

Diseño de una plataforma de aprendizaje basada en tarjetas de memoria

Memoria de Proyecto Final de Máster

Máster Universitario en Desarrollo de sitios y aplicaciones web

Autor: Bruno Orcha García

Consultor: Miguel Calvo Matalobos

Profesor: César Pablo Córcoles Briongos

7 de Junio de 2021

Créditos/Copyright

Licencias



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Abstract

El objetivo de este proyecto es el desarrollo y documentación de una plataforma de aprendizaje basada en tarjetas de memoria. Esta aplicación web permitirá publicar un temario y un conjunto de tarjetas de tipo pregunta/respuesta asociado. Los alumnos realizarán sesiones de aprendizaje mediante el método de repetición espaciada (Leitner), hasta conseguir un determinada tasa de acierto para poder completar todo el temario.

Palabras clave: Plataforma aprendizaje, Tarjetas, Leitner, Repetición Espaciada

Abstract (english version)

The objective of this project is the development and documentation of a learning platform based on memory cards. This web application will allow to publish a syllabus and a set of associated question / answer cards. The students will carry out learning sessions using the spaced repetition method (Leitner), until they achieve a certain rate of success to be able to complete the entire syllabus.

Keywords: learning platform, memory cards, Leitner, spaced repetition

Índice

1	Introducción.....	9
1.1	Contexto y justificación del trabajo.....	9
2	Objetivos.....	11
2.1	Objetivos generales.....	11
2.2	Objetivos específicos.....	11
2.3	Enfoque y método seguido.....	11
3	Planificación del Trabajo.....	12
4	Análisis de mercado y viabilidad.....	15
4.1	Aplicaciones relacionadas.....	15
4.2	Viabilidad.....	17
5	Definición de Personas.....	19
6	Funcionalidades.....	22
6.1	Alta/baja de alumnos.....	22
6.2	Seguimiento de alumnos.....	23
6.3	Gestionar temario.....	23
6.4	Gestionar tarjetas.....	24
6.5	Repaso del alumno.....	25
7	Contenidos.....	26
7.1	Elementos comunes.....	26
7.2	Menú de administradores.....	27
7.3	Menú de usuario.....	28
8	Prototipos.....	30
8.1	Módulo de administración.....	30
8.2	Módulo del alumno.....	34
9	Arquitectura de la aplicación.....	38
9.1	Backend.....	38
9.2	Frontend.....	39
9.3	Base de datos.....	40
10	Plataforma de desarrollo.....	42
10.1	Software.....	42
10.2	Hardware.....	46

11 APIs utilizadas.....	47
11.1 Cloudinary.....	47
12 Usabilidad/UX.....	48
12.1 HomeLink.....	48
12.2 Fat Footer.....	48
12.3 Morphing Controls.....	49
12.4 Navigations Tabs.....	49
12.5 Paginación.....	49
12.6 Input Feedback.....	50
12.7 Completeness meter.....	50
12.8 Table Filter.....	51
12.9 Wysiwyg.....	51
13 Seguridad.....	52
14 Instrucciones de instalación.....	54
14.1 Construcción con Docker.....	54
14.2 Construcción manual.....	55
14.3 Despliegue de kaeme.site.....	56
15 Proyección a futuro.....	59
16 Conclusiones.....	61
17 Anexo 1. Entregables del proyecto.....	62
18 Anexo 2. Código fuente (extractos).....	63
19 Anexo 3. Librerías/Código externo utilizado.....	66
19.1 svg-heatmap.....	66
19.2 Trix Editor.....	66
20 Anexo 4. Guía de usuario.....	67
20.1 Guía de Profesores.....	67
20.2 Guía del alumno.....	78
21 Anexo 5. Bibliografía.....	81

Figuras y tablas

Lista de imágenes, tablas, gráficos, diagramas, etc., numeradas, con títulos y las páginas en las que aparecen.

Índice de figuras

Figura 1: Tarjetas para memorización.....	9
Figura 2: Método Leitner.....	10
Figura 3: Duolingo.....	15
Figura 4: Memrise.....	16
Figura 5: Test DGT.....	16
Figura 6: Test Patrón de Recreo - PER.....	16
Figura 7: Anki.....	17
Figura 8: Modelo de suscripciones.....	18
Figura 9: Personas.....	19
Figura 10: Mapa de historias de usuario.....	22
Figura 11: Elementos comunes del menú.....	26
Figura 12: Elementos del menú de administración.....	27
Figura 13: Elementos del menú de los usuarios/alumnos.....	28
Figura 14: Pantalla de inicio de la administración.....	30
Figura 15: Listado de alumnos.....	31
Figura 16: Creación/Edición de alumnos.....	31
Figura 17: Seguimiento del alumno.....	32
Figura 18: Organización del temario.....	32
Figura 19: Editar tema.....	33
Figura 20: Listado de tarjetas.....	33
Figura 21: Listado de tarjetas.....	34
Figura 22: Activación del alumno.....	34
Figura 23: Pantalla de progreso y de inicio.....	35
Figura 24: Consulta del temario del alumno.....	35
Figura 25: Consulta de un tema por el alumno.....	36
Figura 26: Sesión de aprendizaje del alumno.....	36
Figura 27: Confirmación de superación de un tema.....	37
Figura 28: Arquitectura básica de la aplicación backend/frontend.....	38
Figura 29: Esquema de base de datos con sus relaciones.....	41
Figura 30: Diagrama con librerías usadas en el backend.....	42
Figura 31: Diagrama con librerías usadas en el frontend.....	44
Figura 32: Botón inicio.....	48
Figura 33: Menú inferior.....	48
Figura 34: Botón guardar con cambio de estado.....	49

Figura 35: Componente de pestañas.....	49
Figura 36: Botones de paginación.....	50
Figura 37: Mostrar errores en el formulario.....	50
Figura 38: Barras de progreso.....	50
Figura 39: Filtro de opciones desplegable.....	51
Figura 40: Componente de edición de texto enriquecido.....	51
Figura 41: Página de la aplicación en fly.io.....	56
Figura 42: Página de configuración en namecheap.....	57
Figura 43: Página de configuración en cloudflare.....	58
Figura 44: Niveles de aprendizaje de las tarjetas.....	63
Figura 45: Probabilidad de elección de tarjetas según su nivel.....	63
Figura 46: Proceso de actualización del progreso según Leitner.....	64
Figura 47: Comentarios de código de la implementación Leitner.....	65
Figura 48: Mapa de calor de actividad.....	66
Figura 49: Componente Trix Editor.....	66
Figura 50: Pantalla de selección del módulo alumno/profesor.....	67
Figura 51: Pantalla de acceso de usuarios administradores/profesores.....	68
Figura 52: Pantalla de inicio de profesores.....	69
Figura 53: Listado de usuarios/profesores.....	70
Figura 54: Listado de usuarios/profesores.....	70
Figura 55: Administración del temario.....	71
Figura 56: Edición de un tema/capítulo.....	72
Figura 57: Listado de tarjetas.....	73
Figura 58: Edición de tarjetas.....	74
Figura 59: Listado de alumnos.....	74
Figura 60: Edición de alumnos.....	75
Figura 61: Consulta/seguimiento de alumnos.....	76
Figura 62: Progreso del temario del alumno.....	77
Figura 63: Edición de anotaciones y observaciones del alumno.....	77
Figura 64: Pantalla de activación del alumno.....	78
Figura 65: Pantalla de inicio del alumno.....	78
Figura 66: Consulta del temario del alumno.....	79
Figura 67: Consulta del temario del alumno.....	79
Figura 68: Ejecución de una sesión.....	80
Figura 69: Resumen de la sesión realizada.....	80

1 Introducción



Figura 1: Tarjetas para memorización

1.1 Contexto y justificación del trabajo

El aprendizaje es una de las actividades que requieren mayor esfuerzo y concentración para poder asimilar la información y nuevos conceptos que queremos aprender. Existe bastante literatura sobre el análisis del aprendizaje y los distintos tipos en las que se puede clasificar estos procesos. No todos los procesos son iguales de efectivos ni todas las personas aprenden de la misma manera. Existen numerosos factores que pueden hacer que un proceso de aprendizaje sea más óptimo o no, profesores, estudiantes, la forma de enseñar y explicar, los recursos y materiales disponibles, la estructura y orden de los contenidos, etc.

El auge de las nuevas tecnologías y del acceso a internet, ha aumentado la accesibilidad a fuentes de información de todo tipo, permitiendo la creación de nuevos sistemas de aprendizaje. Sin embargo, como se apuntaba anteriormente, disponer de la información en bruto sin más, no suele ser la mejor estrategia para que el aprendizaje se produzca de forma efectiva. Es por eso que existen distintos modelos y técnicas que estructuran esta información para facilitar a los alumnos la asimilación de los conceptos de forma progresiva.

El concepto e-learning hace referencia a todos estos nuevos métodos, canales y herramientas que permiten el aprendizaje mediante el uso de la tecnología e internet. Para dar soporte a esto aparecen sistemas de información especializados en este campo, son los Learning Management Systems (LMS) que permiten la administración, creación de documentación, registro, puesta a disposición de cursos, seguimiento del progreso de alumnos y profesores. Algunos ejemplos conocidos de estos sistemas son los proyectos open source Moodle, Chamilo, o plataformas propietarias como Udemy, Coursera, o la propia plataforma de la UOC.

Además de estos sistemas de propósito más general y clásico, existen otros productos más específicos a temas muy concretos que utilizan otras técnicas de aprendizaje distintas como pueden ser el caso de las

aplicaciones de aprendizaje de idiomas como Memrise o Babbel. Estos sistemas suelen utilizar la técnica de aprendizaje repetitivo: repetición espaciada.

Esta técnica consiste en mostrar de forma recurrente piezas de información (normalmente en formato pregunta/respuesta). En entornos no electrónicos se suelen utilizar tarjetas, escribiendo en la pregunta en una cara y la respuesta en la otra. Según el usuario vaya acertando estas preguntas, aparecerán nuevas tarjetas con nueva información a memorizar. Las preguntas difíciles (con tasa de error mas alta) aparecerán mas frecuentemente que las tarjetas mas fáciles (con tasa de error menor).

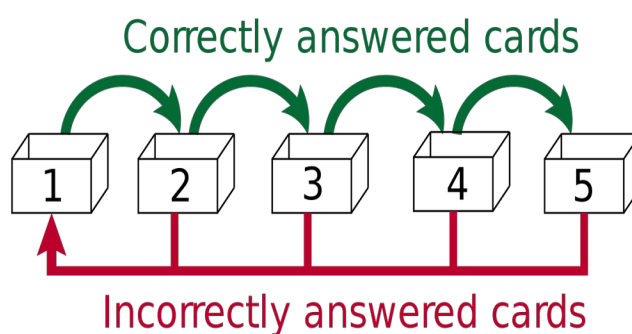


Figura 2: Método Leitner

Existen otros nichos de aprendizaje donde este tipo de sistemas encajarían adecuadamente, sobre todo los que están destinados a ser evaluados mediante exámenes o pruebas tipo test. Es por esto que se propone la construcción de uno de estos sistemas como posible proyecto viable, utilizando los conocimientos adquiridos durante la realización del máster.

2 Objetivos

El objetivo principal por tanto es el análisis e implementación de un sistema e-learning específico utilizando la técnica de aprendizaje repetitivo, basándose en el sistema de Leitner. En este sistema propuesto en los años 70 por un periodista científico alemán del mismo apellido, se clasifican las piezas de conocimientos o tarjetas de memorización en 5 niveles de aprendizaje. De esta forma, cada vez que en una sesión de aprendizaje se acierten las preguntas, estas pasarán al siguiente nivel. Conforme se responda satisfactoriamente a las preguntas, pasarán al siguiente nivel y por tanto la probabilidad de volver a preguntar estas tarjetas en las siguientes sesiones irán decrementándose. Por contra, las preguntas que se falle, volverán al nivel inicial, y tendrán que seguir siendo respondidas satisfactoriamente para conseguir que todas las preguntas estén en el último nivel de aprendizaje, dándose por completado la memorización del tema.

2.1 Objetivos generales

- Aplicar los conocimientos adquiridos durante la realización del Master en la confección de un proyecto real.
- Adquirir experiencia con las herramientas, entornos y tecnologías utilizadas para su desarrollo.
- Crear un producto mínimo viable (MVP, Minimal Viable Product) que permita evaluar el modelo de negocio.

2.2 Objetivos específicos

- Diseñar una aplicación mínima que permita definir un temario y tarjetas de aprendizaje asociadas.
- Creación de una aplicación web que permita a los alumnos memorizar distintos temas o bloques de información usando sus dispositivos móviles.
- Permitir administradores crear contenido, además de gestionar el registro y almacenamiento del progreso de los alumnos.

2.3 Enfoque y método seguido

Una vez elegido la temática de la aplicación y el nicho de aprendizaje, se intentará delimitar el alcance del proyecto definiendo un conjunto de funcionalidades abarcables para la dedicación disponible, y se realizará análisis en el que se definirán los requisitos a implementar. En base a la experiencia propia tanto de haber utilizado anteriormente estos sistemas de aprendizaje de idiomas, y de haber estado intentando aprender de manera independiente el temario del PER (Título náutico de Patrón de Embarcaciones de Recreio), se realizará la planificación de las tareas para la creación de un producto que permita aplicar este tipo de aprendizaje sobre un caso real.

3 Planificación del Trabajo

Para la elaboración de la planificación se ha utilizado el programa Gantt Project para generar el diagrama de Gantt en el que se han incluido las entregas propuestas, y se han desglosado los principales tareas a realizar para la consecución de los hitos.

PEC1	17/02/2021	02/03/2021	
Planificación	17/02/2021	02/03/2021	15 horas
Documentación inicial	17/02/2021	02/03/2021	25 horas
<i>Entrega PEC1</i>	02/03/2021	02/03/2021	
		<i>Subtotal</i>	<i>40 horas</i>
PEC2	03/03/2021	31/03/2021	
Análisis de mercado y viabilidad	03/03/2021	31/03/2021	10 horas
Estudio de arquitectura	03/03/2021	31/03/2021	15 horas
Estudio de usabilidad	03/03/2021	31/03/2021	5 horas
Definición de REQ y CU del backend	03/03/2021	31/03/2021	10 horas
Definición de REQ y CU del frontend	03/03/2021	31/03/2021	10 horas
Análisis y diseño del backend	03/03/2021	31/03/2021	20 horas
Análisis y diseño del frontend	03/03/2021	31/03/2021	20 horas
<i>Entrega PEC2</i>	31/03/2021	31/03/2021	
		<i>Subtotal</i>	<i>90 horas</i>
PEC3	01/04/2021	09/05/2021	
Revisión de la documentación	01/04/2021	09/05/2021	5 horas
Backend	01/04/2021	09/05/2021	20 horas
Gestión del contenido	01/04/2021	09/05/2021	14 horas
Gestión y registro de usuarios	01/04/2021	09/05/2021	14 horas
Frontend	01/04/2021	09/05/2021	20 horas
Registro y autenticación de usuarios	01/04/2021	09/05/2021	14 horas
Consulta del temario/itinerario	01/04/2021	09/05/2021	14 horas
Ejecución de sesiones de aprendizaje	01/04/2021	09/05/2021	20 horas
Despliegue en entorno para pruebas	01/04/2021	09/05/2021	5 horas
Elaboración del video de demostración	01/04/2021	09/05/2021	4 horas
<i>Entrega PEC3</i>	10/05/2021	10/05/2021	
		<i>Subtotal</i>	<i>130 horas</i>

Entrega Final	10/05/2021	07/06/2021	
Revisión de la aplicación y despliegue final	10/05/2021	07/06/2021	35 horas
Revisión final de la documentación	10/05/2021	07/06/2021	25 horas
Presentación o landing de la aplicación	10/05/2021	07/06/2021	20 horas
Elaboración del video del proyecto	10/05/2021	07/06/2021	5 horas
Elaboración del autoinforme	10/05/2021	07/06/2021	4 horas
<i>Entrega PEC4</i>	07/06/2021	07/06/2021	
		<i>Subtotal</i>	89 horas
		Total	348 horas

4 Análisis de mercado y viabilidad

Durante el estudio de diversas materias a lo largo de tiempo, me he visto en la necesidad de utilizar distintas técnicas con el fin de aprender distintos temas o prepararme para la realización de exámenes o pruebas. Existen numerosas técnicas que nos ayudan a asimilar información diversa, como pueden ser en textos de gran extensión el subrayado de las ideas claves o la realización directamente de resúmenes, para su posterior lectura recurrente.

Una de las técnicas más extendidas para el estudio y repaso es la confección de tarjetas. Mediante el uso de tarjetas de papel o cartulina, se escriben por una de sus caras un término o idea clave, y por la parte posterior su definición o explicación. El propósito es realizar sesiones de estudio donde se van sacando tarjetas de forma sucesiva, y se intenta recordar su parte posterior. Como se comentó anteriormente existen técnicas y estudios que permiten optimizar este proceso para maximizar el proceso de aprendizaje, como es el sistema de Leitner o de repaso espaciado.

Este proceso iterativo es mucho más apropiado cuando nuestro objetivo es la superación de algún examen o prueba tipo test. Si disponemos de un conjunto de preguntas y sus correspondientes posibles respuestas, podemos practicar de forma más efectiva para superar la prueba.

4.1 Aplicaciones relacionadas

En el mercado podemos encontrar páginas y aplicaciones móviles centradas en nichos concretos, como por ejemplo el aprendizaje de idiomas (Memrise, Duolingo) o la realización de test de autoescuelas.

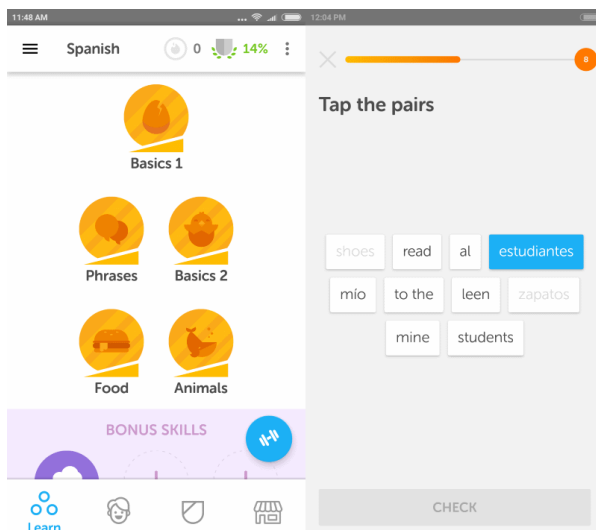


Figura 3: Duolingo

Duolingo

Esta es una aplicación de aprendizaje de idiomas. Dispone de distintos idiomas para aprender, para cada cuál gestiona distintos niveles y un conjunto de contenido (básico, frases, vocabulario, etc).

La aplicación controla el progreso en cada una de estas secciones y niveles para seguir desbloqueando los siguientes.

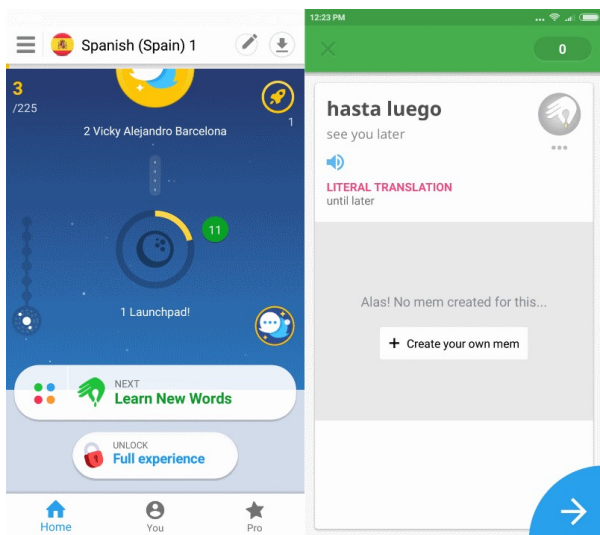


Figura 4: Memrise

Memrise

Otra de las aplicaciones mas populares de aprendizaje de idiomas. Muy similar a la anterior, aunque desarrolla distintos tipos de ejercicios, haciendo un uso mas intensivo de los elementos audiovisuales.

En la pantalla de la izquierda podemos ver un ejemplo del progreso del usuario, donde vemos el nivel de completitud del tema, así como información de los siguientes temas.

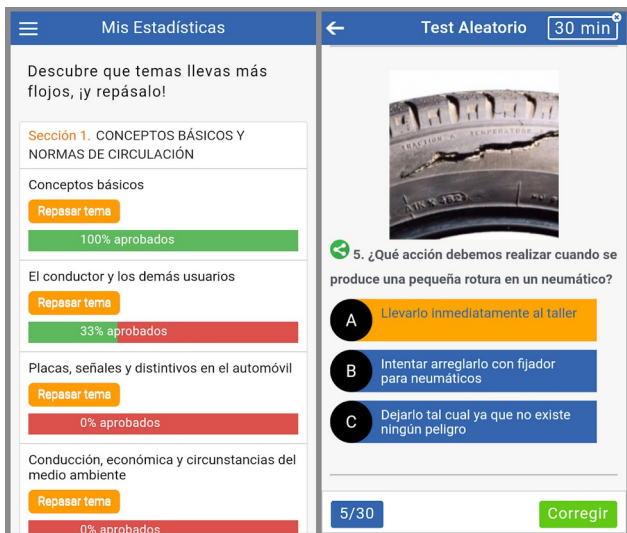


Figura 5: Test DGT

Test DGT

Es una aplicación muy descargada en la tienda de aplicaciones Android destinada a la realización de pruebas y test para la preparación del examen de conducir.

Como en la mayoría de los casos podemos observar de nuevo el mismo esquema donde tenemos secciones y sesiones de aprendizaje y la contabilización del progreso.

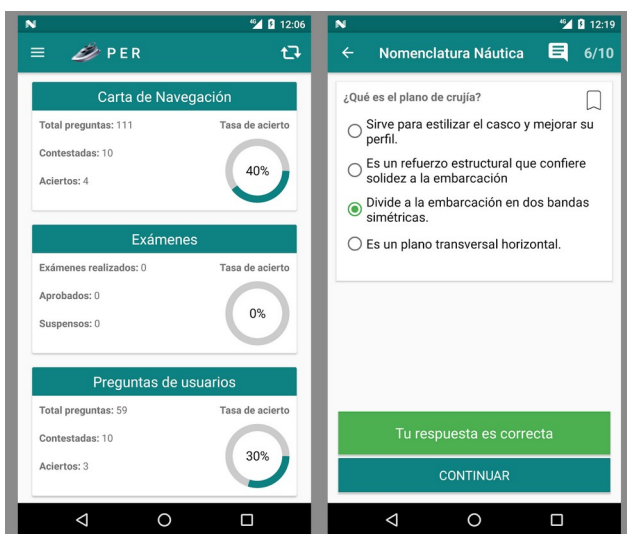


Figura 6: Test Patrón de Recreo - PER

Test Patrón de Recreo – PER

Otra aplicación similar a la anterior solo que enfocado a otro nicho distintos, en este caso, la preparación del examen de Patrón de embarcaciones de Recreo.

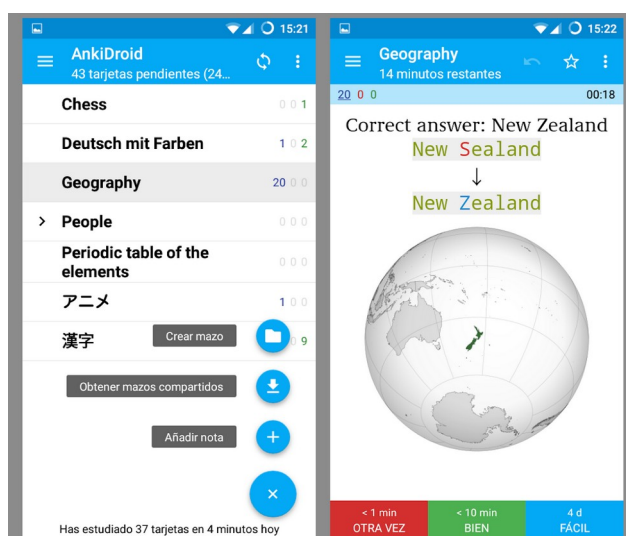


Figura 7: Anki

Anki

Una de las aplicaciones que mas explora este sistema de repetición espaciada es Anki. Además de aplicación móvil dispone de aplicación de escritorio y sincronización web. Permite crear mazos de tarjetas de distintos tipos y poder realizar sesiones.

En la gran mayoría de los casos, estamos ante desarrollos a medida que implementan casuísticas muy concretas para el dominio de aprendizaje que se está tratando. En el caso de idiomas se hace uso de actividades audiovisuales. En otros casos, se intenta intercalar elementos de estudio y repaso con preguntas reales de exámenes. Todas estas aplicaciones y páginas suelen tener un gran componente de gamificación, pensada para motivar al usuario, mostrándole en todo momento gráficas, porcentajes o textos encaminados a retarle, para causar la implicación del usuario.

4.2 Viabilidad

En el caso de este proyecto en particular, no se trataría de un encargo tradicional propuesto por un cliente, si no mas bien un desarrollo que intenta explorar la existencia de un nicho de mercado, que puede llegar a ser una oportunidad de negocio. Se trata por tanto, de evaluar como producto viable la creación de una plataforma que permita a distintas entidades de formación, como academias de distintos tipos, complementar su contenido docente mas tradicional en base a libros, apuntes o plataformas como Moodle, con una herramienta de refuerzo, que usando la técnica de repaso espaciado. Además, pensada para ser usada en sus dispositivos móviles, lo que permitiría a sus alumnos practicar con en cualquier lugar y momento de forma sencilla.

Algunos ejemplos donde este producto podría ser de utilidad podrían ser el caso de autoescuelas, academias para la preparación de exámenes de certificaciones, oposiciones, o pruebas que sigan un formato de pruebas de tipo test. Los administradores de las academias podrán mantener el temario y las preguntas que se adapten al estudio o prueba que se van a preparar de forma sencilla, y permitirá realizar un seguimiento del progreso de sus alumnos.

Además, el producto podría enfocarse como un SaaS (Software as a Service), donde la aplicación desarrollada soportase el sistema de multientidad, donde cada se le ofrecería el servicio personalizado a cada uno de los clientes (academias) y estos tendrían libertad de ofrecérselos a sus clientes de la forma que consideren. Es decir, el SaaS tendría un conjunto de planes destinados a las academias, como puede ser una suscripción anual o un precio por alumno. Pero a su vez, la propia plataforma podría permitir a las academias definir su propio modelo para sus alumnos (contenido temporal gratuito, pago por módulos, suscripción, etc).

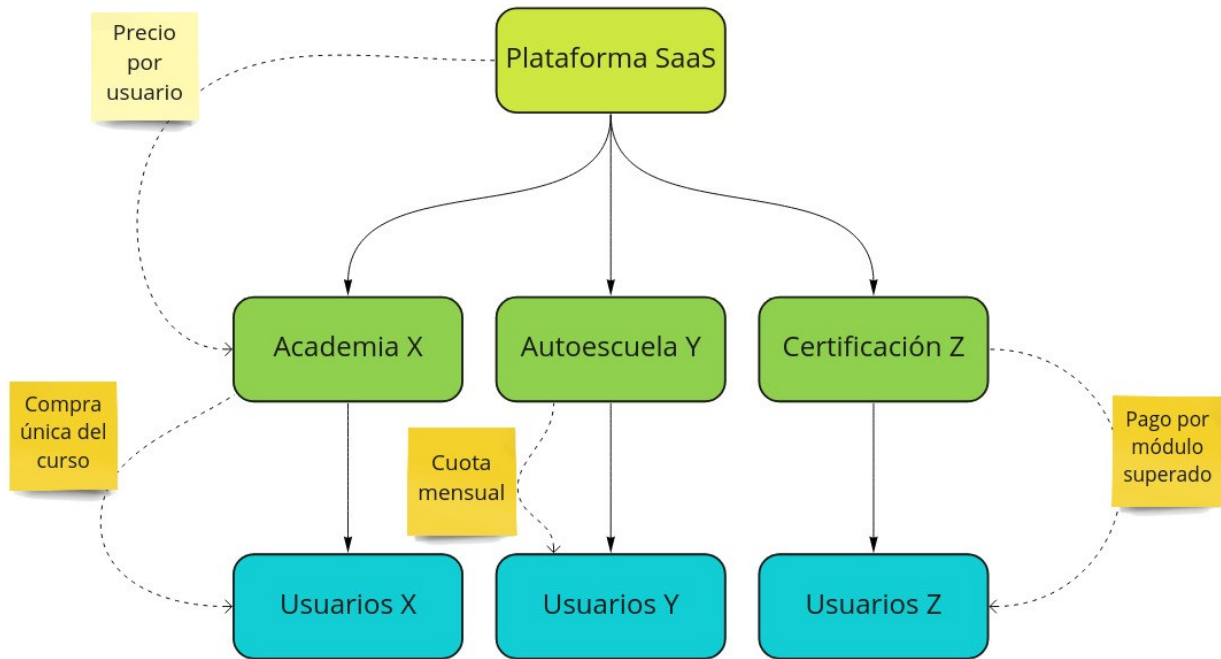


Figura 8: Modelo de suscripciones

5 Definición de Personas

Con el fin de concretar los tipos de usuarios o perfiles que estarían involucrados en el sistema anteriormente expuesto, a continuación se muestran dos fichas correspondientes a estos usuarios siguiendo la metodología de Personas.

El sistema constaría de dos partes bien diferenciadas. Una destinada a nuestro primer tipo de usuario, que serían los administradores o responsables de la entidad que va a crear el contenido de las tarjetas. Siguiendo con los ejemplos expuestos, estos usuarios serían los correspondientes a los profesores de las academias. Además de crear y mantener el contenido, también gestionarían su conjunto de alumnos.

Por tanto, el segundo tipo de usuarios que nos encontramos serían los propios alumnos de la academia. Una vez la academia los registre en el sistema, estos tendrán acceso al contenido creado por los profesores o administradores, y podrán comenzar a realizar las sesiones de aprendizaje.

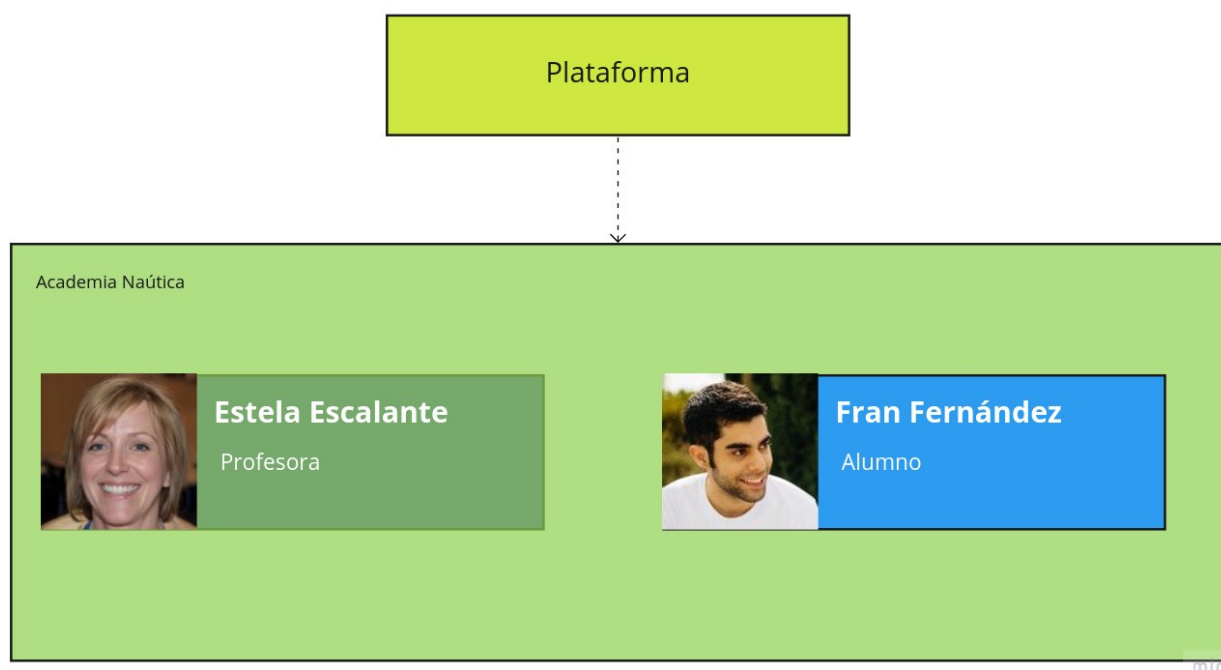




Figura 9: Personas

 <h2 style="display: inline;">Estela Escalante</h2> <p>Claves:</p> <ul style="list-style-type: none"> • Le preocupa que sus alumnos aprendan adecuadamente • Le falta tiempo para realizar un seguimiento 	
<p>Perfil personal</p> <p>Estela es de Cádiz, tiene 48 años y desde hace 15 años lleva su propia academia náutica. Se encarga de labores administrativas y también de enseñanza del temario para sus alumnos. Se encarga de captar nuevos alumnos, mantenerlos implicados y enfocados en obtener resultados para poder superar cada convocatoria.</p>	<p>Demografía</p> <ul style="list-style-type: none"> • Edad: 48 años • Estudios: Empresariales • Trabajo: Profesora de academia • Hobbies: Bicicleta • Personalidad: Extrovertida e inquieta
<p>Objetivo y motivaciones</p> <p>Estela necesita</p> <ul style="list-style-type: none"> • Realizar un seguimiento de sus alumnos • Asegurarse que sus alumnos están preparados • Diferenciarse de su competencia para captar mas alumnos 	<p>Capacidades tecnológicas</p> <ul style="list-style-type: none"> • Usa ordenador habitualmente por trabajo • Ofimática nivel medio • Usa el smarphone para redes sociales y compras
<p>Escenario</p> <p>La asistencia presencial cada vez es mas complicada, y Estela se preocupa porque las clases que realiza sean lo mas provechosas posibles. Cuando prepara a sus alumnos para superar el examen de una titulación náutica intenta repasar de manera general, pero es complicado conocer los puntos fuertes y débiles de cada alumno.</p> <p>De cada grupo suele llevar apuntes sobre el progreso de los alumnos para intentar reforzar algunos temas que sean mas difíciles de asimilar. Para ella es importante la satisfacción de sus alumnos, ya que el boca a boca es muy importante, y necesita buenos resultados, aunque al final queda en los propios alumnos que estudien y se preparen adecuadamente.</p> <p>Navegando por internet encontró una página que le permite agregar sus apuntes y crear un conjunto de preguntas/respuestas y definiciones para complementar lo explicado en clase. Sus alumnos podrán repasar y estudiar utilizando esta plataforma, y ella podrá conocer el avance y el nivel de aprendizaje de cada tema para cada uno de sus alumnos.</p>	

 <h2 style="margin: 0;">Fran Fernández</h2> <p>Claves:</p> <ul style="list-style-type: none"> • Quiere obtener una titulación náutica para coger el barco de su padre • No se le da bien estudiar y memorizar 	
<p>Perfil personal</p> <p>Fran es de Cádiz, tiene 25 años y trabaja como instalador de aires acondicionados. Siempre que puede queda con amigos para pescar. Su padre también es aficionado a la pesca y tiene un barco y le ha prometido que si se saca la titulación le prestará su barco para poder ir a pescar con sus amigos siempre que quiera.</p>	<p>Demografía</p> <ul style="list-style-type: none"> • Edad: 25 años • Estudios: Formación Profesional • Trabajo: Instalador Aire Acondicionados • Hobbies: Pesca, Series • Personalidad: Callado y Obsesivo
<p>Objetivo y motivaciones</p> <p>Fran necesita</p> <ul style="list-style-type: none"> • Obtener la titulación náutica para poder utilizar el barco de su padre • Aprenderse el temario para el examen • Una forma de motivarse para estudiar • Tiempo para poder concentrarse en sus apuntes 	<p>Capacidades tecnológicas</p> <ul style="list-style-type: none"> • Usa ordenador portátil solo para jugar y ver series • Usa el smarphone de forma intensiva, para navegar, redes sociales y compras
<p>Escenario</p> <p>Después de trabajar y tres veces en semana asiste a clases en la academia para prepararse para el examen de la titulación náutica que necesita. Las clases presenciales las entiende bien, pero se le da mal coger apuntes y estudiar por su cuenta. Lleva una vida muy ajetreada y no encuentra tiempo para repasar.</p> <p>En la academia le acaban de informar que tienen un nuevo servicio, que les permitirá estudiar y repasar desde sus propios móviles. Le dan de alta en una nueva plataforma, donde tiene disponible los resúmenes de los temas, y además puede repasarlos realizando sesiones rápidas de preguntas.</p> <p>Ahora puede utilizar cualquier hueco o descanso para estudiar en cualquier parte. Además el sistema le insisten en los puntos del temario donde mas preguntas falla, de forma que puede reforzar de forma precisa para estar mas preparado.</p> <p>tema para cada uno de sus alumnos.</p>	

6 Funcionalidades

Una vez definidas las Personas o perfiles de nuestro sistema, se utilizó la herramienta Miro para simular una sesión de toma de requisitos con estos usuarios utilizando el metodología de mapa de historias de usuarios. Mediante esta técnica se crearon distintos elementos que representan requisitos de alto nivel desde el punto de vista de estos usuarios.

Para reforzar la necesidad de estos requisitos se utiliza el formato de Historias de usuario, donde además de un título que resuma la funcionalidad solicitada, se debe razonar sobre esta, agregando un detalle siguiendo el formato *“Como [Tipo de perfil] Quiero [Funcionalidad] Para [Motivación/Objetivo]”*.

Se ha elegido un conjunto de historias que conforme un MVP (Mínimo producto viable) y que se estima que sea abarcable en el tiempo de desarrollo del TFM disponible. El mapa resultante se muestra continuación como también el detalle de cada una de las historias de usuario:

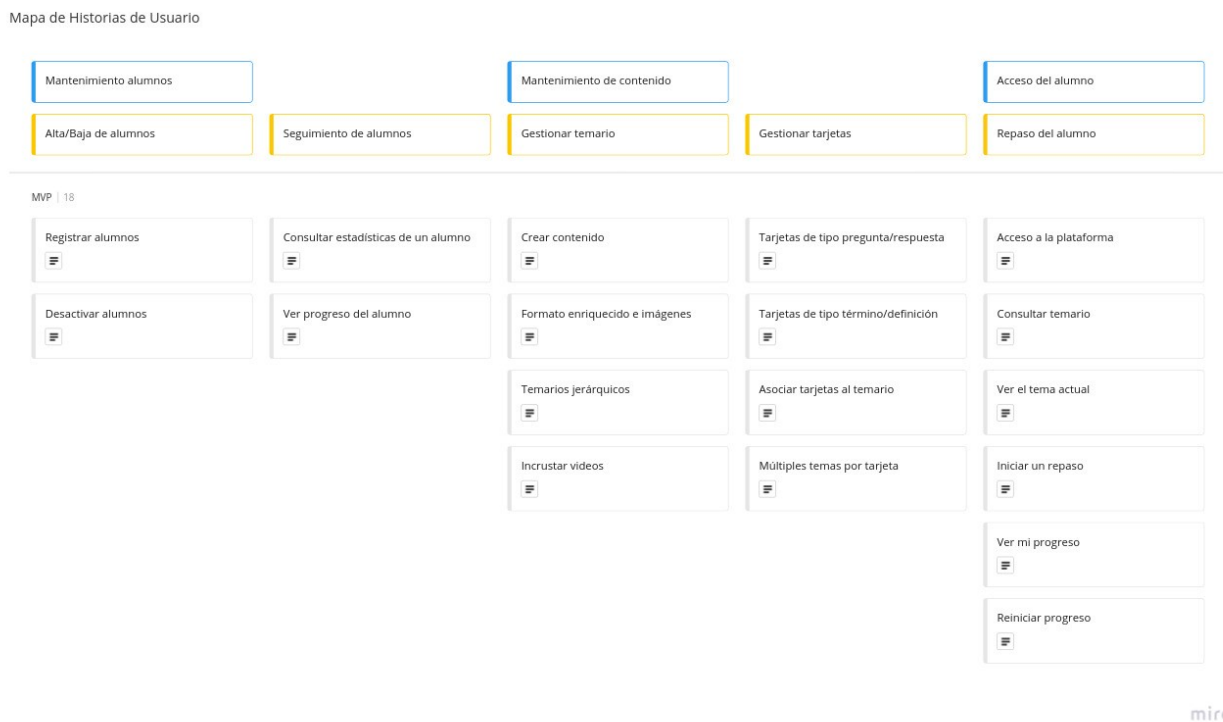


Figura 10: Mapa de historias de usuario

6.1 Alta/baja de alumnos

<p>Registrar alumnos</p> <p>Como administrador</p> <p>Quiero poder registrar nuevos alumnos, indicando sus datos personales</p>
--

Para poder permitirles acceder al temario y hacer el seguimiento de su aprendizaje

Desactivar alumnos

Como administrador

Quiero poder desactivar alumnos y evitar que accedan a la plataforma

Para poder controlar que no usen la plataforma si dejan la academia o adeudan algún pago

6.2 Seguimiento de alumnos

Consultar estadísticas de un alumno

Como administrador

Quiero saber ciertos datos sobre la actividad de un alumno, como el número de sesiones realizadas y las fechas de realización

Para poder realizar un seguimiento del aprendizaje del alumno

Ver progreso del alumno

Como administrador

Quiero poder consultar el progreso de un alumno, mostrando el nivel completitud o conocimiento de cada punto del temario

Para conocer los temas mas difíciles de asimilar por cada alumno

6.3 Gestionar temario

Crear contenido

Como administrador

Quiero crear nuevas entradas en el temario y escribir su contenido

Para que los alumnos puedan ver y acceder a los temas definidos

Formato enriquecido e imágenes

Como administrador

Quiero poder incluir textos con formato e imágenes en la definición del contenido de los temas que forman el temario

Para poder insertar imágenes explicativas de los conceptos del tema

Temarios jerárquicos

Como administrador

Quiero poder ordenar y definir el nivel de los distintos puntos que forman el temario

Para mostrar la estructura jerárquica de los temas

Incrustar videos

Como administrador

Quiero poder añadir videos de plataformas como YouTube o Vimeo como parte de los temas

Para poder completar los apuntes con videos realizados por la propia academia

6.4 Gestionar tarjetas

Tarjetas de tipo pregunta/respuesta

Como administrador

Quiero poder crear tarjetas de tipo test que consten de pregunta y respuestas

Para poder simular las preguntas reales que después formarán parte de los exámenes

Tarjetas de tipo término/definición

Como administrador

Quiero poder crear tarjetas donde se pregunte por un término y se muestre su correspondiente definición

Para que los alumnos puedan asimilar conceptos del temario

Múltiples temas por tarjeta

Como administrador

Quiero poder asignar mas de un punto en el temario para una misma tarjeta

Para poder relacionar distintas fuentes de información relacionada con la pregunta

Asociar tarjetas al temario

Como administrador

Quiero poder asociar una tarjeta a un punto del temario

Para poder agrupar preguntas que tratan del mismo tema

6.5 Repaso del alumno

Acceso a la plataforma

Como alumno

Quiero poder registrarme usando las credenciales facilitadas por el profesor

Para poder acceder a la plataforma, consultar el temario y estudiar

Consultar temario

Como alumno

Quiero poder consultar el temario existente

Para poder acceder a los apuntes teóricos del tema

Ver el tema actual

Como alumno

Quiero ver el punto del temario actual

Para saber que tema se está estudiando de forma activa

Iniciar un repaso

Como alumno

Quiero poder elegir un punto del temario para iniciar un repaso

Para reforzar algunos temas en concreto para mejorar mi aprendizaje

Ver mi progreso

Como alumno

Quiero poder ver el progreso de completitud de cada punto del temario

Para conocer cuánto queda para considerar aprendido un tema

Reiniciar progreso

Como alumno

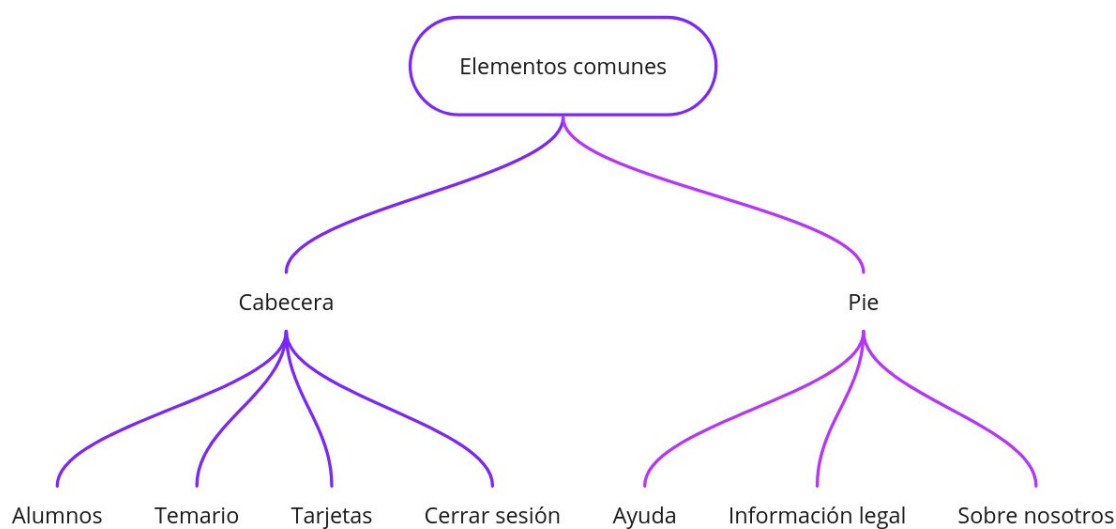
Quiero poder reiniciar el progreso de algún punto del temario

Para poder iniciar el aprendizaje desde cero y ser preguntado por todas las preguntas

7 Contenidos

Una vez descritos estas historias de usuario, se ha diseñado una arquitectura de información con la siguiente estructura y páginas a desarrollar.

7.1 Elementos comunes



miro

Figura 11: Elementos comunes del menú

En las distintas páginas de la aplicación del administrador, tendremos siempre disponible un conjunto de opciones siempre visible para facilitar la navegación. En este caso la cabecera y el pie no son páginas independientes si no que son elementos contenedores de enlaces a otras páginas. La cabecera contiene enlaces a los principales módulos que se detallan en el siguiente punto y la acción de cerrar sesión. El pie de página permitirá el acceso a páginas de contenido general.

7.1.1 Ayuda

Aquí estará la página donde aglutinar la documentación sobre el funcionamiento de la plataforma, donde obtener información sobre los principales opciones, y obtener respuesta a las preguntas frecuentes.

7.1.2 Información legal

La página contendrá información sobre las distintas leyes que aplican al funcionamiento de la plataforma, los términos de uso aceptados por los usuarios, referencia de autoría de librerías de terceros que as lo requieren, así como información sobre RGPD, etc.

7.1.3 Sobre nosotros

Aquí tendremos información sobre la empresa y el servicio. Historia del servicio, equipo, visión y valores, además de datos de contacto que permitan al usuario contactarnos por algún motivo.

7.2 Menú de administradores

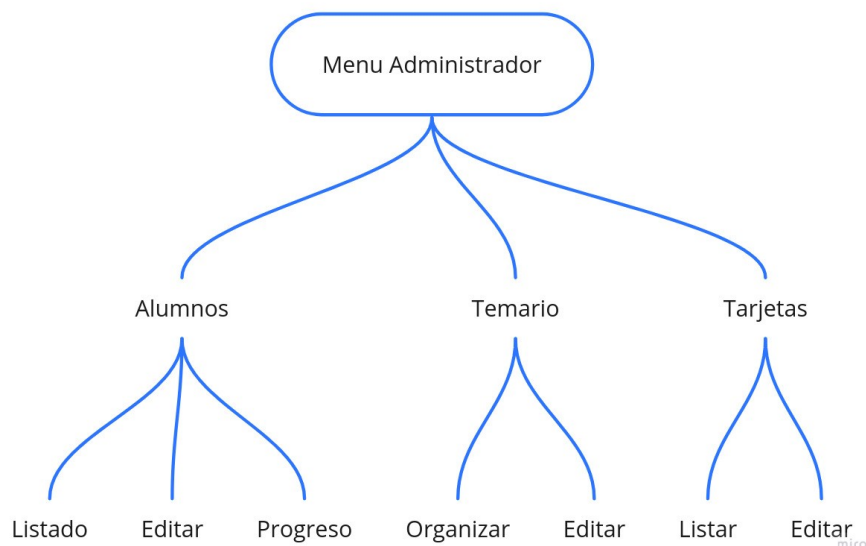


Figura 12: Elementos del menú de administración

La aplicación se separa en dos grandes módulos, uno destinado a los alumnos y otro destinado a los administradores. Aquí nos encontramos con tres grandes grupos en función de los datos que queramos administrar.

7.2.1 Listado de alumnos

En esta página tendremos un listado de los alumnos que tenemos registrados actualmente. Desde esta listado podremos consultar su nombre, estado, y fecha de registro. Desde esta misma pantalla tendremos la posibilidad de realizar algunas acciones que nos navegaran a otras páginas relacionadas con un alumno en concreto.

7.2.2 Edición de alumno

Tanto en el caso de crear un nuevo alumno, como en el caso de la edición visitaremos una nueva página con un formulario que nos permita modificar los datos básicos relacionados con el alumno. Además desde aquí podremos eliminar, deshabilitar o generar un código de activación para el dispositivo que vaya a utilizar el alumno.

7.2.3 Progreso del alumno

Desde la pantalla de consulta podremos realizar el seguimiento del alumno, para consultar el nivel de completitud de cada punto del temario, así como obtener distintas estadísticas con el objetivo de trabajar o enfocarse en las necesidades del alumno.

7.2.4 Organización del temarios

Aquí podremos consultar los puntos o temas de los que consta el temario a aprender por los alumnos. Desde este listado jerárquico podremos modificar el orden y nivel de anidamiento de los puntos que conforman el menú. Además tendremos acciones o enlaces que nos permitan visitar otras páginas para crear nuevos puntos del menú o para operar sobre algún punto existente.

7.2.5 Editar tema

Esta pantalla contendrá una vista jerárquica del temario y donde encaja el tema que estamos editando dentro del temario. Además de un formulario con los campos de gestión necesarios, así como un campo de área editable que nos permita agregar el contenido en texto e imágenes.

7.2.6 Listado de tarjetas

En esta página tendremos una lista con las tarjetas existentes. Además nos permitirá filtrar de manera sencilla por cada punto del temario, para facilitarnos la consulta del número de tarjetas por cada punto. Como en casos anteriores, nos permitirá crear nuevas tarjetas o editar las existentes.

7.2.7 Editar tarjeta

Tanto al crear nueva tarjeta, como al editar accederemos a la misma página. Desde aquí podremos elegir el tipo de tarjeta, modificar su dependencia en el temario, y agregar sus distintas opciones en función del tipo elegido.

7.3 Menú de usuario

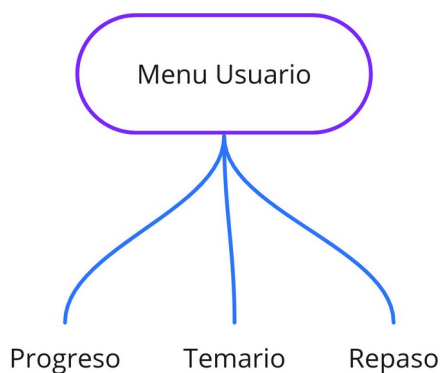


Figura 13: Elementos del menú de los usuarios/alumnos

El módulo destinado a los usuarios, será una aplicación mobile-first destinada a ser utilizada en dispositivos móviles. Debido a esto y a que su propósito es muy específico, la cantidad de páginas de las que dispone es mucho más reducida.

7.3.1 Progreso del alumno

Cuando el alumno accede a la aplicación, lo que veremos siempre es información relacionada con su progreso. Esta será la página de inicio de la aplicación, donde podrá ver el tema actual y podrá iniciar nuevas sesiones de aprendizaje.

7.3.2 Temario

Desde la aplicación del alumno podremos consultar el temario creado por los administradores y profesores. Desde esta pantalla podremos consultar el nivel de completitud de cada tema, si se considera superado, y podremos acceder a algunos de los temas para ver el contenido.

7.3.3 Repaso

A esta pantalla podremos acceder desde distintos puntos. Es la acción principal para la que está destinada esta aplicación del alumno. Desde esta página se gestionará una sesión de aprendizaje que mostrará distintas tarjetas que el alumno tendrá que responder o informar si conocía la respuesta o definición con el fin de obtener estadísticas para implementar el método de repetición espaciada.

8 Prototipos

En esta etapa del diseño de la solución, se han creado una primera versión de las pantallas que hemos analizado en el contenido y arquitectura de información detallado anteriormente. La aplicación está separada en dos grandes módulos, uno destinado a los administradores para ser usada desde un dispositivo con pantalla grande (tipo Escritorio o tablet) y la parte del alumno, diseñado para ser usada desde un dispositivo móvil.

8.1 Módulo de administración

Una vez los usuarios profesores se autenticuen en la aplicación accederán a la página principal de la zona de administración. Desde esta pantalla podrán acceder a los principales módulos que les permitan administrar los alumnos, temarios y tarjetas. También podrán ver la actividad reciente realizada en la plataforma.

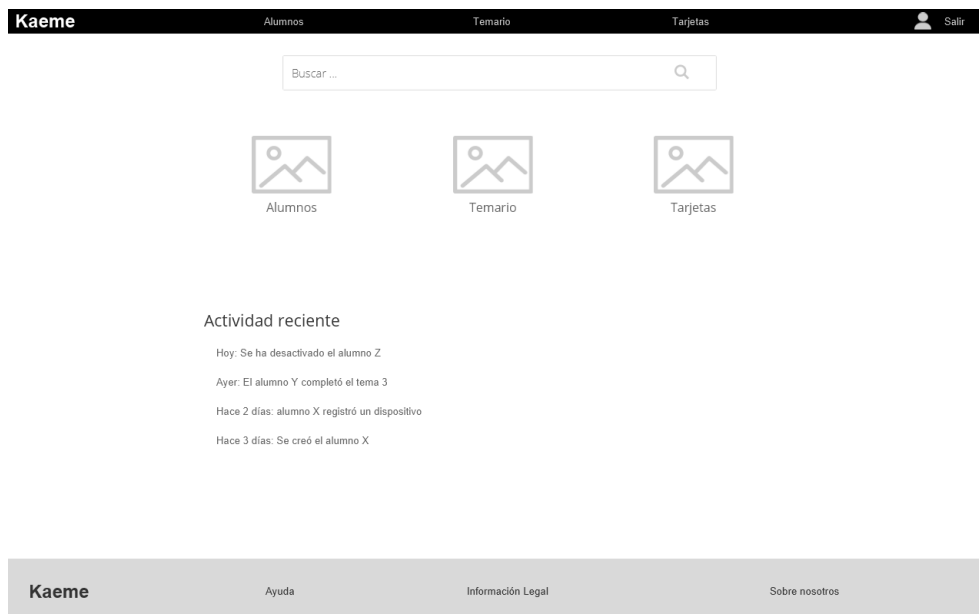


Figura 14: Pantalla de inicio de la administración

Desde el módulo de alumnos se podrá comprobar la lista de alumnos en la aplicación Podremos ver su nombre, apellidos, fecha de inscripción y si están activos o desactivados. Además, desde cada entrada del listado podremos acceder a la consulta de su progreso de aprendizaje o acceder al mantenimiento del usuario.

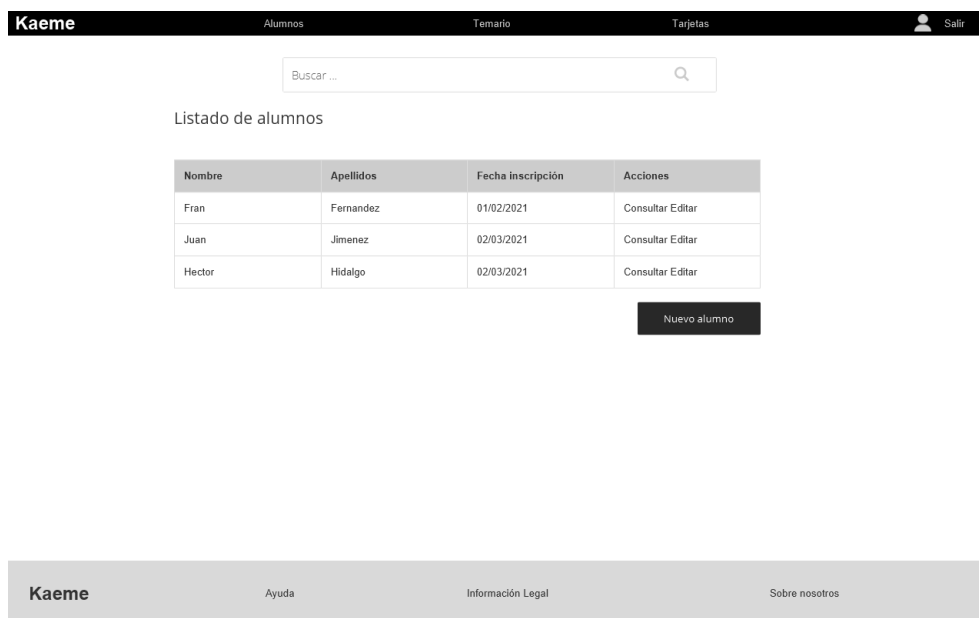


Figura 15: Listado de alumnos

Al editar un alumno podremos editar los datos personales, de contacto, podremos activar o desactivarlo y podremos generar una nueva contraseña o código de activación para que pueda acceder a la plataforma.

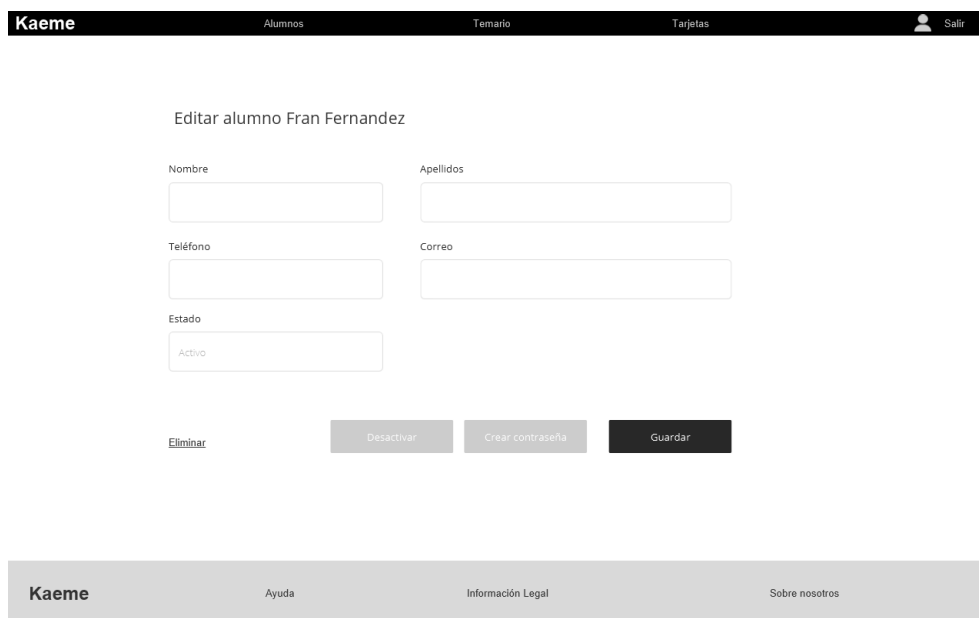


Figura 16: Creación/Edición de alumnos

Desde la pantalla de consulta del alumno podremos comprobar distinta información relativa al progreso del alumno y de las sesiones realizadas.

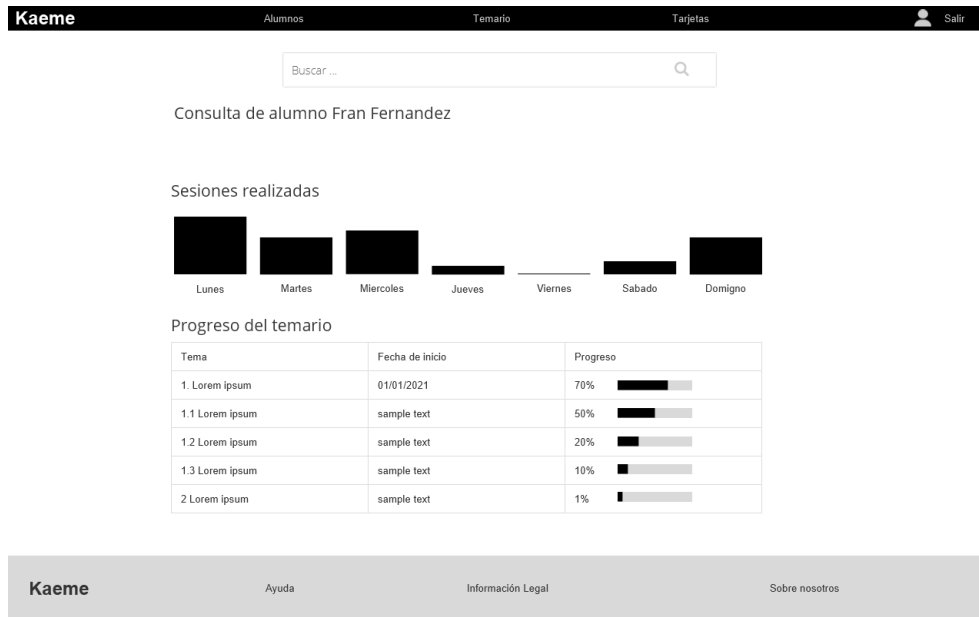


Figura 17: Seguimiento del alumno

Desde la edición del temario podremos acceder al temario definido, con los capítulos y variar su orden y estructura jerárquica.

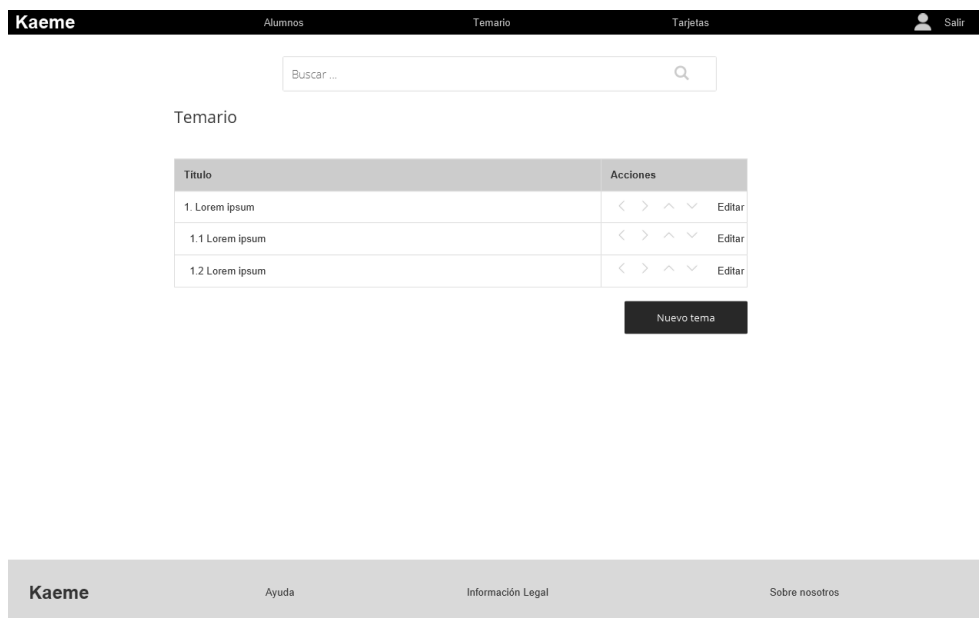


Figura 18: Organización del temario

Al editar un capítulo del temario, podremos editar su nombre y editar y formatear su contenido. Además podremos colgar directamente el capítulo de algún otro punto del menú o eliminarlo.

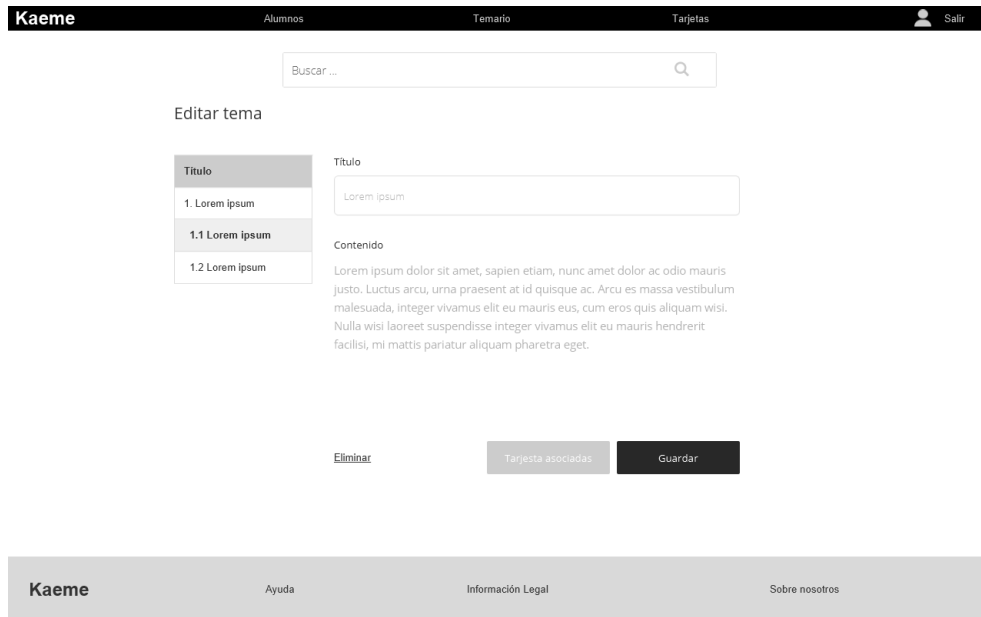


Figura 19: Editar tema

El listado de tarjetas nos permitirá consultar las tarjetas existentes además de poder filtrar según los puntos del temario y ver las tarjetas existentes para cada capítulo.

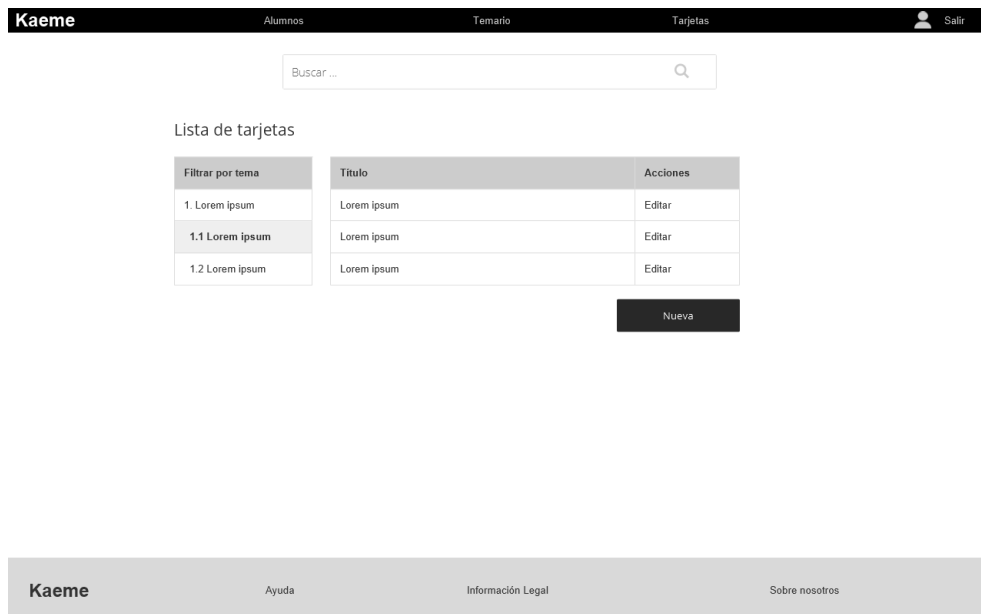


Figura 20: Listado de tarjetas

Al editar la tarjeta podremos indicar el tema al que está asociada la tarjeta. Además podremos indicar la pregunta y la respuesta correspondiente. También podremos eliminar la tarjeta.

Kaeme Alumnos Temario Tarjetas Salir

Editar Tarjeta

Tipo: Tema relacionado:

Pregunta

Respuesta

[Eliminar](#)

Kaeme Ayuda Información Legal Sobre nosotros

Figura 21: Listado de tarjetas

8.2 Módulo del alumno

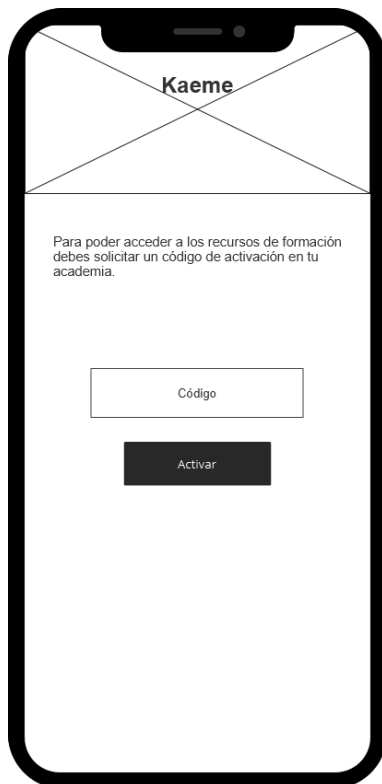


Figura 22: Activación del alumno

Cuando los alumnos acceden inicialmente al módulo de estudio, se les mostrará una pantalla donde deban indicar el código de activación. Este código será facilitado por el profesor y dará acceso al dispositivo a la plataforma.



Figura 23: Pantalla de progreso y de inicio

Una vez los alumnos han activado el dispositivo, no tendrán que volver a autenticarse y se les mostrará la página de inicio, donde podrá ver el progreso realizado y ver el tema actualmente que se está repasando.

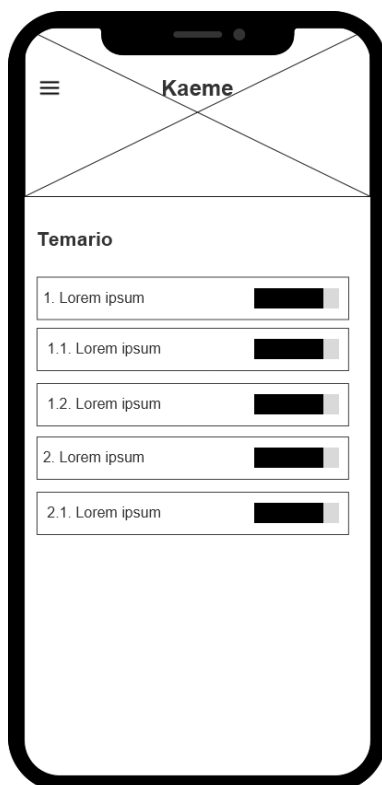


Figura 24: Consulta del temario del alumno

Cuando los alumnos consulten el temario podrán ver el listado de capítulos para ver los recursos de formación y contenido asociado. Además podrán ver el nivel de progreso de aprendizaje de cada capítulo.

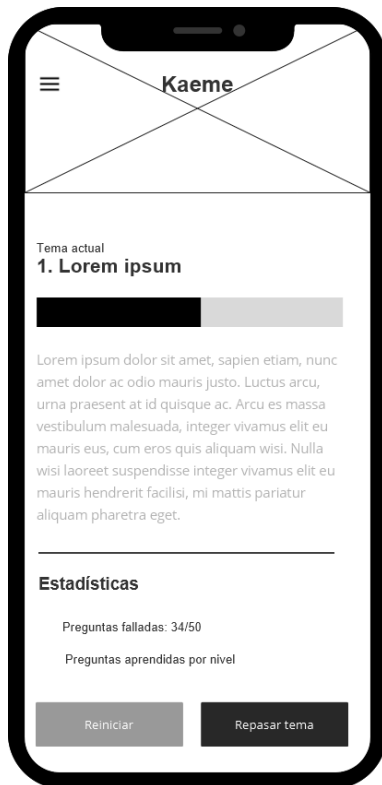


Figura 25: Consulta de un tema por el alumno

Al consultar un tema se mostrará el título y contenido de este. Además se podrá ver el nivel de progreso de aprendizaje de las tarjetas y estadísticas relacionadas con el alumno y el capítulo consultado. Desde aquí podrán iniciar nuevos repaso o eliminar el progreso y comenzar desde el principio.

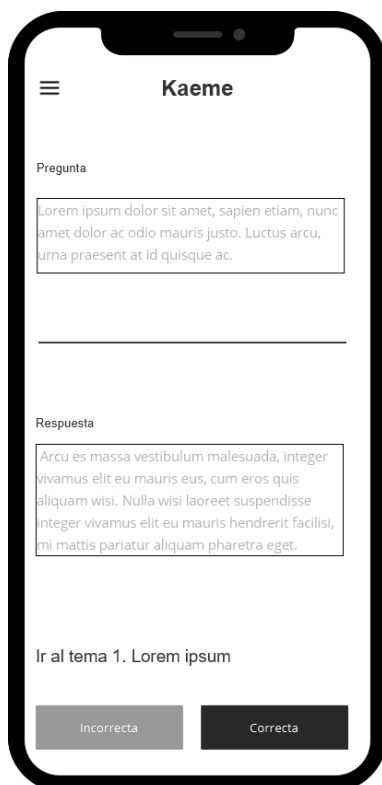
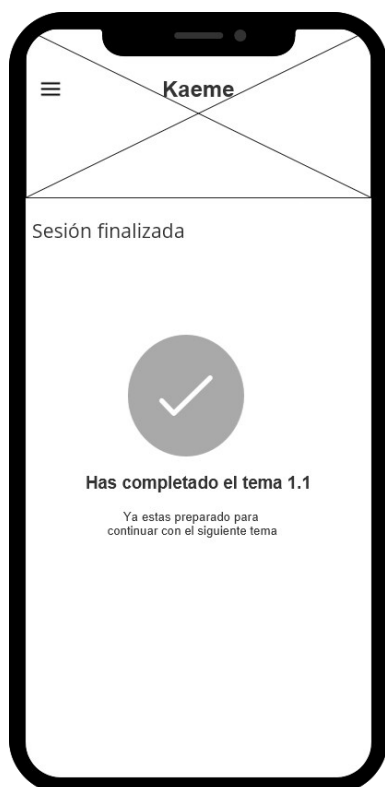


Figura 26: Sesión de aprendizaje del alumno

Cuando se inicia una sesión, se le mostrar a los usuarios la pregunta y los botones para mostrar la respuesta e indicar si considera que la sabía o no.



Al finalizar la sesión de repaso se le muestra información sobre la sesión realizada.

Figura 27: Confirmación de superación de un tema

9 Arquitectura de la aplicación

La aplicación a desarrollar en este proyecto será una aplicación web. Como se ha comentado anteriormente existen dos módulos claramente diferenciados, los destinados a los administradores y a los alumnos. Ambos módulos formarán parte de la misma aplicación web, por lo que sería una instalación, mantenimiento y despliegue único.

Si bien desde el punto de vista funcional estaríamos hablando de dos módulos diferenciados, ambos estarían estructurados desde el punto de vista técnico de la misma forma, separados en dos partes, backend y frontend.

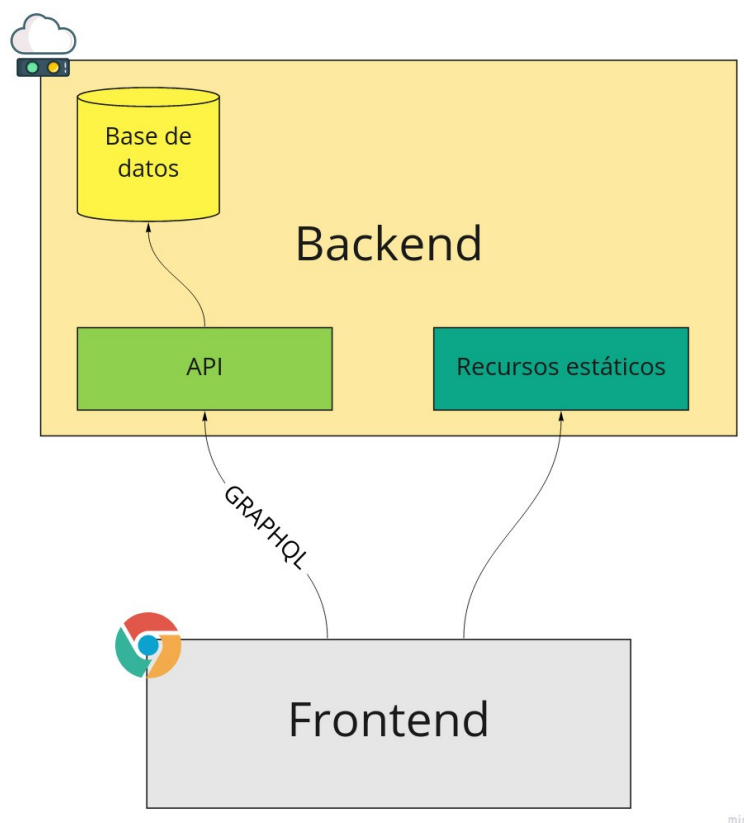
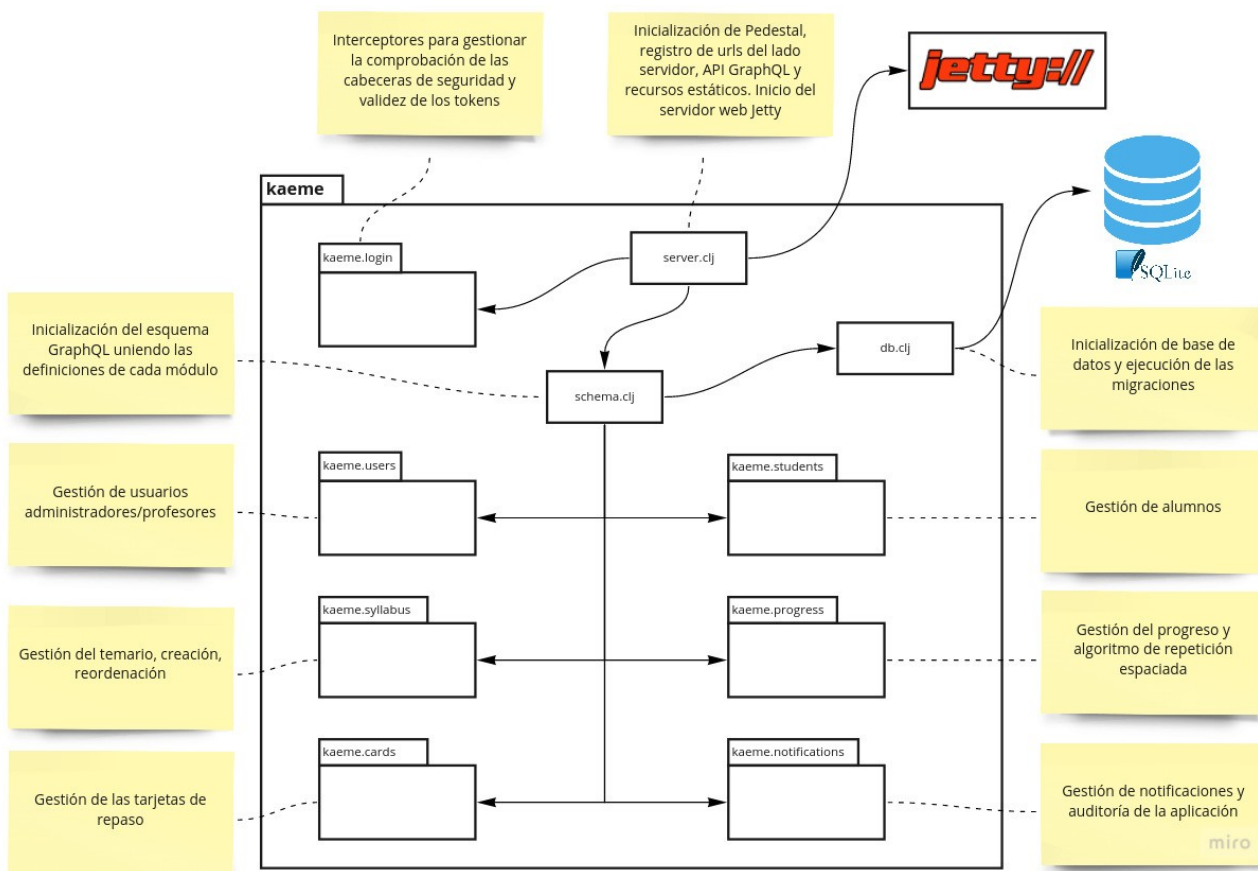


Figura 28: Arquitectura básica de la aplicación backend/frontend

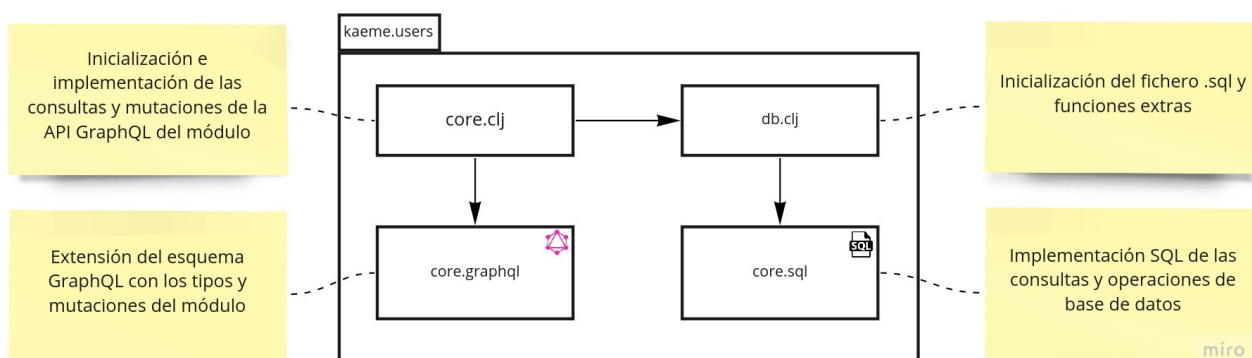
9.1 Backend

Esta parte de la aplicación será la ejecutada en el lado servidor y es la que tendrá acceso a la base de datos para persistir los cambios realizados por los usuarios. Este backend también será el encargado de alojar y servir los elementos estáticos de la aplicación, como imágenes, hojas de estilo y ficheros JavaScript. Por supuesto también ofrecerá la parte dinámica de la aplicación, que consistirá en una API que implementa GraphQL que será consumida por la parte frontend.

A continuación se incluye un diagrama de clases y paquetes con la estructura principal del backend de la aplicación:



Cada uno de los seis paquetes de los principales módulos funcionales de la aplicación siguen el mismo esquema y distribución en cuanto su contenido:

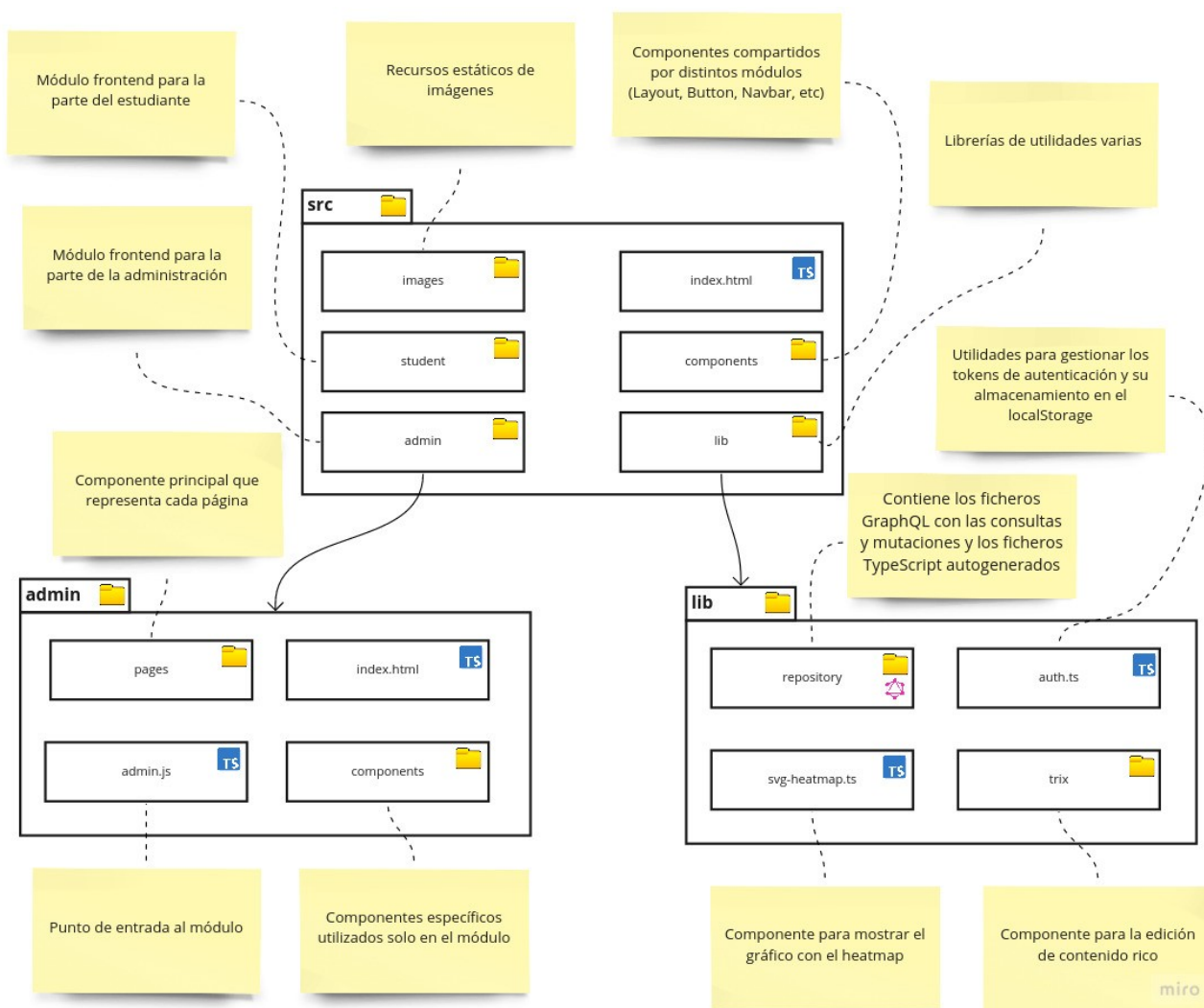


9.2 Frontend

Esta es la parte que se ejecutará en el lado cliente. En este caso la aplicación se ejecuta al completo en el lado cliente, es decir, en el navegador de los usuarios. Durante el primer acceso el backend devuelve el

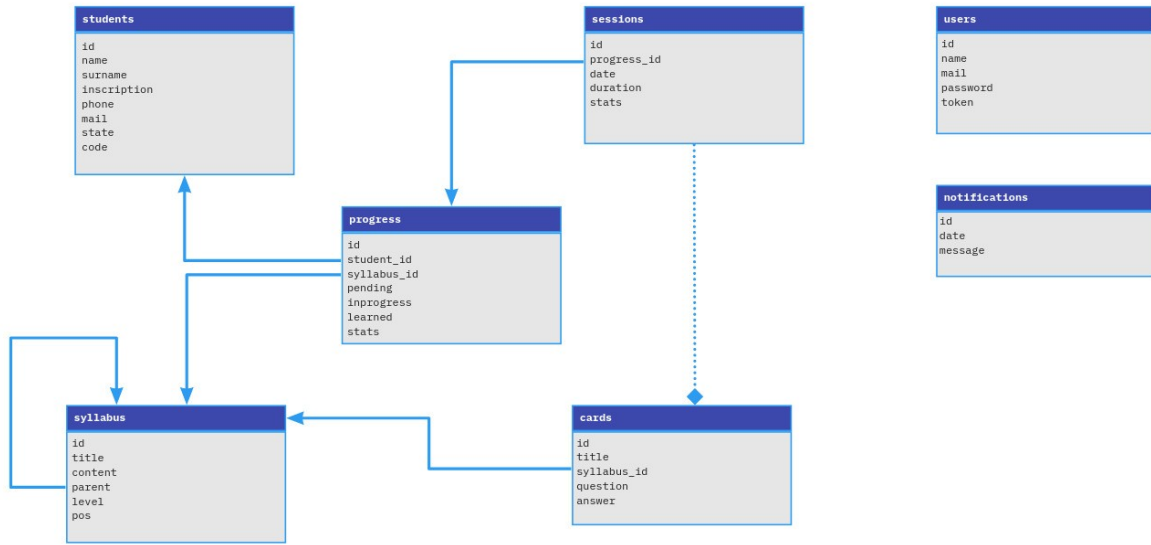
recurso estático de la página de inicio que desencadena la inicialización de la aplicación JavaScript. Esta a su vez solicita el resto de recursos estáticos que necesite. Una vez inicializada hará uso de la API ofrecida por el backend para implementar la funcionalidad de los módulos.

A continuación se muestran un diagrama que muestra la organización de los componentes que componen la parte frontend de la aplicación:



9.3 Base de datos

A continuación se muestran el modelo relacional y las tablas, campos y relaciones implementadas en la base de datos para dar soporte a la funcionalidad requerida. Siguiendo el convenio de nomenclatura, las tablas se nombran con el término plural que modelan, contiene una clave privada autoincrementable (en las bases de datos que no soporten esta característica se pueden implementar usando triggers). Los atributos con claves ajenas se nombran con el nombre de la tabla ajena y el sufijo id.



miro

Figura 29: Esquema de base de datos con sus relaciones

10 Plataforma de desarrollo

A continuación se detallan los principales elementos software y hardware utilizadas para el desarrollo de la solución, junto con algunos consideraciones y justificaciones.

10.1 Software

10.1.1 Herramientas de desarrollo

Como editor de código se utilizará Sublime Text 4, ya que es un editor muy potente y es junto a Emacs el editor con el que más tengo soltura. Mediante el uso de plugins y extensiones podemos añadir nueva funcionalidades para adaptarse a las necesidades y flujos de trabajo que queramos definir. Como sistema de control de código fuente se utilizará Git y como servidor remoto se utilizará la cuenta de Gitlab asociada al usuario de la UOC utilizada en la asignaturas.

10.1.2 Backend

A continuación se muestra un diagrama de las principales librerías utilizadas en la parte del backend. Además se detalla cada una de ellas explicando su propósito:

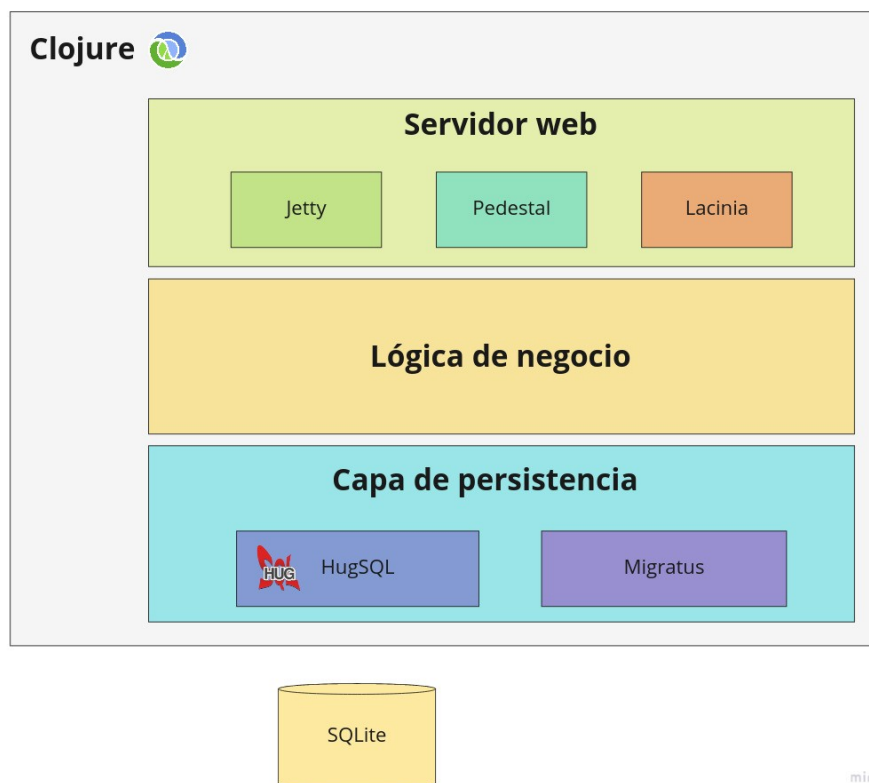


Figura 30: Diagrama con librerías usadas en el backend

Clojure

Clojure es un lenguaje de programación dialecto de Lisp que funciona sobre la máquina virtual de Java. Hace especial énfasis en el paradigma funcional. Su sintaxis y estructura es muy simple, básica y plana. Como el resto de lenguajes que corren sobre JVM, es completamente interoperable con cualquier librería Java existente. Permite utilizar el enfoque de desarrollo "REPL Driven Development".

<https://clojure.org/>

Leiningen

Lein es el sistema de construcción mas extendido para proyectos basados en Clojure. Es muy similar a Maven, donde definimos la configuración de manera declarativa en formato edn, formato nativo del lenguaje. Necesitaremos instalarlo para construir el proyecto.

<https://leiningen.org/>

pedestal

Pedestal es un conjunto de librerías que nos facilita la construcción de servicio y aplicaciones web. Se ejecuta en el backend y puede servir contenido estático, página HTML o gestionar peticiones API.

<http://pedestal.io/>

Lacinia

Lacinia es una implementación idiomática de la especificación GraphQL para Clojure. Su objetivo es ofrecer a las aplicaciones cliente una manera eficiente de obtener los datos del servidor. En comparación con el enfoque REST, GraphQL asegura que los clientes acceden justo a los datos que necesitan (y no mas), disminuyendo el número de consultas y peticiones necesarias. Este proyecto es mantenido y utilizado por Walmart, multinacional americana de venta minorista más grande del mundo.

<https://lacinia.readthedocs.io>

HugSQL

HugSQL es una librería para potenciar SQL. Su premisa es que la mejor herramienta para que las aplicaciones interactúe con bases de datos, mejor que cualquier librería que genera SQL o algún ORM que encapsule o mapee nuestros datos, es mejor utilizar SQL directamente!. Esto nos permite definir en ficheros planos SQL las distintas operaciones y funciones implementadas como sentencias SQL normales. Esta librería se ocupa de exponer de manera sencilla cada operación en estos ficheros como funciones en sus propio namespace, y tratando siempre la seguridad en cuanto a los datos de entrada.

<https://www.hugsql.org/>

migratus

Al igual que ocurre con el código fuente, donde utilizamos sistemas de control de versiones como Subversion o Git, podemos hacer lo mismo para controlar como evoluciona el esquema de base de datos de nuestra aplicación. De esta forma podemos controlar las actualizaciones de base de datos necesarias para cada nueva versión para cada entorno y base de datos. Su funcionamiento es similar a otras librerías como Flyway o liquibase.

<https://github.com/yogthos/migratus>

SQLite

En cuánto a la base de datos, se quiere aprovechar este proyecto para probar un enfoque del que he leído varias referencias. Habitualmente en el desarrollo de aplicaciones web se hace uso de sistemas de bases de datos como MySQL o PostgreSQL. Esto añade un componente de complejidad, ya que estos sistemas son complejos y requieren de una administración adecuada para asegurar la seguridad. Además, si la instalación es propia, esto nos requiere que tengamos que implementar las labores de mantenimiento y copias de seguridad correspondientes. Cuando utilizamos estas bases de datos de terceros proveedores nos podemos librar de estas tareas, pero aumentamos el coste y también aumentamos la latencia, lo que hará que el rendimiento de nuestra aplicación se pueda ver afectada.

<https://www.sqlite.org>

10.1.3 Frontend

A continuación se muestra un diagrama de las principales librerías utilizadas en la parte del frontend y se detallan cada una de ellas:

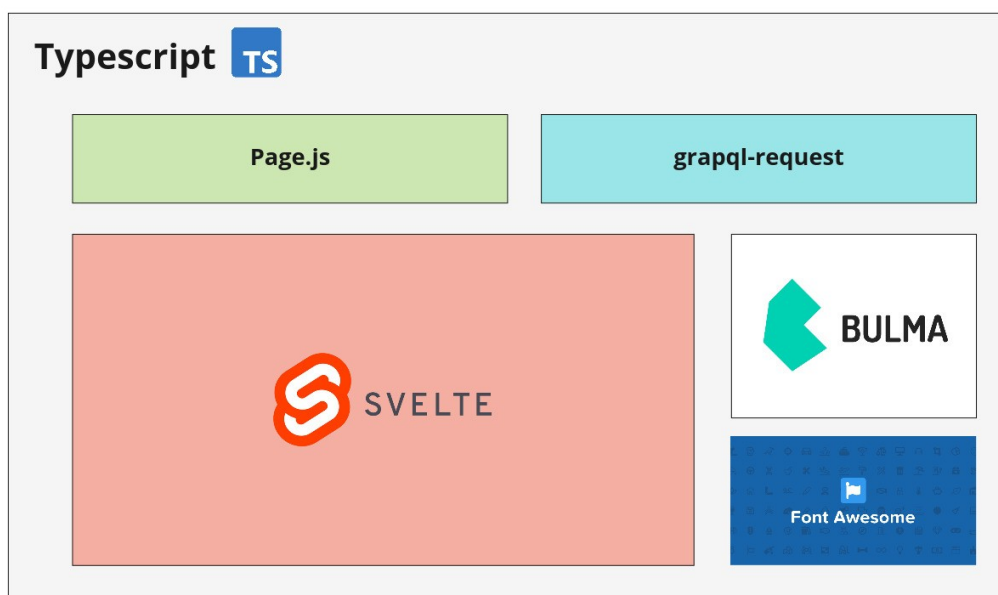


Figura 31: Diagrama con librerías usadas en el frontend

Parcel

Para el desarrollo de la parte frontend se utilizará una de las herramientas utilizadas en las distintas prácticas de las asignaturas del máster, Parcel. Esta herramienta nos permite la construcción de manera sencilla y óptima de los elementos que forman parte del frontend, uniendo todos los fuentes y módulos JavaScript en un único fichero minificado y comprimido, además de renombrar todos los ficheros estáticos con el fin de facilitar el cacheo de los elementos si queremos aprovechar las funcionalidades de algún CDN (Content Delivery Network, como Akami, Cloudflare, Cloudinary, etc).

<https://parceljs.org/>

Typescript

Es un lenguaje de programación libre y de código abierto creado y mantenido por Microsoft. Es un superconjunto de JavaScript que agrega opcionalmente el tipado estático. El compilador Typescript genera código JavaScript, por lo que nos permite utilizar características nuevas y avanzadas y poder seguir ejecutándose en cualquier navegador.

<https://www.typescriptlang.org/>

graphql-code-generator

Como se ha comentado anteriormente se utiliza GraphQL como lenguaje para interactuar con el servidor. Si bien es posible incluir en el código fuente cadenas de texto con la representación de las consultas GraphQL que queremos hacer, esto puede ocasionarnos problemas que no sean detectados hasta en tiempo de ejecución. Esta herramienta utilizada durante el proceso de desarrollo y construcción, analiza el esquema GraphQL expuesto por el backend y lo contrasta con las sentencias GraphQL que queremos ejecutar en cliente, generando código Typescript para ello. Gracias a esto podemos detectar cambios realizados en el backend que pudieran afectar al frontend o errores en cliente donde intentemos solicitar algún campo, opción o filtro inexistente.

<https://www.graphql-code-generator.com/>

graphql-request

Esta es una librería muy sencilla que nos permite realizar simples peticiones GraphQL. Es mantenida por la empresa Prisma, especializada en ofrecer servicios GraphQL en la nube.

<https://github.com/prisma-labs/graphql-request>

Svelte

Se usará el framework de componentes Svelte. Este es un framework reciente, que simplifica el desarrollo y es una alternativa simple a otros frameworks más complejos como Angular, React o Vue.js. Estos frameworks basan su funcionamiento en un cálculo intensivo en el cliente, utilizando la técnica de Shadow

DOM, donde se utiliza árboles de elementos para renderizar y detectar los cambios que se producen en el lado cliente. Svelte utiliza un enfoque distinto, detectando las dependencias en tiempo de compilación, y por tanto generando un código optimizado y con menor impacto de rendimiento. Además para la definición de los componentes, utiliza simplemente HTML, CSS y JavaScript en un único fichero con extensión .svelte.

<https://svelte.dev>

Page.js

Al ser la parte frontend una aplicación completamente renderizada en el lado cliente, necesitamos por tanto una librería que gestione la navegación y las url. Para ello se utiliza esta simple librería que nos permite definir las rutas y parámetros, y qué componentes de la aplicación se deben encargar de renderizarla. Esta librería interceptará los enlaces a otros puntos de la aplicación evitando que se realice una navegación o recarga de la aplicación. Además actualizará la url mostrada al usuario mediante el uso de la API History.

<https://visionmedia.github.io>

Bulma

Para la parte visual se ha utilizado el framework CSS Bulma, que proporciona un conjunto de componentes CSS sencillos y claros con buen soporte responsive.

<https://bulma.io/>

FontAwesome

Para mostrar diferentes elementos gráficos e iconos se ha utilizado la fuente FontAwesome que dispone de un gran número de iconos en su versión libre gratuita.

<https://fontawesome.com/>

10.2 Hardware

Para la construcción del proyecto se ha utilizado de un ordenador portátil y un dispositivo móvil. El ordenador portátil es en el cuál se han instalado las herramientas de desarrollo y elementos necesarios para la compilación, construcción y ejecución de la aplicación.

Para la revisión del responsive se han utilizado las funcionalidades disponibles en los actuales navegadores, además de la extensión <http://re-view.emmet.io/> que permite ver simultáneamente distintos tamaños de la aplicación, y también un dispositivo móvil real.

11 APIs utilizadas

A continuación se detallan los servicios o API utilizadas en la aplicación.

11.1 Cloudinary

Una de los requisitos elegidos de la aplicación contemplaba que en la gestión y redacción del contenido teórico, es decir, en la elaboración de los capítulos del temario se pudiera utilizar texto enriquecido e imágenes. El componente utilizado para editar este contenido permite la definición de unos eventos para integrar este componente con la necesidad de incrustar adjuntos e imágenes al texto enriquecido.

Se ha utilizado el servicio gratuito de Cloudinary como solución sencilla para la subida, custodia y descarga de imágenes. Entre las distintas opciones de integración que dispone este servicio, está la subida de ficheros desde el lado cliente. Cuando un editor copia o adjunta una imagen a un contenido, el componente de edición de texto lanza un evento que es implementado por la aplicación, que realiza una llamada asíncrona desde el lado cliente directamente a la API de Cloudinary y sube el fichero a la carpeta creada para el proyecto, devolviendo la URL final de la imagen subida.

Información sobre el método utilizado se puede encontrar en la siguiente dirección:

https://cloudinary.com/documentation/image_upload_api_reference

12 Usabilidad/UX

A continuación se detallan algunos de las prácticas y patrones de diseño que se han tenido en cuenta en el diseño y construcción de la aplicación.

12.1 HomeLink

Se muestra en un lugar fijo y en todo momento un enlace (normalmente con el logo del sitio) que permite al usuario volver al inicio del sitio.

- **Beneficio:** permitimos un mecanismo que permite al usuario volver al punto de partida para iniciar la búsqueda de información a seguir los patrones conocidos para realizar acciones.
- <http://ui-patterns.com/patterns/HomeLink>

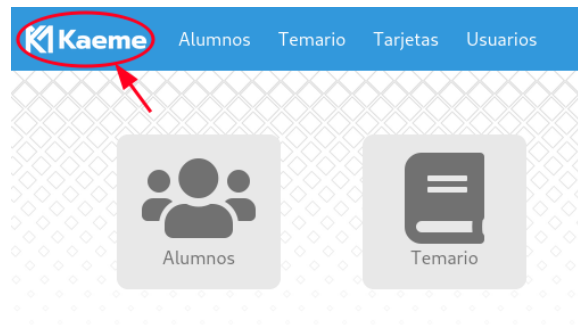


Figura 32: Botón inicio

12.2 Fat Footer

Al final de la página se encuentra siempre una sección amplia, normalmente usando colores inversos para diferenciarlo del contenido principal. En esta sección se encuentra especificado la mayoría de puntos de nuestra arquitectura de información mostrada de forma jerárquica.

- **Beneficio:** Disponer de un lugar visual y conocido donde conocer las distintas opciones existentes y poder acceder de forma directa.
- <http://ui-patterns.com/patterns/FatFooter>



Figura 33: Menú inferior

12.3 Morphing Controls

El usuario solo quiere ver los controles disponibles en cada momento según el modo actual. Algunos controles no necesitan mostrarse al mismo tiempo. La experiencia de usuario tiene distintos modos.

- **Beneficio:** Minimizamos el espacio necesario para mostrar información, y se utilizan distintos estilos sobre los mismos controles para informar de cambios de estado.
- <http://ui-patterns.com/patterns/morphing-controls>

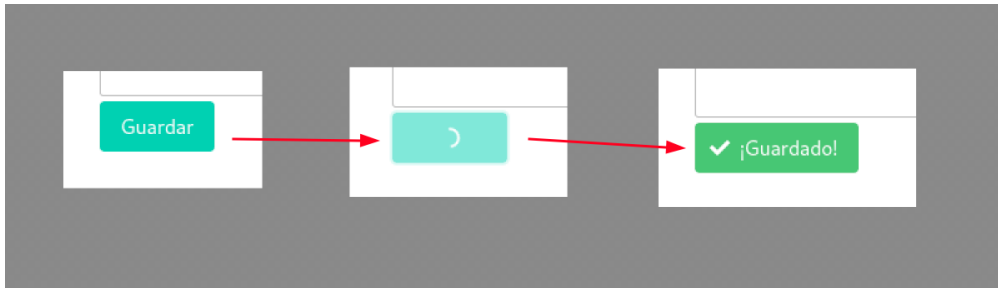


Figura 34: Botón guardar con cambio de estado

12.4 Navigations Tabs

Necesitamos separar el contenido en secciones y que pueda ser accedida de una forma plana y se muestre de forma visual el lugar actual. Usado cuando hay entre 2 y 9 secciones. Es preferible usar nombres de sección relativamente cortos.

- **Beneficio:** Organiza la información de forma que solo se muestre la relativa a la sección activa, ahorrando espacio y reduciendo la complejidad y cantidad de información mostrada simultáneamente.
- <http://ui-patterns.com/patterns/NavigationTabs>



Figura 35: Componente de pestañas

12.5 Paginación

El usuario necesita ver un subconjunto de datos de una tabla con gran cantidad de resultados. Usado cuando no queremos mostrar todos los datos en una única pantalla.

- **Beneficio:** Permitir al usuario mostrar la información en conjuntos mas pequeños y permitir seleccionar fácilmente nuevos conjuntos.
- <http://ui-patterns.com/patterns/Pagination>

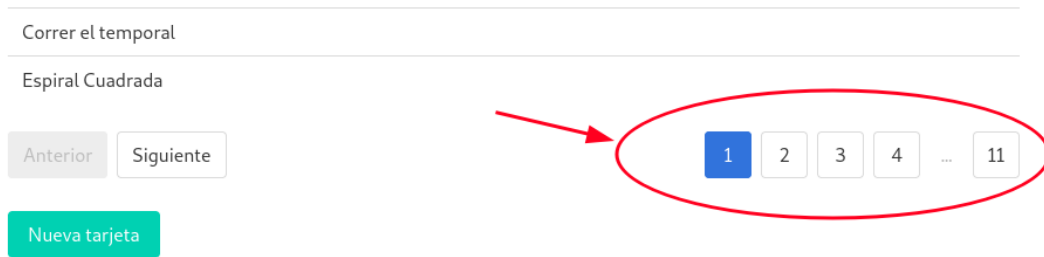


Figura 36: Botones de paginación

12.6 Input Feedback

El usuario ha introducido datos en el sistema y espera recibir información sobre el resultado de su envío.

- **Beneficios:** Informar al usuario de forma directa sobre los campos con datos inválidos.
- <http://ui-patterns.com/patterns/InputFeedback>

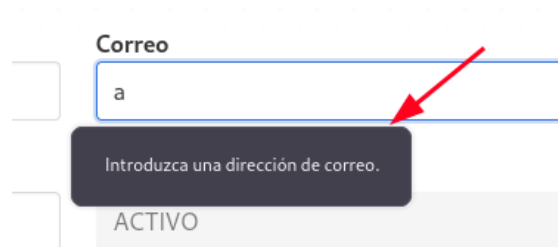


Figura 37: Mostrar errores en el formulario

12.7 Completeness meter

El usuario quiere completar un objetivo y necesita pista y guías para saber cuánto ha alcanzado y cuanto le queda por alcanzar.

- **Beneficio:** Mostrar forma visual e inequívoca el progreso realizado respecto al total posible.
- <http://ui-patterns.com/patterns/CompletenessMeter>

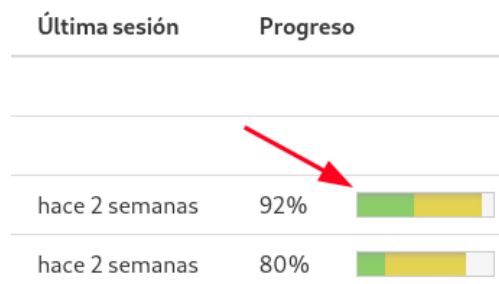


Figura 38: Barras de progreso

12.8 Table Filter

El usuario necesita realizar filtro categóricos en los datos mostrados en una tabla. Se puede usar cuando hay características comunes en las filas que pueden ser filtradas o totalizadas de forma sencilla.

- **Beneficio:** Ofrecer un mecanismo sencillo al usuario para reducir la cantidad de información de una tabla de datos y agrupar filas que estén relacionadas entre ellas.
- <http://ui-patterns.com/patterns/TableFilter>

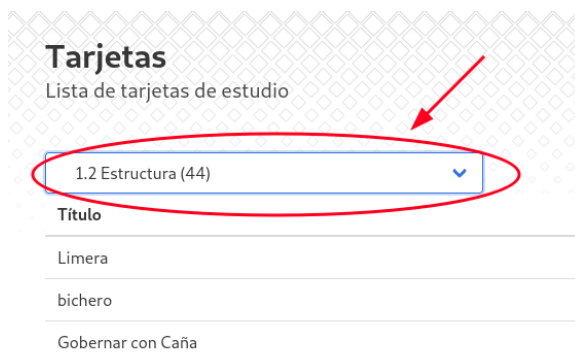


Figura 39: Filtro de opciones desplegable

12.9 Wysiwyg

El usuario quiere crear contenido que contiene texto enriquecido o formateado sin necesidad de conocer como escribir HTML, textile o markdown. Es usado como forma sencilla para que usuarios puedan integrar contenido enriquecido.

- **Beneficio:** Ofrecer una herramienta sencilla que permite ver visualmente el contenido introducido y los estilos (color, fuente, estilo de texto, etc) que se observarán finalmente.
- <http://ui-patterns.com/patterns/WYSIWYG>

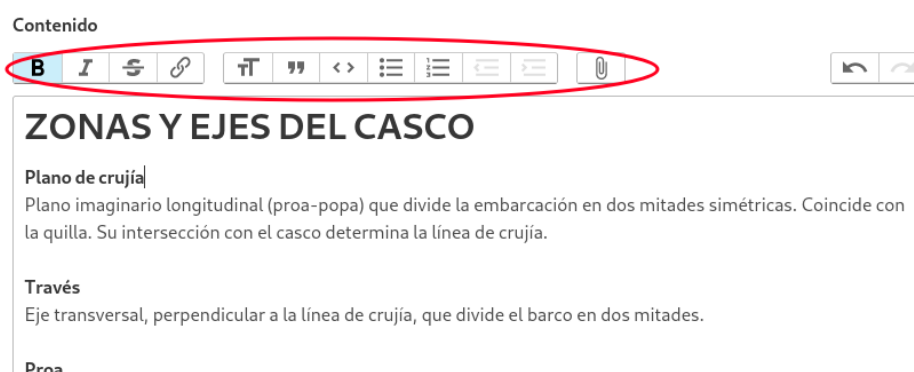


Figura 40: Componente de edición de texto enriquecido

13 Seguridad

A continuación se detallan algunos aspectos que podrían suponer riesgos de seguridad. Se detallan los puntos importantes según cada herramienta o librería utilizada que ofrece opciones o protecciones de seguridad.

ParcelJS es una herramienta de construcción para frontend enfocado en la usabilidad y mejorar la experiencia del desarrollador. Algunos de los aspectos de seguridad relacionados con el uso de esta herramienta serían los siguientes:

- **Evitar exposición de ficheros no deseados.** Una de las ventajas de utilizar esta herramienta es que se crea un árbol de dependencias completa de todo tipo de ficheros. Es decir, ParcelJS escanea y sigue las dependencias o referencias de todo tipo (JS, TS, HTML, CSS ó imágenes), asegurándonos que en la versión final de la aplicación solo esté el código y recursos que realmente estemos usando. Esto quiere decir que una imagen sin utilizar, un JavaScript que no se importe desde ningún módulo, o un fichero HTML que no tengan ningún enlace desde ninguna otra página no formará parte del proyecto de salida. Por lo que esto nos evitaría publicar ficheros por error.
- **Minificación y SourceMaps.** Al desarrollar aplicaciones frontend enviamos al lado cliente todo el código de nuestra aplicación. Aunque herramientas como esta realizan un procesamiento del código para unirlo en un único fichero, realizar cambios (opcionalmente ofuscarlo) y renombramientos en el código con el propósito de reducir su tamaño. Pese a todo es posible analizar el código así generado y obtener información valiosa. Esto es un factor importante a tener en cuenta, ya que todo algoritmo y proceso crítico deberíamos realizarlo en el backend. Durante el proceso de minificación también se suelen generar unos ficheros SourceMaps que guardan la referencia entre el código minificado y el código original para facilitar el proceso de depuración. Estos ficheros pueden ser utilizados para revertir el proceso y por tanto obtener el código fuente original con los nombres de funciones y variables originales. Se ha configurado Parcel para evitar la generación de estos ficheros durante la construcción final del proyecto.

En este proyecto se ha utilizado GraphQL como solución para la exposición y consumo de datos. GraphQL es una definición declarativa de los datos, consultas y operaciones que pueden realizarse entre el frontend y el backend.

- **Entrada de datos correctos.** La propia especificación de GraphQL nos obliga a definir un esquema de los objetos, atributos y tipos de cada uno de ellos. Es por esto, que el propio servidor realiza una validación de los datos de entrada, lo que evita que se agreguen datos no definidos con intención de intentar adivinar o sobrescribir atributos no expuestos en la capa de datos de la API.
- **Exposición de atributos adicionales.** Similar al caso anterior, la propia implementación de la especificación GraphQL del propio servidor, validará los datos de salida, evitando por tanto la exposición de campos internos de los datos que solo existieran en base de datos por ejemplo.

- **Autorización.** La autorización y autenticación son aspectos que no aborda la especificación, por lo tanto debe ser implementada por cada aplicación o por la implementación utilizada. Se pueden utilizar directivas (que serían como metadatos o anotaciones) en los atributos, mutaciones y consultas para restringir o realizar las comprobaciones en la propia implementación de la API. En este caso se utiliza un interceptor que realiza la autorización en busca de un alumno o un profesor y los agrega en el contexto. Según el tipo de consulta u operación, se requerirá de la existencia en el contexto de un profesor o alumno válido según el caso.

Para la autenticación de los usuarios se han definido dos puntos de entrada POST (fuera de GraphQL) para el envío de las credenciales de los administradores y para la activación y registro de los alumnos. En ambos casos, una vez validado los datos se devuelve un **token autogenerados**, que son los enviados por el frontend utilizando las cabeceras **HTTP Authorization** en cada nueva petición a la API. Al hacer uso de este tipo de token podremos implementar ciertas protecciones, como limitar el número de accesos por cada usuario, o asignarle un tiempo de caducidad a los tokens para requerir la reautenticación.

Respecto a la autenticación de los usuarios administradores, no se guardan las contraseñas en texto claro, si no que se utiliza un sistema de cifrado basado en **bcrypt/scrypt**. Esto implica que una vez el usuario elige una contraseña, esta es pasada por un proceso donde se agrega valores aleatorios antes de cifrarla finalmente guardándose en base de datos un valor alfanumérico incompresible. Esto tiene varias ventajas y características especiales. Al utilizarse valores aleatorios en cada contraseña esto hace que si varios usuarios eligen la misma contraseña el hash generado serían siempre distinto, evitando así exponer información de qué usuarios comparten contraseña. Además, el proceso es un solo sentido, imposibilitando intentar descifrar la contraseña se obtiene este hash. Durante el proceso de autenticación lo que se hace es volver a cifrar la contraseña que ha introducido el usuario con los parámetros de cifrado utilizados en su caso y ver si coincide. Este tipo protección es necesario para que en casos catastróficos donde agente maliciosos pudieran extraer información de las tablas de usuarios de base de datos, no podrían averiguar ni analizar sus contraseñas.

Uno de los ataques mas habituales a aplicaciones web es el denominado **SQL Injection**. Este tipo ataques consiste en intentar alterar las consultas SQL esperadas por la aplicación. Para ello los atacantes se aprovechan de parámetros que no son adecuadamente escapados por la aplicación y que además son utilizados de forma insegura en la construcción de estas consultas SQL que hace la aplicación. Esto era muy habitual cuando se intentaba construir las consultas como concatenación de cadenas, incluyendo de este modo valores que vienen sin "sanear". La mayoría de los frameworks de acceso a datos ya ofrecen funcionalidades para evitar este tipo de ataques. En este caso se utiliza HugSQL, que genera sentencias PreparedStatements de JDBC. Mediante este método, las consultas son construidas por un lado con la definición creada por el programador (escritas en ficheros .sql), donde se indica el lugar de los parámetros. Y en una segunda fase, se establecen los parámetros haciendo uso de setParameter de JDBC que garantiza el correcto uso de los parámetros evitando por tanto este tipo de ataques.

14 Instrucciones de instalación

La aplicación está dividida en dos partes: frontend y backend, construidas con tecnologías y herramientas diferentes. A continuación se detallan las instrucciones para la instalación de la aplicación en distintos escenarios.

14.1 Construcción con Docker

Para facilitar la construcción y despliegue de la aplicación se ha utilizado Docker. Esto es una tecnología de creación de contenedores (máquinas virtuales ligeras) que aprovecha alguna de las características de seguridad y aislamiento entre procesos de Linux como cgroups o namespaces. Esto nos permite definir contenedores para construir y desplegar la aplicación de forma que sean reproducibles y portables entre entornos.

La única dependencia por tanto que necesitaremos será instalar docker en la máquina donde queramos construir o ejecutar la aplicación. El único comando que debemos ejecutar será el siguiente:

```
docker build . -t kaeme
```

El comando anterior desencadena la creación de un contenedor específico donde se instalarán todas las dependencias de construcción necesarias e iniciará la compilación de todos los módulos del proyecto. Finalmente se creará un contenedor mínimo para la ejecución y lo etiquetará con el tag kaeme.

Mediante el uso de las opciones save y load podremos exportar este contenedor y llevárnoslo a otra máquina. Por ejemplo, en el caso de querer enviar el contenedor construido a la instancia AWS proporcionada para el proyecto, podemos ejecutar el siguiente comando:

```
docker save kaeme:latest | bzip2 | ssh aws-uoc 'bunzip2 | docker load'
```

Una vez hecho esto tendremos disponible el contenedor en nuestra máquina asignada. Solo necesitaremos iniciar dicho contenedor para iniciar la aplicación (que también lo podremos hacer en nuestro ordenador local). Puesto que los contenedores docker por seguridad están aislados debemos indicar a la hora de ejecutarlo distintas opciones encaminadas a exponer ciertos puertos, ficheros o variables de entorno para configurar ciertas funcionalidades.

```
docker run --restart=always -d -p 80:8888 -v /home/ubuntu/kaeme-db:/app/db/ --name kaeme kaeme:latest
```

El anterior comando es el utilizado en el servidor AWS y las opciones utilizadas se detallan a continuación:

- `--restart=always`, Esta opción indica que el contenedor debe reiniciarse si se para por algún motivo (a no ser que se utiliza `docker stop` explícitamente). Con esta opción también conseguimos que la aplicación se inicia automáticamente al iniciar la instancia AWS.
- `-d`, Esta opción indica que el comando debe ejecutarse en segundo plano. Esto es, libera nuestro terminal en cuanto lo ejecutamos. Podremos consultar los mensajes de log utilizando `docker logs`
- `-p 80:8888`, la aplicación dentro del contenedor escucha en el puerto 8888. Con esta opciones estaríamos mapeando el puerto 80 de la máquina host, con el puerto 8888 dentro del contenedor. Esto nos permite acceder a la aplicación accediendo a la ip de nuestra máquina AWS.
- `-v /home/ubuntu/kaeme-db:/app/db/`, Por defecto nuestro contenedor utilizará la ruta interna `/app/db` como directorio para crear la base de datos SQLite. Esta opción lo que hace es montar un volumen, para montar una ruta en el host (la primera indicada) con una ruta dentro del contenedor (`/app/db`). De esta forma, si destruimos el contenedor o lo actualizamos, los únicos ficheros que persistirán serán los montados de esta forma.
- `--name kaeme`, Con esta opción le damos un nombre a nuestro contenedor en lugar de utilizar uno autogenerado
- `kaeme:last`, este es el nombre o etiqueta de la imagen del contenedor. Esta es la imagen que construimos en el punto anterior con `docker build` y que enviamos a la máquina AWS usando `docker save` y `docker load`.

14.2 Construcción manual

La forma recomendada de construir la aplicación es mediante la utilización de `docker` que se ha detallado en el punto anterior. Todos los pasos que a continuación se detallan son los que se han automatizado con `docker`. Si quisiéramos desarrollar la aplicación en local o construir una versión para desplegar deberíamos instalar manualmente las siguientes dependencias:

- `nodejs >= 14` (<https://nodejs.org>)
- `Java >= 11`
- `lein` (<https://leiningen.org>)

Una vez descomprimido o bajado el código de la aplicación, y una vez instaladas las dependencias anteriores solo deberemos ejecutar el siguiente script sobre un sistemas Linux, MacOSX o Windows con WSL (Windows Subsystem for Linux):

```
sh build.sh
```

Este shell script realiza las siguientes operaciones:

- Comprobar que están disponibles los comandos `npm` y `lein` necesarios para la construcción
- Acceder al directorio `frontend`
- Instalar las dependencias necesarias con `npm install`
- Comenzar la construcción de la parte frontend con `parcel` usando `npm run build`

- Acceder al directorio backend
- Iniciar la construcción de un fichero uberjar

Una vez finalizada la ejecución de este comando se habrá creado un único fichero jar ejecutable que contendrá la parte frontend y todas las dependencias necesarias para iniciar la aplicación. Podremos iniciar la aplicación ejecutando el siguiente comando:

```
java -jar kaeme.jar
```

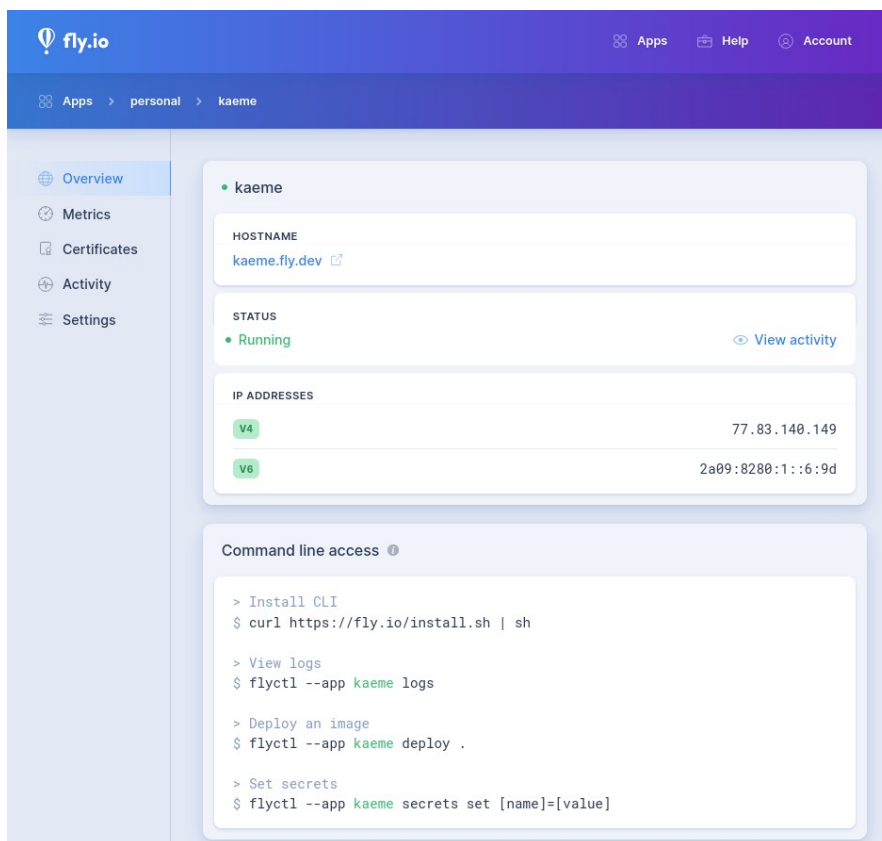
Modificando la variable de entorno DATABASE podremos indicar la ruta donde crear la base de datos SQLite.

14.3 Despliegue de kaeme.site

Con el propósito de enfrentarnos a problemas reales durante un despliegue real se ha procedido a desplegar la aplicación en un entorno de producción real. Para ello se ha evaluado distintos proveedores que soportan docker y se ha optado por el proveedor fly.io

14.3.1 Registro en la plataforma

Nos registramos en la plataforma e instalamos el comando fly siguiendo las instrucciones.



The screenshot shows the fly.io dashboard for an application named 'kaeme'. The interface includes a navigation menu on the left with options like Overview, Metrics, Certificates, Activity, and Settings. The main content area displays the application's details:

- kaeme**: Application name.
- HOSTNAME**: kaeme.fly.dev
- STATUS**: Running (with a 'View activity' link).
- IP ADDRESSES**:
 - V4: 77.83.140.149
 - V6: 2a09:8280:1::6:9d
- Command line access**:
 - > Install CLI: `$ curl https://fly.io/install.sh | sh`
 - > View logs: `$ flyctl --app kaeme logs`
 - > Deploy an image: `$ flyctl --app kaeme deploy .`
 - > Set secrets: `$ flyctl --app kaeme secrets set [name]=[value]`

Figura 41: Página de la aplicación en fly.io

Una vez instalado el comando flyctl, nos dirigimos a la raíz de nuestro proyecto (donde está el fichero Dockerfile) y ejecutamos el siguiente comando:

```
fly deploy
```

Durante la primera ejecución nos autenticaremos de nuevo, y nos preguntará por el nombre de la nueva aplicación, la región donde queremos que se ejecute la aplicación y el puerto por defecto en el que escucha nuestra contenedor.

Una vez finalizada la creación de la aplicación, y cada vez que queramos subir nuevos cambios, se iniciará la construcción del contenedor y se desplegará de forma sencilla y automática en la plataforma.

14.3.2 Registro de dominio

La plataforma fly.io nos genera un subdominio con el nombre de nuestra aplicación, pero con el propósito de mejorar la imagen registramos el dominio **kaeme.site** utilizando el registrado namecheap:

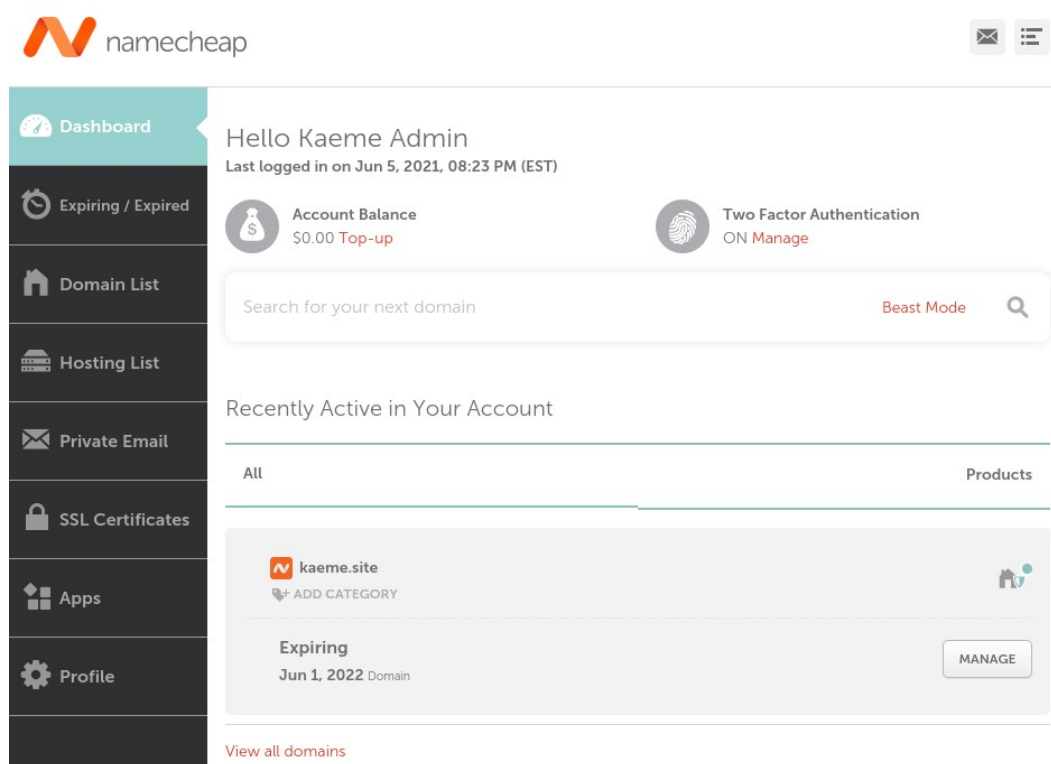


Figura 42: Página de configuración en namecheap

Una vez hecho esto, configuramos en fly.io nuestro nuevo dominio para lo cuál seguimos las instrucciones de la página de fly, que consiste en agregar las ip de fly en los registros A y AAAA en nuestras DNS en namecheap. Finalmente asociamos nuestro dominio a nuestra aplicación con el siguiente comando:

```
flyctl certs create kaeme.site
```

Tras esto, fly realizará comprobaciones y emitirá automáticamente un certificado https usando let's Encrypt de forma gratuita. Por lo que tendremos disponible nuestra aplicación en la url <https://kaeme.site>.

14.3.3 Registro en Cloudflare

Como último paso se activa también el uso de Cloudflare. Durante la activación se copiaron todas las entradas actuales de nuestro dominio y nos indica que sustituyamos los servidores dns de namecheap por los de Cloudflare. Gracias a este servicio, podremos optimizar el rendimiento de nuestras aplicaciones, agregando la posibilidad de cachear recursos y hacer uso de su red distribución CDN para acelerar la carga de nuestra aplicación. Esta servicio ofrece además la opción de forzar la navegación siempre por HTTPS, protección contra ataques DDOS o la activación de cortafuegos WAF para detectar ataques o peticiones sospechosas con el fin de proteger nuestra aplicación.

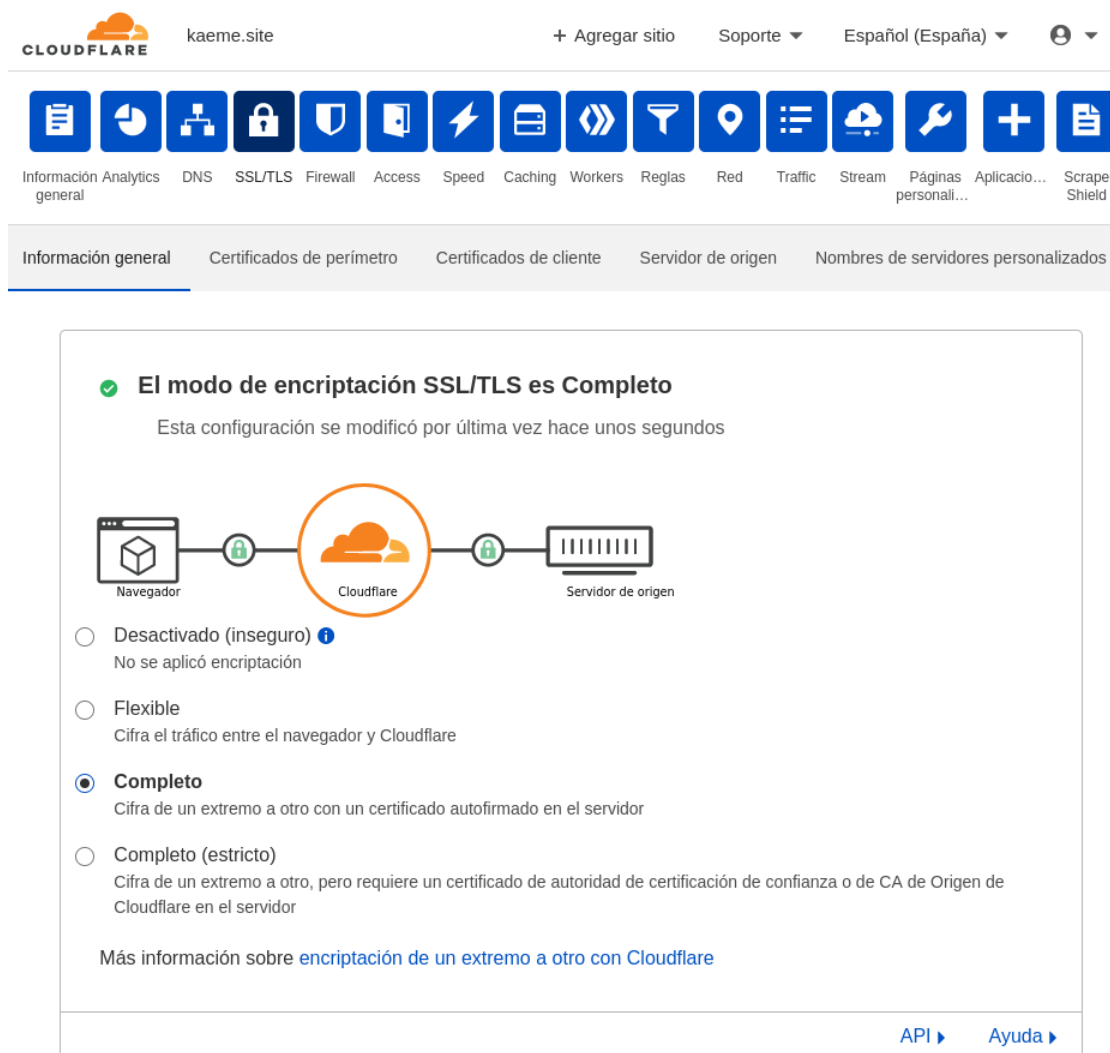


Figura 43: Página de configuración en cloudflare

15 Proyección a futuro

Durante la realización del proyecto se ha tenido que ajustar la cantidad de funcionalidades para que fuera abordable en las horas y créditos correspondientes. Pese a desarrollar un conjunto de funcionalidades correspondientes a un producto mínimo viable (MVP), hay numerosas mejores e ideas que podrían ser interesantes de cara al futuro del proyecto.

Multientidad

Si bien es posible realizar distintas instalaciones independientes para cada cliente, sería más interesante que la aplicación pudiera gestionar la multientidad desde una misma instancia. Esto facilitaría el mantenimiento y el alta de nuevos potenciales clientes, y por tanto orientas el proyecto como un SaaS (Software as a Service).

Dominios personalizados

Relacionado con el punto anterior, una vez que la plataforma pudiera crear entidades sin necesidad de nuevas instalaciones manuales, se podría proporcionar elementos que permitan la personalización de la aplicación, no solo en cuanto logo, nombres y colores, si no también el uso de dominios personalizados. De esta forma academias o clientes que tengan sus propios dominios podrían crear subdominios específicos para ofrecer los servicios de esta aplicación.

Envío de correo

Una funcionalidad sencilla de implementar es el envío de correos. Esto se podrían utilizar para facilitar la recuperación o cambios de contraseñas de los administradores, pero también nos permitiría comunicarnos con los alumnos que hayan informado este dato.

Gestión de clases

Actualmente la lista de alumnos es plana. Sería interesante permitir agrupar los alumnos en grupos o clases que permita facilitar realizar el seguimiento de forma individual o colectiva a nivel de clase.

Gestión de temarios

El MVP actual solo permite gestionar un temario. En una escala superior, es previsible que se tengan que administrar temarios independientes. De esta forma, se podría controlar qué profesores tiene permisos de gestionar los temarios y a que clases o alumnos lo tiene asignado.

Análisis de tarjetas

Si bien el sistema de repetición espaciada intenta optimizar las tarjetas en base a cada alumno, se podría recoger estadísticas globales respecto a las tarjetas. Esto nos permitiría poder determinar que tarjetas suelen fallar más o ser aprendidas antes. De esta forma podríamos agregar nuevas clasificaciones a las tarjetas, como por ejemplo nivel de dificultad en base a los resultados de los propios alumnos.

Permitir alumnos informar de erratas

Es posible que a los profesores comentan errores en la redacción de los temas o en la elaboración de las tarjetas. Implementando un sistema donde los alumnos puedan enviar comentarios sobre el temario o las tarjetas no permitiría detectar erratas o errores de forma más rápida.

Tipos de tarjetas y exámenes

Es posible agregar nuevos tipos de retos o tarjetas, como pueden ser auto completar frases o incluir la selección de posibles respuestas. Esto último podría servir para crear preguntas de exámenes tipo test por ejemplo.

Permitir la importación/exportación a excel/word

Para facilitar la gestión se pueden crear para la opción del temario la opción de exportar el temario en un fichero ofimático con el fin de imprimirse o convertirse en PDF. De igual forma podríamos hacer el caso contrario, importar un temario a partir de un fichero ofimático. Respecto a las preguntas también podrían ser importadas o exportadas desde ficheros u hojas de cálculo.

Integración con plataformas de video

Existen muchas plataformas que permiten gestionar la inclusión de videos y controlar su visionado. De esta forma se podrían facilitar a los clientes la creación de estos videos como explicación de los temas y controlar la visualización de forma privada.

Gamificación

Por último, uno de las funcionalidades más interesantes sería agregar distintos aspectos de gamificación al proceso. El objetivo es involucrar o retar al usuario a conseguir su objetivo. Esto se puede conseguir mostrando progresos relativos a su propio progreso, o también con rankings o clasificaciones dentro de un mismo grupo de usuarios o clase.

16 Conclusiones

El objetivo de este proyecto era el diseño de una plataforma de aprendizaje basada en tarjetas de memoria que permitan a nuestros usuarios gestionar un temario y un conjunto de preguntas y respuestas asociados a cada punto, y poder así registrar y hacer el seguimiento a los alumnos que realicen el proceso de aprendizaje.

La elección de esta temática fue debida a la experiencia propia en el uso de distintas aplicaciones para aprender distintas materias (cursos de inglés, cursos de patrón de embarcaciones PER, cursos de preparación de certificaciones). Todas estas aplicaciones compartían conceptos y funcionamiento similares pero parecían desarrollos a medida. Se pensó que se podría crear una herramienta que pudiera ser lo bastante flexible para ofrecer estas funcionalidades a creadores de contenido (academias) o formadores de cualquier temática. Nuestro público objetivo por tanto son estas entidades que ofrecerán a su vez este servicio a sus alumnos.

Como se ha comentado anteriormente, se ha implementado un conjunto de funcionalidades mínimas que sirvan de mínimo producto viable, y existe gran espacio para mejoras que se pueden abordar. Es este un nicho de mercado que se puede explorar para validar la viabilidad y demanda de esta aplicación con el propósito de ayudar en la digitalización de negocios y academias tradicionales.

Se ha elegido un conjunto de requisitos abordable, realizado la planificación del proyecto, análisis, diseño e implementación hasta conseguir el despliegue de una aplicación funcional. Este proyecto ha permitido poner en práctica muchas de las técnicas, procesos, herramientas y tecnologías que se han estudiado a lo largo del Máster, así como poder evaluar nuevas tecnologías interesantes en el contexto de un proyecto real.

17 Anexo 1. Entregables del proyecto

Junto con este documento se ha hecho entrega del código fuente de la aplicación comprimida en formato .zip. Dentro de este fichero se encuentra un documento README con las mismas indicaciones sobre como construir el proyecto que se explicaron anteriormente en el punto de instalación. Además el código fuente de la aplicación está disponible en el siguiente repositorio GitLab:

<https://gitlab.com/borcha/TFM>

18 Anexo 2. Código fuente (extractos)

Una de las partes más relevantes del proyecto es la relacionada con la implementación del método de repetición espaciada. Todo lo relacionado con este proceso ocurre en el backend y por tanto está implementado en lenguaje Clojure. Inicialmente se intentará explicar el funcionamiento del algoritmo implementado para posteriormente comentar el código Clojure encargado de gestionar esta lógica.

El objetivo del proceso de repetición espaciada es que los alumnos realicen distintas sesiones de pregunta y respuesta con distintos conjuntos de tarjetas hasta que se considere aprendido el conjunto de tarjetas. Las tarjetas no son elegidas simplemente al azar, sino que para el alumno en particular las tarjetas se clasifican en distinto nivel en función del nivel de conocimiento que tenga sobre la tarjeta, y se usa esta clasificación para optimizar la elección de las siguientes tarjetas a mostrar.

A cada tarjeta se le asigna un nivel que va desde 0 hasta 4, siendo 0 las tarjetas que se consideran no aprendidas en absoluto y 4 tarjetas que han sido respondidas satisfactoriamente.

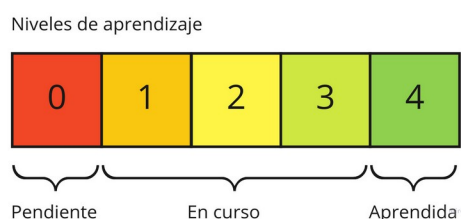


Figura 44: Niveles de aprendizaje de las tarjetas

Todas las tarjetas comienzan en el nivel 0. Cuando un alumno contesta correctamente o indica que sabía la respuesta, la tarjeta en evaluación subiría de nivel, y se le asignaría el nivel 1. En sucesivas sesiones, cuando la tarjeta sigue respondiéndose satisfactoriamente, seguiría subiendo de nivel hasta alcanzar el nivel máximo de 4. En caso de que la tarjeta se indique como incorrecta, ocurriría el proceso contrario y bajaría de nivel. El nivel mínimo que puede alcanzar una tarjeta es de 0.

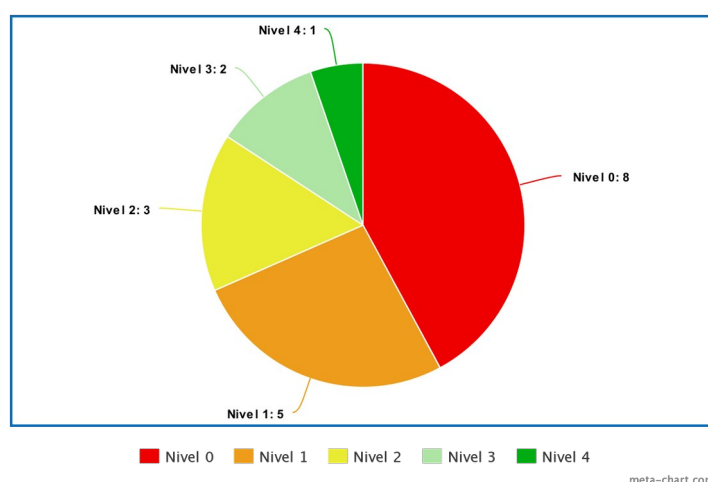


Figura 45: Probabilidad de elección de tarjetas según su nivel

Cuando un alumno quiere realizar una sesión, la aplicación realiza una selección de tarjetas en base a los niveles actuales para este alumno. En el conjunto de tarjetas seleccionadas para la nueva sesión aparecerán con mas frecuencias aquellas tarjetas que tengan menor nivel y con mucho menor frecuencia las tarjetas con niveles mas alto y por tanto consideradas aprendidas. Se hace una selección de tarjetas aleatorias ponderando o asignado mas peso a las tarjetas con menor nivel. El anterior diagrama mostraría la probabilidad de elegir una tarjeta de cada tipo, que si lo asemejamos a una ruleta, sería realizar sucesivas tiradas de la ruleta para elegir que tipo de tarjeta es elegida para formar parte en este nuevo conjunto.

Una vez el sistema elige el conjunto de tarjetas, se produce la sesión por parte del alumno, dando como resultado la clasificación de este conjunto de tarjetas en acertadas/falladas y que servirán para actualizar el nivel de aprendizaje del punto del temario.

En el siguiente diagrama se resumen este proceso que constara de los siguiente pasos:

- Recuperamos los niveles de aprendizaje de las tarjetas que ha estado estudiando el alumno.
- Recuperamos de base de datos el conjunto total de tarjetas asignada al tema repasado.
- Puesto que las tarjetas pueden borrarse, reasignarse a otros temas, o crear nuevas, es necesario realizar una actualización. Por lo tanto las tarjetas que hubieran sido estudiadas por el alumno pero ya no formen parte de tema serían descartadas. Las nuevas tarjetas que se hubieran creado o asignado se incorporaría al estado de progreso del alumno, asignándoles el nivel mínimo de 0.
- Una vez se tiene un conjunto válido de las tarjetas con sus correspondientes niveles se actualizan los niveles de cada tarjeta (siempre teniendo en cuenta el mínimo nivel de 0 y el máximo de 4).
- Realizando el recuento de tarjetas de cada nivel podremos informar al alumno del porcentaje de progreso en el tema que está repasando.

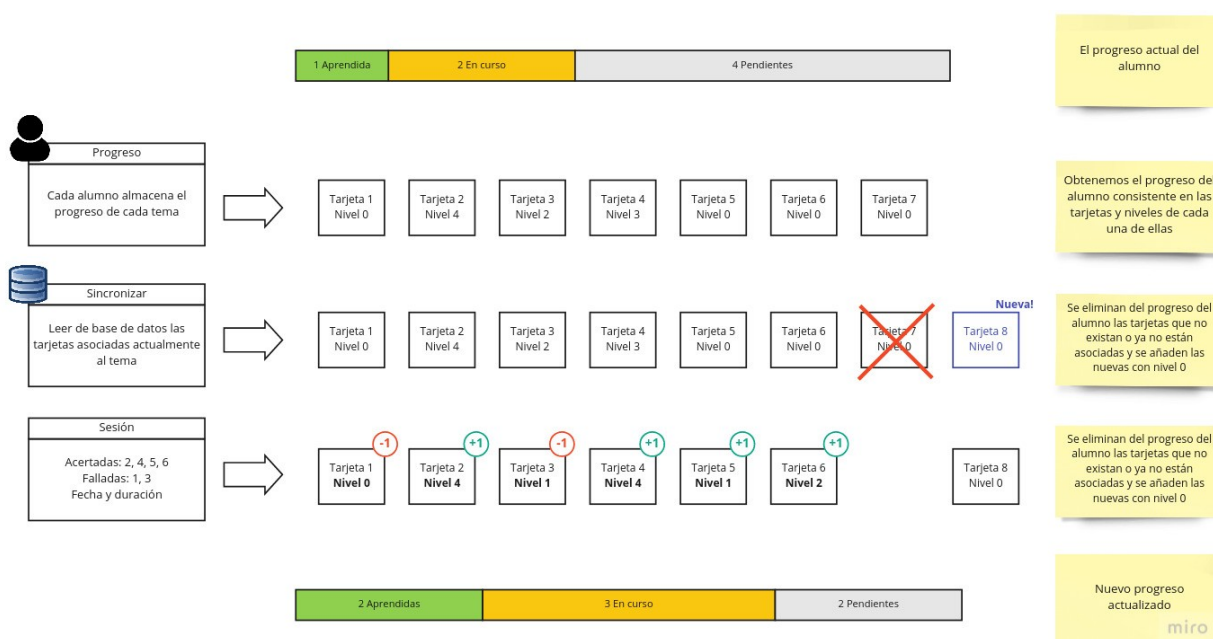


Figura 46: Proceso de actualización del progreso según Letiner

A continuación podemos ver el código Clojure de todo este proceso explicado anteriormente:

```

(defn refresh-progress
  "Actualiza el progreso eliminando y añadiendo nuevas tarjetas que se hayan creado"
  [progress current-ids]
  (let [current (set current-ids)
        valid (filter #(current (:card %)) progress)
        valid-ids (set (map :card valid))
        new (map #(hash-map :card % :level 0) (remove valid-ids current))]
    (doall (into valid new))))

(defn level-up [x] (min (inc x) 4))
(defn level-down [x] (max (dec x) 0))

(defn update-progress
  "Recibe el progreso anterior y el resultado de una sesión.
  Devuelve la lista de cards subiendo o bajando de nivel tarjeta."
  [progress session]
  (let [hit (set (:hits session))
        fault (set (:faults session))]
    (map (fn [{:id :card :as card}]
          (cond-> card
            (hit id) (update :level level-up)
            (fault id) (update :level level-down)))
      progress))

; https://stackoverflow.com/a/15823098
(defn wrand
  "given a vector of slice sizes, returns the index of a slice given a
  random spin of a roulette wheel with compartments proportional to
  slices."
  [slices]
  (let [total (reduce + slices)
        r (rand total)]
    (loop [i 0 sum 0]
      (if (< r (+ (slices i) sum))
        i
        (recur (inc i) (+ (slices i) sum)))))

(def ^:private fibo-weights [8 5 3 2 1])

(defn choose-cards
  "Elige n elementos de forma aleatoria siguiendo la distribución de pesos
  indicada por fibo-weights por cada nivel de aprendizaje"
  [n cards]
  (loop [selected '() cards cards]
    (if (or (empty? cards)
            (= (count selected) n))
        selected
        (let [levels (into (sorted-map) (group-by :level cards))
              weights (mapv fibo-weights (keys levels))
              level (nth (vals levels) (wrand weights))
              card (rand-nth level)]
          (recur (conj selected card))
        ))))

```

Elimina del progreso del alumno las tarjetas inválidas y agrega las nuevas con nivel 0

Las tarjetas solo pueden estar entre los niveles 0 y 4

Sube y baja de nivel las tarjetas acertadas y falladas en la sesión

Elegir un número aleatorio pero utilizando pesos

Pesos actuales, a menor nivel mas probabilidad de ser elegida al componer una nueva sesión

Elegir las tarjetas para la sesión siguiendo la distribución de probabilidad anterior

Figura 47: Comentarios de código de la implementación Leitner

19 Anexo 3. Librerías/Código externo utilizado

A continuación se detallan algunas librerías y componentes que son utilizadas en la aplicación.

19.1 svg-heatmap

El tipo de gráfico heatmap o mapa de calor representa en un diagrama bidimensional los valores de cada coordenada usando un color distinto que representa su magnitud. GitHub popularizó el uso de ese tipo de diagramas al usarlo para mostrar gráficamente el nivel de actividad en un proyecto. Este gráfico consiste por tanto en la representación de los días del año (horizontalmente serían los meses y verticalmente veríamos la semanas). Cada cuadrado sería un día concreto, que tendría asignado un color para cada nivel de actividad.

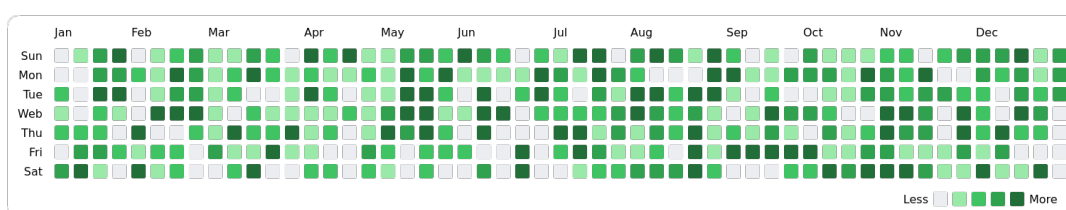


Figura 48: Mapa de calor de actividad

Este mismo gráfico ha sido el elegido para mostrar a los alumnos y profesores el número de sesiones realizadas por un alumno en un periodo de tiempo. El componente utilizado se encuentra disponible en el siguiente repositorio: <https://github.com/jayden-chan/svg-heatmap>

19.2 Trix Editor

Para facilitar la edición de contenido enriquecido para la edición del temario se ha hecho uso del componente Trix Editor. Este es un componente creado y utilizado por Basecamp, empresa referente en el desarrollo web: <https://trix-editor.org/>

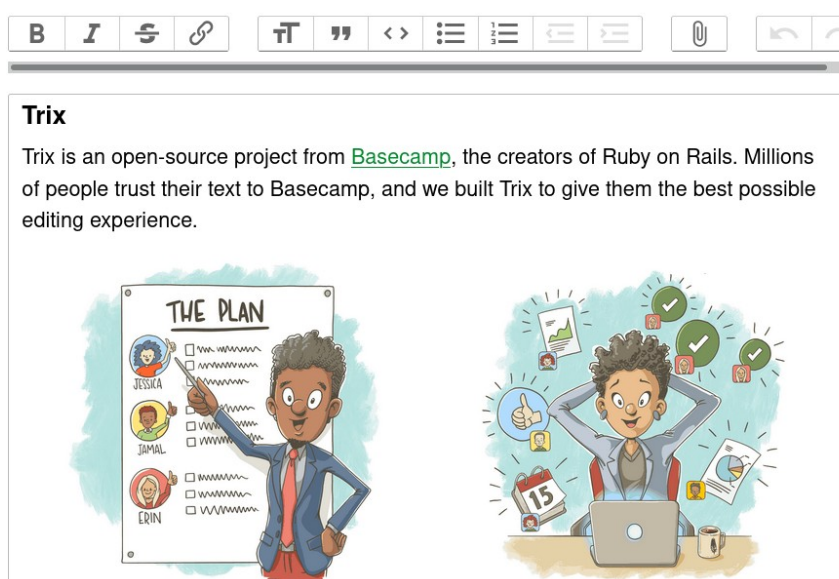


Figura 49: Componente Trix Editor

20 Anexo 4. Guía de usuario

La aplicación consta de dos grandes módulos. El primer módulo va dirigido a profesores y administradores, y el segundo a los alumnos. Una vez accedemos a la aplicación se nos mostrará la siguiente pantalla donde poder seleccionar que módulo acceder.

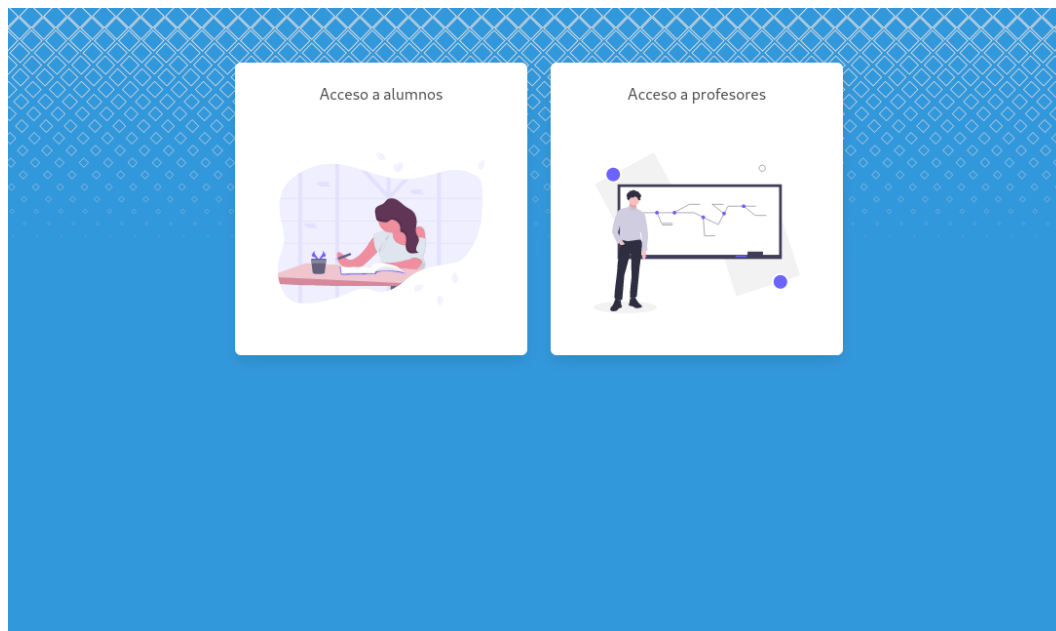


Figura 50: Pantalla de selección del módulo alumno/profesor

20.1 Guía de Profesores

20.1.1 Acceso

Una vez accedamos al módulo de profesores veremos la pantalla de acceso. Aquí deberemos introducir el correo y contraseña asignadas al profesor para poder acceder a la plataforma.

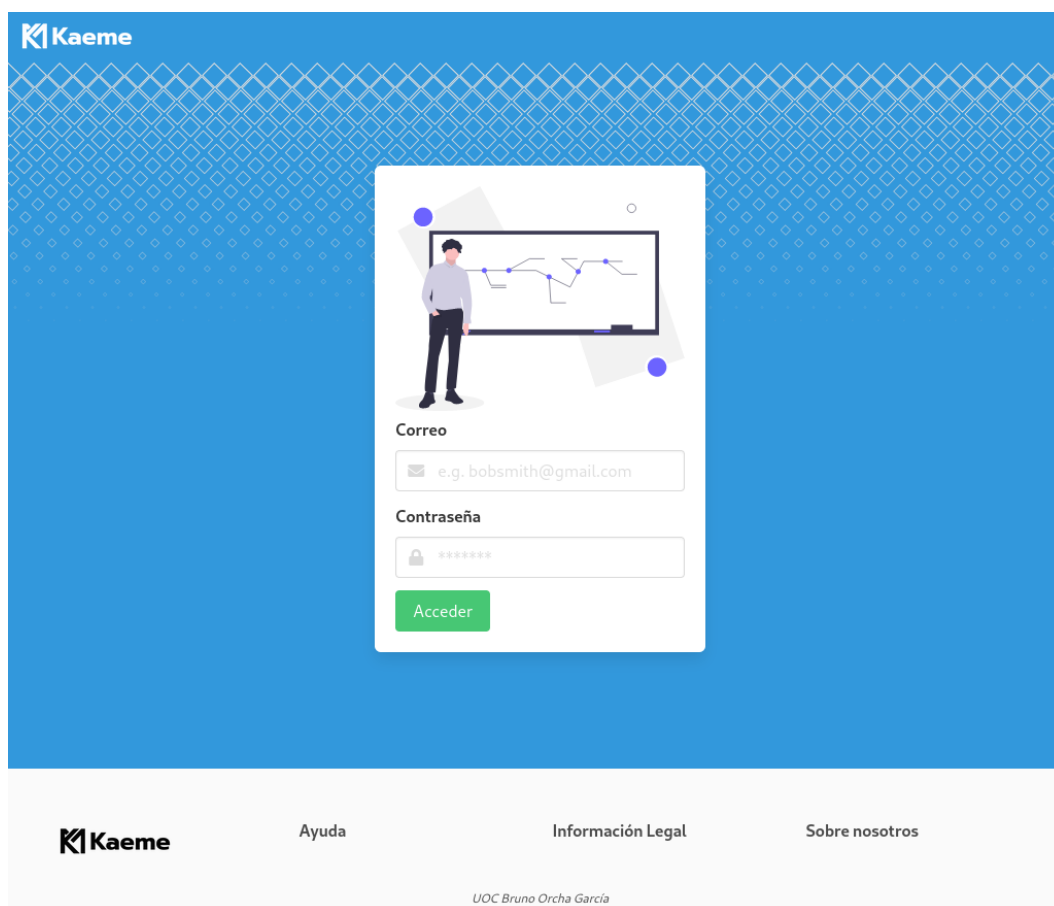


Figura 51: Pantalla de acceso de usuarios administradores/profesores

20.1.2 Inicio

Una vez autenticados accederemos a la pantalla principal de la aplicación desde podremos acceder a los módulos principales de administración. Podremos acceder usando los correspondientes iconos que se muestran como también usando la barra superior para poder acceder directamente. En todo momento podremos pulsar sobre el icono de la aplicación para volver a esta página de inicio.

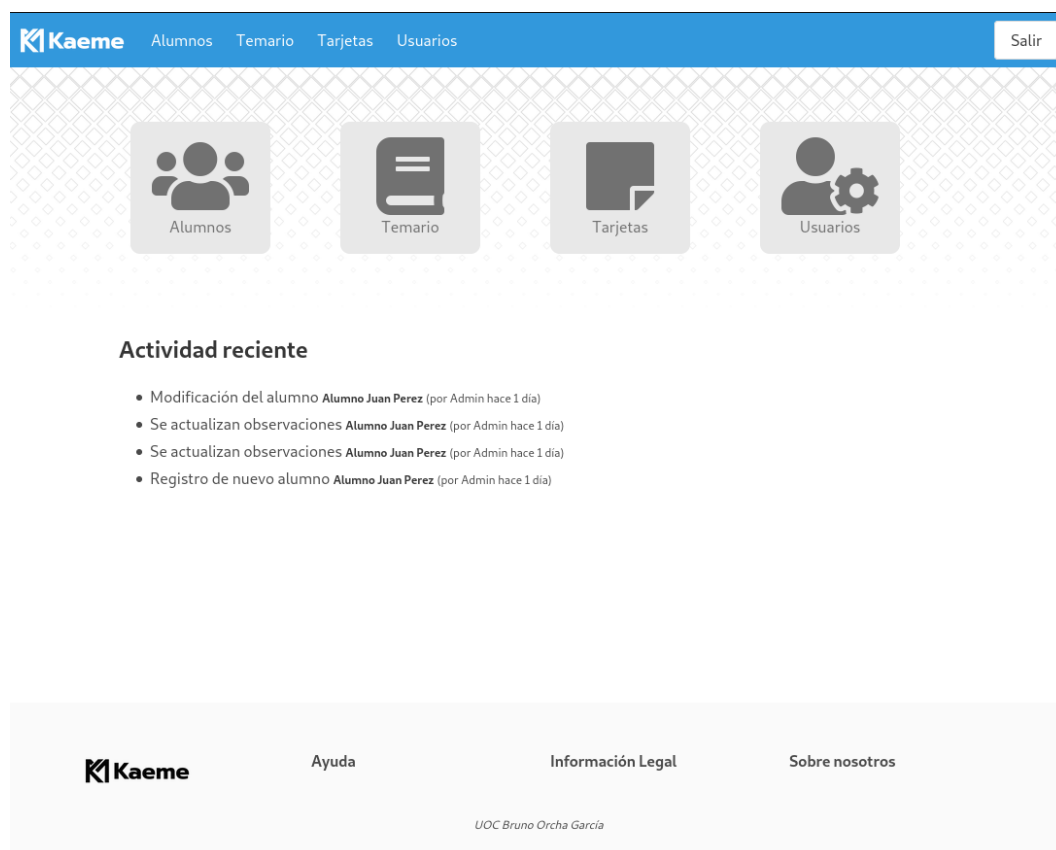


Figura 52: Pantalla de inicio de profesores

20.1.3 Administración de usuarios

Este módulo permitirá consulta, crear y modificar nuevos usuarios administradores o profesores. Todos los usuarios administradores creados tienen el mismo nivel de privilegios y permisos. En cuanto accedamos a este módulo podremos consultar el listado de los usuarios administradores/profesores en el sistema. Desde esta pantalla tendremos la posibilidad de editar un usuario. Para ello pulsaremos sobre la opción editar en el listado. Para crear un nuevo usuario pulsaremos sobre el botón Nuevo usuario.



Figura 53: Listado de usuarios/profesores

20.1.4 Creación/Edición de usuarios

Desde este formulario podremos indicar el nombre, correo y contraseña del nuevo usuario o modificar los datos si es uno existente. Desde esta misma pantalla podremos eliminar el usuario si así lo queremos.

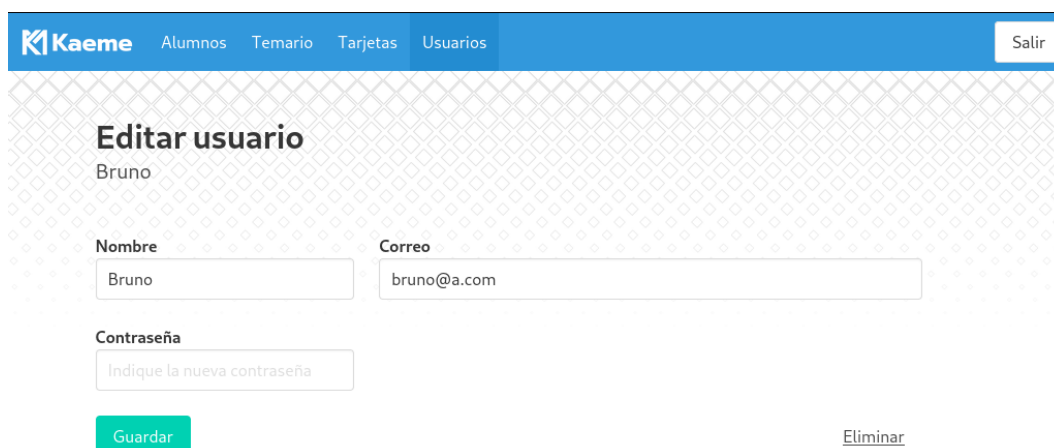


Figura 54: Listado de usuarios/profesores

20.1.5 Administración del temario

Al acceder a este módulo veremos el temario actualmente existente. Desde esta pantalla podremos reordenar los puntos del temario. Para ello deberemos pulsar los botones con las flechas que aparecen en cada fila de cada punto. Podremos mover el punto hacia arriba o abajo, subir nivel de anidación o bajarlo siempre que sea posible. Es decir, solo podremos modificar el orden de un punto dentro de un mismo padre. Al mover un elemento este arrastra todos los elementos hijos consigo mismo.

Desde esta pantalla también podremos crear un nuevo punto del temario mediante el botón “Nuevo tema” o editar uno existente pulsando el enlace “editar” de la fila de la tabla.

Título	Acciones
1. NOMENCLATURA NÁUTICA	
1.1 Casco	
1.2 Estructura	
1.3 Equipo de fondeo	
1.4 Timón	
1.5 Hélice	
1.6 Dimensiones	
1.7 Terminología	
2. AMARRE Y FONDEO	
2.1 Elementos de amarre	
2.3 Fondeo	
2.2 Nudos	
3. SEGURIDAD EN LA MAR	
3.2 Comprobaciones antes de hacerse a la mar	
3.3 Medidas a tomar a bordo con mal tiempo	

Nuevo tema

Figura 55: Administración del temario

20.1.6 Creación/Edición de un tema o capítulo

Al editar o crear un capítulo del temario accederemos a un formulario donde podremos ver los siguientes datos:

- Tema padre: El tema padre del que depende jerárquicamente este punto. Los puntos principales tiene como padre la entrada especial en el menú denominada Raíz. Desde este menú podemos mover fácilmente un capítulo y sus descendientes a nuevo lugar en el temario.
- Título: Es el título que aparece en los listado del temario.
- Contenido: Es contenido en formato texto enriquecido que consiste en los textos e imágenes que conforman el capítulo.

Desde esta pantalla además, podremos eliminar el punto de menú, podremos editar su nombre y editar y formatear su contenido. Además podremos colgar directamente el capítulo de algún otro punto del menú o eliminar

Kaeme Alumnos Temario Tarjetas Usuarios Salir

Editar tema

1. NOMENCLATURA NÁUTICA

Tema padre
Raíz

Título
1. NOMENCLATURA NÁUTICA

Contenido

B **I** **U** **Link** **Text** **Quote** **Code** **Table** **Image** **Undo** **Redo**

Plano de crujía
Plano imaginario longitudinal (proa-popa) que divide la embarcación en dos mitades simétricas. Coincide con la quilla. Su intersección con el casco determina la línea de crujía.

Través
Eje transversal, perpendicular a la línea de crujía, que divide el barco en dos mitades.

Proa
Parte delantera de la embarcación.

Popa
Parte posterior de la embarcación.

Guardar Eliminar

Figura 56: Edición de un tema/capítulo

20.1.7 Administración de tarjetas

Al acceder a este módulo podremos comprobar el listado de tarjetas existentes para cada punto del temario. Mediante el uso del filtro superior a la tabla, podremos comprobar el número de tarjetas existentes para cada tema y mostrar solo los elementos de dicho punto.

Desde esta pantalla podremos crear nuevas tarjetas seleccionando el botón “Nueva tarjeta” o editar las existentes pulsando el enlace “editar” de la fila correspondiente.

Kaeme Alumnos Temario **Tarjetas** Usuarios Salir

Tarjetas

Lista de tarjetas de estudio

1.2 Estructura (44) ▾

Título	Acciones
Limera	
bichero	
Gobernar con Caña	
Manguerote	
Correr el temporal	
Limera	
bichero	
Gobernar con Caña	
Manguerote	
Correr el temporal	

Anterior Siguiete 1 2 3 4 5

Nueva tarjeta

Figura 57: Listado de tarjetas

20.1.8 Creación/Edición de una tarjeta

Al seleccionar la edición o creación de una tarjeta veremos un formulario con las siguientes opciones:

- Tema relacionados: Es el tema asociado a esta tarjeta. La tarjeta debería poner a prueba un conocimiento explicado en el contenido de este punto temático.
- Título: Es un título de uso interno que sirve para identificar la pregunta
- Pregunta: Es el texto de pregunta que se le mostrará a los alumnos
- Respuesta: Es la correspondiente respuesta a la pregunta anterior.

Una vez finalizada la edición podremos guardar los cambios en la tarjeta existen o crearse una nueva. Desde esta pantalla también tendremos la posibilidad de eliminar la tarjeta.

Kaeme Alumnos Temario Tarjetas Usuarios Salir

Editar tarjeta

Limera

Tema relacionado
1.2 Estructura

Título
Limera

Pregunta
Apertura en la parte de popa, encima del codaste, que permite el paso de la mecha hasta la rueda/caña.

Respuesta
Limera

Guardar Eliminar

Figura 58: Edición de tarjetas

20.1.9 Administración de alumnos

Desde este menú podemos consultar la lista de alumnos existentes. Desde aquí podremos ver su nombre, fecha de inscripción y su estado. Aquellos alumnos desactivados aparecerán en color gris. En cada una de las filas que representa un alumno podremos entrar a consultar sus datos de sesiones para realizar el seguimiento o editar el alumno. Desde esta pantalla también podremos crear nuevos alumnos seleccionando la opción “Nuevo alumno”.

Kaeme Alumnos Temario Tarjetas Usuarios Salir

Alumnos

Listado de alumnos registrados

Nombre	Apellidos	Fecha de inscripción	Acciones
Alvaro	Arrimadas	2021-02-06	
Benito	Basallote	2021-02-25	
Carlos	Caronte	2021-05-01	
Daniela	De la Rosa	2021-04-24	
Estefania	Echenique	2021-05-05	
Lunita	asdfasf	2021-05-20	

Nuevo alumno

Figura 59: Listado de alumnos

20.1.10 Creación/Edición de alumnos

Durante la creación de alumnos accederemos a un formulario donde podremos editar los datos personales y de contacto del alumno (nombre, apellidos, teléfono y correo). Además podremos indicar la fecha de inscripción. Una vez pulsemos el botón “Crear” crearemos el nuevo alumno.

En el caso de editar un alumno accederemos al mismo formulario que en el caso anterior, que nos permitirá editar los datos personales y contacto, pero además podremos ver el estado del alumno, además de tener disponibles nuevas acciones:

- **Activar/Desactivar:** Esta opción nos permite cambiar el estado a los alumnos. Los alumnos desactivados no podrán acceder a la plataforma.
- **Código de activación:** Para darle acceso a los alumnos a la plataforma se les facilita un código de activación único. Este código se genera al pulsar este botón. Si un alumno quiere utilizar otro dispositivo para sus sesiones deberá solicitar un nuevo código de activación al profesor.
- **Eliminar:** Podremos eliminar al alumno desde esta misma pantalla.

The screenshot shows the 'Editar alumno' interface. At the top, there is a navigation bar with 'Kaeme' and menu items: 'Alumnos', 'Temario', 'Tarjetas', 'Usuarios', and a 'Salir' button. The main heading is 'Editar alumno' with the name 'Alvaro Arrimadas' below it. The form contains the following fields and controls:

Nombre Alvaro	Apellidos Arrimadas
Teléfono 666777888	Correo alvaro@prueba.com
Fecha de inscripción 06 / 02 / 2021	Estado ACTIVO

At the bottom, there are four buttons: 'Guardar' (green), 'Código de activación', 'Desactivar', and 'Eliminar' (underlined).

Figura 60: Edición de alumnos

20.1.11 Consultar alumnos

Desde esta opción podremos hacer el seguimiento del alumno. Al acceder se nos mostrará la información básica de contacto y tres secciones con la siguiente información:

- **Actividad**
 - **Sesiones realizadas:** Aquí veremos el mapa de calor con la representación del número de sesiones realizadas en los últimos meses.
 - **Tarjetas Totales:** Aquí veremos un resumen del número total de tarjetas que el alumno está estudiando.

- Actividad Reciente: Se muestra un listado de los últimos eventos relacionados con el alumno.
- Progreso: Aquí podremos ver el temario completo junto con datos referentes al número de sesiones realizadas en cada tema, fecha de la última sesión y el estado de aprendizaje de las tarjetas asociadas.
- Notas: Esta opción permite agregar observaciones y notas al alumno que sirva de referencia al profesor.

Consulta de alumno
Benito Basallote (666777888 / beniteo@prueba.com)

Actividad Progreso Notas

Sesiones realizadas

Aquí puede ver una gráfica con la actividad diario del alumno de los últimos meses.

	May	Jun
Lun		
Mar		
Mie		
Jue		
Vie		
Sab		
Dom		

Tarjetas totales

Pendientes	En curso	Aprendidas
7	20	8

Actividad reciente

Última actividad relacionada con el alumno

- Nuevo código de activación Alumno Benito Basallote (por Bruno hace 1 semana)
- Nuevo código de activación Alumno Benito Basallote (por Bruno hace 1 semana)

Figura 61: Consulta/seguimiento de alumnos

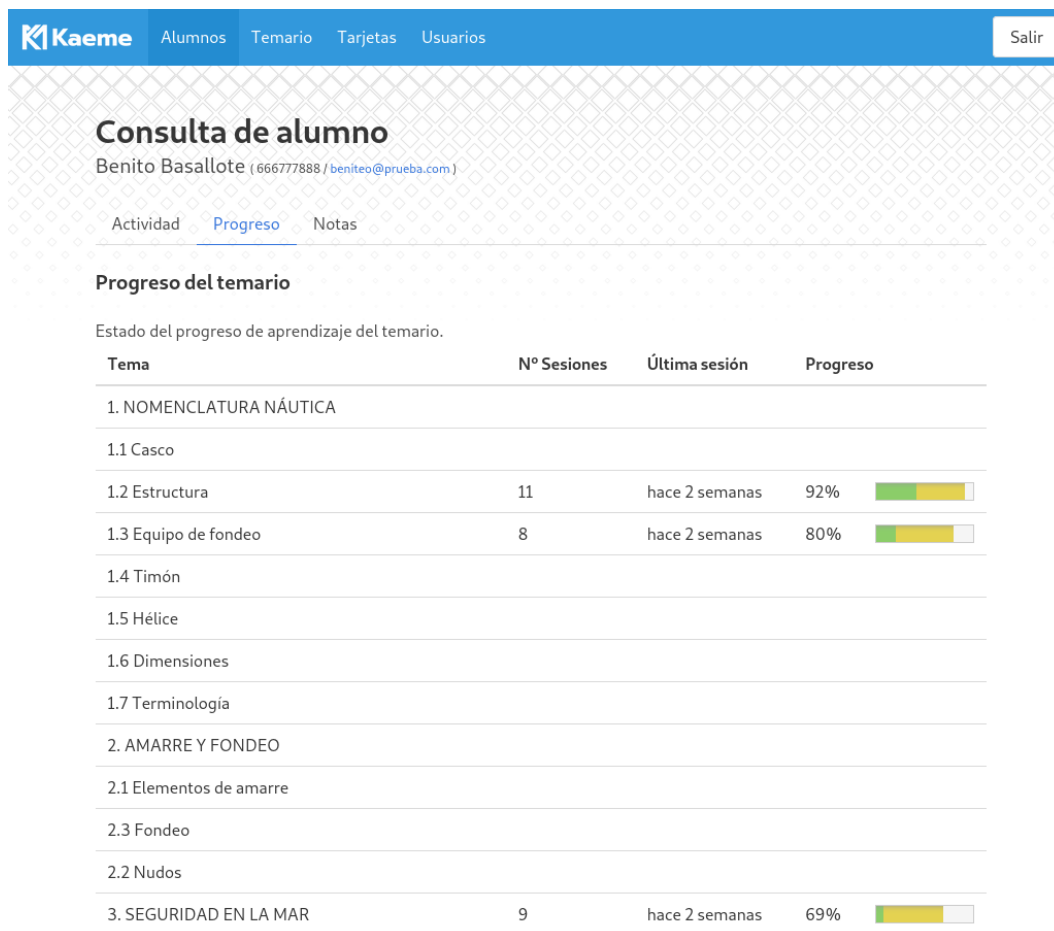


Figura 62: Progreso del temario del alumno

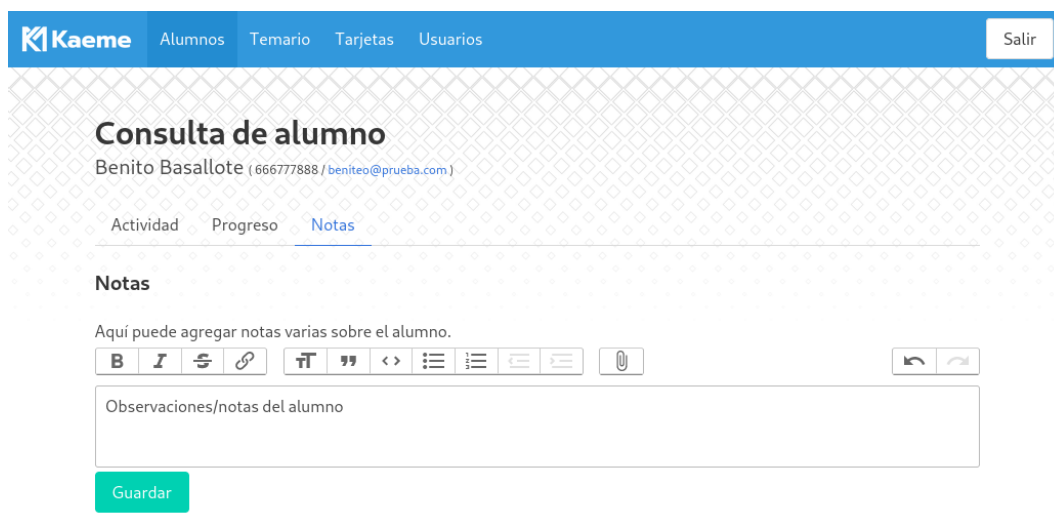


Figura 63: Edición de anotaciones y observaciones del alumno

20.2 Guía del alumno

20.2.1 Activación

Cuando los alumnos acceden por primera vez a la plataforma se les mostrará la pantalla de activación. En este momento deberán introducir el código de activación facilitado por sus profesores para registrar el dispositivo. Solo puede tener un dispositivo registrado a la vez.



Figura 64: Pantalla de activación del alumno

20.2.2 Inicio

En esta pantalla el alumno puede ver el tema que actualmente está estudiando, su nivel de progreso y estadísticas. Respecto a estadísticas se le mostrará el mapa de calor con información de la actividad de los últimos meses, además del total de tarjetas pendientes, en curso y aprendidas del conjunto del temario.

En la parte superior tendría las siguiente información y opciones:

- Tema actual: Muestra el título del último tema del cuál se haya realizado una sesión de aprendizaje.
- Progreso: Muestra la barra de progreso de aprendizaje. En verde se mostrarían el porcentaje de tarjetas que se consideraría aprendidas. En amarillo las que están en proceso y en gris las que todavía están pendientes (es decir, no se ha contestado todavía o siempre se han fallado).
- Ver temario: Desde aquí podremos acceder al temario preparado por los profesores
- Repasar: Podremos iniciar una sesión de estudio del tema actual
- Siguiente tema: Si tenemos completado la barra de progreso del tema podremos ir al siguiente tema.



Figura 65: Pantalla de inicio del alumno

20.2.3 Consulta del temario

Cuando pulsamos sobre “Ver Temario” en la pantalla de inicio navegaremos a la lista completa de puntos del temario. Aquí el alumno podrá ver qué temas tiene completado al mostrarse un icono verde y cuáles están en proceso que se mostrarían con un icono amarillo. Pulsando en alguno de los puntos accedemos al contenido del capítulo.



Figura 66: Consulta del temario del alumno

20.2.4 Consulta de un tema o capítulo

Cuando pulsamos sobre uno de los capítulos desde el temario, podemos ver el contenido creado por los profesores. Al inicio de la página veremos la barra que informa del nivel de progreso. Al final de esta página podremos ver la siguiente información y opciones:

- Estadísticas de tarjetas: El número concreto de tarjetas y en el estado que se encuentra cada una de ellas.
- Repasar: Al pulsar sobre este botón iniciaríamos una nueva sesión de estudio para este tema.
- Reiniciar: Este botón serviría para eliminar todo el progreso realizado en un tema. El propósito puede ser repasar desde el principio todas las tarjetas del tema.

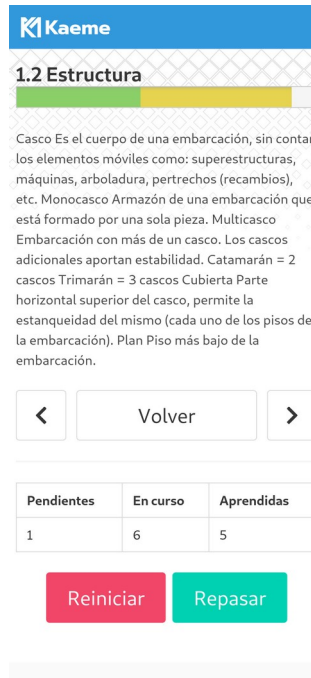


Figura 67: Consulta del temario del alumno

20.2.5 Sesiones

Cuando iniciamos una sesión el sistema comenzará a realizar preguntas al alumno. En la parte superior veremos una barra de progreso para saber cuántas tarjetas quedan para terminar la sesión. Durante la sesión se mostrará la pregunta y un botón “Mostrar Respuesta”. Al pulsar el botón se mostrará la respuesta a la pregunta junto con dos nuevos botones que permiten al alumno indicar que se sabía la respuesta o no.

Una vez termina la sesión, se muestra un resumen con el número de aciertos, fallos y tiempo transcurrido. Tras la realización de la sesión el nivel de progreso asociado a este tema se habrá actualizado.

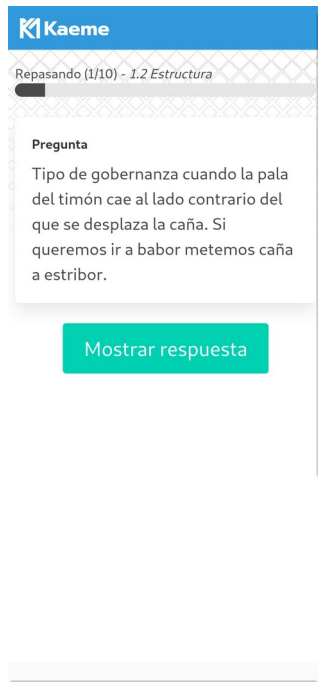


Figura 68: Ejecución de una sesión



Figura 69: Resumen de la sesión realizada

21 Anexo 5. Bibliografía

A continuación se detallan los principales recursos bibliográficos y web consultados durante la realización de este proyecto:

1. Patrones de diseño para interfaces de usuario: <http://ui-patterns.com/>
2. Practical UI Patterns for Design Systems: Fast-Track Interaction Design for a Seamless User Experience. ISBN: 9781484249383.
<https://learning.oreilly.com/library/view/practical-ui-patterns/9781484249383/>
3. Especificación y manual de GraphQL: <https://graphql.org/learn>
4. Programming Clojure, 3rd Edition. ISBN: 9781680502466.
<https://learning.oreilly.com/library/view/programming-clojure-3rd/9781680505719/>
5. Lacinia – GraphQL Clojure Server Implementation Manual. <https://lacinia.readthedocs.io/en/latest/>
6. Framework web Pedestal: <http://pedestal.io/>
7. Plugin para Chrome para facilitar el diseño responsive: <http://re-view.emmet.io/>
8. Framework CSS: <https://bulma.io/>
9. Lenguaje Clojure <https://clojure.org/>
10. Documentación y ejemplos Clojure <https://clojuredocs.org/>
11. Servicio de alojamiento de imágenes Documentación API:
https://cloudinary.com/documentation/image_upload_api_reference
12. Fuente de iconos: <https://fontawesome.com/>
13. Librería para peticiones GraphQL de Prisma: <https://github.com/prisma-labs/graphql-request>
14. Versiónado de Base del esquema de base de datos: <https://github.com/yogthos/migratus>
15. Construcción de proyectos basados en Clojure: <https://leiningen.org/>
16. Herramienta para construcción de proyectos frontend: <https://parceljs.org/>
17. Librería de componentes web: <https://svelte.dev>
18. Librería para enrutamiento local: <https://visionmedia.github.io>
19. Generador de código a partir de esquemas GraphQL: <https://www.graphql-code-generator.com/>
20. Librería para acceso a base de datos SQL: <https://www.hugsql.org/>
21. Motor de base de datos embebido: <https://www.sqlite.org>
22. Lenguaje TypeScript para el lado cliente: <https://www.typescriptlang.org/>
23. Fuente del logo: https://fonts.google.com/specimen/Prompt?preview.text=Kaeme&preview.text_type=custom
24. Recursos de imágenes libres: <https://undraw.co/search>