



UNIVERSIDAD DE MÁLAGA



Graduado en Ingeniería del Software

Desarrollo de Una Aplicación Web para el Apoyo Docente
del Aprendizaje del Inglés en Centros Educativos

Web Application Development for Teaching Support of
English Learning in Educational Centers

Realizado por
García Cabriada, Antonio

Tutorizado por
Pinto Alarcón, Mónica

Departamento
Lenguajes y Ciencias de la Computación
UNIVERSIDAD DE MÁLAGA

MÁLAGA, junio 2022

Resumen

La globalización está cambiando el mundo a un ritmo acelerado, lo que hace que aprender una segunda lengua sea cada vez más importante. Esto ha llevado a la aparición de muchas aplicaciones para aprender idiomas fuera de una academia oficial. El principal problema de este tipo de aplicaciones es que suelen priorizar el atraer a potenciales clientes antes que proporcionar un contenido realmente eficaz y eficiente.

Este proyecto implementa y documenta una aplicación web, llamada Anglika, que permite a sus usuarios aprender inglés usando las metodologías avaladas por los expertos en el ámbito. Anglika propone una forma de estudiar inglés, distinta a la tradicional, incorporando elementos de gamificación, repetición espaciada y adquisición del lenguaje a través de la inmersión. Al unificar todas estas técnicas en una sencilla aplicación web de fácil acceso, permite a sus usuarios mejorar sus habilidades de forma sencilla y eficiente.

Anglika busca facilitar el aprendizaje del inglés lo máximo posible, para ello utiliza tecnologías web que lo hacen fácilmente accesible como por ejemplo *React*, *Google Firebase* y *Heroku*.

Palabras clave: Anglika, Ingles, SRS, Gamificación, Adquisición del Lenguaje

Abstract

Globalization is changing the world at a rapid pace, which makes learning a second language increasingly important. This has led to the emergence of many apps for learning languages outside of an official language school. The main problem with such apps is that they tend to prioritize attracting potential customers rather than providing effective and efficient content.

This project implements and documents a web application, called Anglika, which allows its users to learn English using methodologies endorsed by experts in the field. Anglika proposes a different way of studying English, incorporating elements of gamification, spaced repetition, and language acquisition through immersion. By unifying all these techniques in a simple, easy-to-access web application, it allows its users to improve their skills in a simple and efficient way.

Anglika aims to make learning English as easy as possible by using web technologies that make it easily accessible such as React, Google Firebase and Heroku.

Keywords: Anglika, English, SRS, Gamification, Language Acquisition

Índice

Resumen	1
Abstract	3
Índice	5
Introducción	7
1.1 Motivación	7
1.2 Objetivos.....	8
1.3 Metodología	8
1.4 Estructura de la memoria	10
Tecnologías empleadas	11
2.1 Persistencia de datos.....	11
2.2 Interfaz web.....	12
2.3 Control de versiones.....	12
2.4 Entorno de desarrollo.....	13
2.5 Editor de texto	13
2.6 Despliegue de la aplicación	13
Análisis y requisitos	15
3.1 Análisis	15
3.2 Requisitos funcionales	19
3.3 Requisitos no funcionales.....	20
3.4 Casos de uso	21
Diseño.....	43
4.1 Persistencia.....	43

4.2	Interfaz y navegación.....	46
	Desarrollo	53
5.1	Funcionalidad básica del usuario.....	53
5.2	Añadir contenido	57
5.3	Logros	57
5.4	Estadísticas y profesores	58
	Conclusiones y trabajos futuros	61
6.1	Conclusiones	61
6.2	Trabajos futuros.....	62
	Referencias	65
	Manual de usuario.....	69
A.1	Crear una cuenta	69
A.2	Crear una lección.....	70
A.3	Repasar una lección	75
A.4	Cambiar los ajustes	78
A.5	Editar perfil.....	80
A.6	Añadir alumnos	81
A.7	Eliminar alumnos.....	82

1

Introducción

1.1 Motivación

La motivación de este proyecto proviene de la necesidad que tenemos hoy en día de aprender al menos una lengua extranjera y de lo difícil que es este proceso para el español promedio, lo cual podemos ver al consultar el EF English Proficiency Index [\[1\]](#) que se autodefine como “El ranking mundial más grande según su dominio del inglés” que pone a España en el puesto 33 por debajo de nuestros vecinos europeos como Francia, Portugal o Grecia.

Anglika pretende facilitar el aprendizaje del inglés utilizando metodologías modernas que pese a ser las recomendadas por los expertos no son aplicadas por los profesores ni las escuelas ya que los artículos científicos que existen en este ámbito tienden a ser muy largos y estar llenos de vocabulario técnico [\[2\]](#).

Pretendemos llevar estas técnicas, respaldadas por muchas pruebas y experimentos, a la práctica de una forma que sea amigable tanto para los profesores como los alumnos.

1.2 Objetivos

El principal objetivo de Anglika es impulsar el aprendizaje del inglés en nuestro país. Para conseguir esto necesitamos encargarnos de problemas como encontrar una forma de que los usuarios disfruten del proceso, y que a la vez aprendan de forma eficiente y obtengan resultados reales. Para ello utilizamos estrategias de aprendizaje basadas en la ciencia, en concreto en las teorías de Stephen D Krashen [3] y en los sistemas de repetición espaciada [4], acompañadas de elementos de gamificación [5] que hacen más entretenida la experiencia.

Otro de nuestros objetivos es ayudar a nuestros usuarios a crear hábitos de aprendizaje. Esto es muy importante cuando se trata de aprender idiomas, ya que la constancia es muy importante en este ámbito. Para conseguirlo, Anglika utiliza distintas herramientas de gamificación que incentivan al usuario a estudiar a diario, como un sistema de logros que recompensa al usuario por realizar diversas actividades dentro de la aplicación.

El último de nuestros objetivos es permitir a los profesores crear contenido didáctico de forma fácil, además de darles el poder para monitorizar el progreso de sus alumnos de forma sencilla y centralizada.

1.3 Metodología

La metodología que se ha seguido ha sido una metodología Ágil dividida en 6 iteraciones, la primera estando dedicada únicamente a la identificación de requisitos, la última a realizar una comprobación exhaustiva del correcto funcionamiento de la aplicación y las otras 4 al diseño y desarrollo de cada una de las funcionalidades de la aplicación. Tras cada iteración se ha realizado un conjunto de pruebas manuales sobre la aplicación, en concreto se exploran todos y cada uno de los casos de uso implementados hasta el momento, incluyendo tanto los casos de éxito como los casos alternativos.

Las iteraciones que hemos utilizado han sido las siguientes:

1. Captura de requisitos
2. Implementación de requisitos básicos del usuario
3. Implementación de requisitos para añadir contenido
4. Implementación de requisitos del sistema de logros
5. Implementación de requisitos del sistema de progreso y los profesores
6. Pruebas de funcionamiento global del sistema

Como ya hemos dicho, en la primera iteración se buscó identificar todos los requisitos necesarios para completar la aplicación. Es a partir de esta lista de requisitos que se decidió establecer las siguientes cuatro iteraciones de implementación.

Es importante mencionar que el proceso ha sido flexible y si en algún punto se estimaba que un requisito está mal especificado, es incorrecto o simplemente no es necesario, se ha modificado o incluso eliminado dicho requisito.

Durante el desarrollo de la aplicación se han tenido que realizar múltiples modificaciones a los requisitos, principalmente requisitos que no se habían especificado inicialmente han tenido que ser añadidos, ya que al implementar la aplicación y tener una visión más completa y general se han encontrado varios puntos cuya funcionalidad no estaba completamente especificada.

El proceso de implementación de una funcionalidad empieza con la búsqueda de funcionalidades similares en otras aplicaciones del mismo ámbito, para comparar las distintas formas en las que se puede diseñar una funcionalidad y buscar la mejor forma de aplicar los métodos recomendados por los académicos. Una vez que se ha terminado el proceso de diseño de la funcionalidad se pasa a identificar que campos es necesario modificar o añadir a la base de datos y la mejor forma de organizarlo. Finalmente, una vez que toda la preparación ha sido completada, pasamos a la implementación según el diseño definido. Tras la implementación de la funcionalidad, como ya hemos dicho, se realizan un conjunto de pruebas basadas en los casos de uso identificados, con el objetivo de confirmar el correcto funcionamiento de lo implementado.

Durante la última iteración se realizan un conjunto de pruebas similar al que se realiza tras cada iteración de implementación, pero esta vez se cubren todos los casos de

uso, verificando así que no existen problemas de integración entre las distintas partes de la aplicación y todo funciona correctamente.

1.4 Estructura de la memoria

La memoria se ha dividido en las siguientes secciones:

1. **Introducción.** En esta sección hemos hablado de la motivación del proyecto, sus objetivos, la metodología que se ha seguido para alcanzar dichos objetivos y finalmente la estructura de la memoria, que es el apartado que estás leyendo.
2. **Tecnologías empleadas.** En esta sección se presentarán las tecnologías utilizadas y se compararán con otras alternativas populares.
3. **Análisis.** En esta sección discutiremos los requisitos que se han identificado para la aplicación.
4. **Diseño.** En esta sección se presentará el diseño que hemos seguido, tanto en cuanto a la interfaz de usuario como en la base de datos.
5. **Desarrollo.** En esta sección se comentará el proceso de desarrollo de cada funcionalidad y las decisiones que se han tomado.
6. **Conclusiones y trabajos futuros.** En esta sección veremos las conclusiones del proyecto y comentaremos algunas propuestas para mejorar la aplicación.

2

Tecnologías empleadas

2.1 Persistencia de datos

La persistencia de datos en la aplicación se ha manejado utilizando *Google Firebase* [\[6\]](#) para la información de los usuarios y *Imgur* [\[7\]](#) para el almacenamiento de imágenes.

Hemos decidido utilizar *Firebase* en vez de otras alternativas como *MongoDB* [\[8\]](#) ya que además de darnos acceso a una base de datos no relacional, también nos proporciona herramientas para lidiar con la autenticación de nuestros usuarios de forma segura y sencilla.

La elección de una base de datos NoSQL proviene principalmente de la flexibilidad que esto nos proporciona, ya que no es necesario actualizar el esquema de la base de datos cada vez que se realicen cambios o se añadan nuevos campos. Esto, unido al hecho de que las bases de datos basadas en documentos se integran mejor con JavaScript [9], hace que *Firebase* sea la opción evidente en cuanto a persistencia de datos se trata.

El almacenamiento de imágenes se realiza usando *Imgur*. A pesar de que *Firebase* proporciona también herramientas para almacenar imágenes, estas no son tan cómodas como las que ofrece *Imgur*.

2.2 Interfaz web

Para el desarrollo de la interfaz hemos utilizado el framework *React* [10] junto a la librería de componentes *Mantine* [11].

Existen tres grandes frameworks que dominan el mercado actualmente, *Angular* [12], *React* y *Vue* [13]. Entre ellos hemos elegido *React* por su modelo basado en componentes reutilizables y por su popularidad que se traduce en una vasta colección de librerías y documentación de gran calidad.

La creación del proyecto se ha realizado utilizando la herramienta *Vite* [14] que comparada con otras como *create-react-app* [15] proporciona una mayor velocidad de iteración.

Hemos utilizado la librería *Mantine* que nos proporciona una gran cantidad de componentes compatibles con *React* que nos permiten crear interfaces modernas y atractivas de forma sencilla y rápida. Existen otras librerías similares como *MUI* [16] que cumplen un propósito similar, pero nos hemos decantado por *Mantine* debido a su excelente documentación y a su extensa colección de componentes que nos han sido de gran utilidad a la hora de desarrollar la aplicación como: *RichTextEditor* [17], *RingProgress* [18], ...

2.3 Control de versiones

La herramienta de control de versiones que hemos escogido es *Git* [19], más concretamente *GitHub* [20].

Para interactuar con el repositorio y realizar los “*commits*” hemos usado *GitHub Desktop* que proporciona una interfaz simple y poderosa que se integra de forma excelente con *GitHub*.

Existen interfaces de usuario alternativas para interactuar con *Git* como *SourceTree* [21] o *GitKraken* [22], pero estas alternativas suelen ser o muy complejas o requieren una licencia para ser utilizadas. La sencillez de *GitHub Desktop* la hacen la opción más apropiada.

2.4 Entorno de desarrollo

El código ha sido creado usando exclusivamente *Visual Studio Code* [23]. Esta herramienta creada por Microsoft como alternativa a los *IDEs* [24] tradicionales es perfecta para proyectos como este.

Existen otras alternativas en el mercado como *Atom* [25] o *Sublime Text* [26] que tienen funcionalidades similares, pero la cantidad de extensiones y funcionalidades que provee *VS Code* lo hacen destacar por encima de sus competidores.

Algunas de las extensiones que hemos usado son: *Error Lens* [27], que permite ver claramente los errores de código directamente mientras escribes y *Todo Tree* [28] que nos permite añadir anotaciones en el código y luego visualizarlas de forma global, lo cual es muy útil para marcar secciones del código que están por terminar o contienen errores conocidos.

2.5 Editor de texto

La memoria ha sido escrita en su totalidad usando Microsoft Word [29], basándose en la plantilla ofrecida por la universidad de Málaga.

Hemos decidido utilizar Word debido a su simplicidad y la comodidad de tener la aplicación preinstalada al usar Windows como sistema operativo.

2.6 Despliegue de la aplicación

Para el despliegue de la aplicación existen muchas alternativas, como por ejemplo usar el propio sistema de “hosting” de *Firebase* o crear nuestro propio servidor.

Al final nos decantamos por usar *Heroku* [30], una plataforma que nos permite desplegar nuestra aplicación de forma gratuita y escalable en todo el mundo.

Cabe destacar que *Heroku* tiene la posibilidad de conectarse directamente con *GitHub*, facilitando enormemente el proceso de despliegue después de realizar cambios en la aplicación, ya que se hace automáticamente al subir los cambios a *GitHub*.

también hemos hecho uso de su habilidad para elegir "*buildpacks*" que son plantillas que facilitan el despliegue de aplicaciones con estructuras similares. En nuestro caso hemos utilizado una plantilla orientada a paginas estáticas, que es lo que *Vite* genera al compilar la aplicación.

3

Análisis y requisitos

3.1 Análisis

Nuestro objetivo al desarrollar Anglika no es simplemente crear otra aplicación para aprender inglés, de éstas ya hay muchos ejemplos en el mercado. Lo que nosotros buscamos es traer las teorías de vanguardia en el mundo académico y aplicarlas. Para ellos hemos tenido que investigar sobre estas teorías y en este apartado vamos a presentar cuales son las cosas que hemos aprendido y como las aplicamos en Anglika.

El eje central de nuestra aplicación se basa en las teorías del célebre Stephen D Krashen, un lingüista americano que fue el pionero que impulsó las teorías basadas en "input" y la distinción entre adquisición y aprendizaje [31], que hoy en día son la base en la que se sustentan la mayoría de las teorías actuales.

Según Krashen, el aprendizaje de una segunda lengua funciona igual que el aprendizaje de la primera lengua cuando se es un bebe [32]. Krashen defiende que aprender un lenguaje según los métodos tradicionales como memorizar vocabulario y gramática simplemente no es suficiente, lo que necesitamos es adquirir el lenguaje mediante el consumo de contenido comprensible. El contenido comprensible es el que contiene la

cantidad justa de contenido nuevo que se puede deducir su significado por el contexto, esto es comúnmente conocido como la ley del "i+1" [33].

En los últimos años la Hipótesis de la Lectura está ganando mucha importancia, en concreto la lectura auto seleccionada. En "Guided SSR before Self-Selected Reading" [34] se discute como un sistema que permita al usuario leer cómodamente textos que el mismo seleccione en cualquier lugar y momento del día, junto a un conjunto de lecturas predefinidas que ayuden a hacer la experiencia inicial lo más agradable y sencilla posible, es la forma ideal de aprender una lengua extranjera. Esta filosofía está muy presente a lo largo de toda la aplicación, donde permitimos al usuario crear sus propias lecciones de forma muy sencilla, además de que implementamos un sistema de profesores que pueden proporcionar la parte de guía que comentan en el artículo.

Puede parecer que un enfoque tan centrado en la lectura no sea óptimo a la hora de escribir y hablar, pero como Krashen comenta en su artículo "What about writing?" [35], esto no es necesariamente cierto. Krashen argumenta que escribir no nos hace escribir mejor, sino que es al leer cuando aprendemos como escribir de forma más correcta.

La otra base en la que se fundamenta Anglika es el uso de un sistema de repetición espaciada. Estos sistemas utilizan nuestros conocimientos acerca de la memoria humana para presentarnos el contenido que queremos memorizar justo cuando lo vamos a olvidar, ayudando a reforzar su presencia en nuestra memoria, a la vez que se reduce la cantidad de tiempo dedicada a repasar contenido que ya sabemos.

El sistema de repetición espaciada de Anglika permite a los usuarios repasar de forma eficiente las tarjetas asociadas a sus lecciones. A la hora de repasar el usuario tiene un tiempo límite para recordar el contenido de la tarjeta, y el tiempo que tarde se utiliza para reajustar el intervalo de la tarjeta. La motivación al ahora de utilizar un temporizador es doble, primero actúa como un aspecto de gamificación que es muy común en juegos en la forma de un modo contrarreloj, y segundo, reduce el tiempo potencial que se puede pasar en los repasos, haciendo la experiencia más amena.

Estos sistemas son muy populares a la hora de aprender vocabulario ya que es una tarea que se ajusta muy bien a los objetivos de esta técnica. Cabe destacar que los sistemas

de repetición espaciada requieren que el usuario realice repasos a diario, aunque estos repasos tienden a ser cortos, lo cual ayuda a crear hábitos de aprendizaje.

Finalmente tenemos que mencionar la gamificación que, pese a no ayudar directamente a aprender una lengua extranjera, es muy importante a la hora de motivar al alumno durante el proceso de aprendizaje. De hecho, pensamos que esta es una de las partes más importantes de la aplicación, puesto que no sirve de nada utilizar las técnicas más eficientes de aprendizaje si al final no se le dedica el tiempo suficiente, o peor aún, se abandona el proceso.

Durante el proceso de análisis y diseño hemos referenciado constantemente otras aplicaciones que existen en el mercado, ya que pensamos que nos podemos beneficiar enormemente de su experiencia en este ámbito. Existe una infinidad de aplicaciones orientadas a el aprendizaje de una lengua extranjera, en concreto Inglés, pero nos hemos inspirado principalmente en Duolingo y Anki ya que son muy populares y tenemos experiencia personal usándolas.

Duolingo es probablemente la aplicación más famosa para aprender inglés. Su diseño agradable, su enfoque en clases cortas llenas de elementos de gamificación la hace muy atractiva para los principiantes. También emplea un sistema de repetición espaciada, aunque distinto de los tradicionales y mucho más laxo, priorizando los aspectos de gamificación frente a la eficiencia. Las últimas versiones además incluyen las "stories" que son pequeñas historias a modo de conversaciones entre personajes e incluyen ejercicios para involucrar al estudiante.

Anki, pese a no ser tan universalmente reconocida, es extremadamente popular en la comunidad de estudiantes autodidactas. Al contrario que Duolingo que está completamente orientado al estudio de idiomas, Anki es una aplicación que únicamente te proporciona un sistema de repetición espaciada que puedes utilizar con el objetivo que tu quieras, ya sea memorizar las capitales de mundo, o las 2000 palabras más usadas en inglés. Este poderoso y personalizable sistema de SRS, combinado con su comunidad activa de usuario que crea y comparte mazos con tarjetas que puedes utilizar para aprender todo tipo de cosas, hacen que Anki sea la opción ideal a la hora de memorizar grandes cantidades de información.

Anglika busca combinar lo mejor de ambas aplicaciones y además incluir nuevas funcionalidades que nos permitan aplicar las teorías modernas sobre el aprendizaje de idiomas. De Duolingo hemos decidido adoptar los principales elementos de gamificación, a la vez que eliminamos las limitaciones que impone a los usuarios, al ser un producto comercial que busca beneficios económicos. También hemos mantenido la idea de las "stories" pero las hemos ligado a lecciones que contienen su propio vocabulario y notas, similares a los niveles de Duolingo. Por otro lado hemos buscado traer a Anglika un nivel de personalización similar al de Anki que te permite crear tu propio contenido de aprendizaje y configurar los ajustes del sistema de repetición espaciada para que se adapte mejor a tus necesidades.

	Anglika	Duolingo	Anki
SRS	✓	✓*	✓
Funcionalidad de pago	✗	✓	✗
Puedes añadir contenido nuevo	✓	✗	✓
Configurar SRS	✓	✗	✓
Responder preguntas	✓	✓	✗
Añadir explicaciones a una lección	✓	✓	✗
Ver estadísticas de aprendizaje	✓	✗	✓
Sistema de logros	✓	✓	✗
Lecturas, notas y repasos con contenido multimedia	✓	✓**	✓***
Profesores que aportan lecturas de guía	✓	✗	✗

Los profesores pueden ver las estadísticas de sus alumnos	✓	×	×
Lecturas con vocabulario y notas asociadas	✓	×	×

* Es sistema de repasos de Duolingo aunque técnicamente es un SRS es bastante más laxo, reduciendo la cantidad de repasos que el usuario tiene que hacer diariamente para evitar ser abrumador, pero sacrificando así gran parte de los beneficios de este sistema.

** El contenido multimedia se limita a imágenes ocasionales y pistas de audio.

*** Solo se puede incluir contenido multimedia en los repasos de tarjetas, ya que no existe el concepto de lecciones, ni notas explicativas.

Consideramos que Anglika cumple el objetivo de unificar los mejores aspectos de las aplicaciones que utilizan hoy en día las personas que quieren aprender inglés, y aplicar las metodologías recomendadas por los expertos en el campo. Pese a que la mayoría de las funcionalidades de Anglika están presentes en otras aplicaciones, la forma en la que las suelen implementar no sigue las pautas descritas en los estudios académicos, por ello hemos buscado ceñirnos a la literatura en búsqueda del sistema más eficiente para aprender inglés.

3.2 Requisitos funcionales

1. Requisitos del usuario

- 1.1. El usuario podrá crear una cuenta usando solo un correo y una contraseña.
- 1.2. El usuario podrá crear una cuenta como profesor
- 1.3. El usuario podrá iniciar sesión en la aplicación.
- 1.4. El usuario podrá elegir entre el tema claro o el tema oscuro en los ajustes.
- 1.5. El usuario podrá personalizar la dificultad en los ajustes.
- 1.6. El usuario podrá interactuar con sus temas de las siguientes formas:
 - 1.6.1. Estudiar tarjetas nuevas.
 - 1.6.2. Acceder a las notas.

- 1.6.3. Acceder a la lectura.
- 1.6.4. Realizar una prueba de conocimiento.
- 1.7. El usuario podrá repasar todas sus tarjetas de conocimiento pendientes.
- 1.8. El usuario podrá ver sus estadísticas generales.
- 1.9. El usuario podrá ver sus estadísticas en un tema concreto.
- 2. Requisitos de incorporación de contenidos**
 - 2.1. El usuario podrá crear una nueva una lectura, que puede contener texto, imágenes y videos.
 - 2.2. El usuario podrá añadir tarjetas de repaso asociadas a un tema.
 - 2.3. El usuario podrá añadir notas explicativas, con contenido multimedia, a un tema.
 - 2.4. El usuario podrá añadir preguntas de selección múltiple a un tema.
- 3. Requisitos del sistema de logros**
 - 3.1. El usuario podrá consultar sus logros en la pestaña de logros.
 - 3.2. El sistema de logros lanzará notificaciones cuando se alcance una meta.
 - 3.3. El usuario podrá desbloquear "títulos" al obtener logros.
- 4. Requisitos de monitorización del progreso**
 - 4.1. Si el usuario tiene el rol de profesor podrá añadir alumnos.
 - 4.2. El profesor podrá eliminar alumnos.
 - 4.3. El profesor podrá ver las estadísticas de sus alumnos en un tema.
 - 4.4. El profesor podrá ver las estadísticas generales de un alumno.

3.3 Requisitos no funcionales

- 1. La contraseña del usuario debe contener al menos una letra mayúscula, una minúscula, un número y tener más de seis caracteres.
- 2. Se mantendrá iniciada la sesión aun cuando se cierre el navegador.
- 3. Se utilizará un método seguro para autenticar al usuario.
- 4. Los mazos creados por un profesor se sincronizarán con sus alumnos.
- 5. Los alumnos solo pueden editar sus propios mazos, no los de sus profesores.

6. Todas las ventanas excepto Inicio de sesión y Registro redirigirán a los usuarios que no hayan iniciado sesión a la pestaña de Inicio de sesión para ser redirigidos de vuelta una vez que el usuario haya iniciado sesión.

7. El sitio web estará operativo el 99% del tiempo.

8. El sitio web será accesible desde cualquier parte del mundo.

9. Cualquier acción del usuario recibirá una respuesta en un periodo inferior a 1 minuto.

10. Se incluirá un manual de usuario.

3.4 Casos de uso

1. Crear una cuenta
2. Iniciar sesión
3. Editar una cuenta
4. Eliminar una cuenta
5. Cerrar sesión
6. Crear un tema
7. Editar un tema
8. Eliminar un tema
9. Ver notas
10. Aprender nuevas tarjetas
11. Ver estadísticas de un tema
12. Leer un tema
13. Responder a las preguntas de una lectura
14. Ver logros
15. Ver mis estadísticas
16. Editar ajustes
17. Añadir alumnos
18. Eliminar alumnos
19. Ver estadísticas de un alumno
20. Repasar tarjetas

Identificador	CU_01
Nombre	Crear una cuenta
Actores	Usuario
Precondiciones	El usuario se encuentra en la pantalla de inicio de sesión.
Postcondiciones	Se ha creado una cuenta y se ha iniciado sesión
Descripción	El usuario crea una nueva cuenta
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario pulsa el enlace "¿Aún no tienes cuenta?" 2. El sistema muestra el formulario de registro 3. El usuario rellena sus datos, acepta las políticas de privacidad y pulsa el botón de crear 4. El sistema verifica los datos, crea la cuenta y reenvía al usuario a la ventana adecuada 	
Escenarios alternativos	
<ul style="list-style-type: none"> • Correo invalido <ol style="list-style-type: none"> 3. El usuario introduce un correo invalido 4. El sistema informa al usuario de que el correo es invalido 5. Volvemos al paso 3 • Contraseña invalida <ol style="list-style-type: none"> 3. El usuario introduce una contraseña invalida 4. El sistema informa al usuario de que la contraseña no es valida 5. Volvemos al paso 3 • No acepta las políticas de privacidad <ol style="list-style-type: none"> 5. El usuario no acepta las políticas de privacidad 6. El sistema informa al usuario de que es obligatorio aceptar las políticas de privacidad 7. Volvemos al paso 3 • Error al crear la cuenta <ol style="list-style-type: none"> 5. El sistema encuentra algún error al crear la cuenta y muestra una notificación al usuario 6. Volvemos al paso 3 	

Identificador	CU_02
Nombre	Iniciar sesión
Actores	Usuario

Precondiciones	El usuario se encuentra en la pantalla de inicio de sesión.
Postcondiciones	Se ha iniciado sesión
Descripción	El usuario inicia sesión
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario rellena sus datos y pulsa el botón de iniciar sesión 2. El sistema verifica los datos y reenvía al usuario a la ventana adecuada 	
Escenarios alternativos	
<ul style="list-style-type: none"> • Datos erróneos <ol style="list-style-type: none"> 1. El usuario introduce datos erróneos 2. El sistema informa al usuario de que los datos son erróneos 3. Volvemos al paso 1 • Error al iniciar sesión <ol style="list-style-type: none"> 2. El sistema encuentra algún error al iniciar sesión y muestra una notificación al usuario 3. Volvemos al paso 3 	

Identificador	CU_03
Nombre	Editar una cuenta
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y está en el apartado de "Usuario"
Postcondiciones	Se ha modificado la información
Descripción	El usuario edita su información de perfil
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario pulsa el botón "Editar" 2. El sistema muestra un menú de edición 3. El usuario introduce su nuevo nombre y selecciona su nueva foto de perfil 4. El usuario pulsa "Guardar cambios" 5. El sistema actualiza los datos y cierra el menú de edición 	
Escenarios alternativos	
<ul style="list-style-type: none"> • La foto no es valida <ol style="list-style-type: none"> 3. El usuario una foto no valida (tamaño excesivo, formato no compatible, ...) 4. El sistema informa al usuario de que el archivo no es valido 5. Volvemos al paso 3 • Cancelar <ol style="list-style-type: none"> 4. El usuario pulsa "Cancelar", la "x" o fuera del menú de edición 5. El sistema cierra el menú de edición • Error al guardar los cambios <ol style="list-style-type: none"> 5. El sistema encuentra algún error al guardar los cambios y muestra una notificación al usuario 6. Volvemos al paso 3 	

Identificador	CU_04
Nombre	Eliminar una cuenta
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y está en el apartado de "Usuario"
Postcondiciones	Se ha eliminado la cuenta
Descripción	El usuario elimina su cuenta en la aplicación
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario pulsa el botón "Eliminar cuenta" 2. El sistema muestra un aviso de confirmación 3. El usuario pulsa "Eliminar" 4. El sistema elimina la cuenta y muestra la pantalla de inicio de sesión 	
Escenarios alternativos	
<ul style="list-style-type: none"> • Cancelar <ol style="list-style-type: none"> 3. El usuario pulsa "Cancelar" 4. El sistema cierra el aviso de confirmación • Error al guardar los cambios <ol style="list-style-type: none"> 5. El sistema encuentra algún error al eliminar la cuenta y muestra una notificación al usuario 6. Volvemos al paso 1 	

Identificador	CU_05
Nombre	Cerrar sesión
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y está en el apartado de "Usuario"
Postcondiciones	Se ha cerrado sesión
Descripción	El usuario cierra sesión
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario pulsa el botón "Cerrar sesión" 2. El sistema cierra la sesión del usuario y muestra la pantalla de inicio de sesión 	
Escenarios alternativos	
<ul style="list-style-type: none"> • Error al cerrar sesión <ol style="list-style-type: none"> 2. El sistema encuentra algún error al cerrar la sesión y muestra una notificación al usuario 3. Volvemos al paso 1 	

Identificador	CU_06
Nombre	Crear un tema
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y está en la pestaña de inicio
Postcondiciones	Se ha creado un tema
Descripción	El usuario crea un nuevo tema
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario pulsa el botón "Añadir contenido" 2. El sistema muestra un formulario de creación de contenido 3. El usuario rellena los campos necesarios y pulsa "Guardar" 4. El sistema guarda el contenido y muestra la pantalla de inicio 	
Escenarios alternativos	
<ul style="list-style-type: none"> • Datos incompletos <ol style="list-style-type: none"> 3. El usuario no rellena todos los campos necesarios 4. El sistema informa al usuario de que faltan campos por rellenar 5. Volvemos al paso 3 • Cancelar <ol style="list-style-type: none"> 3. El usuario pulsa "Cancelar" 4. El sistema pide una confirmación 5. El usuario confirma que quiere cancelar 6. El sistema muestra la pantalla de inicio • Error al crear <ol style="list-style-type: none"> 4. El sistema encuentra algún error al crear el contenido y muestra una notificación al usuario 5. Volvemos al paso 1 	

Identificador	CU_07
Nombre	Editar un tema
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y está en la pestaña de inicio
Postcondiciones	Se ha editado un tema
Descripción	El usuario edita uno de sus temas
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario selecciona uno de los temas que ha creado 2. El sistema muestra un menú con varias opciones 3. El usuario pulsa "Editar" 4. El sistema muestra la pestaña de edición de un tema 5. El usuario realiza los cambios que quiere y pulsa guardar 6. El sistema guarda los cambios y muestra la pantalla de inicio 	
Escenarios alternativos	
<ul style="list-style-type: none"> • No me pertenece <ol style="list-style-type: none"> 3. El usuario no puede pulsar "Editar" porque el tema no le pertenece • Datos incompletos <ol style="list-style-type: none"> 5. El usuario no rellena todos los campos necesarios 6. El sistema informa al usuario de que faltan campos por rellenar 7. Volvemos al paso 5 • Error al guardar <ol style="list-style-type: none"> 6. El sistema encuentra algún error al guardar el contenido y muestra una notificación al usuario 7. Volvemos al paso 5 	

Identificador	CU_08
Nombre	Eliminar un tema
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y está en la pestaña de inicio
Postcondiciones	Se ha eliminado un tema
Descripción	El usuario elimina uno de sus temas
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario selecciona uno de los temas que ha creado 2. El sistema muestra un menú con varias opciones 3. El usuario pulsa "Editar" 4. El sistema muestra la pestaña de edición de un tema 5. El usuario pulsa "Eliminar" 6. El sistema muestra un aviso de confirmación 7. El usuario pulsa "Eliminar" 8. El sistema elimina el nivel y muestra la pantalla de inicio 	
Escenarios alternativos	
<ul style="list-style-type: none"> • No me pertenece <ol style="list-style-type: none"> 3. El usuario no puede pulsar "Editar" porque el tema no le pertenece • Cancelar <ol style="list-style-type: none"> 6. El usuario pulsa Cancelar 7. El sistema cierra el aviso de confirmación 8. Volvemos al paso 5 • Error al eliminar <ol style="list-style-type: none"> 8. El sistema encuentra algún error al eliminar el contenido y muestra una notificación al usuario 9. Volvemos al paso 5 	

Identificador	CU_09
Nombre	Ver notas
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y está en la pestaña de inicio
Postcondiciones	El usuario puede ver las notas
Descripción	El usuario accede a las notas de un tema
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario selecciona uno de los temas que ha creado 2. El sistema muestra un menú con varias opciones 3. El usuario pulsa "Ver Notas" 4. El sistema muestra la pestaña de notas del nivel 	
Escenarios alternativos	

Identificador	CU_10
Nombre	Aprender tarjetas nuevas
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y está en la pestaña de inicio
Postcondiciones	Se han aprendido las tarjetas
Descripción	El usuario aprende tarjetas nuevas de un tema
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario selecciona uno de los temas que ha creado 2. El sistema muestra un menú con varias opciones 3. El usuario pulsa "Aprender nuevas" 4. El sistema muestra la pestaña de repasos de un tema con las tarjetas a aprender 5. El usuario realiza el repaso 6. El sistema informa de que se han aprendido las tarjetas y muestra la ventana de inicio 	
Escenarios alternativos	
<ul style="list-style-type: none"> • No quedan tarjetas <ol style="list-style-type: none"> 3. El usuario no puede pulsar "Aprender nuevas" porque no quedan tarjetas por aprender 	

Identificador	CU_11
Nombre	Ver estadísticas de un tema
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y está en la pestaña de inicio
Postcondiciones	Se muestran las estadísticas
Descripción	El usuario consulta sus estadísticas en uno de sus temas
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario selecciona uno de los temas que ha creado 2. El sistema muestra un menú con varias opciones 3. El usuario pulsa "Estadísticas" 4. El sistema muestra las estadísticas 	
Escenarios alternativos	

Identificador	CU_12
Nombre	Leer un tema
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y está en la pestaña de inicio
Postcondiciones	Se muestra la lectura de un tema
Descripción	El usuario lee el contenido de uno de sus temas
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario selecciona uno de los temas que ha creado 2. El sistema muestra un menú con varias opciones 3. El usuario pulsa "Leer" 4. El sistema muestra la lectura correspondiente 	
Escenarios alternativos	

Identificador	CU_13
Nombre	Responder a las preguntas de una lectura
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y está en la pestaña de lectura
Postcondiciones	Se han respondido a las preguntas
Descripción	El usuario realiza la prueba asociada a un tema
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario pulsa "Realizar prueba" 2. El sistema muestra las preguntas 3. El usuario responde a las preguntas y pulsa "Revisar" 4. El sistema comprueba las respuestas, muestra los errores y la puntuación conseguida 	
Escenarios alternativos	

Identificador	CU_14
Nombre	Ver logros
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y está en la pestaña de inicio
Postcondiciones	Se muestran los logros
Descripción	El usuario consulta su progreso en la ventana de logros
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario pulsa el icono de un trofeo en la parte izquierda de la pantalla 2. El sistema muestra los logros y su progreso actual 	
Escenarios alternativos	

Identificador	CU_15
Nombre	Ver mis estadísticas
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y está en la pestaña de inicio
Postcondiciones	Se muestran las estadísticas del usuario actual
Descripción	El usuario consulta sus estadísticas en la ventana de usuario
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario pulsa el icono de una persona en la parte izquierda de la pantalla 2. El sistema muestra su información de usuario y sus estadísticas 	
Escenarios alternativos	

Identificador	CU_16
Nombre	Editar ajustes
Actores	Usuario
Precondiciones	El usuario se encuentra en la pantalla de ajustes.
Postcondiciones	Se han guardado los cambios
Descripción	El usuario realiza modificaciones a sus ajustes
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario realiza los cambios y pulsa "Guardar cambios" 2. El sistema verifica los datos, los guarda y muestra una notificación diciendo que los cambios se han guardado correctamente 	
Escenarios alternativos	
<ul style="list-style-type: none"> • Datos inválidos <ol style="list-style-type: none"> 1. El usuario introduce datos inválidos 2. El sistema informa al usuario de que los datos son inválidos 3. Volvemos al paso 1 • Error al guardar cambios <ol style="list-style-type: none"> 2. El sistema encuentra algún error al guardar los cambios y muestra una notificación al usuario 3. Volvemos al paso 1 	

Identificador	CU_17
Nombre	Añadir alumnos
Actores	Profesor
Precondiciones	El profesor se encuentra en la pantalla de inicio.
Postcondiciones	Se han añadido alumnos
Descripción	El profesor añade usuarios a su lista de alumnos
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario pulsa "Añadir alumno" 2. El sistema muestra la lista de alumnos disponibles 3. El usuario busca a los usuarios que quiera, usando la barra de búsqueda para filtrar por nombre si fuese necesario, los selecciona y pulsa "Añadir" 4. El sistema añade a los usuarios seleccionados como alumnos del profesor y cierra la lista de usuarios 	
Escenarios alternativos	
<ul style="list-style-type: none"> • Cancelar <ol style="list-style-type: none"> 4. El usuario pulsa "Cancelar" 5. El sistema cierra la lista de usuarios 6. Volvemos al paso 1 • Error al añadir <ol style="list-style-type: none"> 4. El sistema encuentra algún error al añadir los alumnos y muestra una notificación al usuario 5. Volvemos al paso 3 	

Identificador	CU_18
Nombre	Eliminar alumno
Actores	Profesor
Precondiciones	El profesor se encuentra en la pantalla de inicio.
Postcondiciones	Se ha eliminado al alumno
Descripción	El profesor elimina uno de sus alumnos
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario pulsa el icono del birrete 2. El sistema muestra la lista de alumnos del profesor 3. El usuario selecciona al alumno en cuestión 4. El sistema muestra un menú con varias opciones 5. El usuario pulsa "Eliminar" 6. El sistema elimina al alumno seleccionado de la lista de alumnos y cierra el menú 	
Escenarios alternativos	
<ul style="list-style-type: none"> • Error al eliminar <ol style="list-style-type: none"> 6. El sistema encuentra algún error al eliminar al alumno y muestra una notificación al usuario 7. Volvemos al paso 3 	

Identificador	CU_19
Nombre	Ver estadísticas de un alumno
Actores	Profesor
Precondiciones	El profesor se encuentra en la pantalla de alumnos.
Postcondiciones	Se muestran las estadísticas de un alumno
Descripción	El profesor consulta las estadísticas de un alumno
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario selecciona al alumno en cuestión 2. El sistema muestra un menú con varias opciones 3. El usuario pulsa "Ver progreso" 4. El sistema muestra las estadísticas del alumno 	
Escenarios alternativos	

Identificador	CU_20
Nombre	Repasar tarjetas
Actores	Usuario
Precondiciones	El usuario se encuentra en la pantalla de inicio.
Postcondiciones	Se han repasado las tarjetas
Descripción	El usuario hace un repaso de sus tarjetas pendientes para el día
Escenario de éxito	
<ol style="list-style-type: none"> 1. El usuario pulsa "Repasar pendientes" 2. El sistema muestra la pestaña de repasos e inicia el temporizador 3. El usuario pulsa "Mostrar respuesta" antes de que el temporizador salte 4. El sistema muestra la otra cara de la tarjeta 5. El usuario autoevalúa si sabía o no la tarjeta 6. El sistema detecta que no hay más repasos pendientes y redirige al inicio 	
Escenarios alternativos	
<ul style="list-style-type: none"> • Tiempo excedido <ol style="list-style-type: none"> 3. El usuario no responde antes de que salte el temporizador 4. El sistema muestra la otra cara de la tarjeta 5. El usuario solo puede pulsar "Mal" 6. Volvemos al paso 2 • Quedan más tarjetas <ol style="list-style-type: none"> 6. El sistema detecta que quedan tarjetas 7. Volvemos al paso 3 	

4

Diseño

En este apartado vamos a discutir el diseño de las distintas partes de la aplicación, tanto de la persistencia en la base de datos como de la interfaz de usuario.

4.1 Persistencia

La persistencia se ha realizado utilizando una base de datos NoSQL, por lo que la estructura se define en términos de colecciones, documentos y subcolecciones.

Los documentos son donde se almacena la información en un formato de clave valor y pueden contener subcolecciones. Las colecciones a su vez no son más que un conjunto de documentos.

En la siguiente imagen podemos ver la estructura final de la base de datos, aunque se ha simplificado ligeramente agrupando varios campos para evitar ofuscar el diagrama, como por ejemplo en usuario donde toda la información sobre el progreso de los logros, que son más de 10 campos, se ha condensado en uno solo.

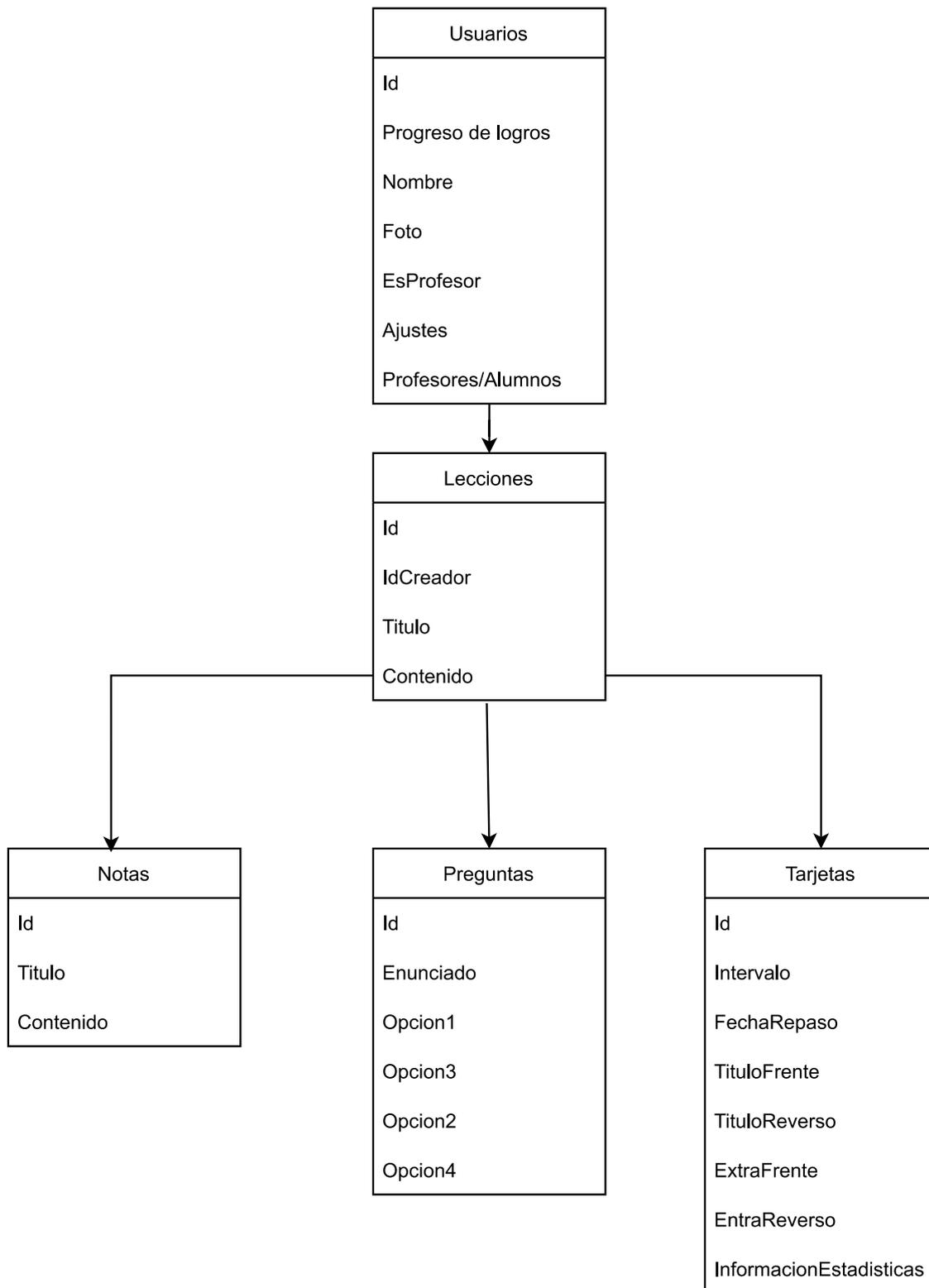


Figura 1. Diagrama simplificado de la base de datos

Tenemos una colección llamada "usuarios", cada documento dentro de esta colección representa un usuario. Estos documentos contienen toda la información de

progreso de los logros del usuario, su nombre y foto de perfil, un valor booleano identificando al usuario como alumno o profesor, una lista con los Id de sus alumnos en caso de ser profesor o de sus profesores en caso de ser alumno.

Además, cada usuario tiene también una subcolección con las lecciones que tiene disponibles. Cada lección contine su título, contenido de la lectura asociada y el ID de su creador que es importante a la hora de permitir al usuario editar o no esta lección.

Cada lección contiene a su vez tres otras subcolecciones, sus notas, sus preguntas y sus tarjetas que contienen la información correspondiente al intervalo y la próxima fecha de repaso.

Las notas contienen dos campos, el titulo que describe el tema tratado en la nota y el contenido que se guarda en formato HTML y permite al usuario utilizar contenido multimedia en sus notas explicativas.

Las preguntas tienen un enunciado y cuatro opciones, cada opción incluye el texto de la respuesta y un valor booleano indicando si esa respuesta es correcta o no.

Las tarjetas tienen cuatro campos de contenido, dos en la parte del frente y dos en el reverso. Estos campos corresponden un titulo que es el contenido a recordar y su traducción en el reverso, además de una frase extra que actúe de contexto y su correspondiente traducción. Además tienen campos para reflejar el intervalo asociado y la próxima fecha de repaso. Finalmente tenemos los campos que se utilizan para contabilizar el progreso de la tarjeta, necesarios para poder generar las estadísticas de aprendizaje.

Esta organización jerárquica y sencilla es el resultado de un proceso de prueba y error a lo largo de todo el desarrollo.

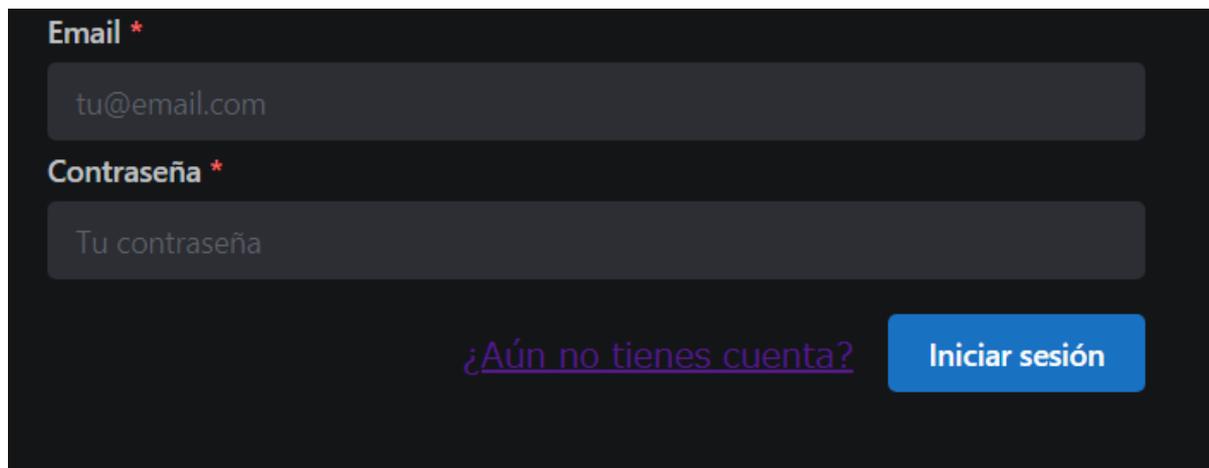
Inicialmente nos planteamos crear una colección dedicada a los logros, donde cada logro guardase una subcolección de usuarios para guardar el progreso individual de cada logro. Esto resultaba en una complejidad innecesaria y un aumento de las consultas a la base de datos, por ello decidimos unificar esta información con el resto de los datos incluidos en el documento del usuario ya que son datos que están muy relacionados y se suelen consultar juntos.

Otra de las decisiones originales que ha sufrido cambios en la forma de almacenar las lecciones que un profesor comparte con sus alumnos. Al principio la idea era que los usuarios tuviesen una referencia a la lección del profesor, pero esto implicaría que tendríamos que cambiar la forma en la que almacenamos el progreso de las tarjetas, ya que se compartirían los mismos documentos para todos los usuarios. Finalmente optamos por duplicar la lección para cada alumno y mantener las copias sincronizadas con el original cada vez que el profesor realice algún cambio.

4.2 Interfaz y navegación

El diseño de la interfaz fue un reto debido a nuestra falta de experiencia en este campo, por ello hemos decidido tomar inspiración de otras aplicaciones exitosas en el ámbito del aprendizaje de lenguas extranjeras. Una de las principales fuentes de inspiración fue *Duolingo* [36] una herramienta con alcance global y una interfaz muy cuidada y agradable. Otra de las fuentes de inspiración fue *Anki* [37] un programa más de nicho, pero muy reconocido y recomendado en la comunidad de estudiantes de idiomas autodidactas.

Las pestañas de inicio de sesión y registro son muy simples, conteniendo cada una un formulario con los datos necesarios exclusivamente. Existe la posibilidad de navegar entre ambas pestañas ya que es una funcionalidad que muy probablemente puedas necesitar.



The image shows a login interface on a dark background. It features two input fields: 'Email *' with a placeholder 'tu@email.com' and 'Contraseña *' with a placeholder 'Tu contraseña'. Below the fields is a link that says '¿Aún no tienes cuenta?' in purple text. To the right of the link is a blue button with the text 'Iniciar sesión' in white.

Figura 2. Inicio de sesión

Nombre *

Tu nombre

Email *

tu@email.com

Contraseña *

Tu contraseña

Estoy de acuerdo con la politica de privacidad

Soy un profesor

[¿Ya tienes cuenta?](#)

Figura 3. Registro

Una vez que se inicia sesión nos encontramos con un diseño basado en cuatro apartados, "estudiar", "logros"/"alumnos", "perfil" y "ajustes". Esta división está inspirada en la forma en la que Duolingo estructura su aplicación, pero ha sido adaptada para ajustarse a nuestras necesidades. La navegación entre estas páginas se puede realizar simplemente pulsando sobre los iconos correspondientes en la parte izquierda de la pantalla.

Otro punto de inspiración se puede ver claramente en la pestaña de "estudiar" donde se hace uso de unos componentes que utilizan anillos para mostrar el progreso, similares a los que utiliza Duolingo, ya que nos parece una forma muy visualmente agradable de presentar el progreso del usuario. Justo en la pestaña de inicio encontramos también un aspecto que está basado en la interfaz de Anki, concretamente los tres números en formato "nuevas/fallidas/aprendiendo" que se pueden ver dentro de cada lección que no esté completa, estos números permiten una forma más precisa de ver el progreso.



Figura 4. Pestaña de estudiar de Anglika

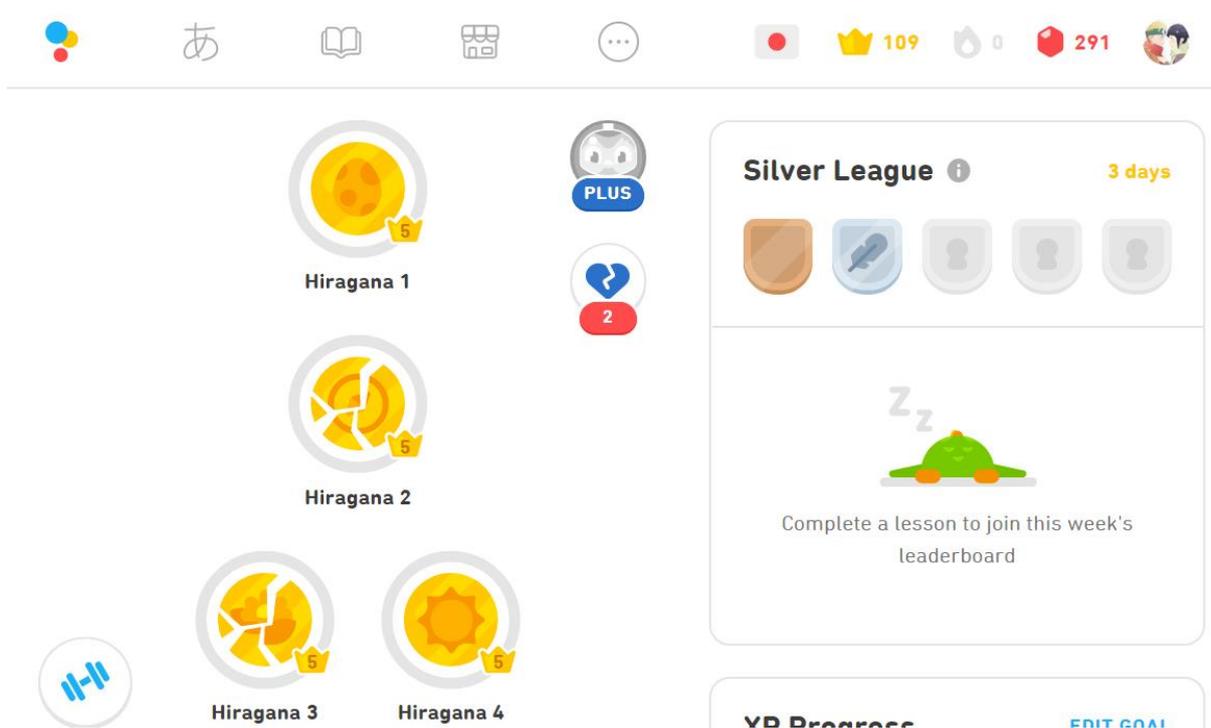


Figura 5. Pestaña de estudiar de Duolingo

Si desde la pestaña de "estudiar" accedemos a los repasos por medio del botón de "estudiar todos", podemos ver una pestaña de repasos basada en la que podemos encontrar en Anki, pero en el caso de Anglika podemos encontrar un temporizador que no existe en Anki, además las opciones con las que el usuario se puede autoevaluar en Anglika se limitan a solo dos, "bien" y "mal", esto se debe a que en caso de pulsar "bien" el grado de

aprendizaje se calcula automáticamente basándose en el tiempo de respuesta, y se puede ver visualmente gracias a la coloración del botón de respuesta que cambia entre verde, amarillo y naranja dependiendo del tiempo transcurrido igual que el propio tiempo del temporizador, que además se torna rojo si el tiempo se acaba.

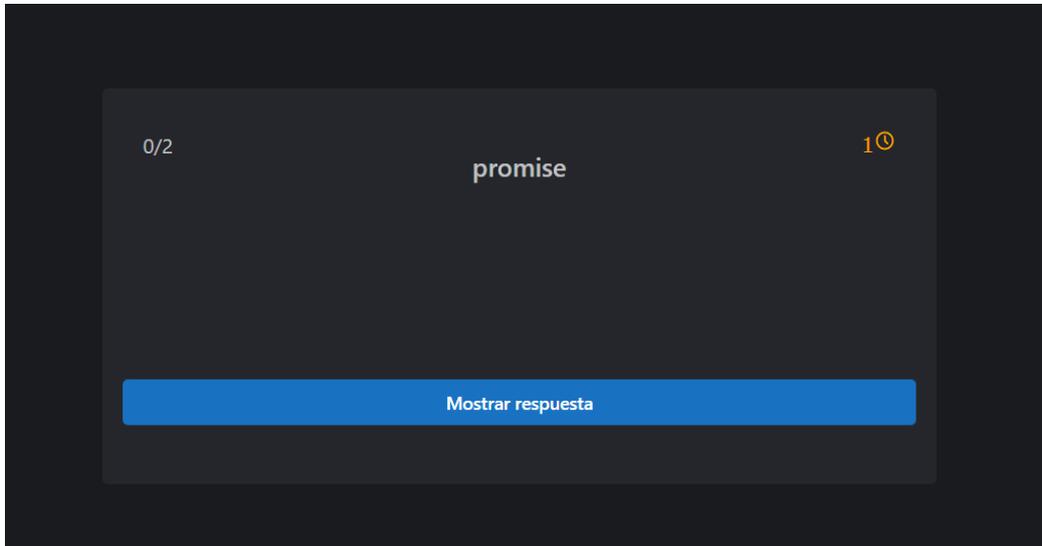


Figura 6. Pestaña de repasos en Anglika

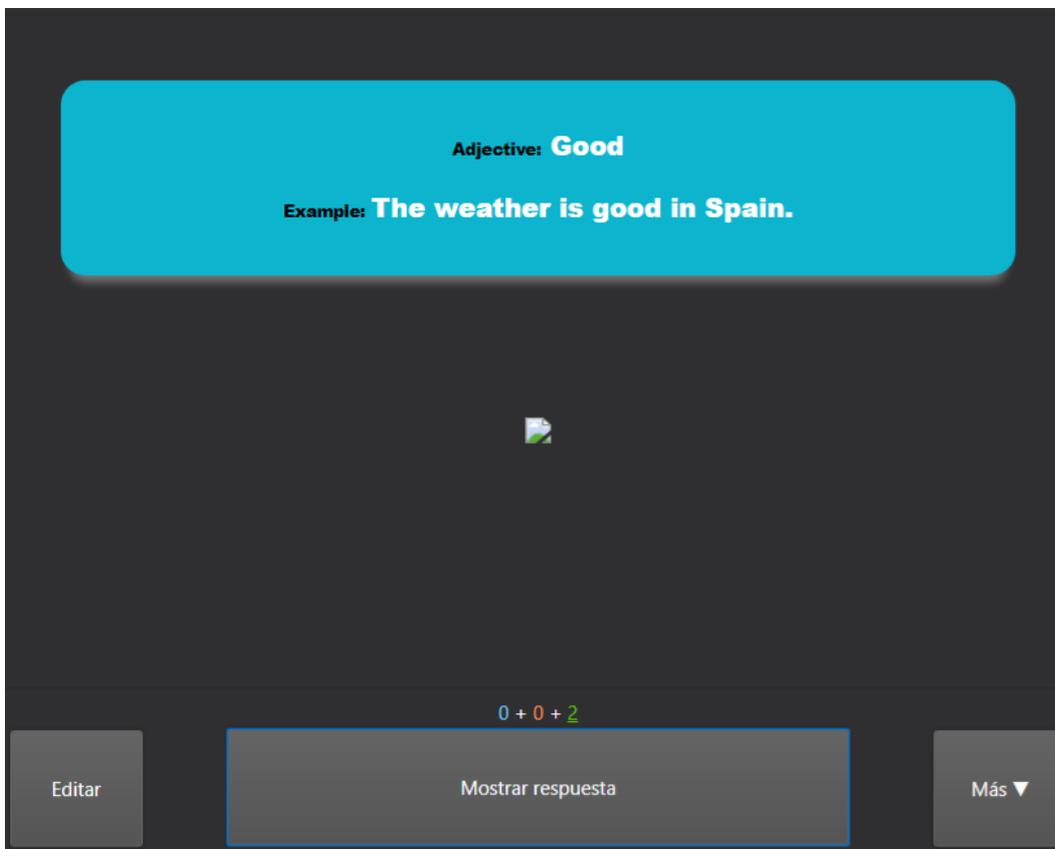


Figura 7. Pestaña de repasos en Anki

El último apartado en el que se cuela la inspiración de otras aplicaciones es en las distintas ventanas de estadísticas, que todas siguen el mismo diseño. Esta ventana está inspirada en su versión en Anki, aunque hemos buscado dar un aspecto más moderno y acorde con el resto de la aplicación.

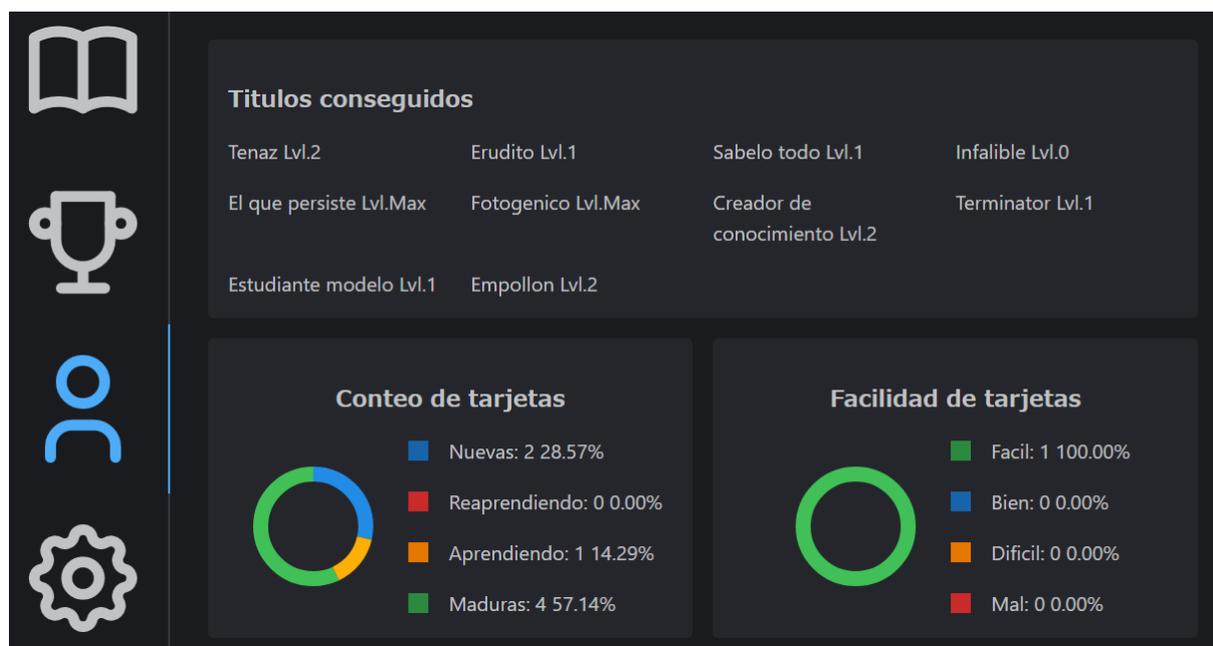


Figura 8. Pestaña de estadísticas de un usuario en Anglika

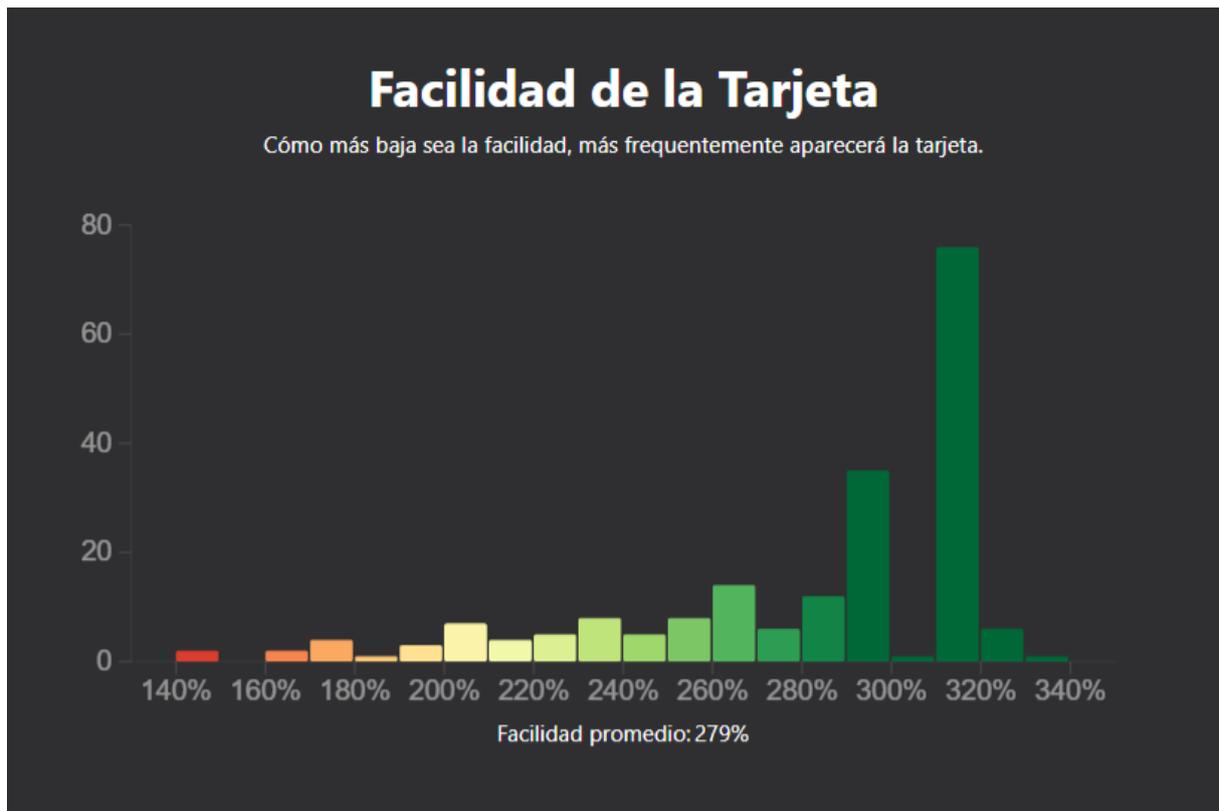


Figura 9. Pestaña de estadísticas de un usuario en Anki

La pestaña de creación de contenido, a la que podemos acceder desde la ventana de inicio o al editar una lección, proporciona una visión global de todos los contenidos de la lección. Pensamos que esta es la forma más intuitiva de crear y editar lecciones, en vez de añadir o modificar los contenidos uno por uno, ya sean tarjetas, notas o preguntas. Si intentas eliminar la lección te encontrarás un mensaje de confirmación que hemos utilizado en varios puntos de la aplicación, siempre que un cambio no fuese reversible.

Titulo

The Frog Prince

Contenido

B *I* U ↶ ↷ ↸ H₁ H₂ H₃ H₄ ☰ ☷ ↵ ↶ ↷ ↸ ↹ ↺ ↻ S² S₂

Once upon a time, a king lived in a large palace with his wife and three daughters. His daughters were all beautiful, but the youngest was so beautiful that even the sun itself, which has seen so much, was astonished whenever it shone in her face. Near to the king's castle there was a great, dark forest, and under an old lime-tree in the forest was a well. One day, when it was very warm, the youngest daughter went out into the forest and sat down by the side of the cool well, and when she was bored she took a golden ball, and threw it up in the air and caught it, for this ball was her favorite toy.

On this occasion the princess's golden ball did not fall into her little hand, but on to the ground beyond, and rolled straight into the water. She followed it with her eyes, but it vanished, and the well was deep, so deep that the bottom could not be seen. At this she began to cry, and cried louder and louder, and could not be comforted. As she was crying, someone said to her, "What's the matter, young princess? Why are you crying so much?"

The End

Tarjetas +

beautiful promise Once upon a time youngest

step by step

Notas +

Some note

Preguntas +

Where all of the daughters beautiful? Another question

Eliminar Guardar

Figura 10. Pestaña de creación de contenido

5

Desarrollo

El desarrollo de la aplicación se ha dividido en 4 iteraciones a las que hay que sumar una iteración inicial en la que se han identificado todos los requisitos de la aplicación y la iteración final en la que se ha realizado una batería de pruebas manuales para comprobar el funcionamiento global de la aplicación.

Tras cada una de estas iteraciones se han realizado múltiples pruebas manuales de cada uno de los casos de uso implementados, probando tanto el caso de éxito como los casos alternativos. también se han revisado los requisitos para corregir o actualizar la lista inicial, lo cual es muy importante, ya que nuestra visión del proyecto ha sufrido ligeras modificaciones a lo largo del desarrollo.

5.1 Funcionalidad básica del usuario

Esta iteración es muy amplia, incluyendo desde el inicio de sesión hasta los repasos de tarjetas. Es por ello por lo que ha llevado mucho tiempo implementar todas las funcionalidades y algunas se han postpuesto a otras iteraciones.

En el apartado de inicio de sesión y registro hemos decidido buscar la simplicidad, pidiendo solo un nombre de usuario, email único y contraseña. Gracias a *Firebase* el manejo de la autenticación de forma segura es trivial, al proporcionar herramientas que nos permiten crear cuentas e iniciar sesión de forma totalmente segura.

La parte más compleja relacionada con el inicio de sesión fue la de evitar que un usuario pudiese acceder a una pestaña sin haberse identificado anteriormente. Para ello envolvimos cada componente dentro de otro componente que comprobase si se ha iniciado sesión, en tal caso se mostrará el componente original, en caso contrario se redirigiría hacia la pestaña de inicio de sesión, manteniendo información de la dirección a donde se quería acceder originalmente, para poder redireccionar de vuelta una vez se inicie sesión. Cabe destacar que toda la navegación dentro de la aplicación se ha realizado usando la librería React Router [38].

```
// Redirects the user to the logging window if not logged in
const Authenticated = ({ children }) => {
  const location = useLocation()
  if (!getAuth().currentUser) {
    return (<Navigate to="/login" element={<Login />} replace state={{ from: location.pathname }} />)
  }

  return children
}
```

Figura 11. Componente que redirecciona a la pestaña de inicio de sesión si no se ha iniciado sesión aun

```
return (
  <UserContext.Provider value={isProfesor}>
    <BrowserRouter>
      <Routes>
        <Route path="login" element={<Login />} />
        <Route path="register" element={<Register />} />
        <Route path="/" element={<Authenticated ><App /></Authenticated>} />
        <Route path="teoria/:tema" element={<Authenticated><Teoria /></Authenticated>} />
        <Route path="reading/:tema" element={<Authenticated><Reading /></Authenticated>} />
        { /* The no params version reviews every thing, the one with a parameter only adds new words to the learning list */ }
        <Route path="review/" element={<Authenticated><Review /></Authenticated>} />
        <Route path="review/:mazo" element={<Authenticated><Review /></Authenticated>} />
        { /* The no params version creates a new lesson, the one with a parameter only edits it*/ }
        <Route path="create/:mazo" element={<Authenticated><Create /></Authenticated>} />
        <Route path="create" element={<Authenticated><Create /></Authenticated>} />
        <Route path="*" element={<Error />} />
      </Routes>
    </BrowserRouter>
  </UserContext.Provider>
)
```

Figura 12. Componente que contiene las rutas de la aplicación

Otro apartado en el que también hemos buscado la simplicidad es el apartado de información del usuario, donde en esta iteración solo se implementó la habilidad de cerrar sesión, eliminar la cuenta o editar el perfil. La edición de perfil hace uso de Mantine Dropzone [39], un componente reutilizable que nos permite seleccionar o arrastrar una imagen directamente en el navegador, esta imagen seleccionada la usamos luego a modo de foto de perfil. El componente se encarga automáticamente de rechazar archivos que no

cumplan con las condiciones de pesar menos de 5mb y ser una imagen. Una vez que el usuario selecciona una imagen esta se sube a la plataforma *Imgur* y un enlace se genera y se utiliza para mostrar la previsualización antes de que el usuario pulse guardar y se almacene la información en la base de datos.

```
<Dropzone
  onDrop={
    async (files) => {
      setLoading(true)
      setImageUrl(await handleImageUpload(files[0]))
    }
  }
  loading={loading}
  onReject={(_) => notifications.showNotification({ message: "Ese archivo no es valido", color: "red" })}
  maxSize={3 * 1024 ** 2}

  accept={IMAGE_MIME_TYPE}
  multiple={false}
  my="md"
>
  {(status) => dropzoneChildren(status, theme)}
</Dropzone>
```

Figura 13. Componente Dropzone que permite al usuario seleccionar una imagen

La pestaña de ajuste fue muy sencilla de desarrollar, al contener solo un formulario en el que los usuarios pueden editar sus ajustes y guardarlos en la base de datos simplemente pulsando un botón.

La pestaña de logros fue igualmente sencilla, aunque debemos admitir que ha sufrido muchos cambios desde su concepción inicial. Durante el proceso de desarrollo hemos jugado mucho con distintas formas de mostrar el progreso de los logros, pero finalmente nos decidimos por usar el componente *RingProgress* incluido con *Mantine*. Este componente nos ofrece una forma muy cómoda y visualmente atractiva para presentar el progreso del usuario.

```

const Achievement = ({ name, logro, milestones, progress }) => {
  const [title, active] = getTitle(name, milestones, progress)
  const target = milestones[active]

  return (
    <Stack justify="center" align="center">
      <h2>{title}</h2>
      <RingProgress
        roundCaps
        sections={[{ value: (progress / target) * 100, color: active == -1 ? "yellow" : "teal" }]}
        label={
          active == -1 ?
            <Center >
              <ThemeIcon color="yellow" variant="light" radius="xl" size="xl">
                <Star size={22} />
              </ThemeIcon>
            </Center >
            :
            <Text align="center" size="xl" weight="bold">{progress} / {target} </Text>
          </label>
        </RingProgress>
      <Text>{target ? logro.replace("%", target) : "¡Has completado este logro!"}</Text>
    </Stack>
  )
}

```

Figura 14. Componente que representa un único logro

La pestaña de inicio fue sin duda una de las más complejas, puesto que necesita información acerca del progreso de todos los temas del usuario y sus tarjetas. Lo primero que hacemos es obtener la información de las tarjetas de la base de datos, luego la procesamos para diferenciar las tarjetas pendientes de las fallidas, nuevas o aprendidas. Una vez que tenemos toda esa información podemos calcular el porcentaje de cada una de ellas para mostrarlo al usuario. También utilizamos esta información para habilitar o deshabilitar ciertas acciones como la capacidad de aprender nuevas tarjetas o editar un mazo.

Finalmente tenemos la pestaña de repaso de tarjetas. En esta pestaña se muestra el frente de las tarjetas hasta que el usuario pulse mostrar, después deberá evaluar como "Bien" o "Mal" dependiendo de si ha recordado correctamente el significado de la tarjeta.

La pestaña de repasos también utiliza un intervalo para llevar la cuenta del tiempo transcurrido, lo cual utilizamos para mostrar un temporizador. Si el temporizador termina se revela la tarjeta y el usuario es forzado a escoger la opción de "Mal". El tiempo transcurrido también se utiliza para seleccionar el aumento de intervalo, mientras menos tiempo se tarda en responder, más aumenta el intervalo.

```

const multiplier = time_passed_percentage > 66 ?
  bonus.easy
: time_passed_percentage > 33 ?
  bonus.ok
:
  bonus.hard

const newInterval = Math.round(multiplier * cards[0].interval)
const newDueDate = new Date()

newDueDate.setDate(newDueDate.getDate() + newInterval)

updateCard(newDueDate, newInterval, false)

```

Figura 15. Lógica encargada de actualizar el intervalo y la fecha de repaso de una tarjeta

5.2 Añadir contenido

Las funcionalidades de añadir contenido están contenidas en una única pestaña que se utiliza tanto para crear como para editar un tema. Hemos elegido combinar estas dos funcionalidades en una sola pestaña ya que nos permite reutilizar la mayor parte de la lógica.

Inicialmente esta iteración parecía sencilla, pero con la adición de los profesores y la necesidad de sincronizar los cambios que realizan a sus temas con los temas de sus alumnos se complicó moderadamente.

El principal problema fue mantener los mismos Ids para las tarjetas, notas, preguntas y los propios temas para el profesor y sus alumnos. Esto es necesario porque a la hora de calcular las estadísticas y sincronizar los cambios necesitamos alguna forma de emparejar las tarjetas del profesor con las de sus alumnos.

Esta iteración, aunque no tan extensa como la anterior requirió muchas más pruebas manuales para poder confirmar que todos los casos funcionaban correctamente.

5.3 Logros

Esta fue sin duda la iteración más liviana al consistir básicamente de encontrar los puntos en el código donde se deben progresar los logros y comprobar si se ha alcanzado alguna meta para mostrar la notificación correspondiente.

Tal vez la funcionalidad más compleja fue la racha de días ya que inicialmente utilizamos la información que *Firebase* nos proporcionaba acerca de la última fecha de inicio de sesión. El problema es que la información de *Firebase* no se actualizaba instantáneamente tras iniciar sesión, lo cual causaba que si recargabas la página rápidamente podías seguir aumentando tu racha de días.

Otro problema que tuvimos fue a la hora de comparar fechas, ya que en algunos casos las fechas se encontraban en formato textual y otras veces en segundos o incluso milisegundos. Lo que acabamos haciendo es convertir todas las fechas a formato textual ignorando la hora y guardando solo la fecha. Es muy posible que esta solución no sea perfecta si el usuario viaja de un país a otro que este en otra zona horaria, pero pensamos que es un problema menor y no impide al usuario promedio utilizar la aplicación de forma satisfactoria.

```
const last = new Date(userDoc.data().lastSignInTime) ?? new Date(user.metadata.lastSignInTime) ?? new Date()
const today = new Date().toDateDateString()

if (isToday(last)) {
  updateDoc(userRef, { lastSignInTime: today })
} else if (isYesterday(last)) {
  const newRacha = (userData.racha ?? 0) + 1
  const batidoRecord = (userData.Tenaz ?? 0) < newRacha
  const newTenaz = batidoRecord ? newRacha : (userData.Tenaz ?? 0)
  await updateDoc(userRef, { racha: newRacha, lastSignInTime: today, learnedToday: 0, Tenaz: newTenaz })

  if (batidoRecord) {
    checkAchievement("Tenaz", notifications)
  }

  if (today.includes("Sun") || today.includes("Sat")) {
    await updateDoc(userRef, { "El que persiste": increment() })
    checkAchievement("El que persiste", notifications)
  }
} else {
  await updateDoc(userRef, { racha: 1, lastSignInTime: today, learnedToday: 0 })
  notifications.showNotification({ title: "Has perdido tu racha", color: "red" })
}
```

Figura 16. Código encargado de comprobar la racha de días seguidos que se ha conectado el usuario

5.4 Estadísticas y profesores

En esta iteración añadimos los profesores a la aplicación, lo cual necesitó realizar cambios a otros apartados implementados en iteraciones anteriores.

Lo primero fue añadir la opción de registrarse como profesor, lo que fue bastante sencillo, requiriendo simplemente una casilla extra en la pestaña de registro. Igualmente,

sencillo fue deshabilitar los logros ya que la lógica encargada de comprobar el progreso estaba autocontenida en una única función, simplemente fue necesario añadir una comprobación de si el usuario actual es o no un profesor.

Un cambio mayor fue el de la pestaña de logros, que fue substituida completamente por una pestaña en la que se muestran los alumnos del profesor y se puede consultar sus estadísticas o eliminarlos de la lista de alumnos. Esta decisión viene a raíz de que los logros tienen el objetivo de motiva a los alumnos a estudiar, pero el rol de profesor no te permite aprender lecciones, si no que solo las puede crear y consultar las estadísticas de sus alumnos.

En línea con lo anterior en la pantalla de inicio tuvimos que eliminar todo lo referente a estudiar tarjetas, leer lecciones o cualquier otra acción innecesaria para un profesor. El botón de repasos se substituyó por uno dedicado a añadir alumnos. Para añadir alumnos decidimos que la manera más adecuada sería mostrar una lista con los usuarios disponibles, es decir que no sean profesores ni estén ya matriculados como alumnos del usuario actual. De esta forma el profesor puede seleccionar de la lista varios alumnos a la vez, e incluso filtrar por nombre a los alumnos. Una vez haya elegido a los alumnos que quiera simplemente pulsa el botón de guardar los cambios y se sincronizarán las lecciones necesarias y se añadirán a la lista de alumnos, todo automáticamente.

La adición de las estadísticas ha involucrado varios cambios, como añadir un botón extra en el menú que se despliega al seleccionar una lección que permite ver las estadísticas de esa lección. Este botón muestra estadísticas diferentes en el caso de ser un profesor o un alumno, si se es un profesor se muestra las estadísticas medias de todos sus alumnos en esa lección, pero si es un alumno solo se muestran sus estadísticas.

Otro lugar donde se pueden consultar las estadísticas es en la pestaña de perfil de un usuario, donde además de la lista de sus profesores se puede ver también las estadísticas promedias de todas sus lecciones. Los profesores pueden acceder a estas mismas estadísticas de cada uno de sus alumnos desde la pestaña de alumnos.

```

isProfesor ?
  <Group position="center" grow>
    <Button onClick={() => setOpen(true)}>
      Añadir alumno
    </Button>
    <Button onClick={() => navigate("create")}>
      Añadir contenido
    </Button>
  </Group>
:
  <Group position="center" grow>
    <Button disabled={pending === 0} onClick={() => navigate("review")}>
      Repasar pendientes {pending}
    </Button>
    <Center>
      <h4>
        Racha de días: {racha}
      </h4>
    </Center>
    <Button onClick={() => navigate("create")}>
      Añadir contenido
    </Button>
  </Group>

```

Figura 17. Ejemplo de lógica que comprueba si el usuario es profesor o no y muestra la interfaz correspondiente.

6

Conclusiones y trabajos futuros

6.1 Conclusiones

La conclusión general es que estoy muy satisfecho con el resultado final, ya que se ha conseguido cumplir de forma satisfactoria los requisitos que se establecieron originalmente.

El nivel de usabilidad y la interfaz de la aplicación es muy satisfactorio, lo cual era un punto de duda inicialmente, al ser la primera aplicación web de gran envergadura que desarrollaba. Es gracias a la librería *Mantine* que se ha podido crear una interfaz moderna y visualmente atractiva.

Las funcionalidades relativas a la creación de contenido cumplen todo lo esperado permitiendo la fácil creación de niveles con contenido audio visual, que es justamente lo que se necesita para poder aplicar los principios de las teorías de aprendizaje basado en "input" y en la adquisición del lenguaje.

Se ha conseguido que los aspectos de gamificación estén presentes en todas las partes de la aplicación, desde la creación de lecciones a través del sistema de logros, hasta los repasos de tarjetas. Esto realmente ayudará a hacer la aplicación más atractiva para los usuarios, fortaleciendo así su habilidad para crear un hábito de aprendizaje.

Los conocimientos adquiridos en el grado me han sido muy útiles en múltiples puntos del desarrollo, aunque haya sido necesario complementarlos con una dosis de autoaprendizaje en línea. En concreto, las metodologías y patrones que he aplicado, de forma exitosa, para manejar la complejidad de la aplicación, se los debo a la enseñanza que he recibido durante estos últimos años.

Durante el desarrollo de la aplicación han aparecido muchas dificultades, desde el diseño de la interfaz hasta problemas a la hora de diseñar la estructura de la base de datos. Puedo decir sin duda que estos problemas se han afrontado sin miedo y ha sido posible abordarlos satisfactoriamente, lo cual me ha hecho más fuerte y capaz de autoaprendizaje y resolver problemas.

Finalmente, resaltar que esta actividad ha sido muy didáctica y me ha permitido crecer como desarrollador y como ingeniero de software. Además, he aprendido muchas técnicas que me pueden ayudar en el futuro a la hora de aprender una lengua extranjera.

6.2 Trabajos futuros

La aplicación está en un estado que considero como completo, pero esto no quiere decir que no existan posibles mejoras. Por ello, en este apartado se presentan unas cuantas ideas que se podrían implementar en el futuro para llevar la aplicación a otro nivel.

- **Añadir más logros y otros aspectos de gamificación**

La gamificación es una parte importante de la aplicación, que en la versión actual se implementa principalmente mediante un sistema de logros que el usuario puede conseguir realizando acciones dentro de la aplicación.

Sería interesante aumentar la cantidad y variedad de logros que el usuario puede obtener, ya que ahora mismo la lista es bastante reducida.

Otra idea interesante sería incluir un sistema de puntos que el usuario pueda conseguir realizando repasos o pruebas de comprensión lectora. Estos puntos se podrían canjear por algún tipo de modificador visual o se podrían utilizar para un ranking online, fomentando la competitividad innata de las personas.

- **Mejorar el algoritmo de retención de conocimiento**

El sistema de repetición espaciada de las tarjetas utiliza nuestro conocimiento de la memoria humana para mostrar las tarjetas a repasar justo cuando el usuario está a punto de olvidarlas, haciendo que los repasos sean mucho más efectivos.

Actualmente el algoritmo que se está utilizando es el más básico, que presupone un aumento exponencial en el intervalo de memoria, pero existen otras soluciones en el mercado más complejas y efectivas que se podrían aplicar. Para ello sería necesario investigar las posibles alternativas y sopesar los pros y contras de cada una de ellas.

- **Traducción automática**

Actualmente los usuarios disponen de múltiple herramienta para crear lecciones, pero el proceso puede ser tedioso, sobre todo a la hora de crear tarjetas de vocabulario que se deben crear manualmente. Una forma de mejorar esto podría ser añadir una opción para generar automáticamente estas tarjetas con traducciones automáticas.

Esto también se podría propagar a las lecturas, donde el usuario podría traducir selectivamente las frases que no pueda entender.

- **Añadir más estadísticas**

La habilidad de la aplicación de recoger estadísticas de sus alumnos tiene un gran potencial para los profesores y para los propios alumnos, ya que les proporcionan una

guía que indica a que apartados debería dedicar más esfuerzo, para optimizar su aprendizaje.

Por ello podría ser muy interesante aumentar el número de estadísticas que se pueden consultar, para incluir cosas como las tarjetas que se fallan con más frecuencia, las preguntas de comprensión lectora que los alumnos consideran más difíciles y todo tipo de métricas que los profesores puedan encontrar útiles.

- **Mejorar la usabilidad en dispositivos móviles**

El desarrollo se ha centrado principalmente en el uso en plataformas de escritorio, lo cual ha facilitado el desarrollo enormemente. Esto no quiere decir que el mercado móvil no sea importante, ni mucho menos. Es por ello por lo que actualizar la interfaz para que sea más amigable para dispositivos móviles es la prioridad número uno a la hora de seguir avanzando con el desarrollo de la aplicación.

Mejorar la usabilidad en dispositivos móviles permitiría a los usuarios aprender en cualquier parte, promoviendo la creación de hábitos y aumentando el número de oportunidades a lo largo del día para estudiar.

Referencias

- [1] EF EPI 2021 EF english proficiency index. (s. f.). EF Education First - España.
<https://www.ef.com.es/eipi/>
- [2] Krashen, S. (2021, 10 de julio). Why don't educators, scholars, and the media pay attention to the research? Books and Articles by Stephen D Krashen.
http://www.sdkrashen.com/content/articles/why_dont_educators_scholars.pdf
- [3] Krashen, S. (s. f.). Books and articles by stephen D krashen. Books and Articles by Stephen D Krashen. <http://www.sdkrashen.com/>
- [4] Contributors to Wikimedia projects. (2001, 9 de agosto). Spaced repetition - Wikipedia. Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Spaced_repetition
- [5] Fitz-Walter, D. Z. (s. f.). What is gamification? Education, business & marketing (2021 examples). Make a Video Game in Minutes Not Months. <https://www.gamify.com/what-is-gamification>
- [6] Google. (s. f.). Firebase. Firebase. <https://firebase.google.com/>
- [7] Imgur. (s. f.). Imgur. Imgur. <https://imgur.com/>
- [8] MongoDB | build faster. build smarter. (s. f.). MongoDB. <https://www.mongodb.com/>
- [9] JavaScript.com. (s. f.). JavaScript.com. <https://www.javascript.com/>
- [10] React – A JavaScript library for building user interfaces. (s. f.). React – A JavaScript library for building user interfaces. <https://reactjs.org/>
- [11] Mantine. (s. f.). Mantine. <https://mantine.dev/>
- [12] Angular. (s. f.). Angular. <https://angular.io/>
- [13] Vue.js - The Progressive JavaScript Framework | Vue.js. (s. f.). Vue.js - The Progressive JavaScript Framework | Vue.js. <https://vuejs.org/>
- [14] Vite. (s. f.). Vite. <https://vitejs.dev/>

- [15] Create React App. (s. f.). Create React App. <https://create-react-app.dev/>
- [16] MUI: The React component library you always wanted. (s. f.). MUI: The React component library you always wanted. <https://mui.com/>
- [17] Rich text editor. (s. f.). Mantine. <https://mantine.dev/others/rte/>
- [18] RingProgress. (s. f.). Mantine. <https://mantine.dev/core/ring-progress/>
- [19] Git. (s. f.). Git. <https://git-scm.com/>
- [20] GitHub: Where the world builds software. (s. f.). GitHub. <https://github.com/>
- [21] Sourcetree | Free Git GUI for Mac and Windows. (s. f.). SourceTree. <https://www.sourcetreeapp.com/>
- [22] GitKraken Legendary Git Tools | GitKraken. (s. f.). GitKraken. <https://www.gitkraken.com/>
- [23] Microsoft. (2021, 3 de noviembre). Visual Studio Code - Code Editing. Redefined. Visual Studio Code - Code Editing. Redefined. <https://code.visualstudio.com/>
- [24] What is an IDE? (s. f.). Red Hat - We make open source technologies for the enterprise. <https://www.redhat.com/en/topics/middleware/what-is-ide>
- [25] A hackable text editor for the 21st Century. (s. f.). Atom. <https://atom.io/>
- [26] Sublime Text - the sophisticated text editor for code, markup and prose. (s. f.). Sublime Text - Text Editing, Done Right. <https://www.sublimetext.com/>
- [27] Error Lens - Visual Studio Marketplace. (s. f.). Extensions for Visual Studio family of products | Visual Studio Marketplace. <https://marketplace.visualstudio.com/items?itemName=usernamehw.errorlens>
- [28] Todo Tree - Visual Studio Marketplace. (s. f.). Extensions for Visual Studio family of products | Visual Studio Marketplace. <https://marketplace.visualstudio.com/items?itemName=Gruntfuggly.todo-tree>
- [29] Microsoft Word - Word Processing Software | Office." Wwww.microsoft.com, www.microsoft.com/en-us/microsoft-365/word.
- [30] Cloud Application Platform | Heroku. (s. f.). Cloud Application Platform | Heroku. <https://www.heroku.com/>
- [31] Krashen, S. D. (1982). Principles and practice in second language acquisition. Pergamon.
- [32] Krashen, S. (2014). Language acquisition without speaking and without study. Books and Articles by Stephen D Krashen.

http://sokrashen.com/content/articles/2014_lao_and_krashen_language_acquisition_with_out_speaking_and_study.pdf

[33] *What is the i+1 Principle? | Oxford Open Learning. (s. f.). Oxford Open Learning | UK Distance Learning | iGCSE & A Level Courses.* <https://www.ool.co.uk/blog/what-is-the-i1-principle/>

[34] *Mason, Beniko. (2019). Guided SSR before Self-Selected Reading.*

[35] *Krashen, S. (2020, 20 de mayo). What about writing? Books and Articles by Stephen D Krashen.* http://www.sokrashen.com/content/articles/what_about_writing_.pdf

[36] *Learn a language for free. (s. f.). Duolingo.* <https://www.duolingo.com/>

[37] *Anki - powerful, intelligent flashcards. (s. f.). Anki - powerful, intelligent flashcards.* <https://apps.ankiweb.net/>

[38] *Declarative routing for React apps at any scale | React Router. (s. f.). Declarative routing for React apps at any scale | React Router.* <https://reactrouter.com/>

[39] *Dropzone. (s. f.).* <https://mantine.dev/others/dropzone/>

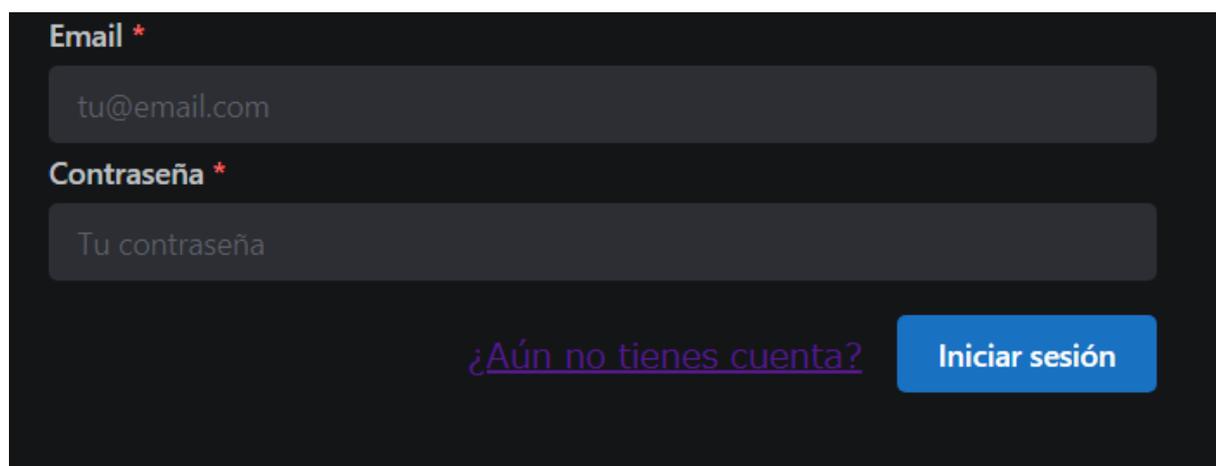
Apéndice A

Manual de usuario

A.1 Crear una cuenta

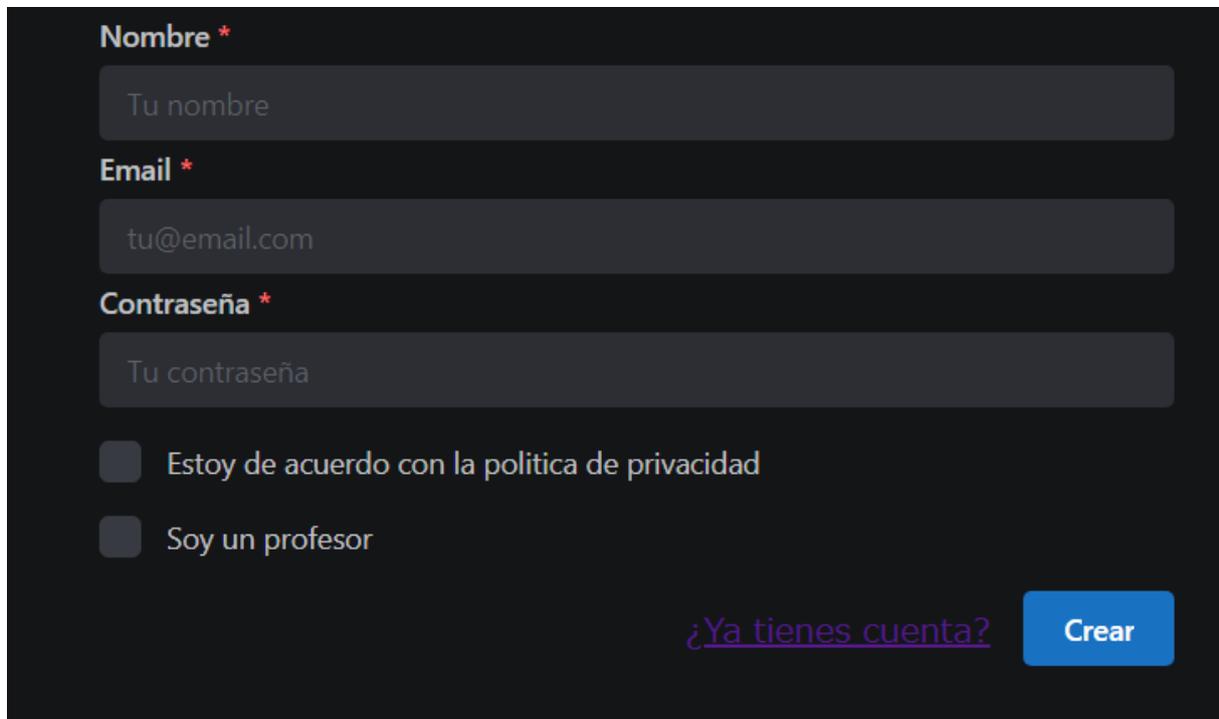
Lo primero que vamos a hacer es acceder a la aplicación utilizando el siguiente enlace: tfg-antonio-gc.herokuapp.com.

Una vez que accedamos al enlace por primera vez nos encontraremos una ventana de inicio de sesión similar a la siguiente:



The image shows a dark-themed login form. At the top left, the label "Email *" is followed by a text input field containing the placeholder "tu@email.com". Below this, the label "Contraseña *" is followed by a text input field containing the placeholder "Tu contraseña". At the bottom right of the form, there is a blue button labeled "Iniciar sesión" and a purple link that reads "¿Aún no tienes cuenta?".

Una vez que tengamos una cuenta podremos iniciar sesión introduciendo nuestros datos en este formulario. Como aun no tenemos cuenta vamos a pulsar sobre "¿Aún no tienes cuenta?", lo que nos llevará a la página de registro:



Nombre *

Tu nombre

Email *

tu@email.com

Contraseña *

Tu contraseña

Estoy de acuerdo con la politica de privacidad

Soy un profesor

[¿Ya tienes cuenta?](#)

En esta pestaña tendremos que rellenar los campos obligatorios y aceptar las políticas de privacidad, una vez lo hagamos y pulsemos "Crear" nos habremos registrado exitosamente. Si quieres crear una cuenta de profesor debes marcar la casilla de "Soy un profesor"

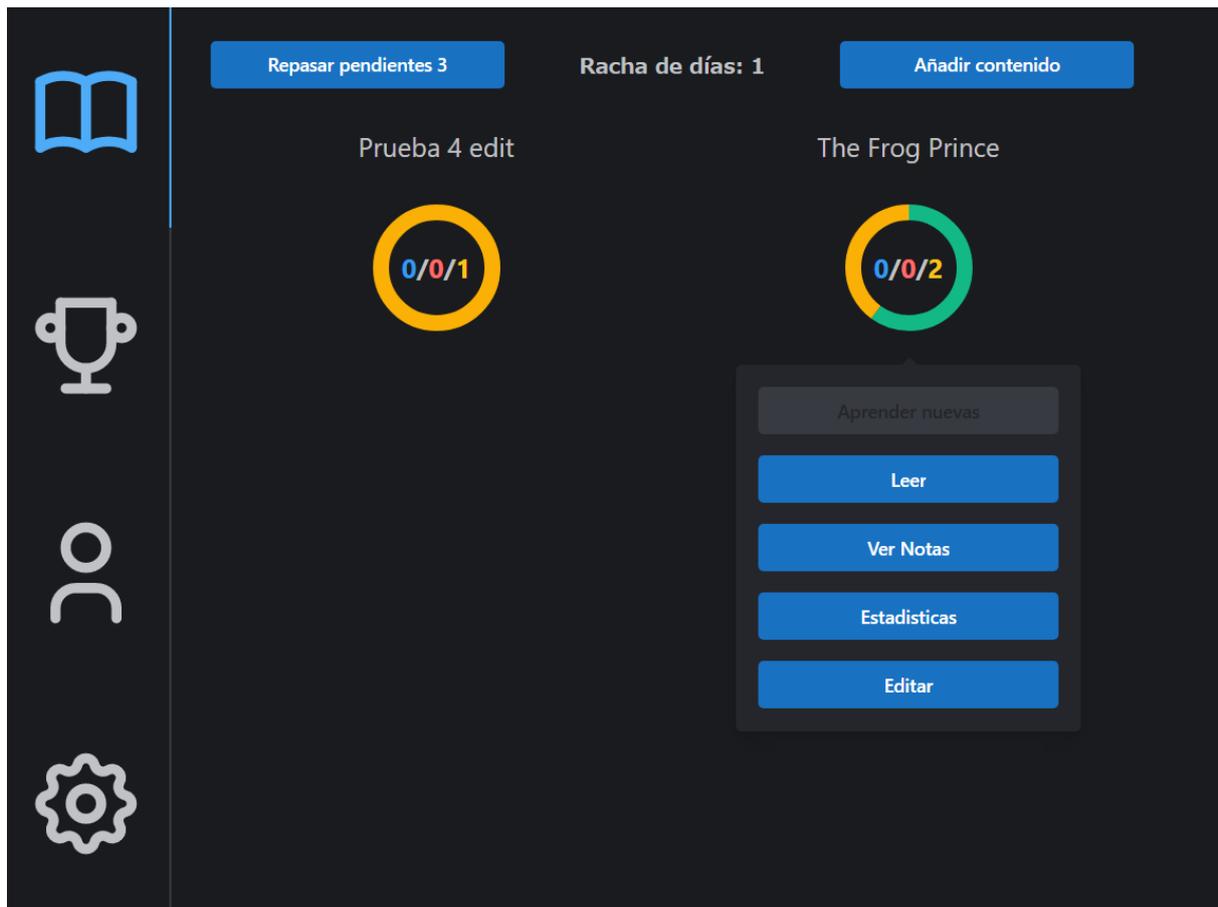
A.2 Crear una lección

Una vez que hayamos iniciado sesión y queramos crear una nueva lección, es tan fácil como pulsar el botón de "Añadir contenido" arriba a la derecha. Si quisiésemos editar una lección en vez de crear una nueva simplemente pulsamos sobre ella y luego el

botón

de

"Editar".

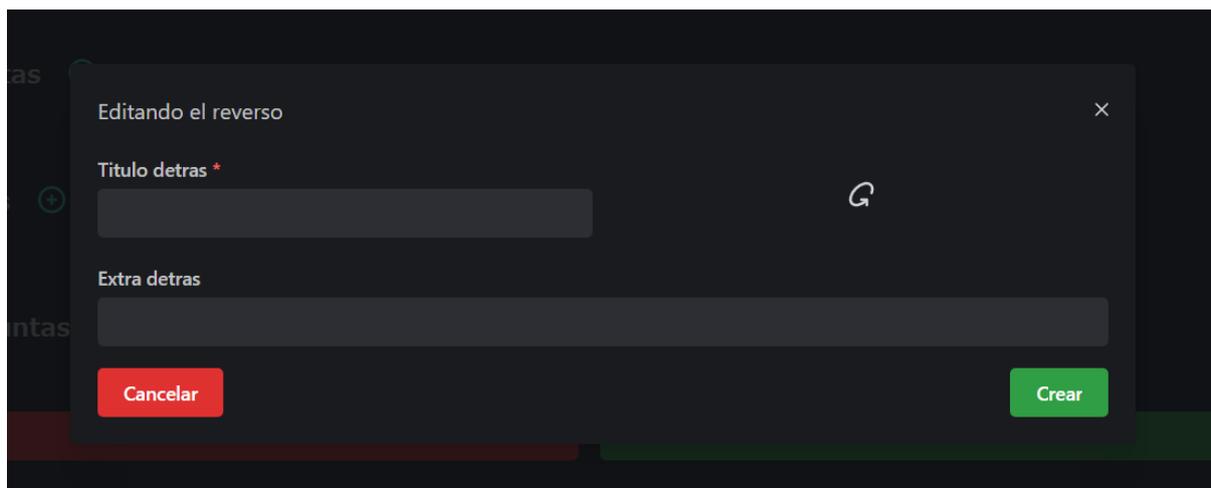


Una vez que estemos en la pestaña de creación/edición de una lección nos encontraremos con varias partes.

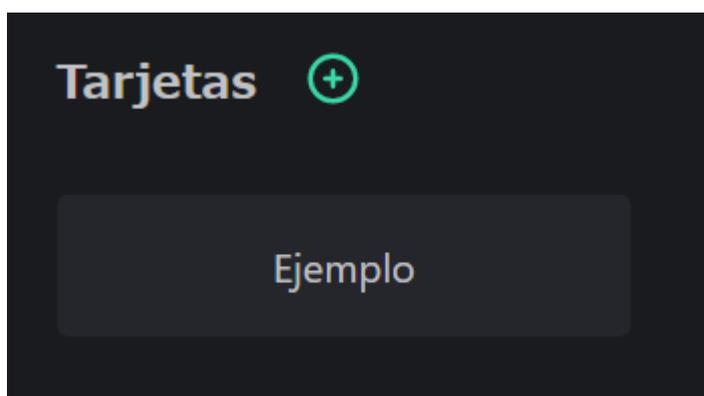
El título es el nombre que se mostrará asociado a la lección, como por ejemplo "The Frog Prince" en la captura anterior.

El contenido es un editor de texto enriquecido que nos permite crear una lectura que contenga imágenes, videos y cualquier otro tipo de contenido multimedia. Para añadir imágenes puedes simplemente arrastrarlas encima del componente.

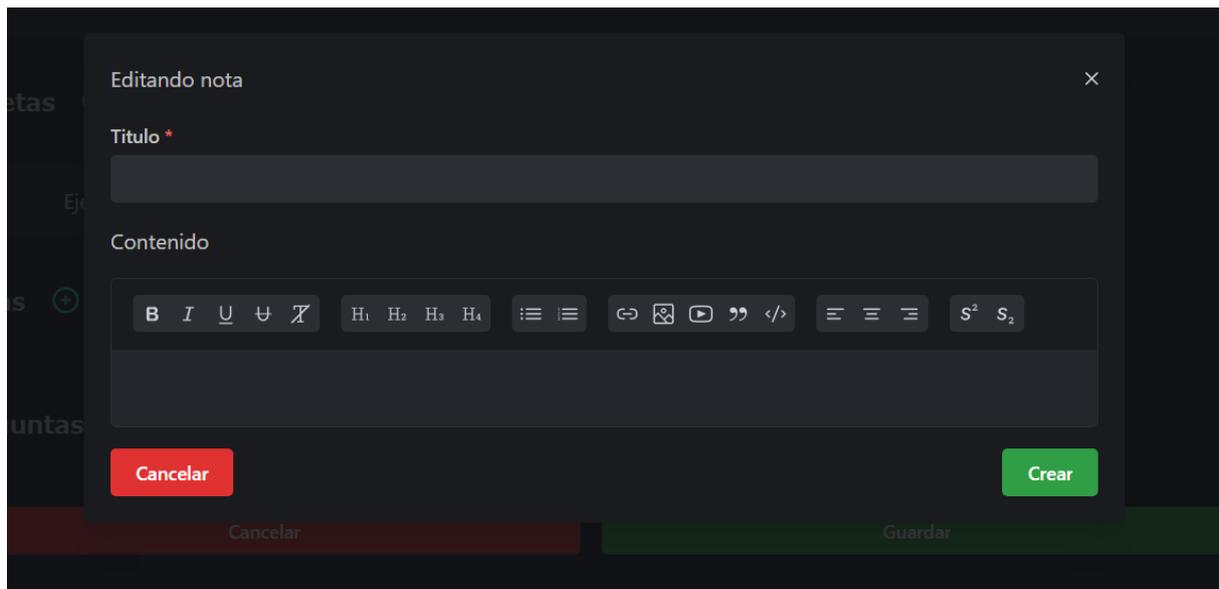
Luego encontramos las tarjetas, notas y preguntas, que inicialmente estarán vacías, pero podemos añadir elementos pulsando el símbolo de + a la derecha de cada una.



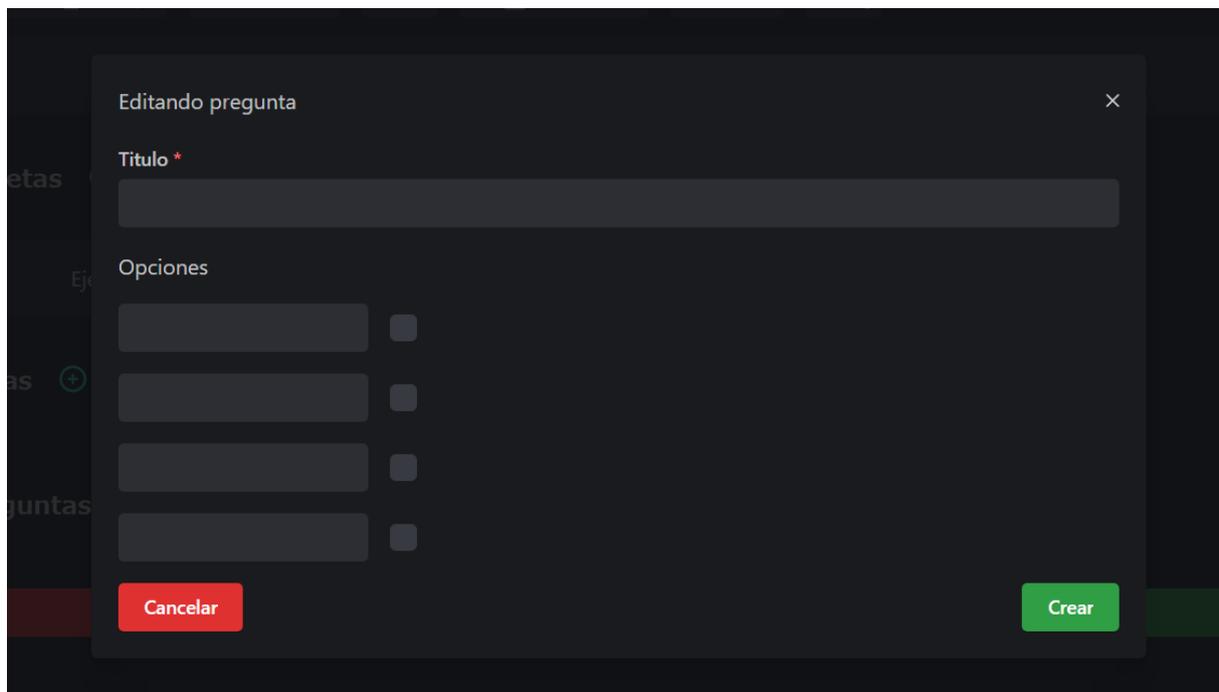
Una vez creada aparecerá en la lista de tarjetas, para realizar cambios simplemente tienes que pulsar sobre ella y volverás a encontrar el mismo menú de edición/creación. Este proceso es el mismo para las notas y las preguntas.



Las notas son una forma de añadir explicaciones adicionales acerca de la lección, ya sea sobre algún punto de gramática, o sobre contexto para entender mejor la lección. Para ello contienen un título que debe ser una pequeña descripción del contenido y luego el contenido como tal, que vuelve a ser un texto enriquecido, permitiendo el uso de contenido multimedia.



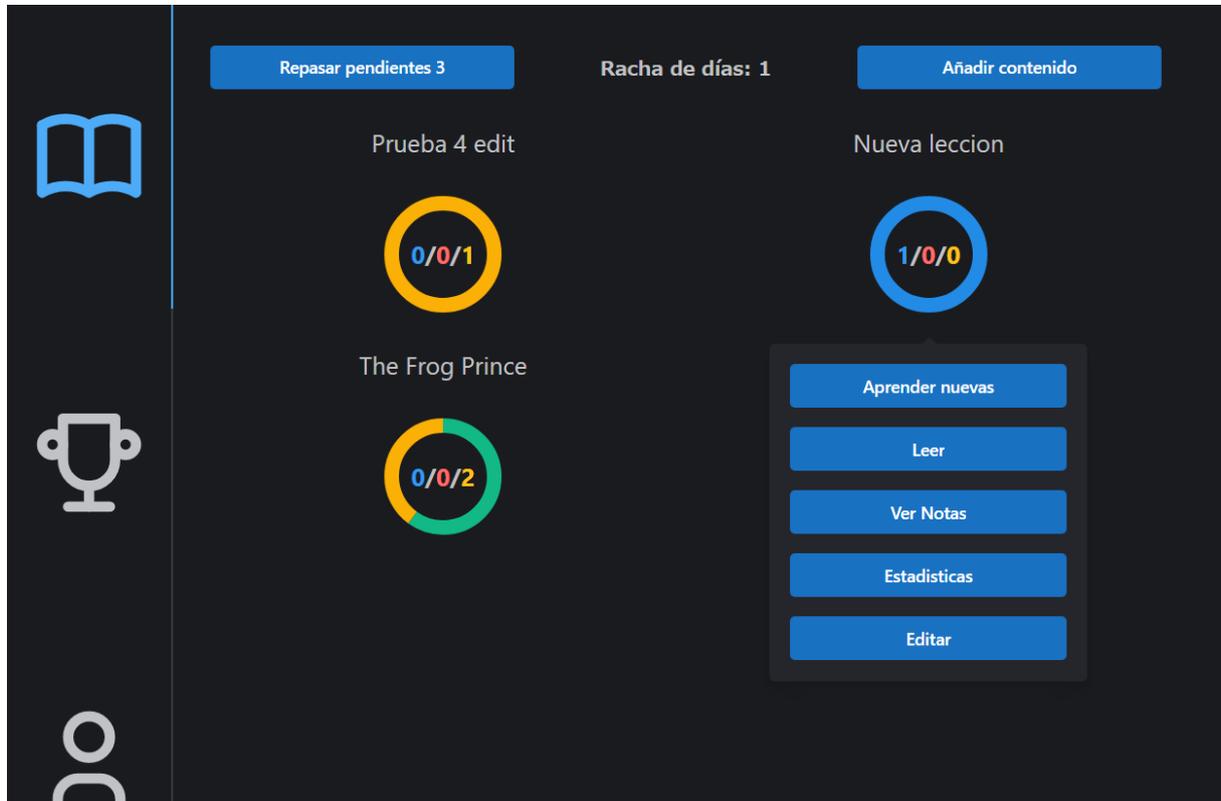
Finalmente, tenemos las preguntas que son pequeñas cuestiones sobre la lección que ayudan al alumno a evaluar su comprensión del texto. El primer campo es el enunciado de la pregunta, estas preguntas son de múltiple opción, por ello hay que rellenar las cuatro opciones posibles y marcar las casillas de las opciones que son correctas.



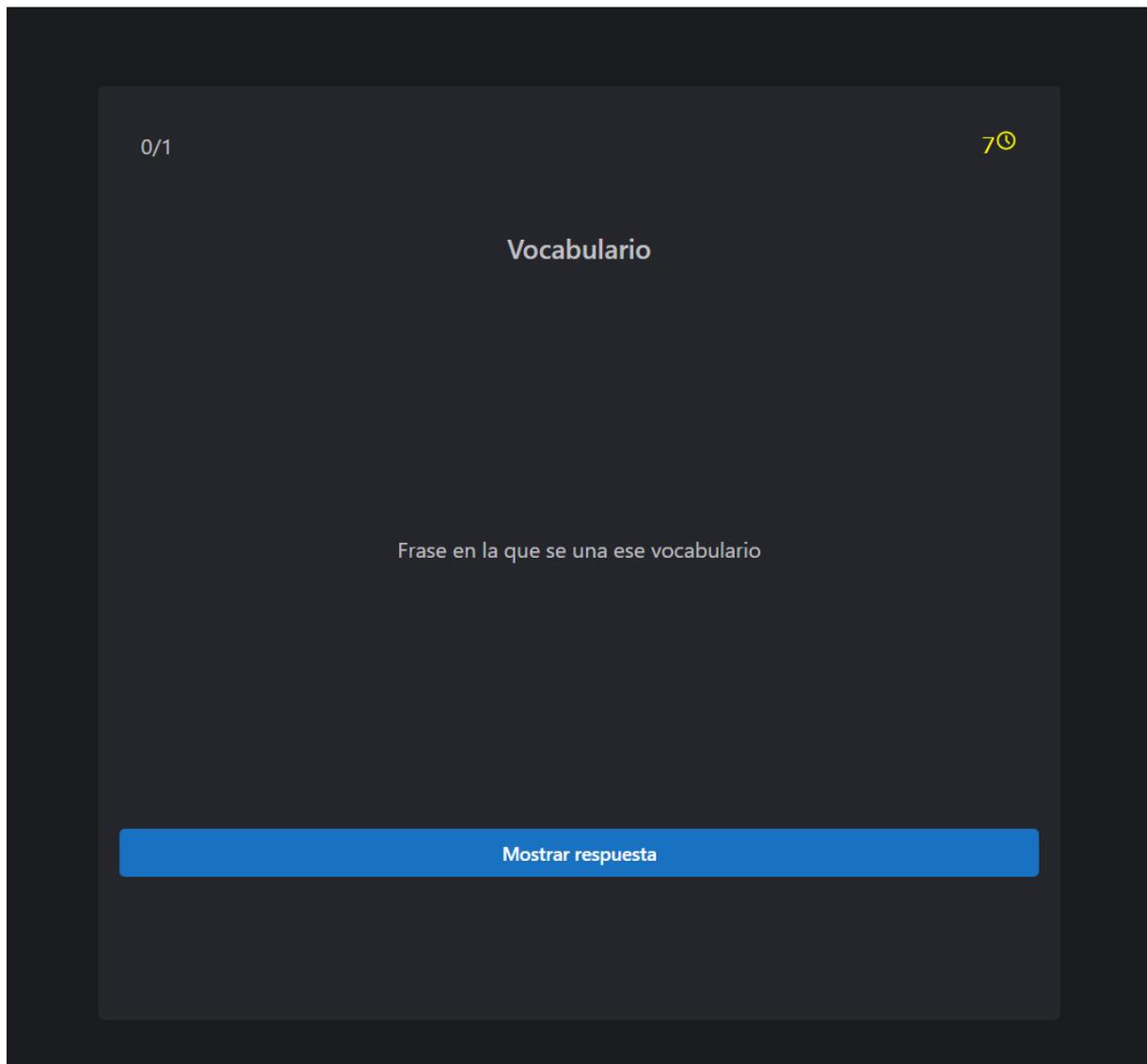
Una vez que hemos terminado podemos pulsar "Guardar" para conservar estos cambios o "Cancelar" para cancelarlos. En el caso de que estemos editando un mazo tendremos la opción de eliminarlos en vez del botón de cancelar.

A.3 Repasar una lección

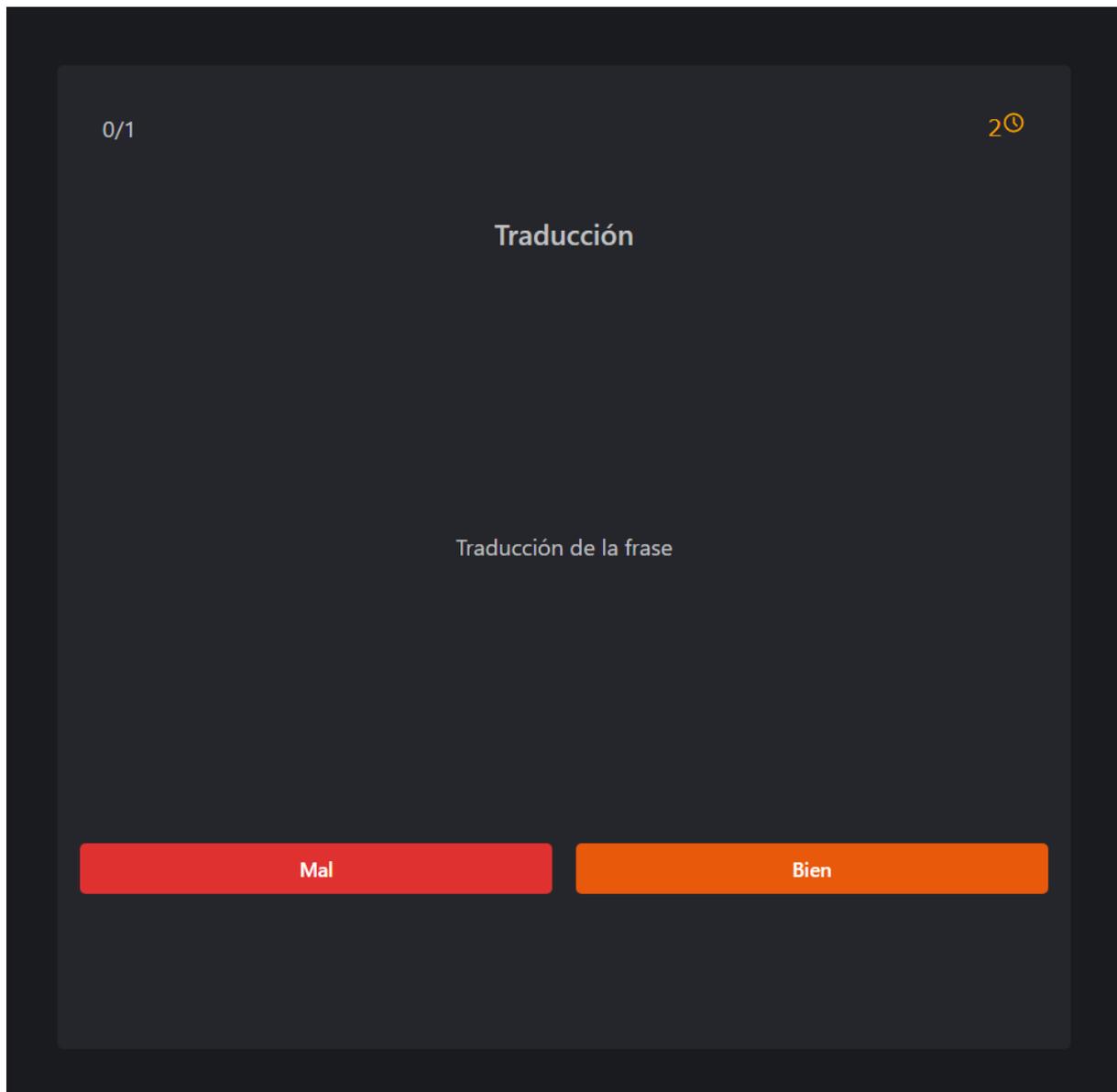
Una vez que tenemos mazos que repasar podemos acceder a los repasos pulsando el botón de repasar pendientes o si queremos aprender palabras nuevas de un mazo podemos pulsar sobre él y seleccionar "Aprender nuevas".



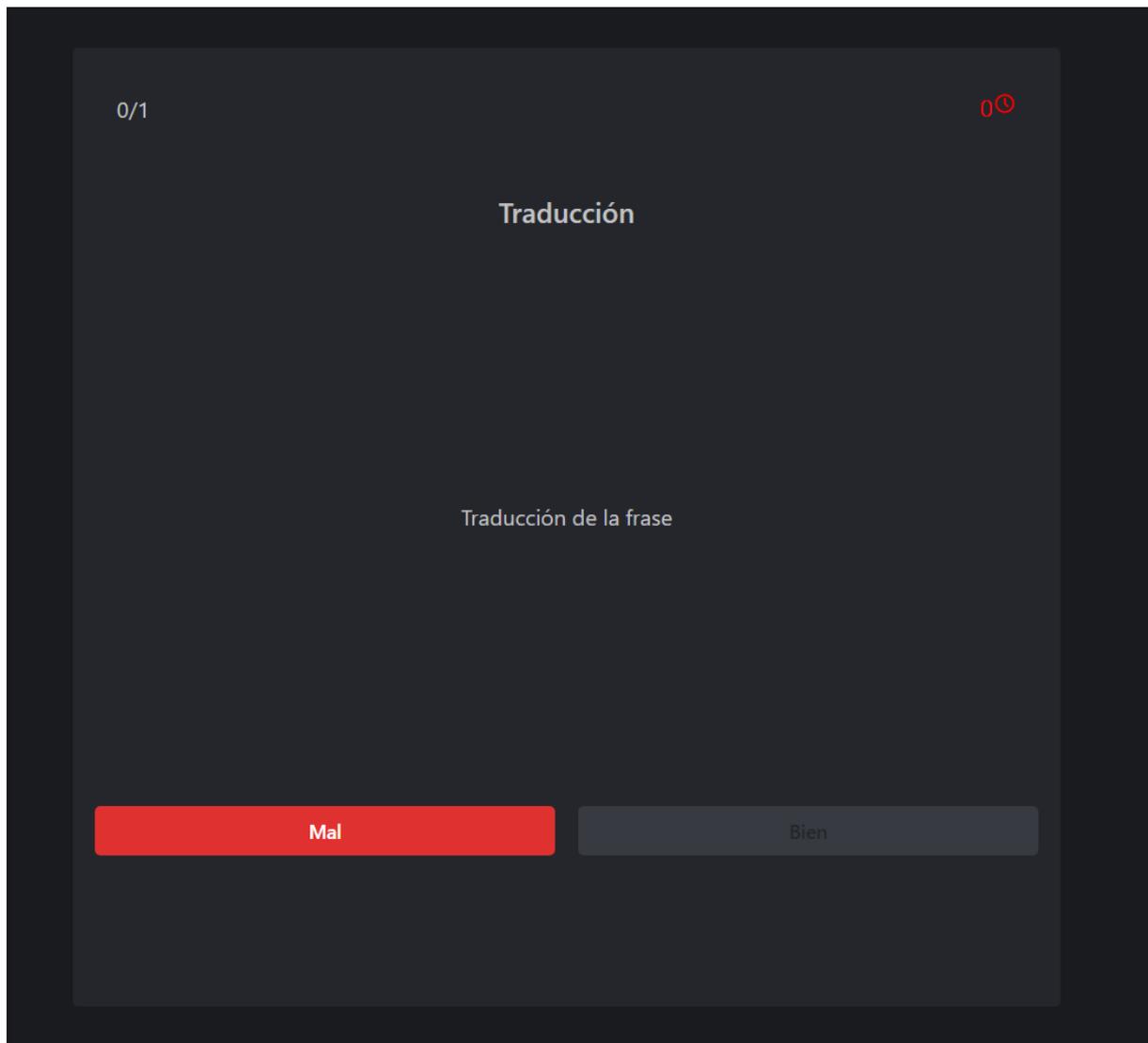
Una vez en la pestaña de repasos nos encontraremos con un temporizador que empieza a contar, antes de que se acabe el tiempo debes intentar recordar el significado de la tarjeta en pantalla y pulsar el botón "Mostrar respuesta"



Una vez que pulses el botón de "Mostrar respuesta" se mostrará el reverso de la tarjeta y podrás seleccionar "Mal" o "Bien" dependiendo de si has podido recordar bien el significado o no.



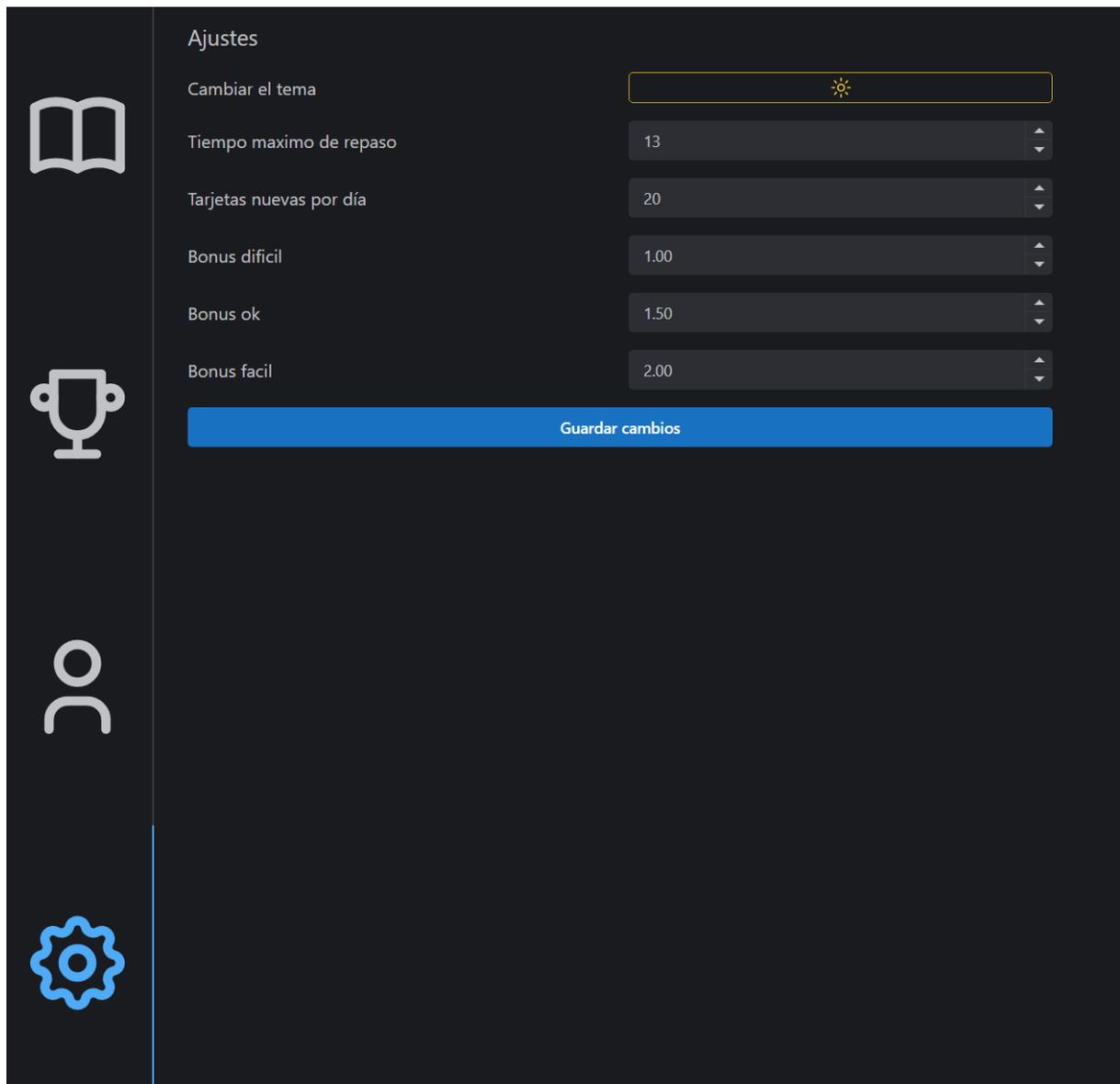
El color del botón de "Bien" indica lo bien que recuerdas la tarjeta y depende del tiempo que has tardado en contestar, esto se utiliza para decidir cuanto tiempo hay que esperar antes de volver a mostrar la tarjeta. Si te quedas sin tiempo te veras obligado a pulsar "Mal".



Este proceso se repite hasta que no queden más tarjetas por repasar o aprender.

A.4 Cambiar los ajustes

Existen varios ajustes que puedes modificar, para ello solo tienes que acceder al apartado de ajustes, indicado por un icono de un engranaje.



Cambiar el tema te permite alternar entre el modo claro o el modo oscuro de la aplicación.

Tiempo máximo de repaso es el tiempo con el que se inicializa el contador de los repasos.

Tarjetas nuevas por días es el límite de tarjetas que puedes aprender por día, es importante no aumentarlo mucho puesto que el efecto que tiene en el tiempo que tendrás que dedicar diariamente a los repasos es exponencial.

Los bonuses de difícil, ok y fácil indican como debería crecer el intervalo dependiendo de cómo de bien te sepas una tarjeta. En este caso el bonus de difícil es de 1 indicando que las tarjetas difíciles no aumentan su intervalo, sin embargo las tarjetas fáciles duplicarán su

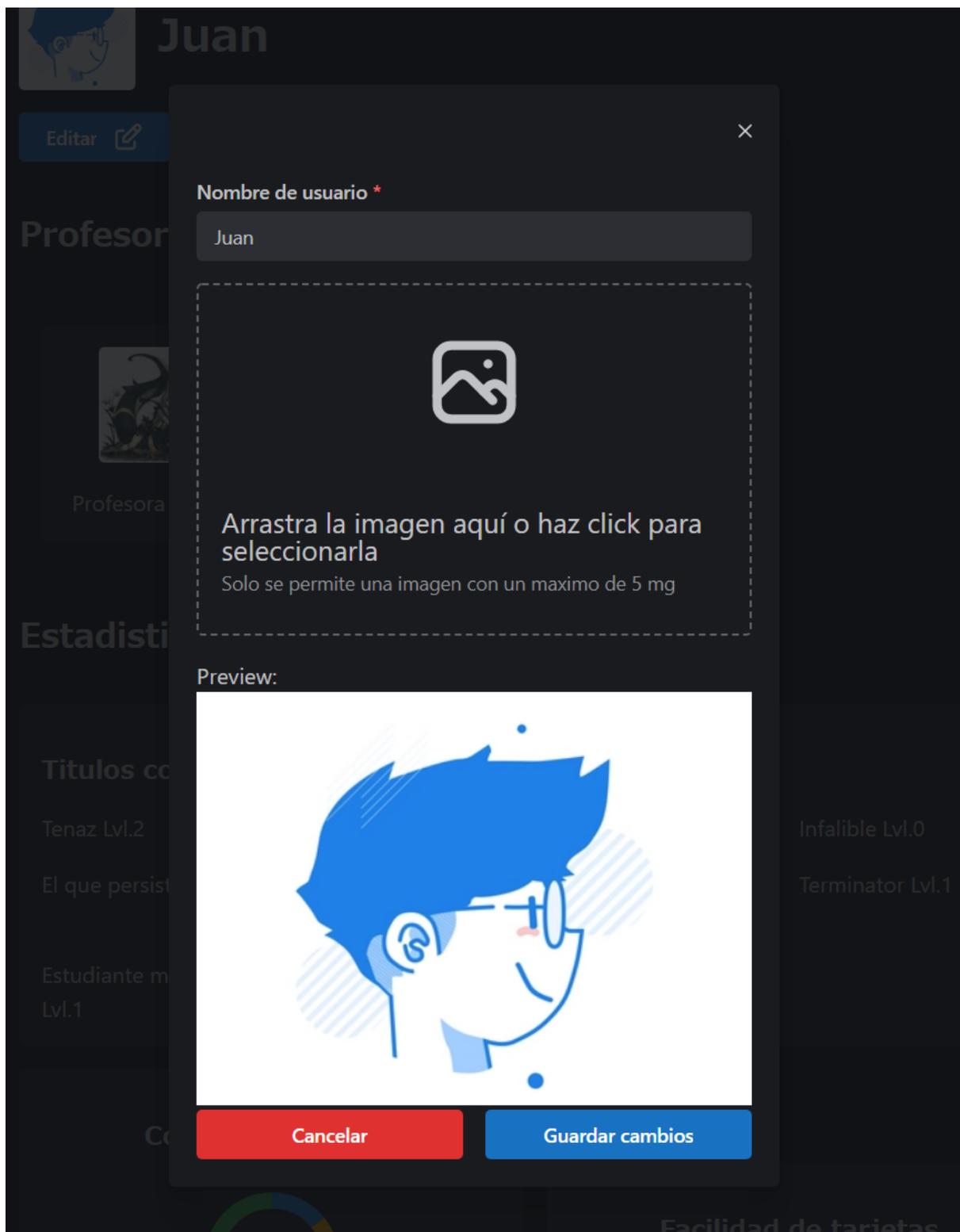
intervalo, es decir, si la repasaste hace 4 días y hoy la has acertado en el tiempo considerado como fácil no la volverás a ver hasta dentro de 8 días.

A.5 Editar perfil

Si quieres modificar tu perfil solo tienes que dirigirte al apartado de usuario indicado con un icono de una persona. Aquí podrás eliminar tu cuenta o cerrar sesión si quieres, además de consultar tus estadísticas y editar tu perfil pulsando el botón de editar.

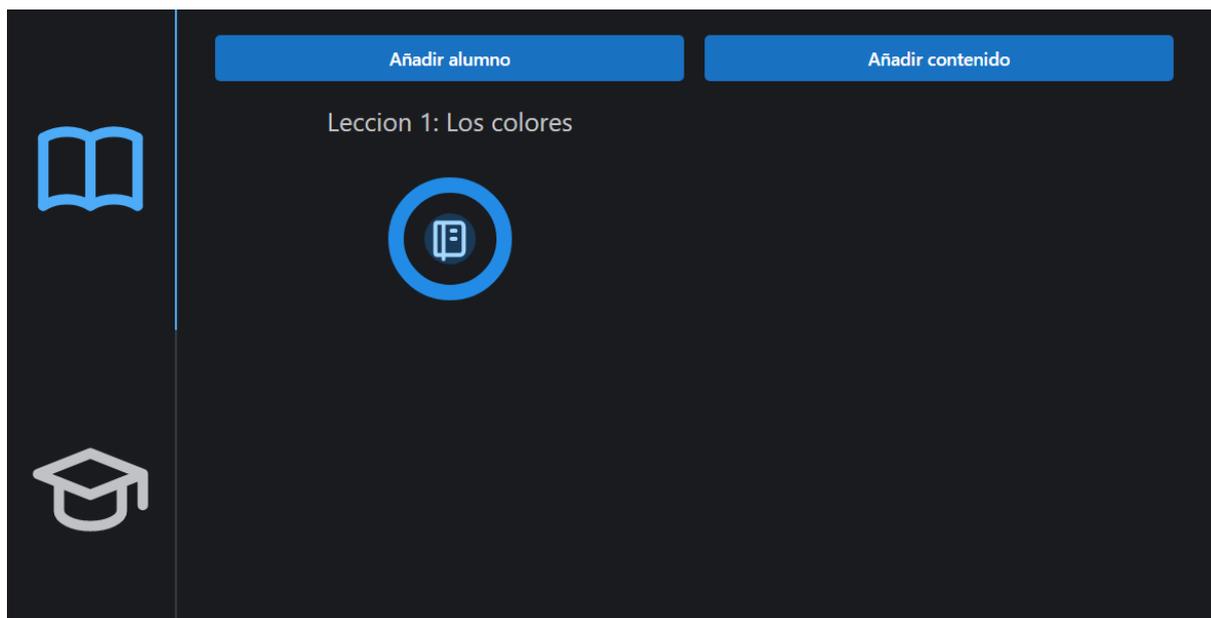
The screenshot shows a user profile interface for 'Juan'. On the left is a dark sidebar with icons for a book, a trophy, a person, and a gear. The main content area includes a profile header with a blue profile picture, the name 'Juan', and three buttons: 'Editar' (blue), 'Cerrar sesión' (blue), and 'Eliminar cuenta' (red). Below this is a 'Profesores' section with a card for 'Profesora Pamela' featuring a dragon illustration. The 'Estadísticas' section is divided into two parts: 'Títulos conseguidos' and two donut charts. The titles section lists: Tenaz Lvl.2, Erudito Lvl.1, Sabelo todo Lvl.1, Infalible Lvl.0, El que persiste Lvl.Max, Fotogenico Lvl.Max, Creador de conocimiento Lvl.2, Terminator Lvl.1, Estudiante modelo Lvl.1, and Empollon Lvl.2. The 'Conteo de tarjetas' chart shows: Nuevas: 1 (14.29%), Reaprendiendo: 0 (0.00%), Aprendiendo: 2 (28.57%), and Maduras: 4 (57.14%). The 'Facilidad de tarjetas' chart shows: Facil: 3 (100.00%), Bien: 0 (0.00%), Dificil: 0 (0.00%), and Mal: 0 (0.00%).

A la hora de editar tu perfil te permitimos modificar tu nombre de usuario y tu foto de perfil.

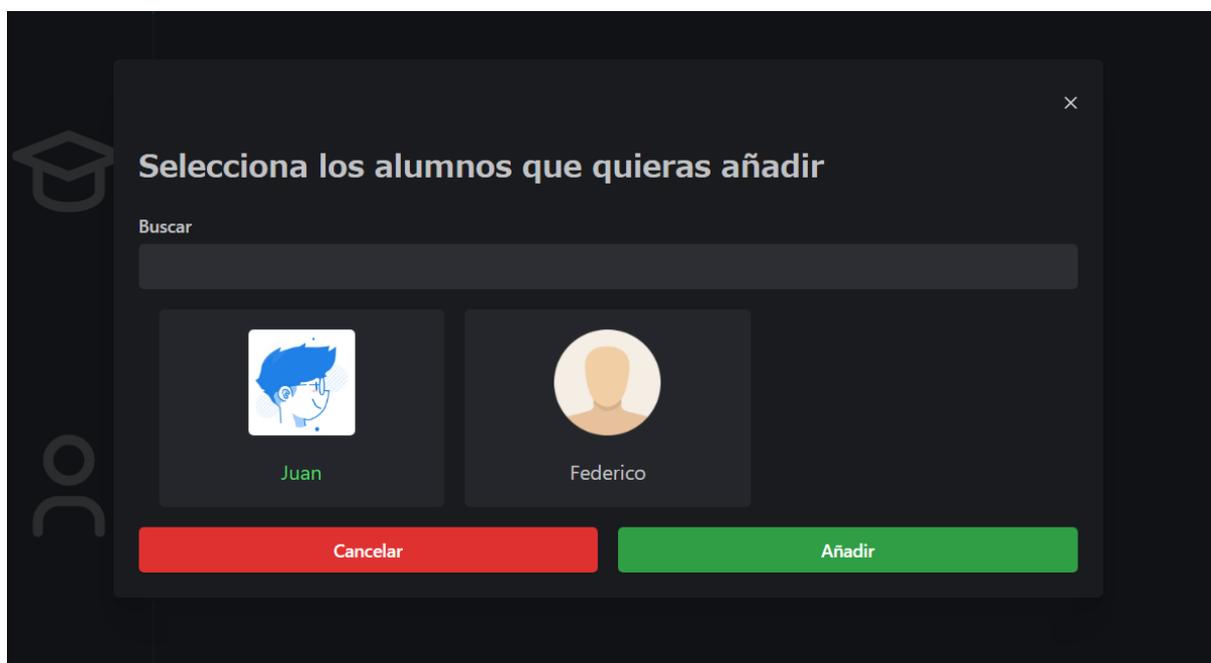


A.6 Añadir alumnos

Si eres un profesor y quieres añadir nuevos alumnos bajo tu tutelaje es tan sencillo como desde la página de inicio pulsar el botón "Añadir alumno", esto desplegará un menú con todos los alumnos disponibles.

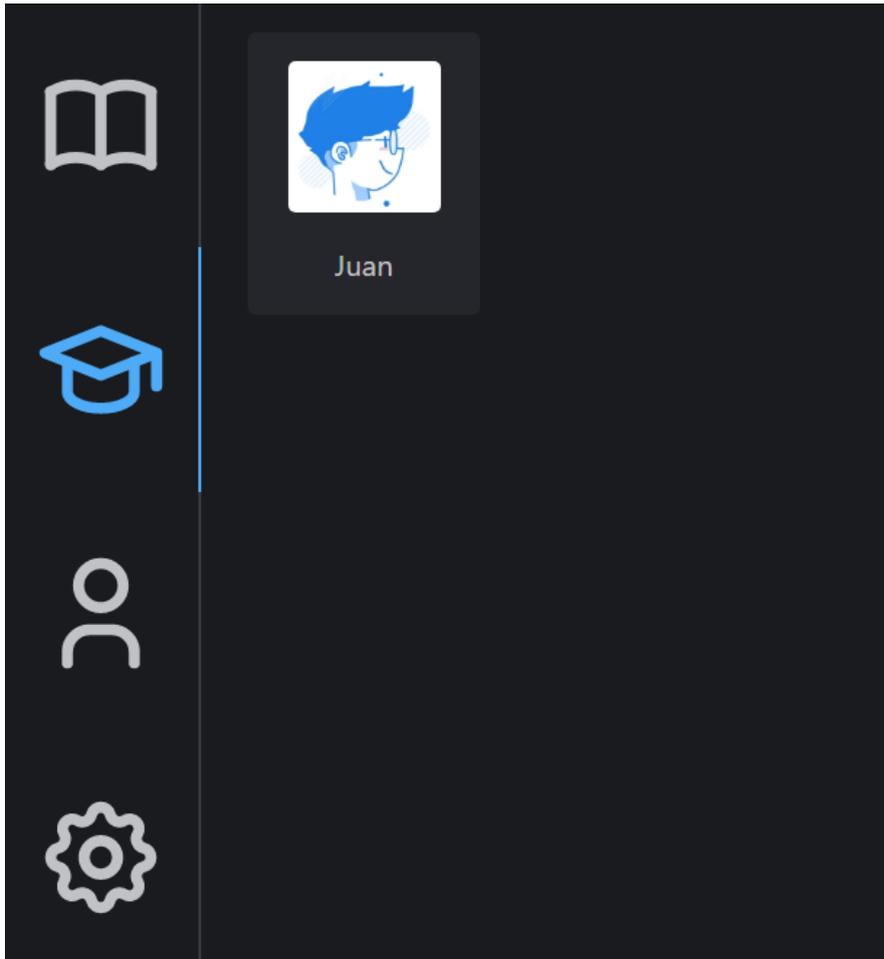


De esta lista puedes seleccionar a todos los alumnos que quieras añadir, además de que puedes buscar a un alumno concreto por nombre usando la barra de búsqueda. Una vez has seleccionado a los alumnos que quieras, que se mostraran con el nombre en verde, puedes pulsar añadir para añadirlos todos a la vez.

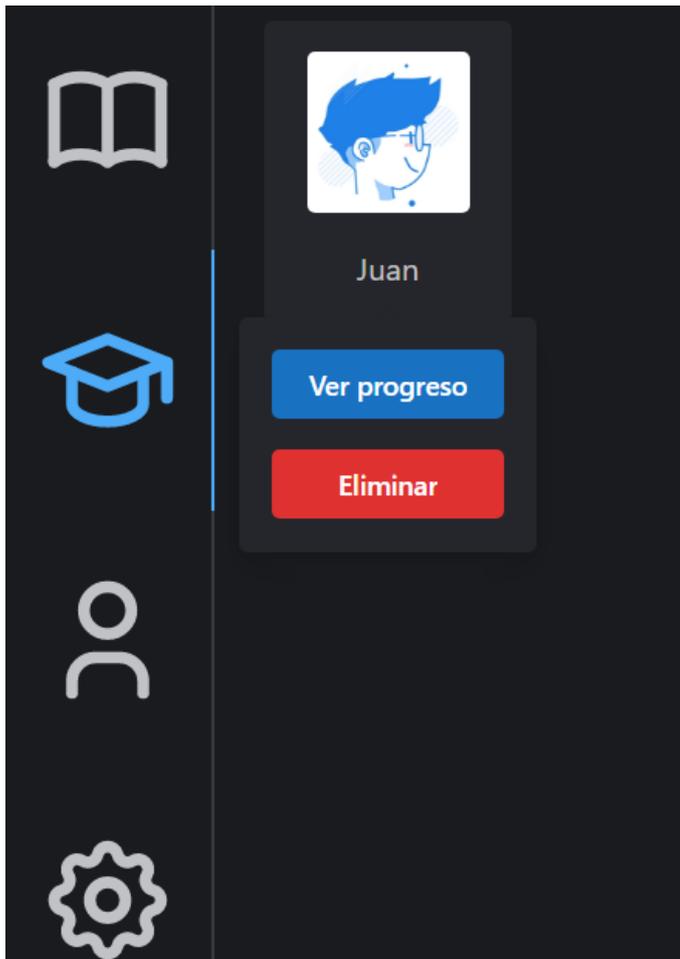


A.7 Eliminar alumnos

Si deseas eliminar alguno de tus alumnos, el proceso a seguir es muy sencillo. Lo primero será dirigirnos al apartado de alumnos, indicado con un icono de un birrete. En este apartado podremos ver la lista de nuestros alumnos.



Al seleccionar uno de ellos aparecerán dos opciones, la primera te permite ver el progreso de ese alumno en forma de sus estadísticas, y la segunda te permite eliminar al alumno.



Si no tienes ningún alumno se mostrará el siguiente mensaje

