...	No existe evidencia...	Los fármacos antic...	La utilización de or...	El tratamiento quir...	Los espaciadores l...	1	Tratamiento: La decisión...	 

Ilustración 43: Sección de preguntas

Por último, en el menú se encuentra la sección de Perfil, que permite editar el perfil del usuario registrado y cerrar sesión. La sección de editar perfil permite cambiar nombre y contraseña.



Ilustración 44: Editar perfil

7 DEFINICIONES Y ABREVIATURAS

- **Framework:** es una estructura de software predefinida que proporciona una serie de herramientas, librerías y/o convenciones para facilitar el desarrollo de aplicaciones o sistemas de software. Un framework típicamente incluye una arquitectura de software base que define la organización y los componentes principales de una aplicación, y proporciona una serie de funcionalidades comunes y reutilizables que los desarrolladores pueden utilizar para construir aplicaciones de manera más rápida y eficiente.
- **DOM:** *Document Object Model*, o Modelo de Objetos del Documento en español, es una interfaz de programación que proporciona una representación estructurada en forma de árbol de un documento HTML o XML, permitiendo a los programas acceder y manipular el contenido, la estructura y el estilo de un documento web. Es una tecnología fundamental para la creación de aplicaciones web interactivas y dinámicas.
- **Frontend:** El frontend es la parte de una aplicación que interactúa directamente con los usuarios finales. El frontend es el componente visible de una aplicación y se compone de elementos como los diseños, los componentes de UI, los formularios, los botones, las animaciones, las transiciones, y otros elementos que permiten a los usuarios interactuar con la aplicación.

- **Backend:** El backend es la parte de una aplicación que se encarga del procesamiento y la gestión de los datos y la lógica empresarial que no se muestra directamente al usuario final. Se compone de componentes y servicios que se ejecutan en un servidor o en un entorno en la nube, y que permiten a la aplicación almacenar, procesar y recuperar datos, interactuar con otros sistemas y servicios, y proporcionar servicios y funcionalidades complejas.
- **SDK:** Software Development Kit, o Kit de Desarrollo de Software en español, es un conjunto de herramientas y recursos que los desarrolladores utilizan para construir aplicaciones para una plataforma o tecnología específica.
- **POST:** En el protocolo HTTP (Hypertext Transfer Protocol), POST es un método de solicitud (o petición) que se utiliza para enviar datos a un servidor web para su procesamiento. El método POST es ampliamente utilizado en aplicaciones web para enviar datos de formularios, realizar acciones que modifican los datos del servidor, enviar mensajes y realizar otras tareas que requieren la transferencia de datos desde el cliente al servidor.
- **API REST:** Una API REST (Representational State Transfer) es un tipo de arquitectura de software para sistemas web que utiliza el protocolo HTTP para acceder y manipular recursos de manera uniforme. La API REST define un conjunto de restricciones y principios de diseño que se basan en la representación de recursos, y proporciona una interfaz uniforme para acceder a estos recursos a través de operaciones HTTP estándar, como GET, POST, PUT, DELETE y otras.

8 BIBLIOGRAFÍA

[1] Moodle. (s.f). *Actividad de examen*.

[https://docs.moodle.org/all/es/Actividad de examen](https://docs.moodle.org/all/es/Actividad_de_examen)

[2] StackScale. (12 de abril de 2022). *Navegadores más usados: características, comparativa y estadísticas*. <https://www.stackscale.com/es/blog/top-navegadores-caracteristicas-comparativa-estadisticas/>

[3] Openinnova. (s.f). *Angular vs React vs Vue. ¿Cuál Elegir?*

<https://www.openinnova.es/angular-vs-react-vs-vue-espanol-cual-elegir/>

[4] Firebase. (s.f). *Firebase Products*. <https://firebase.google.com/products>

[5] HubSpot. (10 de marzo de 2022). *Qué es Trello, para qué sirve y cómo funciona*.

<https://blog.hubspot.es/marketing/que-es-trello>

[6] Microsoft. (27 de septiembre de 2022). *Características de Visual Studio*.

<https://learn.microsoft.com/es-es/visualstudio/ide/advanced-feature-overview?view=vs-2022>

[7] Carlos Mayanga. (19 de febrero de 2020). *¿Qué es SmartGIT? (requisito de empleo)*

https://www.carlosmayanga.com/2020/02/que-es-smartgit-requisito-de-empleo_19.html

[8] Firebase. (s.f). *Firebase pricing*. <https://firebase.google.com/pricing>